Feedback 2.0: An Investigation into Using Sharable Feedback Tags as Programming Feedback

Ph.D. Thesis

August 2010

Stephen Cummins

Technology Enhanced Learning Research Group
School of Engineering and Computing Sciences
Durham University

Abstract

Objectives: Learning and teaching computer programming is a recognised challenge in Higher Education. Since feedback is regarded as being the most important part of the learning process, it is expected that improving it could support students' learning. This thesis aims to investigate how new forms of feedback can improve student learning of programming and how feedback sharing can further enhance the students' learning experience.

Methods: This thesis investigates the use of new forms of feedback for programming courses. The work explores the use of collaborative tagging often found in Web 2.0 software systems and a feedback approach that requires examiners to annotate students source code with short, potentially reusable feedback. The thesis utilises a variety of research methods including questionnaires, focus groups and collection of system usage data recorded from student interactions with their feedback. Sentiment and thematic analysis are used to investigate how well feedback tags communicate the intended message from examiners to students. The approaches used are tested and refined over two preliminary investigations before use in the final investigation.

Results: The work identified that a majority of students responded positively to the new feedback approach described. Student engagement was high with up to 100% viewing their feedback and at least 42% of students opting to share their feedback. Students in the cohort who achieved either the lower or higher marks for the assignment appeared more likely to share their feedback.

Conclusions: This thesis has demonstrated that sharing of feedback can be useful for disseminating good practice and common pitfalls. Provision of feedback which is contextually rich and textually concise has resulted in higher engagement from students. However, the outcomes of this research have been shown to be influenced by the assessment process adopted by the University. For example, students were more likely to engage with their feedback if marks are unavailable at the time of feedback release. This issue and many others are proposed as further work.

Declaration of Authorship

I, Stephen Cummins, declare that this thesis titled, 'Feedback 2.0: An Investigation into Using Sharable Feedback Tags as Programming Feedback' and the work presented in it are my own. I confirm that no part of the material provided has previously been submitted by the author for a higher degree in Durham University or any other University. All the work presented here is the sole work of the author.

This research has been documented or is related, in part, within the publications listed below:

- Cummins, S. (2008), 'TR-TEL-08-05: Changing Programming Feedback Using Web 2.0 Technologies', Technology Enhanced Learning Research Group Technical Report Series, Durham University, August 2008
- Geldart, J. and Cummins, S. (2009), 'The Automatic Integration of Taxonomies with Folksonomies using Non-Axiomatic Logic', Information Systems Development: Towards a Service Provision Society
- Cummins, S., Burd, L. and Hatch, A. (2009) 'Folksonomies Of Feedback: Tagging Source Code', Higher Education Academy Subject Centre for Information and Computer Science, 25th-27th August 2009, University of Kent in Caterbury
- Cummins, S., Burd, L. and Hatch, A. (2009) 'Tag Based Feedback for Programming Courses', ACM SIGCSE inroads Bulletin, 41 (4), December 2009
- Cummins, S., Burd, L. and Hatch, A., (2010) 'Using Feedback Tags and Sentiment Analysis to Generate Shareable Learning Resources', In Proceeding of the 10th IEEE International Conference on Advanced Learning Technologies, July 5-7, 2010, Sousse, Tunisia
- Cummins, S., Burd L. and Hatch, A, (2010) 'Tags as a Feedback Mechanism in Programming Courses: Analysis to Support Just-in-Time Teaching', Higher Education Academy Subject Centre for Information and Computer Science, 24th-26th August 2010, Durham University

Acknowledgements

I would like to acknowledge and thank my friends (especially Emma Back, Andy & Tam Burn, Ryan Ford, Richard Delley, David Lewis and Adam Low) and all of my family, for understanding and respecting my need to work. I could not have finished without your support.

Thank you to Professor Liz Burd and Doctor Andrew Hatch, for providing invaluable support and technical advice on the direction of this research.

I would like to thank Doctor Patricia Shaw for allowing me to test the SWATT approach for programming feedback within her course. Without her support, I would have been unable to conduct a large portion of this research.

Special thanks to Andy Burn, Joan Llewellyn and Adam Low for either proof-reading this thesis or helping with the manual sentiment / thematic analysis validation; both were of great help.

I would like to acknowledge the Centre for Excellence in Teaching and Learning: Active Learning in Computing for partially funding this research.

I would finally like to thank, the Ogden Trust, County Durham Economic Partnership, One North East and Durham University for funding this research for one year and giving me the opportunity to work with the fantastic Tanfield School as an Outreach Teaching Fellow. Thank you to all of the science department at Tanfield school for making me feel welcome and a part of the team; working with you was a great experience and was hugely rewarding.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Learning and teaching computer programming is widely recognised as being a challenging undertaking within Higher Education (DuBoulay, 1989; Robins et al., 2003; Winslow, 1996) and does not appear to have become any easier over time.

Feedback "... is the life blood of learning" (Rowntree, 1987), without meaningful feedback there cannot be any learning. Regardless of whether the feedback is generated internally from the learners' past experiences or externally from a lecturer assessing a software project, feedback is an exceptionally important aspect of the learning process.

Students often identify the quantity (Weaver, 2006) or quality of feedback they receive as being below expectations (NSS2009, 2009; Rowe and Wood, 2007). Often, this can be as a direct result of strict time constraints being placed on examiners, combined with the large number of assignments that must be assessed.

It is the importance of feedback and the recognised difficulty of learning programming that provide justification for this thesis investigating a new approach to generation, dissemination and interaction with feedback. Logically it follows that if feedback is how we learn, then by improving it we will in

some way improve our learning. This is the premise of the work presented within this thesis.

All of the research presented in this thesis was carried out in one institution, Durham University, using staff and students from a variety of cohorts as participants. Within the Computer Science and Software Engineering courses at Durham University, there are a variety of modules that require undergraduates to be able to develop software in various programming languages. Of these modules, two are of particular interest, the level 1 "Introduction to Programming" course, where students are not required to have any prior programming knowledge and the level 2 "Software Engineering Group Project". Both of these courses involve a number of student programming activities, which will be used to investigate the new approach to programming feedback presented in this thesis. The two modules under investigation both use the Java programming language and as such, this thesis focuses on investigating feedback for Java programming assessments only. However, the techniques presented are not specific to Java and should be usable with any programming language.

1.2 Research Objectives

This thesis investigates the effects of utilising the popular Web 2.0 tagging paradigm as a means of providing feedback tags for student programming projects. The notion of sharing the feedback generated in tag form will be explored within this thesis, in order to identify how students interact with their own and each others' feedback and whether they perceive any benefit in receiving sharable feedback tags. It is hoped that by providing feedback in a novel and more interactive form, students will engage more with their feedback and this increase in engagement may support them in modifying their learning.

It is expected that using feedback tags will be beneficial in the analysis of individual and cohort feedback as it enables analysis techniques that are

regularly used in tagging systems, for example tag clouds and co-occurrence analysis of tags. This may provide additional information about the specific strengths and weaknesses of students, which otherwise may remain undiscovered.

The main objective of this research is to investigate whether or not providing feedback in the form of tags associated with source code fragments and the ability to share these anonymously, is perceived as being beneficial by students or examiners. The importance of student perception of feedback is a key focus of this thesis, since it is arguable that the learner is in the best position to evaluate how useful feedback received is. The aim of this study is to provide justification for or against the use of this approach to feedback based on both quantitative and qualitative results.

1.2.1 Research Contributions

This thesis provides the following research contributions:

- Development of a system to support generation and dissemination of tag based feedback for source code assessment.
- Quantitative data on how the participating students interact with the system.
- Qualitative data on whether students perceived any benefit when using the new system.
- Discussion of the system as well as application of different analysis techniques that can be used on the resulting feedback to gather information that may support learning or teaching.
- Answers to the Research Questions outlined in Table 1.1.

1.2.2 Criteria for Success

The provision of satisfactory answers to the 6 Research Questions (RQ), located in Table 1.1, is the fundamental factor that will determine successful completion of the investigation. Detailed information on each research question and how it will be answered can be found in Chapter 4.

The research questions introduced in this chapter can be categorised as either:

- Category 1: Investigating Feedback Tags.
- Category 2: Sharing of Feedback Tags.

RQ	Research Question	Category
RQ1	Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?	Category 1
RQ2	Do students opt-in to share their code and associated feedback?	Category 2
RQ3	Which students tend to opt-in? E.g. weaker or stronger students?	Category 2
RQ4	Do students perceive benefit from having access to other students' code and associated feedback tags?	Category 2
RQ5	Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?	Category 1
RQ6	How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?	Category 1

Table 1.1: Research questions

1.2.3 Category 1: Investigating Feedback Tags

Investigating feedback tags is an important consideration of this thesis, the research questions in Table 1.1 relating to this category aim to investigate how successful tag based feedback is as a mechanism for providing feedback for programming assignments. RQ1 is focused on student perceptions of the feedback they have received. It is important to gauge student perception in order to determine whether this feedback strategy has been useful in some way for their learning.

RQ5 refers to the ability for students or lecturers to be able to gain useful information from the analysis of feedback tags. The additional information will be generated from sentiment analysis and thematic analysis of the feedback tags and may highlight strengths or weaknesses in an individual's or a cohort's learning. This question aims to investigate if any such patterns can be identified. Thematic and sentiment analysis methods are described in Chapter 4.

RQ6 focuses on investigating how well feedback tags can communicate sentiment information from examiners to students.

1.2.4 Category 2: Sharing of Feedback Tags

Research questions under this category aim to investigate the effect of students being able to share feedback and associated source code and whether or not they:

1. opt in to this scheme and use it

2. and whether they perceive a benefit in doing so

The rational for providing sharing features with this type of feedback is that not only are you encouraging students to engage with their own and others' feedback but you are also providing a mechanism for them to increase the amount of feedback they receive in total. Students are also given the opportunity to be exposed to more source code annotated with feedback.

This can provide them with examples of different ways their peers have thought about solving the same problem and examiner feedback on these. In addition, many social networking systems such as Facebook, del.icio.us and Twitter are based on sharing information and are continuing to attract users. The approach investigated by this thesis aims to capitalise on this increased popularity.

1.2.5 Scope of the Research Questions

The utilisation of tags as a form of feedback and the concept of sharing this information between students is part of the core originality presented in this thesis. A difficulty apparent in this research area is that the concepts involved have been seldom discussed in existing literature and as such many research questions could be included. This thesis aims to provide a foundation for future work exploring feedback tags as a mechanism for feedback, particularly in programming courses. The research focuses on investigating whether feedback tags are suitable as a form of feedback for programming students. Determining student perceptions of using feedback tags as a feedback technique will be considered along with evidence of how students react to the prospect of sharing their feedback.

Other interesting research questions which involve quantifying, for example, how much students are able to improve their learning through the use of feedback tags or comparisons of feedback tagging to existing techniques have been excluded from this study. The focus on investigating student perceptions, usage and analysis of the feedback has been selected in order to determine whether further research would be appropriate. Another important question is how the technique can be applied to peer assessment. This is also excluded from this thesis and is included as further work. This is primarily due to the desire to establish whether the technique itself is perceived as being useful before any additional experimental or comparative research is undertaken. It seems logical to evaluate the feedback technique in a simplified situation before investigating its application to more complex scenarios.

1.3 Thesis Outline

The remainder of the thesis is outlined below:

Chapter 2: This chapter discusses the pedagogic issues surrounding the teaching of programming and related literature. It begins by giving an overview of some of the general educational theories that are applicable to teaching programming and continues to outline some of the reasons programming is such a difficult skill for novices to master. Finally it describes some of the current approaches that aim to mitigate these problems.

Chapter 3: This chapter details the various technologies that inspired the design and implementation of the prototype system used to generate results for this investigative thesis. This chapter also discusses the system and the impact it may have on programming feedback as a process.

Chapter 4: This chapter presents the research design and methods used within the investigation. This chapter also describes each research questions motivation and how they will be answered using the described research methods.

Chapter 5 and 6: Due to the iterative nature of this investigation there are two preliminary investigations used to direct the methods employed in the final investigation. These are presented in these chapters and are used to form a justification of the final research approach.

Chapter 7: Reports the results of the final investigation. The results are presented in accordance with the recommendations from the preliminary investigations but with more emphasis on answering the research questions posed in Table 1.1.

Chapter 8: The final chapter presents the conclusions and future work that could be developed as a result of this investigation.

Chapter 2

Pedagogy of Computer Programming

2.1 Introduction

For the purposes of this thesis, the activity of programming is defined as "...the act of assembling a set of symbols representing computational actions" (Kelleher and Pausch, 2005) that can be interpreted by a computer. Whilst the definition is clear, the processes involved in programming are far from simplistic. The task of teaching programming to novices is itself one that is inundated with difficulties (Robins et al., 2003); these difficulties are identified by a variety of sources (Kelleher and Pausch, 2005; Kölling et al., 2003; Kölling, 1999; Lahtinen et al., 2005; Winslow, 1996). This chapter begins by presenting an overview of some of the important educational theories relevant to teaching programming to novices. It then describes why programming is such a difficult topic to teach. Finally, this chapter presents a detailed description of current approaches used to support generation of feedback for programming students.

2.2 Educational Theories

In order to gain a better understanding of learning and teaching programming, a general overview of some of the prevalent educational theories is necessary. This section will outline some of the theories that are particularly applicable to the teaching of novice programmers. There are many theories, based on educational psychology, that contribute to this research area including Behaviourism (Watson, 1997) and Constructivism (Piaget, 1947).

2.2.1 Behaviourism

Behaviourism is a theory based on the psychological notions of conditioning. Proponents of this theory see the human mind as a black box that, when supplied with a stimulus, produces a response which can be quantitatively measured (Watson, 1997). One of the most famous behaviourists is the well known psychologist Pavlov with his classic experiment using dogs. His experiments showed that the dogs could be conditioned to salivate when a bell rang if it was associated with the concept of food, even if no food was presented (Knowles et al., 2005). This demonstrated that behaviour could be changed overtime if the correct stimulus and response mechanism was applied.

This theory, when applied to learning and teaching, suggests that positive and negative consequences could be used to reinforce or discourage learner behaviour. Simple types of reinforcement can occur from praise to expressions of disapproval. Learning activities can even be designed to inherently contain consequences that affect learner behaviour. An example of negative reinforcement may be a situation where a student achieves a good enough mark in an initial assessment and as a result is rewarded by not having to sit the final examination. This uses the possibility of using a negative consequence to encourage students to achieve a better mark in the initial assessment. Behaviourism is a theory that is not often cited as being utilised in the teaching of programming, however historically it has been used as a

fundamental teaching theory.

2.2.2 Constructivism

Constructivism has roots in many disciplines including philosophy, psychology, and cybernetics (von Glasersfeld, 1989b). Constructivists hold the belief that learning is an active process of constructing knowledge (Lefoe, 1998) and that learning requires a direct and active involvement from learners. In constructivism learners are responsible for constructing new knowledge, this is in contrast to historic methods of teaching that involve the teacher broadcasting or delivering course content, in the hope that learners passively absorb knowledge or skills. Constructivism as a theory has many forms which includes radical (Roth, 2000), cognitive (Doolittle, 1999) and social (Roth, 2000) variants. However, despite the differences in perspective all constructivists share the same view which is that learners construct new knowledge and meaning from their experiences. The intricacies of each different constructivist perspective are not relevant to the discussion within this thesis and so have been omitted.

Von Glastersfeld makes the distinction between training and teaching. Training is considered as being synonymous with 'Rote Learning' or 'Surface Learning'. The methods used in training are often repetitive and do not encourage a 'deep' understanding of the relevant concepts. The aim of teaching is to encourage the construction of new concepts. This is central to constructivist theory since it is believed that the construction of new concepts by a learner leads to true understanding of said concepts (von Glasersfeld, 1989a).

This theory can easily be applied to learning how to program. For example, it is often noted that programming cannot be learnt exclusively through lectures or reading a book (Jenkins, 2002). At some stage the learner must attempt to actively engage in using the skills required in programming activities to fully construct the knowledge and processes needed. The theory of constructivism holds a particularly good synergy to learning programming

partly because students must, particularly in Object Orientated languages, construct a mental model of the programming structures available in order to understand and manipulate them.

2.2.3 Problem Based Learning

Programming assignments often lend themselves to practical approaches to learning founded from constructivist ideas, an example of which is Problem Based Learning (PBL). PBL is an experiential learning technique. This means that students derive meaning directly from their experiences of a situation. In PBL, students are set a meaningful problem that is based on a real world situation and are expected to use whatever resources are available in order to solve it. It is the process of solving this problem and reflecting on the practical experiences gained that causes the student to learn. Often in PBL tasks, students work in small collaborative groups in order to learn what they need to know in order to solve the problem.

Another key difference in how PBL is implemented when compared to traditional teaching methods is that the teacher or facilitator "is no longer considered the main repository of knowledge" (Hmelo-Silver, 2004). They are more concerned with facilitating the collaborative element of the learning process. Students involved in PBL are often presented with a very minimal outline of the problem. The onus is on the student to identify what and how information should be used. However, the facilitator can ask a series of open ended questions designed to encourage students or to consolidate the groups' thinking.

Learning how to program is a good candidate for PBL strategies and is one which has been reported on a number of occasions throughout the literature (Ryoo et al., 2008; Delaney et al., 2003). The affinity of programming to PBL is partially due to how applied software engineering often operates in industry. It is often the case where the customer does not know exactly what they need from a software system to actually solve their particular problem, and as a result there is a need for requirements engineering.

It is only through experiencing different situations, and relating new experiences to past ones, that learners gain new knowledge (Duffy and Cunningham, 1996). Therefore, adoption of a constructivist teaching method implies that teaching should be learner-centred and goal directed (Sun and Williams, 2004) with the aim of using 'real world' problems to help students facilitate knowledge construction. Using 'real world' problems in order to stimulate learning is a logical activity for those who agree with constructivist theory, especially when considering the belief that learning only occurs when individuals interact and get feedback from their environment. It is the feedback from the aforementioned interaction that may cause a perturbation or a conceptual change within the learner and it is this conceptual change that results in the construction of new knowledge (Piaget, 1947; von Glasersfeld, 1989a).

2.2.4 Blooms Taxonomy of Learning

A commonly cited theory within educational literature is Bloom's taxonomy of learning (Bloom et al., 1956). His taxonomy describes six of the major categories of cognitive learning and is often visualised as a pyramid with the more difficult cognitive skills being at the top. The taxonomy introduced the following high level cognitive skills, these were broken down into a number of sub categories, however these have been omitted for brevity.

1. **Knowledge** - The lowest in the cognitive skills. This represents remembering facts, figures and being able to engage in simple recall of material.

2. **Comprehension** - This skill category involves students being able to understand the meaning of material or problems.

3. **Application** - This skill is demonstrated when a student uses a technique or concept in a new or unrelated situation.

4. **Analysis** - Students engaging in this skill are able to separate concepts or problems into separate conceptual entities so that the overall structure can be understood.

5. **Synthesis** - This skill relates to the creation of structures or patterns from a variety of conceptual sources.

6. **Evaluation** - This skill is the highest in the taxonomy reflecting its difficulty. Students are at this level expected to make judgements about the value of ideas or material using a reasoned approach.

These cognitive skills are often used as a basis for designing assessment activities to ensure that they correctly examine the appropriate skills for the task at hand. A revised taxonomy was developed by Krathwohl (2002) and is based on the original six cognitive skills but with some modification using ideas developed in modern cognitive psychology. The revised taxonomy renames some of the high level skills and these changes are listed below (Krathwohl, 2002). Once again the subcategories have been omitted for brevity because, for the purposes of this report the high level skills are sufficient for discussion.

1. **Remember** - Recalling knowledge from memory.

2. **Understand** - Being able to determine the semantics or meaning of sources.

3. **Apply** - Carrying out or following a procedure in a given situation.

4. **Analyse** - Dividing material or knowledge into its constituent parts and being able to detect how these interrelate.

5. **Evaluate** - Being able to make judgements about material or ideas whilst following criteria or standards.

6. **Create** - Putting different elements or ideas together to form new knowledge or products.

Another key difference presented in the revised model is that it is a two dimensional model where the so called "Knowledge Dimension" is included along with the cognitive skills listed above. The knowledge dimension allows for Knowledge to be categorised according to type. The types presented are as follows, again these are taken from (Krathwohl, 2002):

1. **Factual Knowledge** - The basic, most low level knowledge which is required for students to be able to solve problems in a given discipline.
2. **Conceptual Knowledge** - This is the knowledge of relationships between factual knowledge. It is this knowledge that allows students to identify which concepts within a domain interrelate and function together.
3. **Procedural Knowledge** - Essentially this is knowledge of how to follow a process, for example using algorithms or scientific methods.
4. **Metacognitive Knowledge** - This is the understanding of how one learns about learning. It is through knowing how one learns that one is able to adapt to new situations or problems.

It is clear that if feedback activities or processes can encourage any development of these cognitive skills whilst serving its primary purpose as feedback then it could be advantageous to learners. One particular example of these high level cognitive skills being stimulated through an assessment and feedback process is that of peer assessment, which allows learners to access the higher level skills such as Evaluate and Analyse.

2.2.5 Approaches to Learning

There are two main approaches to learning, these are often referred to Deep and Surface approaches (Biggs, 1979; Marton and Säljö, 1976b,a; Entwistle, 2001; Heinström, 2000). The concepts of deep and surface learning were first used by Marton and Säljö in 1976 to describe students' approaches to learning.

Marton and Säljö's experiment identified two groups of students. The first group studied with the intention of remembering facts from textual resources, this is often categorised as a surface learning approach. The other group attempted to understand the principal ideas and understand specific concepts of a given text; this is often considered a deep learning approach.

Surface learners learn the text or sign of the material with the aim of being to be able to reproduce it. This often means that they fail to understand the underlying concepts behind the material. This approach to learning often causes students to be trapped into this rote learning strategy (Marton and Säljö, 1976a). Surface learners often focus on what they consider to be a balance between doing the minimum required not to fail and working too hard (Kember et al., 1995). Their learning technique focuses on replication and reproduction, and generally surface learners prefer to limit their reading to be essential material only (Kember et al., 1995). In contrast students using deep approaches to learning will attempt to read widely and gain a deeper more detailed grasp of the subject matter.

Learning the skills involved in programming actually require an approach grounded somewhere in the middle of the two extremes (Jenkins, 2002). Jenkins argues that both rote learning and deep approaches are useful for different aspects of learning to program. For example, learning syntax and language constructs requires a rote learning strategy and learning how to design algorithms or debug code requires a much deeper grasp of programming concepts. This requirement for students to learn the concepts using a deep strategy as well as being able to almost rote learn the syntax could compound the difficulties experienced by students learning to program. This is especially relevant when considering that some students will not be accustomed to using a deep approach and others may not be accustomed to using the surface equivalent (Marton and Säljö, 1976b).

2.2.6 Communities of Practice

Communities of practice are formed by groups of people who hold some shared interests, problems or concerns and can benefit from interacting with one another (Wenger et al., 2002). Communities of practice is an idea based around the concept that social learning occurs between people who have common interests (Wenger, 2000; Roth, 2000) and work together.

The pedagogic benefits of communities of practice are based on the fact that members of the community gain the ability to tap into both common knowledge of the group and the diverse knowledge of each individual. Therefore, members of the community can get help when they need it and the strength of the group can support individuals' learning.

Communities of practice have recently been seen moving from the physical world to virtual ones (Johnson, 2001). Online social networking type environments can help facilitate the communications required for a community of practice to develop. This type of communication is consistent with that found in large scale software development projects. As such, exploration of how these types of community can be fostered during programming courses could yield a benefit to learning or teaching.

2.3 Assessment

Assessment is the process of measuring the extent to which a student or group of students has met the learning outcomes of a particular course. This is often done by a lecturer or teacher critically evaluating students work such as essays, presentations, reports, examination scripts or in this case source code.

There are two main approaches to assessment; these are criterion referenced assessment and norm referenced assessment (Brown, 1997). Criterion referenced assessment is focused on measuring whether or not students have met pre-specified criteria and is used to determine how well a student has performed against a static objective as opposed to in comparison to another student. Norm referenced criteria are designed to permit comparisons between

students and to allow rankings to be generated. This thesis is concerned primarily with criterion referenced assessment as this is the most commonly used in Higher Education in the United Kingdom.

The functions of assessment tasks are often distinguished as being either formative or summative (DES/WO, 1988). This report briefly discusses these different types of assessment and how they relate to learning programming.

2.3.1 Formative and Summative Assessment

The purpose of formative assessment is to allow "positive achievements of students to be recognised" (DES/WO, 1988) and to highlight where improvements can be made. It is useful for giving learners a chance to improve before attempting an assessment that contributes to their final qualification. Formative assessment is designed to take place at regular intervals throughout the course. For an assessment to qualify as being formative the feedback derived from it must contain information that enables students to improve on their performance (Wiliam and Black, 1996). Feedback from formative assessment should be used to highlight problems in students learning so that remedial action can be taken (Harlen and James, 1997). If this feedback is provided immediately before a lecture via, for example, a class questionnaire or online test, it is sometimes known as just-in-time teaching (JiTT) (Novak et al., 1999). Just-in-time teaching refers to a process whereby the lecturer uses formative feedback on how a cohort has understood some element of a course in order to guide or modify the content or pace of the future lectures (Bailey and Forbes, 2005).

Summative assessment is the type of assessment used to measure students' learning so that students as well as stakeholders (e.g. funding bodies, parents and the institution) can record and compare achievement in an objective way. Often summative assessment results are in the form of a grade or percentage and contribute to the students' end qualification results. More often than not summative assessment does not have a significant contribution to learning (Knight, 2002), instead simply acts as a measurement of achievement.

Harlen and James argue that there is often a blur between formative and summative assessment and that there is a definite need to ensure the distinction is maintained. The distinction between formative and summative types of assessment is essentially that of timing and purpose (Harlen and James, 1997). Formative assessment is designed to be regular and to contribute to the students' learning, whereas summative assessment is designed to be a measure or summation of the students' achievement at a certain point. Furthermore, there is a difference in perception for the different types of assessment. There is the perception that formative assessment should be a dialogue between tutor and student (Knight, 2002), where there is an opportunity to clarify and negotiate meanings and concepts to do with the assessed work. In contrast summative assessment represents a judgement, where there is an imbalance of power between the assessor and the assessed (Higgins et al., 2001; Knight, 2002). As a result there is no longer the perception of a dialogue but more of a unidirectional communication from the tutor to the student. Wiliam and Black disagree and suggest that all assessment can in fact serve a summative purpose as long as it leads to interpretable evidence of student performance being generated. It is the additional quality of generating feedback which can be used to improve student performance in some way that makes an assessment capable of serving a formative purpose (Wiliam and Black, 1996).

Unfortunately, the process that has been adopted for assessment has become one that is expensive for both tutors and students (Knight, 2002). Students invest significant time and emotion into their work and tutors are investing ever more time to mark it. Time pressures often encourage surface approaches to learning as it is often quicker to rote learn than it is to develop a deeper understanding of the topic (Knight, 2002). Recent studies suggest that assessment is becoming more and more central to education, in so far as, if you wish to change the way students learn, then changing the methods of assessment is the best way of doing so (Brown, 1997). This is incongruous as the purpose of assessment as measuring learning outcomes. The change in

student learning should originate from the other direction. That is, to change the way students learn you should change the learning outcomes and then the assessment. Assessment should not be used as the primary driver of teaching. It should only be used to generate feedback and to measure whether students have met the learning outcomes.

2.3.2 Peer Assessment

Peer review or peer assessment (Dochy et al., 1999) is a technique familiar to most people within academia. It is the way we encourage good scholarship and expand the human body of knowledge (Gehringer et al., 2006). In a learning environment, peer assessment activities are operated on a compressed scale where each student occupies both the role of an author and a reviewer. The idea here is to increase the amount of feedback circulated between students. It is clear that the amount of feedback that can be delivered by other students is significantly higher than the amount that can be feasibly delivered by the relatively few teaching staff (Gehringer et al., 2006). More benefits derive from peer feedback, of these, one of the most important is that of comprehension. Students, when talking to one another, use familiar vocabulary and are less likely to use language that is not mutually understood, whereas lecturers and academics often use a very specialised vocabulary that can exclude students from understanding the feedback (Carless, 2006). This means that the feedback exchanged from peer assessment is likely to be better comprehended by the students involved (Sitthiworachart and Joy, 2008).

Another benefit of these activities, besides the increased amount of feedback being circulated, is that students are able to access skills that relate to the higher levels of Bloom's revised taxonomy such as analyse and evaluate (Carlson and Berry, 2007; Gehringer et al., 2006). The skills developed in peer review activities include critical analysis, ability to diagnose misconceptions, general evaluation skills and communication of suggestions for improvement (Gehringer et al., 2006), all of which are valuable to student learning.

Whilst peer assessment may seem like the 'silver bullet' of assessment

and feedback for students, there are significant criticisms of it as a technique. One of the most important is that peer review at undergraduate level can be an example of the 'blind leading the blind' (Carlson and Berry, 2007). This suggests that students who have misconceptions relating to the work propagate these misconceptions to other students and therefore damage others' learning. Other criticisms are that students have bias during the peer marking process. They often will be more generous to their colleagues and sometimes do not take the assessment process seriously. Whilst some students accept peer feedback as being valuable some of the more cynical complain that they are 'paying' to be taught by experienced lecturers and want their feedback to come from them. This complaint alludes to the conception that Higher Education is becoming more and more consumer driven (Rowe and Wood, 2007; Dochy and McDowell, 1997).

Within the context of programming, peer assessment fits particularly well. An example of a similar technique being used in industry comes from the agile methods of software development, which utilise the technique of pair programming to increase accuracy of source code developed. This technique involves two programmers sharing one computer and having to negotiate and discuss the source code as it is written. One of the more important benefits of paired programming approaches and peer assessment is that they encourage the programmers to make the source code they write easily comprehensible, particularly as another programmer is going to have to understand the source code and give feedback on it immediately.

Tools to support peer assessment in programming courses have been developed and are sometimes used to assess learning outcomes in a summative way. A majority of these tools permit students to fill out an online proforma sheet for one of their peers. Various mechanisms have been used to ensure that the feedback delivered in a peer assessment situation is fair including taking the standard deviation of particular students marks and putting a summative weighting towards accuracy of peer marking (Sitthiworachart and Joy, 2008). That is, the student marking will be assessed on how appropriate

their marks are. In most usages of peer assessment students are given some form of rubric to support them (Carlson and Berry, 2007).

2.4 Feedback

Arguably the most important aspect of the educational process is the provision and use of feedback. It allows students to get a commentary about their work and enables them to adjust their mental model in the light of the communication received. Any learning activity without some form of associated feedback is essentially useless to the learner (Laurillard, 1993; Haines, 2004). This is because human beings learn through interacting with the external world and getting some sort of feedback from it (Laurillard, 1993). Feedback in Higher Education can be thought of as a dialogue between the examiner and the student (Higgins et al., 2001) whereby the examiner attempts to reinforce or elicit a change in the mental model of the student which results in some improvement to the student's learning.

There are two high level types of feedback called intrinsic and extrinsic (Laurillard, 1993). Intrinsic feedback is the type of feedback received as a "natural consequence of your action" (Laurillard, 1993), for example you know what will happen when you go near a fire. The fact that you have experienced it before and have felt the heat leads to the conclusion that it will burn you. This is intrinsic feedback because it is a natural response to an action. Extrinsic feedback is feedback that is introduced usually outside the situation as a description of the action, for example, receiving comments of approval or disapproval from another person or group of people (Laurillard, 1993).

Feedback can be delivered in forms as simple as a grade or a percentage to as complex as annotations, comments and conversation. It is not uncommon for feedback to be issued on paper, via e-mail or verbally. The differing opinions as to what feedback is constitute another major problem in its delivery. What one person finds useful as feedback may not be as useful to another. There also appears to be a link between a student's learning

approach and the type of feedback they prefer (Rowe and Wood, 2008).

Programming students often receive feedback from automated sources, for example the compiler or development environment will provide limited feedback on whether a program is syntactically correct or not. This is one of the most frequently used methods of feedback for a novice programmer as it is given every time they compile their program. However, feedback delivered via automated approaches tends to focuses on 'low level' concepts such as use of syntax and not the higher level concepts such as overall program design (Butler and Morgan, 2007).

Feedback in programming assignments from teaching staff can come in various formats including: e-mail, social networking, audio recordings (Chapman and Busch, 2009) and written proforma or summary sheets. The primary criticism of feedback given for programming work is that if it is given via a medium that is physically separated from the student's original work, it adds a cognitive load to interpreting it (Sweller, 1994; Plimmer and Mason, 2006). This suggests that feedback issued through annotations and in-line comments is potentially more valuable to students than general comments given in isolation from the student's original work.

2.4.1 What Feedback Do Students Want?

Each student has individual needs and preferences when it comes to learning (Biggs, 2003; Felder and Silverman, 1988; Jackson, 1995; Miller, 2002). There is no exception when considering the feedback process. Each individual prefers their feedback to be delivered in different ways, for example visually, textually or verbally.

Rowe and Wood questioned students in a higher education institution in order to determine what they actually want in terms of feedback on their learning. Of the many suggestions from students some of the more interesting ones include: personalised feedback, feedback that relates their performance to that of their peers, and for feedback to be delivered in alternative ways that are not always in written form. They also highlighted problems concerning

how students understand and interpret feedback received. They suggest that students regularly misinterpret the feedback they receive and this significantly limits its usefulness to aid learning. Finally they suggest that feedback should always relate to the learning objectives being assessed (Rowe and Wood, 2007). This can seem quite obvious since assessment inherently aims to identify whether particular learning objectives have been met. It should then follow that the feedback should provide commentary as to how well the learning objectives have been met with regards to the evidence provided by the assessment.

The problem of feedback misinterpretation means that sometimes students are unable to understand the feedback they receive. A common reason for this misunderstanding derives from the fact that the feedback is delivered in academic discourse to which students have restricted access (Carless, 2006; Weaver, 2006). If a student's only form of feedback is a short, written comment on a proforma sheet that they may or may not be able to understand, then there is a clear need to make feedback more meaningful and accessible.

2.4.2 How Do Students Use Feedback?

For the most part, when asked, students respond saying that they use feedback mainly as a tool to aid revision for final examinations. The underlying principle here is that they identify problems in their learning by reviewing the feedback and address them so that in their final assessment they do not make the same mistakes again (Rowe and Wood, 2007).

Students often indicate that they mainly take mental notes of their feedback and do not take direct corrective action (Orrell, 2006). This may be due to how the feedback is delivered or what form it is delivered in. If feedback is delivered, for example, via a piece of paper that is isolated from the student's original source code submission, it is easy for the student to glance over it briefly and throw it away. Whereas, if it were delivered within the context of their original work, the student would be able to interpret how the feedback relates directly to a given aspect of their work and perhaps use

it more comprehensively. As previously stated, the timeliness of feedback also determines how or indeed if students use the feedback provided (Rowe and Wood, 2007).

One student in the Carless' study (Carless, 2006) suggested that they re-read old assignments and feedback, both to find good aspects and to see how much she improved, thus building her self confidence. This study was carried out in Hong Kong so whether or not this attitude is generalisable across different cultures is unknown. This particular student mentions how feedback helps her confidence. This is a very important aspect of feedback. It should not always be negative; in fact positive feedback can have more effect than negative in changing the way students learn. Positive feedback has also been noted as improving student satisfaction and the general student mood (Rowe and Wood, 2008). As such good use of positive feedback is an important way of improving university ratings (Rowe and Wood, 2008). This so called sentiment of feedback is an important factor in determining how students engage with their feedback.

Some students acknowledge that they sometimes do not collect paper feedback (Winter and Dye, 2004) left for them by examiners. However, they usually cite delays in it becoming available as being the primary reason for non-collection. This highlights how a quick turn-around for the assessment and feedback cycle is critical for student engagement and subsequent improvement to learning.

Not only is there a need to ensure rapid release of feedback but it appears as though the order of feedback release is also important (Black and Wiliam, 1998; Winter and Dye, 2004). Students appear to use their feedback commentary less if it is provided at the same time as their marks or summative grades. Black and Wiliams study demonstrates that the providing normative feedback i.e. the marks, alongside formative feedback can in fact cause a negative effect or even cause the student to ignore the comments (Black and Wiliam, 1998). Therefore, to increase the likelihood of student engagement with feedback, the summative marks should not be released until students have had enough

time to interpret the formative comments given about their work.

2.4.3 Sentiment of Feedback

The sentiment is a measure of how positive, negative or neutral a segment of feedback is, and can be critical in determining how students use the feedback. In educational theory there is a notion of a "feedback sandwich" (Haines, 2004) which aims to balance the feedback given to students in such a way that negative or critical comments are given when surrounded by positive ones. This enables the student to identify both areas which require improvement and those which they have succeeded in. Balancing the sentiment of feedback given to students ensures that they are not presented with exclusively negative comments which is known to lead to disengagement with feedback and could cause it be disregarded entirely, or worse, the students performance could deteriorate (Gee, 1972; Hyland and Hyland, 2001). An example of a study conducted with 11th grade students studying English highlights the case where receiving exclusively negative or critical comments had a detrimental impact on students ability to write essays in future exercises (Gee, 1972). The negative comments essentially demotivated students to the point that their ability to write essays actually deteriorated as a result of the feedback (Gee, 1972).

Manual sentiment analysis of feedback has been discussed in prior educational research, however research attention in this area is limited. Brown and Glover's research on categorisation of feedback includes a category on whether the comments could act to motivate or demotivate students (Brown and Glover, 2006). This has clear links to sentiment analysis, as ultimately the purpose for balancing the sentiment of feedback given to students is to ensure they remain motivated yet are still able to improve.

It is apparent that a majority of students crave positive feedback in addition to the negative (Weaver, 2006). The use of automated sentiment analysis tools can help examiners to monitor the feedback generated before releasing it to students. This can help to make sure the feedback delivered

does not damage a student's confidence by being overly negative. The use of automated sentiment analysis as a means of verifying the sentiment of feedback has been used by the CAFEX2 project (Gillam et al., 2009).

Learning to program is a process that requires not only logic and good problem solving skills but also creativity and finesse. As in Gee's study it could be the case that if a student received exclusively negative feedback that they may loose motivation and consequently their ability or confidence in programming could be negatively affected. This is why, carefully considering the underlying sentiment of feedback delivered to students is important.

2.5 Difficulties in Learning Programming

This section discusses the factors that cause programming to be recognised as such a difficult skill to learn. To put the difficulty into perspective, it often takes at least 10 years for a novice programmer to develop into an expert (Robins et al., 2003; Winslow, 1996). This statement illustrates how challenging it is to learn to program as few professions, save those in the medical field, have this large a learning overhead.

2.5.1 Threshold Concepts of Learning to Program

One explanation as to why learning to program is such a difficult activity to master is that their are a number of, so called, threshold concepts that must be grasped in order for students to progress. A threshold concept was first introduced by Meyer and Land (Meyer and Land, 2003) and is used to describe a concept or idea that is pivotal to effective learning in a particular discipline. Often failure to effectively understand a threshold concept can form a barrier to success for students within a field.

Threshold concepts have a number of common features as outlined by Meyer and Land and these are summarised in the list below. The list was adapted from Meyer and Land (2003).

- **Transformative**, once understood a student can see the subject in a new light. Often this means that their understanding has changed so much that the concept can shift the personal values or the student is able to see situations in the light of the newly constructed knowledge. An example of this could be the concept of Object Orientated Programming (OOP). Once a student understands OOP, they may become accustomed to analysing problems in terms of object interactions.

- **Irreversible**, this simply means that threshold concepts are notions that are difficult to be forgotten by students. They are usually such

important concepts that once a student truly understands them, they make a lasting impression.

- **Integrative**, this refers to the uncovering of a significant synoptic link between two concepts. A threshold concept may therefore expose a previously hidden link.
- **Bounded**, threshold concepts can sometimes be used to mark the boundary between academic disciplines.
- **Troublesome**, a threshold concept is often well known within its discipline as being difficult for students to learn. An example of this is learning to program.

The domain of programming has been identified as having a number of threshold concepts that are a barrier for novices (Drummond and Jamieson, 2005; Eckerdal et al., 2006a). Two particular threshold concepts are commonly cited in programming courses, these are abstraction and object orientation (Eckerdal et al., 2006a).

It is important to discuss what makes OOP such a troublesome set of skills or concepts for novices to learn. The following sections discuss some of the reasons that have been identified as contributing to the difficulties experienced when students are learning to program.

2.5.2 Novelty

The nature of programming as a discipline can make it a popular choice for students. This is because competent programmers tend to exhibit an affinity for solving, often very complex, problems using a logical approach. These skills are often valued highly by potential employers. Additionally, students may see programming as a novel topic, one that they may not have explored before. Some may be attracted with promise of being able to develop creative and useful software systems and others with the hope of lucrative career opportunities. It is possible for students to be naïve to the challenges

that are involved in learning to program and therefore be unprepared for the difficulties they may face.

Dijkstra discusses the concept of 'Radical Novelty' (Dijkstra, 1989) in relation to computer science and computer programming. He explains that teaching material that is radically novel is particularly problematic because students normally learn new topics by relating them to previous knowledge and experiences. This, however, is not possible with a topic such as programming as for many students it is so novel and unfamiliar that it is difficult to make links to existing knowledge (Dijkstra, 1989). The virtual constructs used by computer programmers can sometimes be so alien that linking them to knowledge that is familiar to a novice is challenging. This novelty, while being useful for exciting students is also a possible explanation for the difficulty experienced in the learning and teaching of it.

2.5.3 Many Skills

One central difficulty for novices learning how to program is the fact that programming is not a single skill. It is actually a combination of multiple skills. These skills tend to be such that undergraduates are unlikely to have prior detailed experience of using them together to solve programming problems. Competent programmers are expected to be able to move through the following phases that make up a standard programming work flow - "problem representation, program design, coding and debugging" - (Bishop-Clark, 1995) as well as using a variety of testing methods for program verification. Expecting students to learn how and when to interchange the particular skills involved in the aforementioned phases is not a trivial task.

Compounding this difficulty is that certain learning styles and approaches are more relevant to the different skills required. This means that students must be able to select the best approach for the corresponding phase of the programming activity. See the Theories of Learning as discussed in the Sections 2.2.1 and 2.2.2.

2.5.4 Program Design and Program Comprehension

Software design is a complex activity and is recognised as being challenging for novice programmers at the beginning of their course, and in some cases it has been found to be problematic for graduating students as well (Eckerdal et al., 2006b). This is partially due to the requirement for a deep learning strategy to be employed and the threshold concepts involved in object orientated design e.g. abstraction to be understood. In addition to this, students appear to be unable to successfully decompose problems from a specification into useful programmable objects. This is a crucial skill for the novice programmer to master and explains why providing practice in problem solving techniques is important in programming courses.

Comprehension difficulties arise when students are learning to program. Not only do students need to be taught how to express computational instructions in a formal way that a compiler can recognise, but they also have to be taught to read and comprehend existing programs. It has been proven that just because someone can write a program it does not necessarily imply that they can read or comprehend one and vice versa (Winslow, 1996; Robins et al., 2003). This means that two separate skills must be developed by novices simultaneously, which further compounds the challenges that they face.

Experts in programming tend to be capable of using an as needed strategy of program comprehension (Littman et al., 1987), where they focus on understanding different aspects of the program to the detriment of others (Koenemann and Robertson, 1991). This enables them to comprehend complex systems quicker than novices who attempt to understand the entire program immediately. This can result in students being overwhelmed by the information and subsequently becoming demotivated. Novices also tend to focus on understanding the program domain as opposed to the problem domain. This causes great difficulties when trying to use Object Orientated languages where the entire premise of the language is to enable the programmer to focus on the problem domain (Robins et al., 2003; Wiedenbeck et al., 1999).

2.5.5 Choice of Language

Choice of programming language is also a factor in determining the difficulty novices may face in their learning. Object Orientated (OO) programming languages have been introduced as an alternative to procedural languages, with the hope being that using the notion of objects will make programs easier to read and write. The original concept behind OO languages was that by creating virtual objects that map to aspects of the problem domain the conceptual difficulties inherent in programming would be alleviated. In theory programmers would only have to understand the problem domain in order to understand how the software should work. However, research studies have shown that this is not the case (Robins et al., 2003). Programming in OO languages can in fact be regarded as being more difficult than in standard procedural languages. In fact the activity of mapping objects from the problem domain to the program domain is not trivial for novices at all. Novices can become confused when identifying the objects and sometimes may identify objects that are not useful to solving the specific problem (Robins et al., 2003).

Rist argues that OO languages actually add an overhead to programming because not only do users have to be familiar with procedural programming techniques, but they also have to be experienced in using them to construct conceptual entities in the form of objects (Rist, 1996). This means that OO programmers essentially have to be capable of interchanging their usage of these two different programming paradigms. This is a high level skill that may be difficult for novices to master.

Wiedenbeck et al (1999) suggest that the distributed nature of OO programs combined with the complications in the control flow leads to higher comprehension overheads. This is especially problematic for novices who can struggle to understand program flow (Wiedenbeck et al., 1999) even in simple procedural languages.

2.5.6 Timing and Course Structure

In terms of how programming courses are taught, there are a variety of constraints that can cause difficulties. For example, Computer Science or Software Engineering courses can often be timed so that programming components are in the first year of undergraduate courses (Jenkins, 2002; Joy and Luck, 1996). This means that undergraduates who have not lived away from home or have not had to manage their own finances before are immediately being challenged with course material that is widely recognised as being difficult. The fact that students are often in this so called 'transition phase' of life makes the whole process even more difficult (Jenkins, 2002). The reasons that programming courses are often scheduled in the first year of degree programmes vary between institutions, but certainly it is recognised that this may not be the best time for undergraduates to learn programming.

The way courses tend to be structured leads to very carefully planned ordering of material that enable one skill or concept to build upon another. While this is a positive point for many students, should a student fall behind, it can be particularly difficult for them to recover. Due to time constraints in terms of when a course must end, students are often denied the flexibility to learn at their own pace (Jenkins, 2002). They must keep up with the lecture materials in order to achieve the learning outcomes within the specified time frame. This lack of flexibility will disadvantage some students and, as a result, add to the difficulties students can experience when learning to program.

Since learning to program is such a difficult topic to teach, a variety of competing approaches to teaching it have developed since 2001. The main three are: "imperative-first", "functional-first" and perhaps the most popular at the moment "objects-first". These three primary teaching strategies were described in the ACM Computing Curricula in 2001 (Joint Task Force on Computing Curricula, 2001). This thesis is only concerned with the "objects-first" teaching strategy as it is the one used in the higher education institution where the research was conducted. The "objects-first" approach, as the name suggests, focuses on teaching students the object orientated paradigm, starting

with basic objects and then object interaction and inheritance. The teaching strategy is not the primary focus of this thesis, however interested readers are recommended to read (Cooper et al., 2003) as a good introduction.

2.5.7 Difference in Abilities

A difficulty inherent in many courses at undergraduate level is that students will enter the course having different levels of experience (Jenkins, 2002), this is especially so in programming courses. This makes determining the pace of the course particularly difficult as those who are entirely new to programming may struggle, whereas those who have in depth experience will find the course boring. Finding a balance of difficulty yet keeping the content stimulating for each individual student is a huge challenge.

2.6 Tools to Support Assessment and Feedback

A variety of tools to support programming students have been developed (Deek and McHugh, 1998) from teaching Integrated Development Environments (IDEs) (Kölling et al., 2003; Goldman, 2004; Kelleher and Pausch, 2007) to electronic tutoring systems (Daly and Horgan, 2004). However, this thesis is concerned particularly with tools to support assessment and feedback. This section discusses different software systems designed to support feedback generation.

In programming courses, assessment comes in a variety of forms from examinations where students are expected to design systems or write snippets of code, to perhaps the more common model of students submitting coursework projects for assessment.

Throughout the assessment process for written work, examiners often make notes and annotations directly on the work they are assessing, highlighting aspects that are can be improved by the student. This is often carried out on paper for assignments such as reports or essays. However, delivering feedback for programming work in this manner is more of a challenge due to the verbose nature of printed source code. As such, a number of software tools to support assessment and feedback have been developed to support delivery of feedback via an electronic medium.

There are three general approaches to using technology to handle assessment feedback (Plimmer and Mason, 2006). These are summarised as the following: using software to alter the existing document by insertion of comments, using software to simulate writing in ink over the top of students work or by delivery of a separate document that contains comments related to a piece of work (Plimmer and Mason, 2006).

Software that permits manual annotation of students' work using either free form hand written annotations or typed ones can be considered ink-over feedback systems. These systems essentially simulate traditional approaches of

marking work by enabling staff to write over students work without restriction. These methods offer the most freedom (Plimmer and Mason, 2006) and hence are more likely to be used in practice. The benefits of these systems focus mainly around the improved traceability of the resulting marked work. The work is now electronically captured and can be delivered to students immediately and with less chance of it being lost. The main drawback of this approach is that now not only do the concepts described in the comments have to be understood and interpreted by the students, who may not be familiar to the academic discourse in which the comments are written in, but also the handwriting has to be deciphered. The 'Penmarked' system by Plimmer and Mason attempts to mitigate this by operating a limited form of handwriting recognition. They allow markers to add score details by writing them in electronic ink and then this information is recorded as a figure on the students mark sheet. Sadly full text handwriting recognition is still not reliable within this system without significant training of the software and so the comments cannot be further interpreted by computer. This limits the amount of automated analysis examiners can do on the electronic feedback given.

Another approach to feedback is the system of issuing a separate document containing the feedback. This is by far the weakest of the systems because if references are to be made to the students original work they must be made with a navigational commentary as well (Plimmer and Mason, 2006), for example 'On page 23 paragraph 4 you should ...'. This increases the cognitive load required for the student understand their feedback as they must refer to two documents simultaneously. It is clear that students benefit most from feedback that clearly relates to specific aspects of their work and as such delivering it in an isolated form may not be as useful to their learning.

2.6.1 Automated Assessment Tools

One feature of programming is that it is very easy to check a program for correctness using automated unit testing. Many tools utilise these automated

testing methods to assess program correctness (Benford et al., 1995; Higgins et al., 2002, 2005; Jackson and Usher, 1997). However, these are sometimes unable to test some of the other important features of a software project for example maintainability or comprehensibility, elegance of solution, modularity and so on. Largely these automated systems run a series of tests on the students' code with various input values and compare the output with some model output generated by the examiner; submissions that match tend to get a 'pass' for that assessment criterion. Submissions that do not match are usually flagged to the examiner or marked as a 'fail'.

These automated assessment tools provide various benefits including the ability for feedback to be generated extremely quickly and automatically distributed to students. Furthermore, some assignments may allow students to check their work for correctness for a limited number of times before final submission, allowing students to gauge whether their code is correct or not.

One system is called 'Scheme-Robo' (Saikkonen et al., 2001). This system facilitates automated assessment of small programming exercises written in the Scheme functional programming language. The feedback delivered by this system is largely concerning the correctness of the solution based on runtime and memory constraints placed by the examiner. A major criticism of this system is the rigid nature of the error messages presented to students as feedback. The approach adopted by this system ignores aspects such as style and elegance of the solution.

A different approach to assessing algorithms without using a specific programming language comes from the work highlighted by Malmi et al. They used the system TRAKLA and TRAKLA2 to assess whether students understood the fundamental concepts involved in their algorithms course (Malmi et al., 2005). They specifically focused on teaching and assessing the underlying concepts of specific algorithms without using a specific programming language. The aim was to teach the fundamentals of algorithms so that they can be implemented in any language later in the students' course. They assessed their course by using a number of exercises in the form of

dynamically generated, interactive Java applets (Malmi et al., 2005). These allowed students to simulate algorithms by hand and submit the output to the system for assessment. The success was measured in whether the manually generated outputs matched those of the computerised version. The feedback issued to students was visual and immediately delivered. However, it was still fully automated and limited in that the system is incapable of providing personalised corrective feedback for students.

Some automated tools do attempt to assess aspects of source code that are typically done by humans (Berry and Meekings, 1985). Readability of students' source code is an important aspect of programming, one which often contributes to the students final grade. Automated tools sometimes attempt to assess these aspects by applying sometimes arbitrary selected rules. For example, one may be to check that no method is longer than an arbitrary number of lines or that comments are present for each method (Berry and Meekings, 1985; Hung et al., 1993; Venables and Haywood, 2003). This aims to limit the comprehension difficulties that may be experienced by a human reader. However, there is a problem with this technique, particularly in languages such as Java where it is possible to write an entire program on one line of code since white space has little syntactic meaning. Another problem is that these tools tend not to differentiate between group developed projects and smaller individual projects. It is reasonable to assume that different projects would have different expectations associated with them in terms of applied style (Berry and Meekings, 1985). In some cases a programming standards document may have been used in some aspects of a project which may conflict with stylistic guidelines set by automated assessment tools. Furthermore, it is sometimes unreasonable to impose arbitrary limits for judging readability of source code. In most cases human judgement is required to decide whether source code is readable or not, it is a very subjective process but one that is important none-the-less, as ultimately humans need to read and maintain software systems.

Automated tools can be criticised as being impersonal with a large focus

on reporting correctness or incorrectness of students' work and little regard to providing individual feedback to students on how they can improve or provision of guidelines on the quality of the students work.

2.6.2 Semi-automated Assessment Tools

The use of semi-automated software throughout the marking and feedback process for programming work has become more and more popular. This is mainly due to its improved traceability and its ability to expedite aspects of the entire process (Joy and Luck, 1998). Furthermore, semi-automatic systems often have fewer constraints imposed on the examiners, this makes them a much more flexible alternative to fully automated systems. However, examiners are required to be more involved in the assessment process than in automated systems of assessment.

One of these systems is the BOSS system for electronic assessment of java programming code (Joy et al., 2005). This system operates by running the student's code through pre-specified test cases and automatically assigning marks based on these results. The system does not aim to replace the examiner; on the contrary the examiner is still an integral part of the system as they must judge the quality and style of the work submitted. The automatic assessment is completed largely by using a unit testing software called JUnit, however batch testing has been implemented for the lecturer to use if more complex tests are required. The feedback provided by the BOSS system for students is useful as it can provide an immediate insight as to whether or not their submission passes the test cases. The limiting factor of this approach is that the feedback appears to be delivered as a separate conceptual entity isolated from the student's original work. As highlighted earlier, this could cause a cognitive overhead in students having to map their feedback to specific aspects of their own work.

The 'submit' system (Venables and Haywood, 2003) uses some of the automated comprehension assessment strategies outlined previously but also acknowledges that a human examiner can give additional feedback as a

separate comment online. This allows assessment of the higher level skills used in programming. However, the submit system does again present the human generated comments in isolation from the student's original work. This means that the system also suffers from the cognitive overhead of students having to map tutors' comments to specific aspects of their work.

Baillie-de Byl describes a criteria based system and how it can be used to reduce the overhead inherent in the assessing Java source code. The system supports annotation of code submitted to enable in-line feedback to be issued to the student (Baillie-de Byl, 2004). This type of assessment system is useful for enforcing a structured marking approach. However, with such rigid structure comes an inherent lack of flexibility which may cause problems when it is appropriate to reward students who do extra work or research. As with all electronic feedback dissemination systems, the problem of late delivery of feedback is mitigated by enabling instant transmission of feedback to the students.

The Environment for Learning to Program or ELP system enables delivery of feedback in the form of a dynamic discussion that appears annotated within students' programming work (Bancroft and Roe, 2006). This system is a particularly good example of how to provide feedback on programming work that is traceable and is based around the student's originally submitted work. Preserving the context of the feedback by storing it as a discussion overlaid on top of the student's original submission reduces the cognitive overhead and allows the student to see exactly what aspect of their work is being discussed. The ELP system has demonstrated the positive impact of providing in context feedback for programming work.

A more detailed review of both automated and semi automated assessment and feedback systems can be found in the technical report (Cummins, 2008) written in advance of this research.

2.7 Chapter Overview

This chapter has provided a high level overview of the literature surrounding educational theories and how some of these can be applied to the scenario of teaching novices to program.

In this summary of the literature, a set of technical solutions that aim to support delivery of assessment feedback to students have been discussed. However, for most of these, the ability for examiners to analyse and identify the aspects of programming that students need support with is constrained in some way. Providing feedback within the context of a student's originally assessed work has been identified as being important in increasing the student's ability to comprehend the feedback and see its relationship to their own work.

There is a lack of high level analysis possible with existing manual feedback delivery approaches. That is, it is difficult to amalgamate the information to get a bigger picture of how students are performing in their programming work from using the text feedback provided. This additional analysis capability may be able to support both lecturers and students in directing their teaching or learning.

Recognising the importance of assessment and the feedback generated from it, it is critical to ensure that students have the best possible chance of success in learning the skills involved in becoming a successful programmer. It is the recognised importance of feedback that has led to this thesis focusing on investigating a novel way of formatting, delivering and analysing programming feedback.

This chapter has also highlighted the importance of the sentiment of feedback delivered to students. The fact that exclusively negative feedback can actually damage students' future performance means that it is important to ensure the feedback from examiners is interpreted as it was intended by students receiving it.

The next chapter discusses two competing information management theories and introduces a prototype feedback system that aims to provide an alternative to some of the semi-automated assessment strategies discussed.

Chapter 3

Technology: Information Management and Feedback

3.1 Introduction

This chapter reflects on existing technologies that support information management and how they have been used to direct the investigation presented within this thesis. Towards the end of the chapter, a new software system is introduced. It is this prototype system that implements the new approach to feedback and the results presented in the subsequent chapters are generated from the usage of it.

This chapter will discuss two competing knowledge management approaches; that of the semantic web, with its controlled vocabularies, versus collaborative tagging (Macgregor and McCulloch, 2006), an approach often used in Web 2.0 systems. Knowledge management is a very abstract and diverse field of research. Its links with education and informatics are among the reasons it is discussed within this thesis.

3.2 Evaluation of Approaches to Information Management

Feedback, in a simple sense, can be considered as being metadata for a particular student submission. As a result, a review of current approaches to information management is included in this section, and is used to direct the development of a prototype feedback approach.

There are effectively two approaches to information organisation that this thesis considers. These are controlled vocabularies (formal ontologies) and uncontrolled ones (community tagging systems). This section discusses and compares these two information management approaches.

3.2.1 The Semantic Web and Controlled Vocabularies

The 'Semantic Web' is a concept that was first described by Tim Berners-Lee as a stage in the evolution of the World Wide Web. The semantic web is described as a "web of data with meaning in the sense that a computer program can learn enough about what the data means..." (Berners-Lee, 1999) in order to process it intelligently. In his recent publications Berners-Lee describes 'The Semantic Web...' as his vision for the future of the World Wide Web. He describes a time when humans can simply ask a question in natural language and the semantic web would give a natural language response based on all the information available. Subsequent visions for the future of the World Wide Web have been discussed, for example, that of the Web of Active Knowledge (Geldart et al., 2008). The semantic web is the most commonly cited modern knowledge management approach.

The concept of an ontology is core to the semantic web. An ontology is defined as an "explicit specification of a conceptualization" (Gruber, 1993). In other words, an ontology can be seen as a collection of terms, attributes and relationships (McGuinness and Van Harmelen, 2004) within a specific domain. Not only can the use of ontologies facilitate searching of terms (or concepts) but it can allow artificial agents to make inferences using the

relational metadata. Technical languages such as the Web Ontology Language (OWL) (Horrocks et al., 2003; McGuinness and Van Harmelen, 2004) and Resource Description Framework (RDF) (Brickley and Guha, 2004) have been developed to facilitate the creation and definition of ontologies that can be interpreted by computers. Some ontologies may place restrictions on who can make changes to them, such as adding entries, making edits and removing entries. An ontology that has one or more of these restrictions can be considered as being a controlled vocabulary. The creation or management of an ontology requires significant expertise and training. This in itself restricts the ability of the general user to contribute to the management of ontologies.

A controlled vocabulary refers to any knowledge management approach where the control of how resources are classified and annotated is held by an individual or an organisation and not by the information consumers. This idea of a small number of people performing the expensive task of resource annotation and management for the benefit of a larger user base is a common practice in libraries and other information repositories.

There are substantial benefits of using controlled vocabularies, including the fact that the vocabulary being used does not include slang, non-standardised metadata or unreliable sources. It is also more likely that resource metadata held within a controlled system will be complete and to a specified standard. Additionally, language features can be considered and handled in order to improve the quality and consistency of the metadata. For example, controlled vocabularies can implement standards such as stemming (removal of plurals) or adding clarification to unusual language features. One example of an unusual language feature is a homonym: a word that has the same spelling and sometimes the same pronunciation but a different meaning. These language phenomena and relationship to information management is discussed in greater detail by Cummins (2008).

The primary criticism of formal classification techniques is that they are too restrictive in terms of who can contribute to resource annotation. The inherent "drawbacks not only limit the amount and quality of ontological

metadata created but also who can be involved in its creation and therefore the overall usefulness of the approach." (Bateman et al., 2006) It is a combination of the general restrictive nature of controlled vocabularies and the over flexibility of uncontrolled ones that has led to a variety of hybrid approaches (Tijerino et al., 2006; Spyns et al., 2006; Passant, 2007; Specia and Motta, 2007; Geldart and Cummins, 2008), where controlled vocabularies have been merged with ideas found in Web 2.0 style tagging systems. Many of these hybrid approaches require the user to not only tag resources, but to position them in an ontology as well. Since hybrid approaches usually require application of both paradigms and ultimately increase the workload of the user; the desirability of these approaches is reduced.

A secondary criticism is from the fact that people do not always share the same vocabulary (Carless, 2006). An example is with students and lecturers. It is clear that a student may not be familiar with the academic discourse that a lecturer will be accustomed to and as such they will use different vocabulary. This can become a barrier to understanding and communication. Controlled vocabularies are often managed by experts in the field and as such resources will be annotated from the perspective of the expert and not necessarily the information consumers.

3.2.2 Web 2.0 Applications

Web 2.0 is a popular term that refers to a collection of internet based technologies that facilitate dynamic user-generated content. Some examples of this type of technology can be found with the likes of wikis, Weblogs (Blogs), internet forums and social networking platforms such as Facebook and MySpace. These technologies and platforms allow users to create dynamic and persistent content in a communal environment without the need for explicit training. A key philosophy that many Web 2.0 systems tend to adopt is that data management and organisation is done democratically without a need for a central controlling authority.

This model not only decreases the cost of having to employ controlling

bodies or librarians but in the case of Wikipedia, the online community based encyclopaedia, it has demonstrated that for the most part the knowledge of the community can be managed to a relatively high standard democratically. However, as it is well known that the information stored in these environments is susceptible to abuse and delivering misinformation or low quality content.

The focus of this thesis is on feedback and as such, information management aspects of Web 2.0 systems are of particular interest. The following section discusses some of these approaches to communal information management.

3.2.2.1 Collaborative Tagging & Folksonomies

The concept of community organised information or resource tagging is one that is becoming more and more ubiquitous on the internet. For example, even some online shops such as Amazon.com have allowed customer generated tags (or keywords) to be attached to the descriptions of products sold.

A tag is often a short fragment of human readable text, which acts as a form of searchable metadata when it is attached to a resource. Tags can often be considered as being keywords that describe a resource. A key distinction between a tag and traditional notions of metadata is the fact that there are no formal restrictions on the format of tags. As a result, tagging resources is a particularly easy and flexible process that requires almost no training or instruction (Gruber, 2007). This means that tagging is accessible to, and is often performed by, the user community and not a central authority. Therefore, there is little need for 'power users' who traditionally would be tasked with policing or dictating the structure of the information. It is the end-users, as a community, who coordinate the cataloguing, ordering and most other tasks involved in the management of the information resources.

From an individual user's perspective, a tag can be seen as a personal marker to enable the user to relocate information previously found (von Glasersfeld, 1989a). In terms of resource discovery, tags allow other users to locate new possibly related information by searching for the tag.

One particular system of collective tagging is called a Folksonomy. Folk-

sonomies can be thought of as systems of information organisation where user-generated keywords are used to describe the meaning of a particular document (Al-Khalifa and Davis, 2007). The word Folksonomy is a portmanteau of the word folk, meaning people, and taxonomy, meaning a system of classification. The word was originally coined by the information architect Thomas Vander Wal in 2004 (Al-Khalifa and Davis, 2006). A folksonomy can be seen as a system of collective tagging involving three conceptual entities: users, resources and tags. The users of a folksonomy contribute by tagging or annotating resources and sharing this annotation data within the folksonomy. They typically do this primarily for their own personal benefit, as a means of bookmarking interesting resources. However, this tagging data is often shared so that it can be used to support discovery of interesting resources by other users. This process is in slight contrast to collaborative tagging systems, where users are all working towards a common goal (Vander Wal, 2007) of generating meaningful metadata for information resources.

There are two distinct types of folksonomy as shown in Figure 3.1: a broad folksonomy and a narrow folksonomy. A broad folksonomy is characterised as having multiple users who tag resources with their own tags that are meaningful to the individual user (Vander Wal, 2005). An example of a broad folksonomy is http://del.icio.us; the social bookmarking website. Here many users can tag the same URL with the same or different tags. There are often multiple instances of the same tag being attached to the same URL resource.

A narrow folksonomy, in contrast, is one that has fewer people involved in the tagging process and more people involved in searching through the tagged resources (Vander Wal, 2005). In these folksonomies, tagged resources generally have only one instance of each tag applied to them. An example of one such narrow folksonomy is Flickr; the photograph tagging system. In this case the primary purpose of the tags is to help other people locate images of interest. In Flickr, users tag uploaded photographs to enable other users to locate them. This is a particularly useful application of a folksonomy because information resources in the form of images do not inherently have a

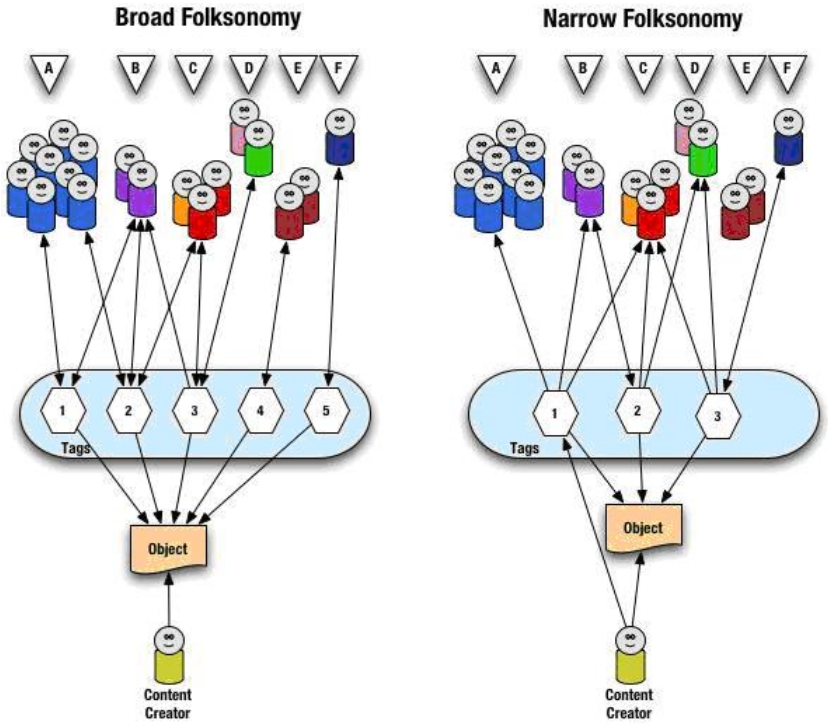

Figure 3.1: Diagram depicting the two types of folksonomy (Vander Wal, 2005)

meaning that can be extracted automatically by computer. The subject of an image, when described in text form, is useful in supporting comprehensive media searching. Currently, the most effective approach to extracting textual descriptions of images is by asking people. However, in a broad folksonomy such as del.icio.us, users tag resources primarily to help themselves relocate them at a later date, it is by coincidence that it benefits the entire user community in terms of resource discovery.

Tag based systems of organisation are becoming more and more popular, partially due to the growing amount of information that is available via the internet and the resulting need to organise this information in a flexible ad-hoc way.

While this notion of tagging systems may seem chaotic in comparison with controlled vocabularies, there are useful techniques that provide users

Figure 3.2: Example of a tag cloud produced using the Wordle.net generator

with additional search (Hotho et al., 2006), recommendation and analysis capabilities. Of the techniques available, the most common is frequency analysis. Inevitably, in a tagging system, more than one user will tag a resource with the same tag. On these occasions, tags act as a vote and therefore the more users who tag a resource with the same thing, the more meaningful that tag can be considered for the particular resource. This situation is where the concept of tag visualisations (Dubinko et al., 2006) such as tag clouds becomes useful. A tag cloud is a weighted list of tags (Sinclair and Cardew-Hall, 2008) in which the frequency of tags allocated to a particular resource is represented by changes in the font size of the text. In a tag cloud, the most frequently occurring tags appear in a larger fonts or in a more vibrant colour, as shown in Figure 3.2.

The other type of analysis, known as co-occurrence analysis, can become very useful in allowing tagging systems to cope with unusual language features like homonyms. These language features would cause ambiguous tags in collaborative tagging systems (Au Yeung et al., 2007). An example of an ambiguous tag could occur when there exists an article about fishing tagged

with 'Bass' and an article about a musical instrument tagged with the same word but with a different meaning intended. Co-occurrence of tags can help in discovering the difference by considering the other tags a resource may have. For example, the article about a fish may also be tagged with 'fish', immediately giving the reader the context they need to decide whether they are interested in it. Co-occurrence is a useful analysis tool especially for clustering of tags to form visualisations or logical models of information. Distinguishing between ambiguous tags like 'Bass' can also be done by visualising the relevant resources as a tripartite graph (Au Yeung et al., 2007) and analysing the clusters that form.

Co-occurrence analysis can be used for the purpose of disambiguation. However, it is also possible to use folksonomies as recommendation systems or even as a means of connecting users with similar tag-resource associations. This can all be done by calculating similarities or differences between users' tags and resources. Analysing user profiles can help to identify or be used to create communities of practice (Diederich and Iofciu, 2006).

The primary benefits of tagging systems are their flexibility and ease of use. Additionally, users as consumers of information are empowered to annotate resources with tags that are most meaningful to them. This means that folksonomies can take advantage of the vocabulary of its entire user base instead of a small subset of users, as often is the case in a controlled vocabulary.

With this high degree of flexibility comes a price and the major disadvantage of tagging systems. Since users are free to add any tag that they desire to any resource, some users can apply overly personalised tags that only have meaning to the individual. Examples of these are 'toread' or 'me', which may be useful for personal information management but will not be for the rest of the user base. This metadata noise can affect searching and be tedious for general user as they try to navigate the tag space. There have been attempts to mitigate this by using techniques such as non-axiomatic logic as described in (Geldart and Cummins, 2008) and the FolksAnnotation system described

by Al-Khalifa et al (2007).

Some users adopt tagging conventions which try to embed a primitive hierarchy into their personal tags. For example "Programming/C++, Programming/Java, Programming/XHTML" (Guy and Tonkin, 2006). These kinds of tags endeavour to create a pseudo hierarchy (Guy and Tonkin, 2006) which allows users to bring some form of organisation to their personal tags. This is very useful for the individual again but probably would not be very useful for those not utilising the same tagging conventions. In fact sometimes these pseudo hierarchies can contribute to the problem of metadata noise for other users.

An additional limitation of simple tag based systems is that often no formal process of data sanitation occurs, meaning that stemming or merging of similar metadata is not always a formal part of the annotation procedure. This means if for example a user annotates some resource with the tag "horse" and another uses the term "horses" one search term may return different results to the other. However, despite this disadvantage, the flexibility of tag based systems and the reduced need for user training may in certain circumstances outweigh the disadvantages discussed.

3.2.2.2 Assessment 2.0

The popularity and usefulness of Web 2.0 systems has been recognised within the domain of education and in particular assessment. This has led to new methods of assessment, sometimes referred to as Assessment 2.0, which involving the use of Web 2.0 tools for collecting the evidence required to measure learning outcomes. This thesis aims to contribute to Assessment 2.0 as a group of techniques by investigating a novel approach to feedback generation, in particular for assessment of programming source code.

Existing techniques described within Assessment 2.0 are not directly relevant to this thesis as none are focused specifically on feedback generation; most concentrate on collecting evidence for assessment. Detailed examples of technologies and how they can be used in assessment can be found in recent

literature (Elliott, 2007, 2008; Cummins, 2008).

3.2.3 Hybrid Approaches

Hybrid approaches also exist that attempt to leverage the flexibility of Web 2.0 tagging systems and combine or convert them to a controlled vocabulary or ontology (Bateman et al., 2006; Angeletou et al., 2007; Echarte et al., 2007; Laniado et al., 2007; Van Damme et al., 2007). One approach to combining folksonomy and ontology information management strategies for managing metadata of learning objects is called the CommonFolks system (Bateman et al., 2006). This approach involves requiring users, at the time of tagging, to position the resource in the ontology. CommonFolks uses and extends the ontology and lexical English language database known as WordNet (Miller et al., 2006). By requiring both approaches of metadata creation the authors have ensured that the metadata attached to learning objects can be used effectively both by human users but also by automated systems. The additional relational information captured by ontologies is primarily intended to allow automated systems to perform enhanced analysis on the data captured. This additional information is less necessary for human interpretation since they are often able to infer the relationships codified by the ontology just from viewing the tag metadata and using prior knowledge.

While the approach outlined by Bateman et al provides a clear strategy for using the WordNet ontology to classify learning objects, it is clear that the amount of time and effort involved by the user is far greater than that of an equivalent simple tag based solution. Furthermore, it is unrealistic for users to be able to intelligently position terms accurately within a complex ontology such as WordNet without formal training. By using ontologies in this way there is scope for logical disagreements within the ontology from users' annotations. In CommonFolks these disagreements are either ignored or prevented. Sometimes, these disagreements can be central to a discipline, for example, the different and sometimes conflicting social perspectives in sociology.

3.2.4 Overview

This review of existing approaches to information management has led to the decision to investigate how folksonomy style tagging can be used as a method of feedback to programming students. The ethos of sharing often found in Web 2.0 systems, in addition to the flexibility and scope for detecting interesting patterns in visualisations such as tag clouds has motivated this decision.

It has been decided that controlled vocabularies are too restrictive and should be discounted from this investigation. Ontology based solutions require relatively high amounts of training as well as prior knowledge of how different concepts can interrelate. Ontologies also struggle to cope with representing or resolving disagreements which may occur within the domain of programming feedback especially if the feedback strategy is expanded to include peer review exercises.

Hybrid approaches have also been ignored from this investigation because an overhead would manifest if users were required to add additional metadata to describe each resources position in an ontology. This would over-complicate the feedback process and one of the purposes of this thesis is to investigate the use of a simplified form of feedback.

3.3 The SWATT System

A novel prototype software system was developed in support of this thesis. This system, known as the SWATT system, was used as a tool to facilitate the application of tag based feedback to source code assessments.

The SoftWare Assessment Through Tagging (SWATT) system utilises ideas often seen in Web 2.0 technologies in order to deliver feedback in a novel way to students. The system supports the generation and dissemination of feedback in the form of tags and permits these to be associated with code fragments which can be displayed in-line within the context of a student's original work.

The requirements for the SWATT prototype, based on the research questions under investigation, are summarised as follows.

- The system must facilitate the annotation of student submitted source code using the Eclipse development environment for the feedback tagging process. Examiners must be able to use this feature. Provision should be made that students may be able to use this functionality in a peer assessment situation in future.

- Students must be able to view their own feedback tags both in summary form and with the option of viewing the annotations in the context of their original submission.

- Students should be given the opportunity to share their feedback and associated source code with their peers and in so doing be granted access to all other shared feedback and code.

- Students should be able to search through the shared feedback on the system and view some simple analysis of their feedback, for example tag cloud generation or co-occurrence data.

- The viewing and analysis of feedback should be done in a web based environment.

- The SWATT system must be able to record student usage data.

3.3.1 Design and Implementation

The design and implementation of this prototype is not the focus of this thesis, however a brief outline of how it operates may be useful for the reader in understanding the overall technique.

The SWATT system consists of a series of object orientated PHP web services developed using the CakePHP framework. The CakePHP framework implements the Model View Controller (MVC) pattern of software development. The MVC pattern forces separation of the business logic, the user interface and the controlling functionality and aims to reduce the burden of maintenance by modularising the software. Figure 3.3 illustrates how the MVC design operates for the general case. The SWATT system is comprised of a collection of web services which allows student and staff interaction using a web based front-end. The students' submissions are stored on the file system of the web server whilst the feedback tag and user data is stored in a secure MySQL database.

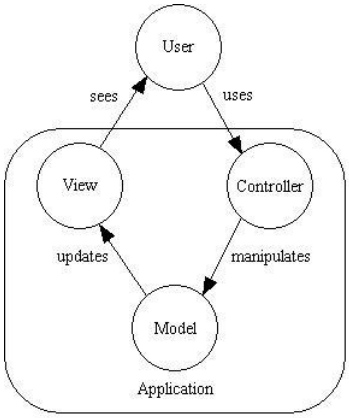

Figure 3.3: MVC explanatory diagram.

In order to annotate the students' original source code, a plugin was developed for the eclipse Integrated Development Environment (IDE) and

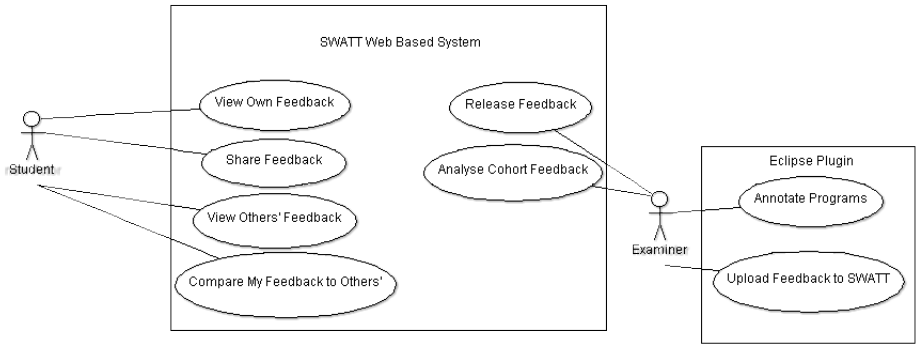

Figure 3.4: Use case diagram of for the SWATT system

facilitates annotation and uploading of the raw feedback data to a web service. The reason an eclipse plugin was developed, was that examiners in the local institution had experience in using the Eclipse IDE. It was, therefore, deemed useful to reduce the learning overhead for examiners by using a familiar extendible IDE as a platform for annotating students' source code submissions.

A high level use case diagram of how different user groups can interact with the SWATT prototype is shown in Figure 3.4. The diagram distinguishes between the aspects of the system different users typically use. An overview of the process for using the SWATT system for examiner-to-student feedback is as follows.

1. Students submit their completed source code to the online SWATT system.

2. Examiners utilise the Eclipse plugin as a means of downloading, annotating and uploading the annotations for students' software projects.

3. Examiners can make the feedback visible after moderation has occurred. At this moment students can view their feedback online.

4. Students can then opt to share their feedback and associated work anonymously and in so doing are allowed to view the shared feedback of their peers.

3.3.2 Receiving Feedback through SWATT

As soon as the examiners have annotated the student's work and the feedback has been made visible, the student is immediately able to explore their feedback tags in the context of their originally submitted source code. They will be presented with a feedback summary view as shown in Figure 3.5 and can opt to view the tags in line with their original work as shown in Figure 3.6. The feedback summary starts by presenting a high level overview of the student's feedback via a simple tag cloud visualisation. The tag cloud is calculated using the frequency of the tags that have occurred within the student's feedback. The tag cloud can be defined simply as:

$$\begin{aligned} textSize(String) &= k \cdot frequency(String) \\ TagCloud &= \{f(tag) | \, textSize(tag) \geq BaseTagSize\} \end{aligned} \tag{3.1}$$

Students can then elect to view the tags annotated throughout their original source code or try and explore the meaning of their tags by clicking on them. As a student clicks on a particular tag they are presented with the tags profile page. If they have shared their work, the tags profile page will show other occasions where the given tag has been used. Additionally, students are shown a discussion board where participants can discuss the meaning of the tag. A key feature is the fact that other uses of the tag along with associated source code fragments would be presented to those who have opted into the sharing aspects of the system. These tag uses can provide a greater context to the student allowing them to see how the feedback they have received has been applied to the work of their peers.

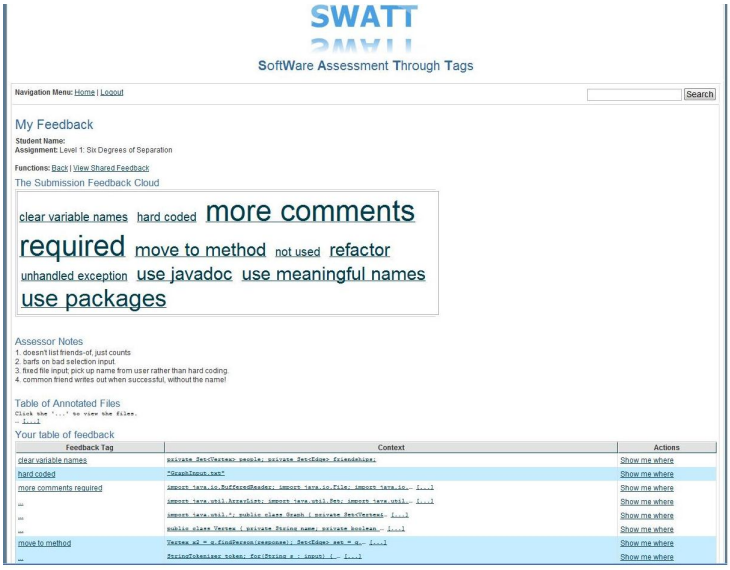

Figure 3.5: Screenshot of the SWATT system showing an individual's feedback

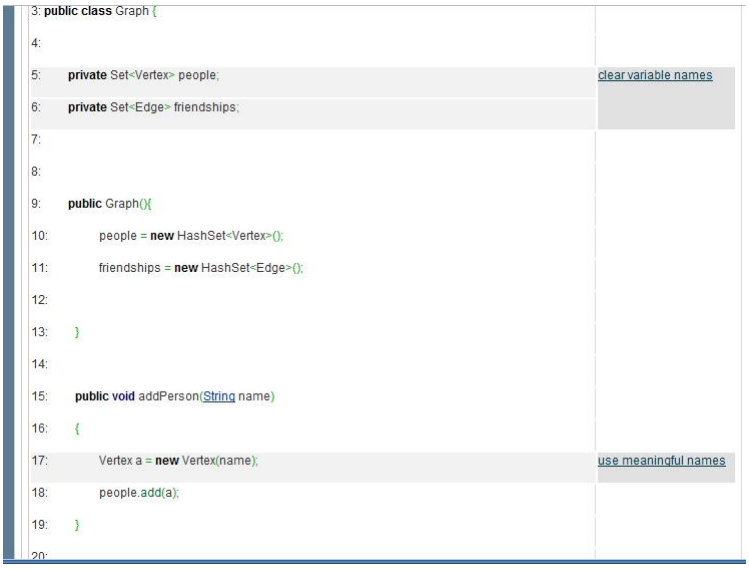

Figure 3.6: Screenshot showing the SWATT system and tags associated inline with source code.

3.3.3 Sharing Feedback with SWATT

Another key aspect of the SWATT feedback approach is that students are capable of, but not obligated to, share their feedback and associated source code in an anonymous way. By default, students' feedback and work are private and can only be viewed by the person who created it. Students who elect to share their feedback and work are rewarded by being given access to all other students' shared feedback. This means that students can compare their feedback to that of their peers as well as gain more information about their own feedback by seeing how similar tags have been applied to the work of their peers.

In order to encourage students to use the sharing functionality of the system, a high level similarity metric was implemented and was provided to all students. This allows students to see how similar their feedback is to all of the other submissions from the cohort. The intention was to increase students' interest and encourage them to opt-in to the sharing functionality to find out exactly how their submission was similar to another students'. The problem with this similarity metric is it is not intelligent and will not detect tags that are similar or related by co-occurrence metrics. Despite this, the similarity metric was used to provide an indication of similarity and for no other purpose. This similarity metric was defined for each student's submission simply as:

$$MySimilarityPercentage = \frac{|(MyTags \cap OtherPersonsTags)|}{|MyTags|}100$$

As soon as a student opts into the sharing aspect of the system, their abilities to interact with the system are less restricted and more information is available to them. As a safeguard to students who may temporarily share their work just to view everyone else's, students are unable to unshare their work and are informed of this in advance. This means that students must commit to sharing their work indefinitely to be given a higher level of access. Amongst other things the higher level of access includes the ability for a student to compare their feedback with one of their peers as shown in Figure

3.7.

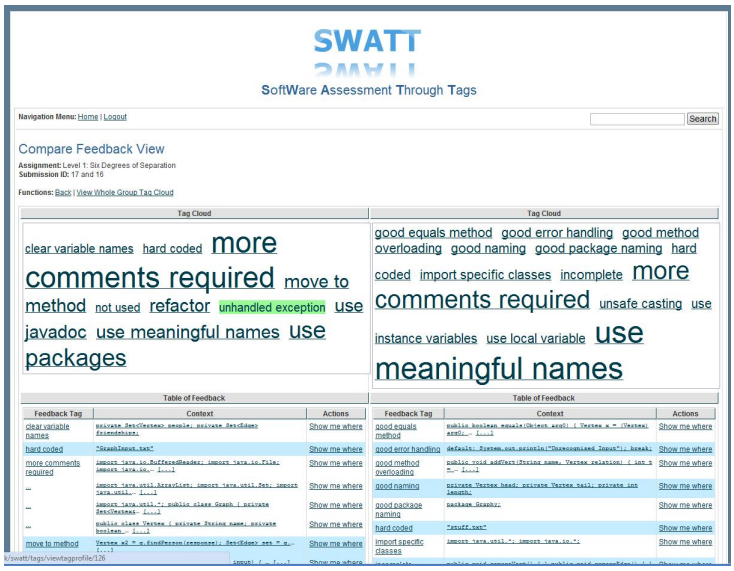

Figure 3.7: Screenshot of the SWATT system where a student is comparing their feedback with one of their peers.

There are a number of anticipated benefits of sharing feedback. The fact that students will be exposed to a significantly larger amount of feedback which, when put into context of their own work, may help them to better understand how to improve. Students may also be able to gain benefit from discussing the meaning of each other's feedback in an anonymous way thus forming a community around the feedback. Another proposed benefit of sharing feedback is the fact that there will be more engagement from students. The aim is that feedback will no longer be a throw away piece of paper but a dynamic and social aspect of learning how to program.

3.3.4 Feedback Tags as Reusable Learning Resources

Tag based feedback when combined with additional information can be considered a type of reusable learning resource. For example, if a feedback tag and source code combination is given additional metadata such as details regarding the intended sentiment of the feedback, it becomes generally more

useful to other users and for analysis purposes. It is important to note that these feedback resources may be highly dependent on their original context. For example, a feedback tag may be associated with a code fragment that only really makes sense in the context of the entire source code file. With the SWATT system this context is preserved and if the feedback had been shared, users can access this wider context to help in feedback comprehension. With additional metadata added to this feedback a more comprehensive learning resource could be formed. For example, automated addition of whether the feedback is positive or negative could provide students an easy way of identifying aspects of good and bad practice in programming. It is suggested that these resources may not be limited to one cohort or even one assignment, the corpus of feedback information will be useful across cohorts as common programming mistakes are highlighted and areas of good practice shared.

3.3.5 Limitations

The SWATT system is based upon collaborative tagging systems such as those described in Section 3.2.2 and as such shares their limitations. Since SWATT generated feedback has no formal restrictions imposed upon the style, size or format of the tags created, there is scope for the problems such as metadata noise and the associated problems with searching through the corpus of feedback tags that are generated. However, due to the small scale and specialist focus of the SWATT system, this is not anticipated as being a problem for this research. In the future, if a large scale version of the SWATT system was used, it is likely that an automated solution (Geldart and Cummins, 2008) might be employed to help mitigate the problems with metadata noise and searching.

The SWATT approach and its current design operate under the premise that examiners are the only users who actually annotate the student submissions. This is currently much like a situation that may be found in a controlled vocabulary. A primary focus of this thesis is to determine whether use of sharable feedback tags is beneficial to either examiners or students and

as such it was decided to simplify the prototype and only allow examiners to perform the annotations. If it is the case that the approach is deemed beneficial then the SWATT system would ideally be extended for peer assessment activities, thus moving away from the controlled vocabulary paradigm and firmly making the system more consistent with Web 2.0 information management approaches.

3.4 Chapter Overview

This chapter has introduced some of the competing high level information management strategies and has summarised the positive and negative aspects of them. The tag based solutions have been used as inspiration for developing the prototype SWATT system which is to be used during the assessment of student developed programming code. One of the key features of the system is the ability of students to view summaries of their feedback as a tag cloud. They are then able to focus in on the lower level feedback tags and code fragments in order to take corrective action. In addition to this, users are given the opportunity to share their feedback and code snippets with their peer group. The idea of sharing information is exploited by many Web 2.0 systems where users contribute such as Wikipedia, the online community driven encyclopaedia.

The SWATT system utilises ideas from the ELP system (Bancroft and Roe, 2006) of embedding feedback within the student's original source code but modifies the approach to be consistent with tag based feedback instead of full textual discussion. The SWATT system improves on existing feedback systems by encouraging students to not only engage and explore their own feedback but also that of their peers. This thesis will evaluate if using the SWATT approach makes feedback more reusable and less likely to be thrown away.

This next chapter will present the research methods that are employed along with the use of the SWATT system to investigate student perceptions of

the tag based feedback approach as well as their engagement or disengagement with the sharing functionality provided by the system.

Chapter 4

Research Methods

4.1 Introduction

This chapter describes the research methods employed to answer the research questions introduced in Table 1.1 of Chapter 1. This section will initially present a general overview some of the research methods central to this study before moving on to describe how they will be used in the different aspects of the investigation.

4.2 Research Methods

The exploratory nature of this thesis, along with the focus on human factors, such as perceived ability to learn, has led to the selection of research methods which are largely qualitative. This is because such research methods are able to collect student opinions and explore aspects of the new feedback approach which the researchers may not have considered or anticipated.

A variety of research methods are required in order to answer the research questions that are central to this thesis. These include, Questionnaires, Focus Groups, Automatically Collected Usage Data and even analysis techniques often found predominately in social sciences disciplines, for example Thematic Analysis (Flick, 2006). The use of multiple qualitative methods enables a

process of triangulation, where the data from one research approach can be used to explain or justify the results of another. Triangulation of research methods provides a higher degree of validity for the results presented.

4.2.1 Rejected Research Methods

A variety of other research methods exist from experiments to participant observations. The research methods included in this study have been selected due to their exploratory function. The research questions used to focus this thesis are investigatory in nature. This means that formal experiments are not necessarily appropriate or useful.

One reason for the exclusion of formal experimental approaches is that there is very little existing research that can be used to help formulate a hypothesis that can be tested in an experimental setting. Furthermore, the difficulties inherent in using control groups would cause ethical concerns. For example, the control group could be at a disadvantage in their learning as a result of not being given access to the same feedback treatment. This, when combined with the associated time constraints, has resulted in the decision to exclude experiments from this thesis. The methods proposed by this thesis intend to provide the necessary information such that a hypothesis could be formed for future experimental research.

Observational methods such as participant and non-participant observations have been discounted; the distributed nature of the research makes this approach unfeasible. Use of the technique is expected to be carried out electronically and independent of geographical location. Furthermore, each individual is expected to use the system in different ways which would be difficult to capture using qualitative analysis derived from researcher perceptions. Instead a non-intrusive approach of automated data collection has been used to capture more quantitative results of how the students used the SWATT system.

4.2.2 Questionnaires

Questionnaires are one of the fundamental methods of gathering qualitative and semi-quantitative results from participants within a research sample. Some of the research questions, in particular RQ1 and RQ4, focus explicitly on investigating human perceptions of feedback. Therefore, questionnaires have been selected as a good method of gauging opinions and perceptions on the topics of interest. More specifically, electronic questionnaires are used to provide a quick and convenient mechanism for users to respond in their own time.

The most commonly cited risk with using questionnaires is that there is a tendency to have a low response rate. The primary reason questionnaires have been selected is that the use of anonymous questionnaires reduces the pressure students feel when compared to an interview situation. Questionnaires also allow students to express themselves anonymously, which reduces the risk that they may feel compelled to respond in a way that they think the researcher wants to hear.

Research into questionnaire design has been extremely influential in the development of the questionnaires used for this thesis. Results from questionnaires are very sensitive to how questions are worded and presented to the participants. In this study, consideration of the neutrality of questions, as well as literature on whether to provide students with middle options in numeric scale questions, has been of significant importance.

There is a debate (Kalton et al., 1980) in questionnaire design that queries the effect of offering participants a middle response in scale questions, for example "Please rate the extent you think X from 1 to 5 ...". One aspect of the debate is whether to provide participants an even number of options or an odd number of options, essentially determining whether students can select a middle or neutral option i.e. 3 in a scale of 1 to 5.

This thesis has adopted the view as proposed by Kalton et al; which states that using a middle option is primarily dependent on what the study is trying to measure. If the study is aiming to measure definitive answers from the

respondents then the full range of the scale should be made available so a neutral option should be given, whereas if the study aims to measure "leanings" then the middle option can be omitted (Kalton et al., 1980). This concept of measuring leanings is useful if you wish people to express an opinion one way or another and you are not concerned with how definitive the answers are.

The open text responses from all questionnaires conducted are analysed by pragmatically grouping related comments into topics and selecting the most relevant for discussion. This process is made especially convenient thanks to the use of an electronic questionnaire system which separated the scale responses from the open responses. This pragmatic approach is expected to be sufficient as the population being sampled is not very large. Should this investigation be applied to a larger population then a more formal thematic analysis process could be applied for the open text responses.

An example questionnaire is included in Appendix A.

4.2.3 Focus Groups

Focus groups are a type of group interview (Morgan, 1988) that allows immediate responses from participants through informal discussion. The benefits of using focus groups are not limited to the content discussed in the meeting, even the interactions between participants may be recorded and analysed. The ability for focus groups to facilitate the identification of "participants' experiences and perspectives"(Morgan, 1988), is ideal with respect to the research questions being investigated in this thesis.

Focus groups were selected over other interview techniques because the use of them enables participants to present their opinions and discuss them with each other. Another benefit is that focus groups can be self contained, meaning that the results from the research can stand on their own without further need for data collection (Morgan, 1988). Focus groups can be used as a means of explaining or reinforcing the results collected from questionnaires. This triangulation of data collection methods can help improve validity of results presented. It is also clear that focus groups are particularly suitable

for highlighting why participants think as they do (Morgan, 1988) which is particularly important given the nature of the research questions.

The focus groups used in this research are exploratory in nature. This means they are used to explore the results collected from the other research methods; primarily the questionnaires and the automatically collected usage data. Since focus groups are used as a means of exploring the results collected from other research methods, there is a limit to what can be planned before the research has commenced.

The number of focus groups run are entirely dependent on the results collected from the other research methods. Should all of the results be explained sufficiently in a single focus group then no further groups are necessary. However, if there are still some questions that require clarifying then further groups may be planned.

Due to the time consuming nature of focus groups and their relatively high cost in terms of participant and researcher time, only the final investigation utilises them. The preliminary investigations outlined in Chapters 5 and 6 focus on the primary research methods.

An important weakness exhibited by the use of focus groups is the fact that the researcher has only a limited control over the subject of the data collected (Morgan, 1988). This is typically because participants have the freedom to discuss high level topics instead of answering specific interview like questions.

The data collected from focus groups is analysed using a similar method as described in Section 4.2.2. That is, the transcripts from the focus groups were analysed by grouping relevant responses into themes and selecting those relevent to discussion. Should a large amount of focus groups be used or those with long durations, a more formal thematic analysis approach would be more applicable.

4.2.4 Automatic Collection of Usage Data

Automatic usage data is being collected in order to gather quantitative data that can be used to reinforce the data collected from the largely qualitative methods employed by this study. The SWATT system has recording elements embedded in the software and the data collected from these can be used to observe how participants engage and interact with tag based feedback.

The usage data gathered is independent of questionnaire data and users cannot be linked due to the anonymity of the two data collection methods. The automatically gathered data, however, can be used to identify general trends or even specific usages of the system. This may reinforce or contradict questionnaire results and give the research a higher degree of validity.

The types of user interactions that may be recorded include:

- Logging into the system
- Viewing one's own feedback
- Sharing of one's own feedback
- Viewing someone else's shared feedback
- Viewing tag profiles for more information
- Recording the viewing of tags

The focus of some of the research questions is on sharing and on how tags are used as a feedback mechanism. Therefore, the automatically collected data focuses primarily on sharing feedback and how students interact with shared feedback. The evaluation of how feedback tags perform as a form of feedback is primarily based on results from questionnaires and other qualitative methods.

4.2.5 Sentiment Analysis

Sentiment analysis is an analysis technique whereby the underlying opinion of text can be mined and its connotation examined. It works by asking the

analyst to make a judgement on whether a particular text is positive, negative or neutral. The sentiment of an extract of text is defined, in this thesis, as a measure of how positive, negative or neutral the underlying phraseology appears to a human reader. Typical short examples include "good work" which implies a positive sentiment and "bad work" a negative sentiment.

Sentiment analysis is a mature field of research, however recently it has received increased research interest and continues to grow (Pang and Lee, 2008). A variety of terminology exists to describe this field of research which spans linguistics, computational sciences and social sciences. Some of these terms include: opinion mining, sentiment analysis, subjectivity analysis, review mining and occasionally automated approaches can be considered a form of affective computing (Pang and Lee, 2008).

A variety of automated sentiment analysis tools exist. One particular system is developed by the National Centre for Text Mining (NaCTeM) (Piao et al., 2009) and is the system used in this thesis. NaCTeM is a blackbox tool and has been selected as, at this time, it is the only freely available system. Automated sentiment analysis is a difficult activity to achieve computationally. This is primarily because interpreting human opinion, which is naturally subjective, is reliant on a number of factors including context, prior knowledge and even how the human is feeling at the time. As a result, this field of research has yet to develop a system that is completely reliable.

The manual process of sentiment analysis adopted within this thesis requires a number of human participants, from both student and examiner groups, to record their perceptions of the sentiment of a sample of feedback tags. These are then compared to determine any similarities or differences in perception. Ultimately, this thesis uses sentiment analysis as a tool to identify how useful feedback tags are at conveying sentiment to different users. However, by combining the sentiment analysis results with thematic analysis, additional information may be identified that could be useful to learning or teaching. The application of sentiment analysis to feedback described forms part of the novel contribution presented in this thesis.

The discussion of how automated sentiment analysis tools operate computationally has been omitted from this thesis as it is not directly relevant. Interested readers are directed elsewhere (Piao et al., 2009; Pang and Lee, 2008; Agarwal, 2005) for technical details on the operation of sentiment analysis tools.

Within commercial environments, sentiment analysis is a powerful tool in determining how well a marketing project is being executed. Using online public discussion systems such as Twitter.com, a company can search for their name or a name of a product and run the comments through an automated sentiment analysis system to gauge public opinion and modify their marketing strategy accordingly.

The pedagogic importance of sentiment in feedback has been highlighted in Section 2.4.3, with studies showing that the wrong balance of positive and negative feedback can actually adversely effect students' future performance (Gee, 1972).

4.2.6 Thematic Analysis

Thematic Analysis (Flick, 2006; Braun and Clarke, 2006; Burn, 2008) is an analytical approach often used in the social sciences to analyse narratives, often in the form of interview transcripts, to identify patterns or trends in the form of themes. This section describes a general application of the analytical approach and the types of results that are likely to be collected.

The initial phase of carrying out thematic analysis is for one or more researchers to review the dataset and derive a set of themes that appear throughout. The derived themes then can be coded within the dataset. After the entire dataset has been encoded by the primary researcher, a series of validation operations take place, which strengthens the validity and reliability of the generated themes. To do this, one or more reviewers must be given the themes and the original dataset. They are then asked to re-encode a sample of the data. The results of this are then compared with the initial researcher's encodings and an agreement rating is calculated.

Normally, prior to beginning the thematic analysis process, an agreement rate is defined. Usually this is at least 80%(Burn, 2008). This is because it is generally accepted, due to the subjective nature of thematic analysis, that some reviewers will disagree to some extent, and this is recognised as being unavoidable. The agreement rate represents confidence on the reliability and repeatability of the study.

Should the reviewers' encodings differ more than the acceptable agreement rate, then an iterative process of negotiation and potentially modification of the theme definitions or names begins. After which, the data would be re-encoded by the researcher with the new themes and rules and the review process repeats until the required agreement rate is achieved.

Among the benefits of thematic analysis is that of providing a general overview of the data, as well as providing the ability for researchers to detect high level trends. This is useful in directing future research and generation of research questions. Particularly in the case of interview transcripts, thematic analysis enables the data to be summarised in fewer more succinct themes.

In terms of feedback, category analysis is not a novel analysis technique. Studies investigating the distribution of feedback into predefined categories (Brown and Glover, 2006) have been done before. Brown and Glover's study presents a generic categorisation framework for feedback that can be used regardless of discipline. The use of thematic analysis in this thesis focuses on the subject-specific themes or categories with the aim being that examiners may gain an insight into how students' feedback tags relate directly to high level programming skills or concepts.

The high level overview of the data provided through thematic analysis complements the focused nature of feedback tags when presented in the form of a tag cloud. This is because tag clouds show reoccurring feedback tags, which may represent very specific items of feedback. Thematic analysis provides an indication as to the reoccurring themes within the feedback. The combination of thematic analysis and standard tag frequency analysis could provide the analyst with a more complete picture of the data.

4.3 Investigation Design

The investigations presented in this thesis employ an iterative experimental process, as shown in Figure 4.1. This means that more than one preliminary investigation is used to derive and focus the experimental methods used in the final investigation. Within this thesis, a number of preliminary investigations, determined by the amount of time available, are used to justify and direct the methods applied in the final investigation. The process and results from these are documented in Chapter 5 and 6.

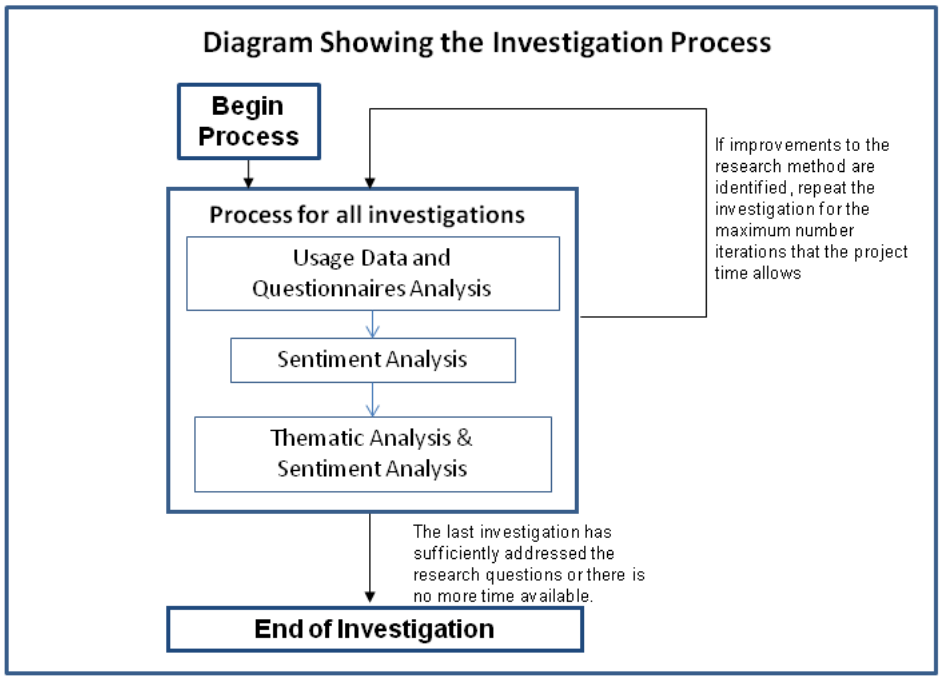

Figure 4.1: Diagram showing the investigation process

4.3.1 Planned Investigation Format

For the series of preliminary investigations, a structured approach is adopted that uses three sub-investigations with each data set to test the research methods and gain some preliminary data. The phases consist of the following.

1. Usage Data and Questionnaire Analysis - Investigate how many students share their work along with students' reasoning for deciding to share, or for abstaining from sharing. This investigation will contain questionnaire results and details of how sharable tag based feedback has been perceived by students.

2. Sentiment Anlaysis - Sentiment analysis is used to generate data that may help direct teaching by highlighting (along with thematic analysis) the topics that a significant amount of negative feedback has been issued. It is also of particular interest to determine whether students and examiner perceptions as to the sentiment of feedback tags agree. This will inform future research on whether additional metadata are required for tag based feedback.

3. Thematic Analysis and Sentiment Analysis - Combined analysis of the data collected from the aforementioned sentiment analysis as well as data collected from thematically analysing the feedback tags. The results of this analysis may help direct remedial teaching by identifying strengths and weaknesses of a cohort in terms of the high level themes identified.

The investigation format may be altered or slightly modified after the preliminary investigations to refine the final investigation.

4.4 Research Questions

In order to ensure each Research Question (RQ) is addressed this thesis summarises how each is answered in terms of which research method is used. The motivation for each research question is included in this section.

4.4.1 Research Question 1

RQ1: Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?

RQ1 is concerned with investigating the effect of applying the new approach in terms of whether students perceive any benefit in receiving feedback in the form of tags for source code assessment. Perceived benefit is defined in this thesis as the following.

- Perception, in this study, is measured in the form of student responses via the media of questionnaires and focus groups.
- Benefit is defined as each student's perceived ability to improve based on the feedback given and their ability to understand the feedback. Further to this, students are asked whether they have received enough feedback, in their opinion, and whether it is of high enough quality. These aspects of the feedback determine how much benefit the new form of feedback has contributed.

In order to answer RQ1, student opinion was gathered using questionnaires and focus groups where possible. In addition to the student responses, sentiment analysis helps to determine the suitability of feedback tags as a method of expressing feedback and whether this expression is conveyed clearly between examiners and students.

This notion of 'perceived benefit' is important for determining student satisfaction, since it is a measure of the students' overall perception of the feedback issued.

4.4.2 Research Question 2

RQ2: Do students opt-in to share their code and associated feedback?

RQ2 looks specifically at whether or not students volunteer to share their feedback and associated source code. These data will be collected by recording mechanisms built into the SWATT system.

This research question is important as it determines how useful the approach is to participating students and gives a clear indication on whether this type of sharing activity is interpreted as being worthwhile by students.

The notion of sharing feedback in this form is a high contributor to the novelty of this thesis. This is why an entire research question is devoted to identifying whether students share their feedback or not.

4.4.3 Research Question 3

RQ3: Which students tend to opt-in e.g. weaker or stronger students?

RQ3 is directly linked to RQ2 as it investigates which of the student population decide to share their work and feedback. This question specifically focuses on whether there is any noticeable link between students' performance in the assessment task and their desire to opt-in to the sharing aspects of the system. This is measured by comparing a student's decision to share their work with their assessment results given as a percentage. This is only possible with summative assessment as formative assessments do not usually generate a quantitative measure of students' performance.

The motivation for inclusion of this research question is to identify whether or not students who share their work tend to be the ones who perform well or otherwise. This will identify whether the system supports the stronger students more than the weaker ones or vice versa.

4.4.4 Research Question 4

RQ4: Do students perceive benefit from having access to other students' code and associated feedback tags?

RQ4 directly links to RQ2 again and considers the sharing aspects of the system. It aims to investigate whether students see any benefit to sharing their work and feedback as well as seeing the feedback of their peers. This was investigated by using questionnaires and focus groups to gather student responses and gauge perception.

The motivation for inclusion is to gauge student opinion of the sharing functionality and to identify the reasoning why students do or do not decide to share their feedback. This RQ aims to provide some explanation for the

results of RQ2 and RQ3.

4.4.5 Research Question 5

RQ5: Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?

RQ5 focuses on two specially selected analysis techniques that can be used with feedback tags. These two analytical techniques have been selected due to how easily the results of one (Sentiment Analysis) may be used along with the other (Thematic Analysis) to infer additional pedagogic information. It is of particular interest to investigate the themes within learning programming that students are struggling with on a particular assignment.

The thematic analysis results may identify high level themes which the feedback tags for a cohort are associated with. This data, when combined with information on the tag's sentiment, can highlight both positive areas and negative ones, which when visualised may be useful for an examiner in modifying their teaching approach to suit the needs of their students. The results on which aspects of the course had most negative feedback, according to sentiment analysis, provide an indication as to the concepts which students may need additional support with.

The motivation for inclusion of RQ5 is explained by the desire to see what added benefits to teaching and learning can be derived as a result of using feedback tags and the analysis techniques that can be applied to them.

4.4.6 Research Question 6

RQ6: How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?

RQ6 endeavours to determine whether feedback communicated in the form of tags can effectively transfer the sentiment information as intended by the examiner when the tag was used/created. This has been deemed as

being of high importance based on existing literature on provision of balanced feedback as discussed in Chapter 2.

4.5 Chapter Overview

A variety of research approaches and data collection mechanisms have been discussed in this chapter, in addition to a more detailed look at each research question. The next chapters describe the preliminary investigations and detail how the research methods discussed in this chapter are applied.

Chapter 5

Preliminary Investigation Using a Group-based Assessment

5.1 Introduction

In order to trial the research methods described in Chapter 4 and to ensure they are suitable for addressing the research questions, a series of exploratory preliminary investigations were carried out. These are presented in this chapter and in Chapter 6. The purpose of these preliminary investigations was to highlight any potential improvements to the research procedure in advance of the final investigation. Each preliminary investigation should yield results which can contribute to answering the core research questions, however the research questions are not fully addressed until the final investigation.

To test the SWATT approach to feedback delivery, it was decided to use the system to give feedback to students who, at the time of development, were working on a year long software engineering group project. This project finished just as the initial prototype of the SWATT system became ready for testing. This made the group project ideal for a dry run investigation and for initial user testing.

5.2 Investigation Context

This preliminary investigation involved participants from a second year undergraduate, software engineering group project module. The source code submitted by the students for this module was assessed using the SWATT approach as outlined in Section 3.3.1.

There were a total of 12 groups which were allocated by the module coordinator for the purpose of the group project. Each of the groups consisted of between 5 and 6 students. In total there were 67 students involved in the group project module, all of which were participants in this investigation.

Two examiners were involved in annotating two files from each of the group projects submitted. The process for selecting the two files for assessment was based on information taken from the configuration management system, Subversion. Subversion stores information regarding the number of changes made to each file. The two files with the greatest number of revisions were selected. This selection mechanism was used simply because it generated a convenient sample for testing the SWATT approach. It is also expected that the most edited files would also be among the most interesting files for marking. The summative assessment was conducted independently of this investigation. Unfortunately, summative marks were generated by the examiners and released prior to the application of the SWATT feedback approach. This may make the feedback generated less likely to be used by students as they have already had their assessment results (Gibbs and Simpson, 2004).

After both examiners independently generated the feedback tags for each submission, the feedback from the two examiners was combined and released to the individual members of each group. Each group's usage of the system was recorded in addition to gathering questionnaire data from individuals.

A total of 100 unique tags were generated for all of the feedback delivered to students for this assignment. Some tags were used more than once in different groups' feedback, a total of 295 tag associations were made which equates to an average of 25 feedback tags per group's feedback.

The data collected from this particular investigation was used in a number of smaller investigations, each one will be described in turn in the following sections.

5.3 Investigating Sharable Feedback Tags

This investigation evaluates student perceptions of tag based feedback. Perceptions are important to evaluate both positive and negative aspects of the SWATT approach.

The secondary emphasis of this investigation was to determine which students and how many opt to share their feedback. Identifying student motivations behind these decisions is also important as they may provide some insight in to how they use feedback given in this form.

The purpose of this investigation is to gather results to support the answering of the following research questions.

- *RQ1 Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?*
- *RQ2 Do students opt-in to share their code and associated feedback?*
- *RQ3 Which students tend to opt-in? E.g. weaker or stronger students?*
- *RQ4 Do students perceive benefit from having access to other students' code and associated feedback tags?*

5.3.1 Investigation Method

The results of this investigation were collected by recording how students interact with the SWATT system once the feedback is released to them. One particular interaction is that of sharing feedback and work. As described in Chapter 3, the act of sharing is permanent and causes an anonymised form of the students work and feedback to become visible to all students who have also shared their work for a given assignment.

After the students have had some time to use the system, a questionnaire is sent out electronically to ask them about their perceptions of the new technique. The students were given two weeks to respond to questionnaires and to use the system before the results were collected and the questionnaire closed.

This investigation utilises the SWATT process of feedback generation as outlined in Section 3.3.1. As the students view and interact with their feedback, the system records these interactions for analysis. This investigation focuses on evaluating the results of these interactions and identifying student perceptions of the SWATT approach. A combination of the aforementioned electronic questionnaires and usage data is used to investigate the effects of students opting in or out of sharing.

The questionnaire specifically asks students to give their perception of the tag based feedback on the following issues:

- Ease of understanding
- Usefulness of in-line feedback
- Ability to improve based on the feedback
- Usefulness of sharing
- Overall quality of the feedback
- Satisfaction over the amount or quantity of the feedback
- Thoughts about tag based feedback in comparison to traditional mechanisms of feedback such as proformas and summary sheets

In this investigation the questionnaires were designed to measure 'leanings' as described in Chapter 4 and the final investigation would focus on gathering more definitive answers. As a result, many of the scale questions are given with a scale of 1-4, with 1 = Very Poor, 2 = Poor, 3 = Good, 4 = Very good and as such do not include a neutral option.

5.3.2 Results

5.3.2.1 System Usage Results

A total of 58% (39/67) of students involved in the project logged in to the system at least once, however the mean number of students to login from each group was 3.25. There were a total of 55 student logins in to the system, indicating some students logged in more than once.

The distribution of student logins and groups is shown in Table 5.1, along with information about the groups final score and whether or not a member of the group opted in to sharing their groups work and feedback. The final score presented represents only the programming aspect of the module and has been verified as part of the module shadow process operated by the university. The groups in Table 5.1 have been anonymised and assigned letters so that the data cannot be linked to student participants.

Group	Group Size	Number of Logins	Group Feedback Shared?	Assignment Score %
A	6	8 from 6 users	No	78
B	6	8 from 5 users	Yes	64
C	6	7 from 4 users	Yes	84
D	5	6 from 3 users	Yes	88
E	6	5 from 4 users	No	92
F	6	4 from 3 users	No	80
G	5	4 from 3 users	No	78
H	6	4 from 3 users	No	75
I	5	3 from 3 users	Yes	75
J	5	3 from 2 users	Yes	92
K	6	2 from 2 users	No	66
L	5	1 from 1 user	No	65

Table 5.1: System usage data by group

A total of 42% (5/12) of groups decided to share their feedback and source code. This represents the groups who had full access to the system's sharing functionality and who were able to view other shared feedback using the SWATT system.

5.3.2.2 Questionnaire Results

The questionnaire response rate was low with only 21% (14/67) students responding. This is possibly due to the release date of the questionnaire coinciding with examination revision time. The questionnaire results are summarised below:

- 71% of respondents reported that the feedback issued was "Very Easy" to understand.
- 50% of respondents stated that the quantity of feedback received was "Very Good" or "Good". However, the remaining respondents reported it as being "Poor".
- 36% reported that the feedback was of a good quality with the remaining reporting it as being either "Poor" or "Very Poor".
- 93% of respondents said that being able to see their feedback tags along side the associated source code was "Very Helpful" or "Helpful" to their learning.
- 86% of respondents said that they thought this approach to feedback would be useful when applied to individual projects.
- 36% of respondents reported that it was useful to see other groups' feedback and associated source code.
- There was divided opinion as to whether the respondents thought they could improve from their feedback. 29% said yes, 29% no and 43% said maybe.

Some respondents explained why they were interested in exploring other groups' feedback and work. "I wanted to see what other groups did wrong compared to us...". This student exhibited an underlying competitive desire to see how other groups had performed in comparison their own. This seemed to be a common reason that groups decided to share.

Another student stated that the feedback was "useful as to see comparison of work, and quality of feedback, plus common pitfalls." This statement suggests a number of things including that students want to see that the quality of feedback is consistent between groups, almost a desire to check up on the examiners. Another issue presented in this comment was that of detecting common errors or pitfalls in the programming assignment. This is important as the student has identified that the system can help students in detecting common errors across the cohort and of the opportunities to learn from them.

5.3.2.3 Investigating Differences Between Sharers and Non-Sharers

The mean assessment score of groups who did not opt in to sharing was 76.29% with a standard deviation of (SD=9.14); the median score is 78%. The mean score for those who did share is slightly higher at 80.60% but does also have a higher standard deviation (SD=11.22); the median score is 84%. This suggests that whilst students who shared their work scored higher on average, there is a slightly higher deviation between the data points suggesting relatively few groups skewed the average.

Using an independent samples t-test it is clear that there is not a statistically significant difference between the marks of those who did and did not share; $t(10)=-0.74$, $p=0.479$. However, due to the small population of interest, the statistical significance may not highlight some of the more subtle patterns in the data. For example, Figure 5.1 shows each group's summative marks and highlights those groups who shared their feedback. Those who opted in to the sharing are bold and positioned below the continuum line. One can see a trend that suggests that a majority of groups who opted not

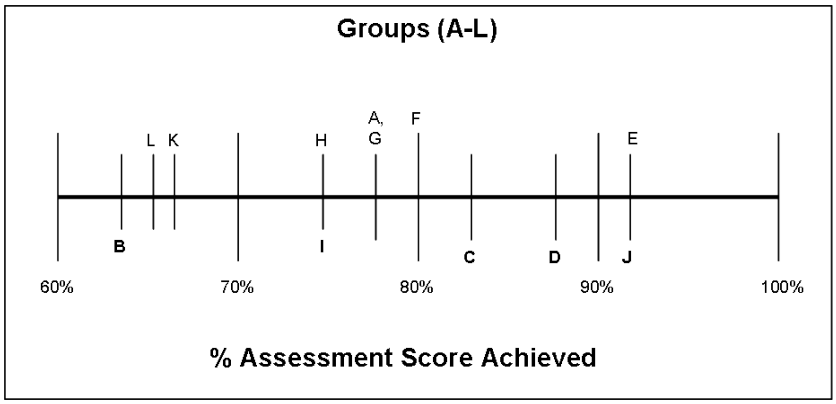

Figure 5.1: Continuum showing distribution of shared work

to share their work appear in the middle of the continuum and those who share are at either end of it. Also, Figure 5.2 confirms that more students who achieved higher marks overall in their project opted in to the sharing scheme. The figures suggest that those groups who have either high or low marks, in comparison to the rest of the cohort, tend to share their work more often than those who receive middle ranged marks.

Independent t-tests were run on the questionnaire data to determine whether there were any significant statistical differences between those groups who reported they had shared their feedback and those who did not. The statistical analysis is hampered by the low return rate for the questionnaires as well as the small population of interest and as such is included purely for completeness.

Table 5.2 shows results of independent sample t-tests for the questionnaire responses of those who opted to share their feedback (S) and those who did not (NS). All of the results analysed were collected from the questionnaire and were answered using a Likert scale of 1 to 4; with 1 being very poor and 4 being very good. Table 5.2 presents how each group S and NS rated the feedback in terms of understandability, perceived quality and perception of how sufficient the quantity of feedback given was.

Table 5.2 indicates that there are no statistically significant differences in

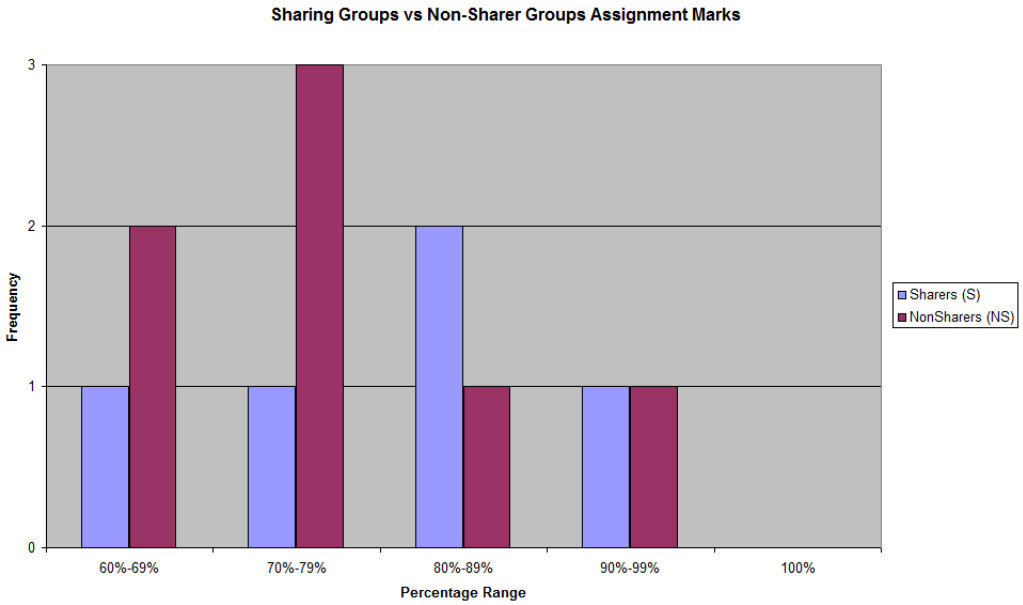

Figure 5.2: Graph showing sharers and non-sharers' assessment marks

terms of students' perceived ability to understand their feedback or perceived quality of the feedback between S and NS. This means that there is no apparent difference between S and NS in terms of how well they thought they understood their feedback. However, it is should be noted that the average score is higher for those who did not share. This statistical analysis of perceptions of feedback quality is consistent with the fact that 50% said the quality of feedback was 'good' or 'very good' and the remaining said it was 'poor' or 'very poor'. However, those who did share tended to have a higher mean score than those who did not.

Table 5.2 shows that there is a significant difference in terms of how satisfied with the amount or quantity of the feedback students received between S and NS. It should be noted that those who shared generally were more satisfied with the amount of feedback they received than those who did not. This could be because those who shared had the opportunity to have been exposed to their peers' feedback and hence felt as though they received more overall.

	Understandability	Quality	Quantity
Sharers (S)			
Mean	2.60	2.20	2.80
SD	0.55	0.84	0.45
Non-Sharers (NS)			
Mean	2.78	1.89	2.00
SD	0.44	0.93	0.87
t-test			
Result	t(12)=0.67	t(12)=-0.62	t(12)=2.28
Significance	n.s.	n.s.	p=0.04

Table 5.2: Table showing statistical tests run on sharers vs non-sharers

It is clear from Table 5.2 that the average scores for most of the questionnaire results are quite low. This could be for a number of reasons, including the fact that students had already received their summative results and could have expressed any dissatisfaction with these in their responses to the SWATT questionnaire. Furthermore, being the first release of the software, a number of requests were made by students to improve the usability of the system. Therefore, the quality of the initial release of the prototype system could have also had a negative impact on the results. It should be noted that the mean scores used in the t-test in some instances have a high standard deviation, for example the test concerning feedback quality had a standard deviation of 0.93, so the mean as a statistical model may be considered as being less representative.

5.3.3 Threats to Validity

The primary threat to validity of this investigation is that of the questionnaire response rate. As the rate was low the results are not necessarily representative of the cohort's perceptions of the system. However, it does provide an

indication of how some students perceived the new feedback technique in a group situation.

Individuals in a group could have discussed their feedback whilst using another individual's SWATT account, thereby skewing the results gathered from the system's usage data.

A possible factor impacting the number of groups who shared their feedback could be that since project groups were formed across friendship groups, two friends in different groups may have shared their group's feedback informally. This will not have been recorded by the system.

Students were given summary feedback and assessment marks for this project before the results were released using the SWATT system. This may have meant that students had less need to engage with the feedback because they had already received it in another form.

5.3.4 Evaluation

A possible explanation of why there were less than half of all groups opting to share their work is that they were not informed clearly that their work would be anonymised. Another reason could be that in a group scenario no individual would want to share the work on behalf of the group without complete consensus. If just one member objected then it would be morally difficult for a group to share the feedback and disregard an individual's feelings.

The questionnaire results that indicate only 36% of respondents found the sharing functionality useful could be misleading for a number of reasons including the fact that only 42% of groups opted to share their work. This means that some of the respondents could have been in groups that did not opt in to the sharing scheme; this would result in their access being restricted, which would result in the respondents being unable to view other groups' work.

Another reason sharing could be undervalued in a team project is that by its nature a group project already provides a means for students to share

their work. To some extent students in a group project can receive feedback from the other members of their team. Therefore, there may be a reduced desire to share work and feedback electronically in group situations since the information they would gain from doing so is already provided from within the group.

The questionnaire results also highlight that 50% of the respondents thought the amount of feedback generated was either "Poor" or "Very Poor", with the remaining thinking it was "Good" or "Very Good". A total of 85% of students, who negatively commented on the quantity of feedback they received, did not elect to share their work. This means that they would not have had access to all of the additional feedback shared by their peers, which may have been useful to them. This difference is reinforced by the statistical analysis presented in Table 5.2.

A factor that limits the usefulness of the feedback sharing feature provided by the SWATT system is that it relies on many groups opting to share their work in order to gain the most benefit, much like many other Web 2.0 systems. If, for example, only two groups opt to share their work then the two groups will only be able to see one other group's feedback and work to compare with their own. Whereas, if all groups shared their work, a diverse and larger dataset becomes available for students to explore.

The most positive result from this investigation is that respondents praised the fact that they could see the comments along side their original code as being particularly useful. With 93% of students reporting it was either 'Very Helpful' or 'Helpful' to their learning.

Students responded to whether they preferred the SWATT approach to other feedback mechanisms. A total of 21% thought that the SWATT system was better and 64% thought that the SWATT approach is useful when combined with traditional approaches. The remaining 14% preferred traditional approaches to SWATT. Within this study, a majority of the respondents from the questionnaire perceived a benefit to the SWATT system but most thought it should be used together with traditional approaches and

not as a replacement.

Students who participated in the questionnaire indicated that they would have found more value in receiving feedback using the SWATT system if the project was an individually assessed one and not a group one. This is a key finding and one which will direct the future investigations presented in this thesis.

5.3.5 Section Overview

This section has presented evidence that can be used to address the research questions presented in Table 5.3. The remaining questions are considered in the subsequent sections.

RQ	Research Question	Considered
RQ1	Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?	✓
RQ2	Do students opt-in to share their code and associated feedback?	✓
RQ3	Which students tend to opt-in? E.g. weaker or stronger students?	✓
RQ4	Do students perceive benefit from having access to other students' code and associated feedback tags?	✓
RQ5	Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?	
RQ6	How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?	

Table 5.3: Research questions considered in section 5.3

Less than half of all groups opted to share their feedback and work electronically using the SWATT system. This may be due to the investigation taking place in a group scenario, since working by teams students are already provided with an environment for sharing feedback. Due to the results of this study, it is recommended that future investigations use individually assessed projects. It is expected that more students will opt to share their feedback in

a project without the group work element. This is based on the notion that within the group, individuals can discuss and explore their feedback, whereas this is perhaps not the case in individual work.

Questionnaire results indicate that half of the respondents were unhappy with the quantity of feedback they received. This may be because a majority (85%) of them had not opted in to the sharing portion of the system. In fact only 1/7 respondents who had shared their feedback was unhappy with the amount of feedback they received.

The findings of this investigation serve as a foundation for further research with the primary conclusion being that the technique is less useful to students when used with a group project.

5.3.5.1 Relevance to Research Questions

This section discusses the findings of this investigation in the context of the research questions.

RQ1: Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?
The preliminary results reported by this investigation, based on the metrics defined by the research question in Section 4.4.1, show that there is a mixed feeling as to how beneficial the approach has been.

The questionnaire results indicate that the students perceived ability to improve using the feedback received from the SWATT system is divided. Only 29% (4/14) categorically said that they could improve based on the feedback they received. The same proportion said that they could not, however a majority of respondents were uncertain. This result indicates that the feedback generated using feedback tags is useful only some of the time, however student responses in this study have not given a definitive answer. Explanations to these types of responses would be sought from focus groups, should the same result appear in the final investigation.

The perceived quality of the feedback from those who responded to the questionnaire was largely poor. Only 36% of respondents were satisfied with

the quality, the remaining students were not. This indicates that the students who responded to the questionnaire thought that the quality of the feedback received was not within their expectations.

With 50% of the respondents reporting that they were satisfied with the amount of feedback received, it is difficult to gauge how successful this technique has been at increasing the amount of feedback students receive. However, it is worth noting that 85% of those who were unsatisfied with the amount of feedback received did not opt-in to the sharing aspects of the system. Therefore, they did not receive access to the feedback of their peers which could have been helpful.

Overall, student perceptions of the feedback, given in the form of tags, was negative in this investigation. However, this could be symptomatic of the fact the system was used for group work. Perhaps it could be that students simply had no need to engage with the feedback when their marks had already been issued in advance.

RQ2: Do students opt-in to share their code and associated feedback? A total of 42% of all groups opted in to the sharing aspects of the system. For this research question, in a group based situation, it is clear that less than half of the groups were interested in exploring the feedback of their peers.

However, those who did share their work did comment on the benefits, for example one student stated in the questionnaire that they shared their work "Because it helps others see what markers are picking up, and it costs nothing to share the knowledge... Only reason not to would be some misguided sense of privacy towards code... In fact, I'd argue you shouldn't get a choice [in sharing]."

Further research is required to answer the question of how students react to the opportunity of sharing individually assessed work.

RQ3: Which students tend to opt-in e.g. weaker or stronger students? This investigation shows that a majority of groups who opted to share their work were at either end of the continuum diagram as shown in Figure 5.1. The early indications from this study show that mainly the strongest and weakest

groups opted in to the sharing feature, with the groups with mid range marks not engaging with the system. Again this could be because the grades were released prior to the feedback tags and groups who were happy with their marks may not have been inclined to investigate the system's sharing features. This would be another useful topic to investigate using focus groups, should a similar result appear in the final investigation.

RQ4: Do students perceive benefit from having access to other students' code and associated feedback tags?

The questionnaire data indicates that all those respondents who opted in to the sharing scheme also found seeing other groups' work and feedback useful. The remaining students did not opt in to the sharing scheme and so would not have been able to see other groups' feedback.

5.4 Sentiment Analysis of Feedback Tags

This analysis has the primary focus of collecting data to be used along with Thematic Analysis to support Learning and Teaching. However, the secondary focus is on determining whether or not tags are an effective mechanism of conveying balanced feedback for students. The importance of sentiment analysis to feedback is discussed in Section 2.4.3.

The research questions to be focused on in this section are as follows:

- *RQ5 Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?*
- *RQ6 How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?*

5.4.1 Investigation Method

In order to investigate the sentiment of feedback when it is delivered in the form of tags, a short study was carried out using the data gathered from the group project introduced in Section 5.2. This study involved recruiting two students, two examiners and using an automated sentiment analysis tool to compare how the feedback was perceived. This preliminary study investigated whether students and examiners agreed or disagreed with regards to the overall sentiment of given samples of feedback tags.

A sample of 45 feedback tags, over one third of all tags, were selected from the global corpus of feedback generated from the group project for sentiment analysis. As shown in Table 5.4, all tags were analysed using the National Centre for Text Mining (NaCTeM) sentiment analysis tool. The human participants' tag samples were limited due to the time consuming nature of manual sentiment analysis and as such only two students (S_1 and S_2) and two examiners (E_1 and E_2) analysed two of three samples (T_1 T_2

T_3) of the 45 tags. The distribution of the tags to participants is depicted in Table 5.4.

	S_1	S_2	E_1	E_2	NaCTeM Tool
T_1	✓	✓	✓	✓	✓
T_2	✓		✓		✓
T_3		✓		✓	✓

Table 5.4: Distribution of tags for analysis

The effects of the distribution shown in Table 5.4 was to ensure that at least the same 15 tags (T_1) were reviewed by every participant and a further 30 tags (T_2 and T_3) were reviewed by at least one examiner and one student. The primary purpose for limiting the amount of feedback tags seen by each human participant was to ensure that issues such as fatigue and time required for post-analysis questioning was reasonable.

5.4.2 Results

The results of this preliminary investigation indicate that there is a relatively high level of agreement between the human participants and the NaCTeM tool. After comparing the results between the participants and the tool an average of 88% agreement was reached. This percentage is calculated using the average agreement percentage for each tag.

Sentiment	Students				Examiners				NaCTeM Tool			
	T_1	T_2	T_3	Mean	T_1	T_2	T_3	Mean	T_1	T_2	T_3	Mean
Positive	10%	7%	14%	10%	7%	7%	14%	9%	13%	13%	7%	11%
Negative	27%	33%	36%	32%	17%	33%	22%	24%	13%	47%	7%	23%
Neutral	63%	60%	50%	58%	77%	60%	64%	67%	73%	40%	86%	66%

Table 5.5: Distribution of tags according to respondent group

The results indicate that only 64% of tags had complete agreement between all respondents in addition to the automated system. This suggests that the

disagreements on the tags usually came from only one of the respondents or the software and not all of the respondents disagreeing with each other.

Table 5.5 shows the percentage tags as they were distributed according to each group of respondents. It is clear from the table that examiners perceived fewer tags as being positive when compared to the students and the automated tool. This is demonstrated also in samples T_1 and T_3 in Figure 5.3 and 5.5. Additionally, examiners report fewer tags as being negative than students on average.

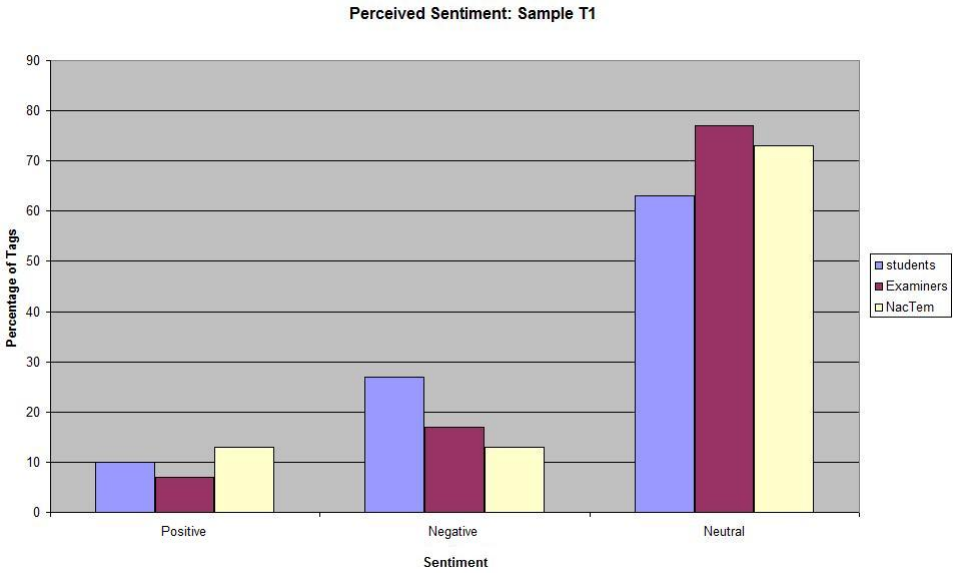

Figure 5.3: Graph showing distribution of perceived sentiment in sample T_1

In Figure 5.4 both human participants were in complete agreement. The automated tool appears to have to associate a positive or negative sentiment with feedback which was largely perceived as being neutral by students and examiners.

In Figure 5.3 and Figure 5.5 it is clear that students perceived more positive and negative tags, whereas Examiners perceived a higher proportion of them as being neutral.

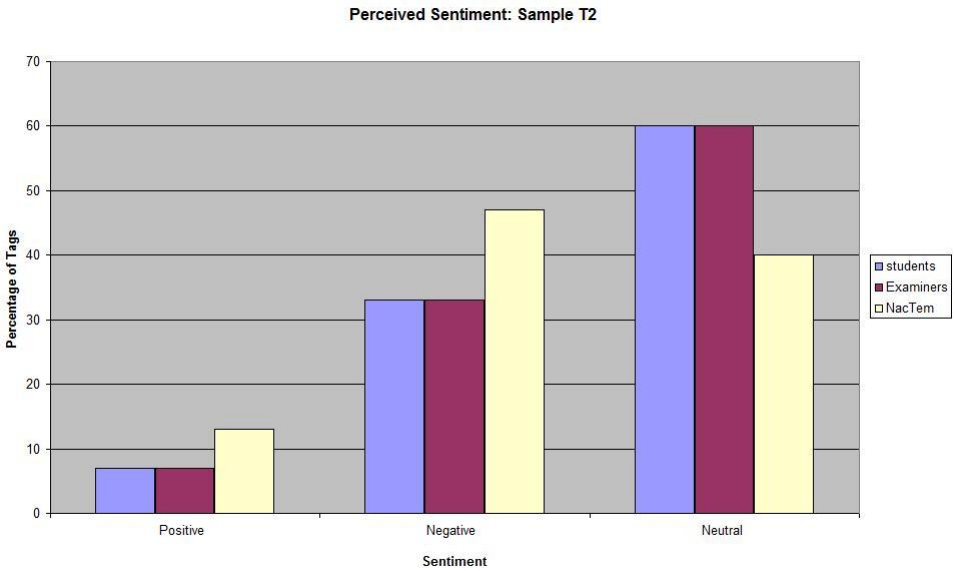

Figure 5.4: Graph showing distribution of perceived sentiment in sample T_2

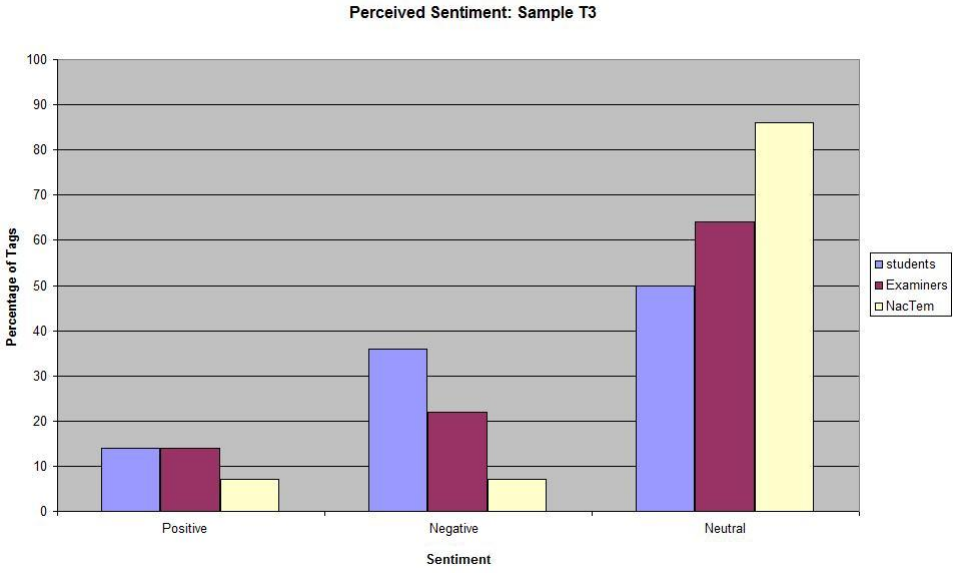

Figure 5.5: Graph showing distribution of perceived sentiment in sample T_3

5.4.3 Threats to Validity

The primary threat to validity in this study is that the number of student and examiner participants involved in the sentiment analysis is small; this is due to the time consuming nature of manual sentiment analysis. Due to the small sample the results presented may not be applicable outside the context of this investigation without further research.

5.4.4 Evaluation

A total of 16 (36%) of tags had some disagreement from students, examiners or the NaCTeM tool. Of these 16 disagreements, 6 were completely due to the automated tool disagreeing with all human participants. These disagreements can therefore be ignored as the the automated system was unable to respond to feedback that contained domain specific vocabulary. For example, "ensure threadsafe" is a tag that was marked as being neutral by all human participants but the NaCTeM tool reported it as being positive. It is important to note that the automated tool was not designed to cope with short domain specific text and was more designed for working with longer, more continuous prose.

Out of the remaining 10 disagreements, 6 were from student participants disagreeing with the examiners and the automated tool. Most of these were instances where students perceived a tag as being either positive or negative and the examiners and NaCTeM reported it as being neutral.

The remaining 4 disagreements were where students and the NaCTeM tool were in disagreement with the examiners. In one of these instances the tag "needs comments" was perceived by examiners as being neutral but students and NaCTeM reported it as having negative connotations.

It should be noted that, in this preliminary investigation, examiners classified fewer tags as being positive compared to the students and the NaCTeM tool; this is particularly noticeable in Figures 5.3 and 5.5. These results indicate that examiners were more likely to identify neutral and negative tags; with students being more likely to identify tags as being positive.

However, students are also more likely to perceive tags that examiners think are neutral as being negative.

T_1 represents the tags that were reviewed by all human participants and therefore is the most representative data set. This dataset confirms the trend that students identify 3% more tags as being positive and 10% more tags being negative, whereas examiners consider a majority of the feedback as being neutral.

Ultimately the results presented indicate that in most cases (T_1 and T_3) students and examiners hold different perceptions of what constitutes as positive, negative or neutral feedback tags. Since only 6/45 tags yielded a disagreement between all human participants and the automated system an approximate rate of error is 13.33%. However, since the tag sample and number of human participants is small, this rate may not be meaningful outside of the context of this investigation.

5.4.5 Section Overview

This section has provided an indication as to the answers to the research questions shown in Table 5.6. The remaining questions are considered in the subsequent sections.

This investigation highlights how different short comments in the form of tags can be interpreted in conflicting ways between students and examiners. These differences become particularly problematic when an examiner is trying to convey a particular sentiment and the students perceive it differently to how it was intended. If a student perceives the sentiment of feedback as being radically different to an examiner, there is a risk of the student disengaging completely from the feedback and it being disregarded.

As a result of this investigation, it is clear that careful consideration as to the sentiment of the feedback tags delivered to students is required to reduce ambiguity in its underlying sentiment.

The ability for feedback tags to communicate sentiment from one user to another when considered outside of their created context is not high.

RQ	Research Question	Considered
RQ1	Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?	
RQ2	Do students opt-in to share their code and associated feedback?	
RQ3	Which students tend to opt-in? E.g. weaker or stronger students?	
RQ4	Do students perceive benefit from having access to other students' code and associated feedback tags?	
RQ5	Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?	✓
RQ6	How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?	✓

Table 5.6: Research questions considered in section 5.4 and 5.5

Therefore, feedback in this form may benefit from additional metadata about the intended sentiment that can be combined to make feedback tags inherently clearer with regards to sentiment. Automated sentiment analysis tools may provide a convenient mechanism for generating this data, especially as only relatively few of the disagreements on sentiment were due entirely to the automated system.

Future investigations could involve more participants or further development of the automated sentiment analysis engine so that it performs better with programming specific terminology.

5.4.5.1 Relevance to Research Questions

This preliminary investigation has provided an indication as to the answers to RQ5.

RQ5: Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching? This investigation has demonstrated how sentiment analysis both automated and manual can be used to determine the sentiment of feedback tags. The pat-

terns highlighted in this investigation include the fact that students perceive tags differently to examiners. This suggests a need for additional metadata that clarifies the intended sentiment to be included with the feedback tags.

The data collected from this investigation could be combined with the thematic analysis data to allow patterns in the sentiment of cohort feedback to be detected.

RQ6: How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?

According to this investigation, it is clear that there are fundamental differences in how students and examiners perceive the sentiment of feedback expressed using tags. This is not to say that feedback delivered in traditional formats has a higher degree of clarity when it comes to communicating sentiment information. In order to determine whether this is the case, additional research would be required which is outside the scope of this thesis.

It is clear that with the help of automated sentiment analysis tools it is possible to provide additional metadata for the feedback to help overcome this limitation to a reasonable level. This investigation has noted there are limitations to using automated approaches. For example, the NaCTeM automated sentiment analysis tool fails to correctly identify the sentiment of tags that refer to higher level concepts or very technical terminology. However, despite this, automated tools may provide a convenient way of detecting the sentiment of feedback if used in a semi-automated process.

5.5 Extending Sentiment Analysis of Feedback Tags: Using Thematic Analysis

In order to extend the information gained from the sentiment analysis run on the feedback tags in Section 5.4, a process of thematic analysis was utilised. This process is introduced and described in Section 4.2.6.

It is anticipated that by combining the data gathered using the two analysis techniques, sentiment and thematic analysis, additional information may be uncovered which could prove useful for lecturers or course directors. This information may help in determining which aspects of the course received high concentrations of positive or negative feedback tags.

This section focuses exclusively on addressing:

- *RQ5 Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?.*

The results will be collected by combining the data from the sentiment analysis in Section 5.4 and thematic analysis collected in this section.

5.5.1 Investigation Method

The process for thematic analysis described in Section 4.2.6 was followed as a guideline and the following specific process was used in this preliminary investigation.

- Derive the initial themes based on software engineering theory and examiners past experiences.
- Code the feedback tags into the themes that fit most appropriately based on the tags' perceived semantics.
- Give a 30% sample of the feedback tags and the themes derived, to examiners for encoding.

- Calculate the reviewers' percentage agreement rate.
- If the agreement rate is less than the 80% agreed threshold (Burn, 2008), then reduce or refine tag themes in negotiation with reviewers and then repeat the blind review and refinement process until agreement rate is achieved.

Two examiners, who were not involved in the original group work assessment process, were recruited as reviewers for the thematic analysis process. Since each feedback tag is to be considered in isolation from its associated source code fragment, the reviewers had no pre-knowledge of each tag's specific context.

The review stages used a blind review technique where the reviewers would follow the same analysis method as the original researcher. The reviewers were tasked with using the themes to attempt to replicate the same tag and theme associations. This review stage is crucial for providing a level of reliability and repeatability for this analysis technique. The agreement rate was calculated based on whether the reviewers' tag to theme allocations matched or agreed with the original researchers'.

The initial thematic analysis used the following themes and definitions which are derived using knowledge from common software engineering theory and past experience of students programming. A majority of the themes were taken from concepts discussed in the course text books (Sommerville, 2004; Pressman, 2004). No reviewers had been consulted about the themes selected prior to the first review phase of the thematic analysis process.

- **Best Practice** - refers to tags that identify areas of good practice or bad practice for programming activities in general. These are not specific to a particular programming language. For example, one might consider that hard coding all user input variables is bad practice irrespective of what programming language it is done in.
- **Completeness** - refers to tags that identify how complete the student's work is in terms of functionality and the task set.

- **Comprehension** - refers to any tags that highlight issues of understanding the source code for example comments, documentation and ease of understanding.
- **Design** - refers to any tag that makes reference to the use of particular design elements or patterns
- **Efficiency** - refers to tags that comment on the computational complexity of code, or code redundancy.
- **Maintainability** - this theme relates to tags that comment on how easy it would be for another competent programmer to extend the software.
- **Object Orientation** - refers to tags that make comment on how well the student's code uses object orientation, e.g. class structure and encapsulation.
- **Testing** - refers to tags relating to students use of test cases or automated testing strategies as identified through their submitted source code.
- **Use of Syntax** - this theme is very similar to Best Practice however it refers to tags that focus on students use of programming language specific features. An example is, if a student did not use the 'synchronised' keyword and a tag said "use synchronised", it would fall in to the use of syntax category as it refers to a java specific language feature. However, if the tag said "make threadsafe" it could be any programming language and would be in the 'Best Practice' theme. The aim of distinguishing the two themes is to determine if the student's feedback relates to Java specific issues or programming in general.
- **Miscellaneous** - refers to tags that have a very specific meaning or do not fit in to any of the other themes.

The results of the first review phase reported only a 66% agreement rate between the reviewers and original researcher. This was partially due to there being too many themes, some of which were overlapping. This caused significant confusion between the reviewers and hence resulted in a low agreement rate.

A refinement and reduction of themes took place which was aimed at improving the quality of the themes. The low agreement in the initial review phase forced the focus on fewer, higher level themes. This, as a consequence, enables a broader understanding of the feedback generated. The description of the themes that were agreed are as follows:

- **Completeness** - this theme represents feedback tags that describe how finished the students work is.
- **Comprehension** - refers to any tags that highlight issues surrounding the understanding of source code, for example comments, documentation or ease of extendibility.
- **Design** - is any tag that refers to design elements or patterns which can be seen in the examined source code.
- **Programming Standards** - this theme represents any feedback tag that identifies how students source code aligns to accepted programming standards within the field. It can also refer to tags that suggest improvements to code involving use of more appropriate language features or techniques.
- **Miscellaneous** - this theme represents any tag that cannot fit in to any other theme. Tags common to this theme tend to be non-specific such as "good", which when isolated from the original source code is not very meaningful.

The second review phase using the new themes revealed that an average agreement rate of 90.45% was achieved; this satisfied the desired 80% limit

and as a result the thematic analysis process could be considered as being completed.

An example of how two themes were amalgamated is the "Best Practice" and "Use of Syntax" themes. It was decided to combine them due to the difficulty of separating specific language features from general programming practices and thus a higher level more general theme of "Programming Standards" was formed.

These final themes specifically relate to programming feedback and were chosen to provide high level information that may help direct learning or teaching. It would be possible to thematically analyse the feedback tags from a purely educational perspective and use themes constructed out of the types of feedback. However, the focus of this particular analysis is on how the feedback relates to the various aspects of programming as a skill.

Now that the thematic analysis process has concluded, the resulting data can be used in combination with the data collected by sentiment analysis in Section 5.4 to attempt to gain a better insight in to the cohort's learning.

5.5.2 Results

The distribution of feedback tags according to the different themes as agreed by the review process are shown in Table 5.7. It is clear that there is a significant imbalance in the distribution of tags across the themes, with a large majority of tags being associated to the 'Programming Standards' theme.

Table 5.8 presents the themes along with the proportions of tags in each theme according to the recorded sentiment from the NaCTeM tool and a human participant. It was decided to focus on the sentiment as interpreted by the NaCTeM tool since it was demonstrated in the Section 5.4 as being adequate at providing an indication of human perception for this data set. However, due to the restrictions of the system being unable to interpret the sentiment of specific technical terms, it was decided to include the perception of the primary researcher alongside the results of the automated system for

added perspective.

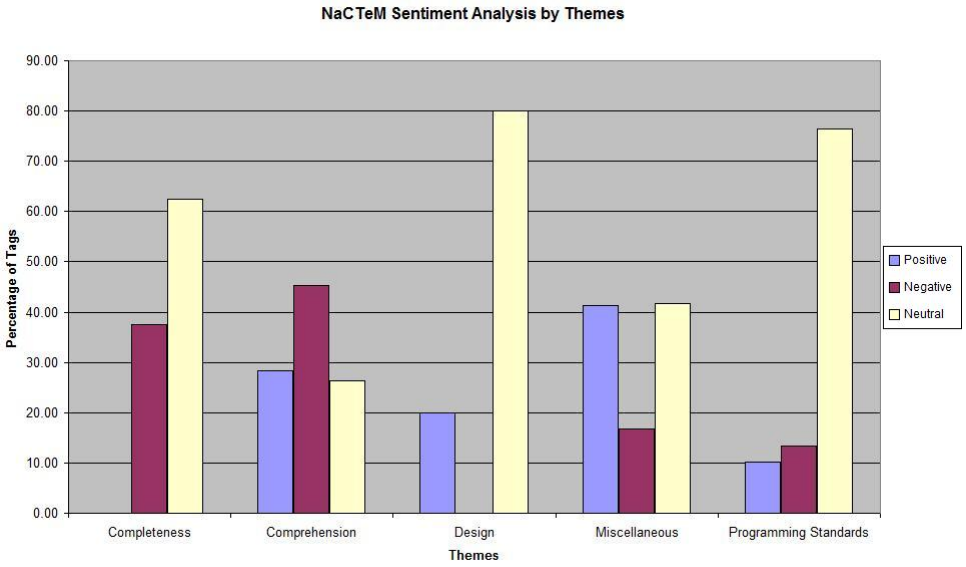

Figure 5.6: Graph presenting the NaCTeM sentiment analysis thematically

The primary disagreement between the automated system and the researchers' perceptions is in the 'Completeness' category. The human respondent reported that all of the tags were negative, whereas the NaCTeM system reported less than half of the tags as being negative in that category and the remainder being neutral. The secondary disagreement is within the 'Design' theme of feedback. The human respondent has identified one quarter of the

Theme	Unique Tags	% of tag uses
Completeness	3	5.42%
Comprehension	26	32.20%
Design	8	5.08%
Miscellaneous	10	4.07%
Programming Standards	53	53.22%

Table 5.7: Distribution of feedback tags in to themes

	% Positive		% Negative		% Neutral	
Theme	NaCTeM	Human	NaCTeM	Human	NaCTeM	Human
Completeness	0.00	0.00	37.50	100.00	62.50	0.00
Comprehension	28.42	23.16	45.26	45.26	26.32	31.58
Design	20.00	20.00	0.00	26.67	80.00	53.33
Miscellaneous	41.37	33.33	16.67	33.33	41.67	33.33
Programming Standards	10.19	0.00	13.38	6.37	76.43	93.63

Table 5.8: Sentiment analysis presented in context of thematic analysis data

tags as being negative whereas the automated system has not identified any.

Figure 5.6 shows the percentages of tags in each theme according to the sentiment analysis as performed by the automated NaCTeM sentiment analysis engine. It is clear from the graph that a majority of tags within most themes are reported as being neutral, apart from the 'Comprehension' theme which has a higher proportion of negative tags associated with it.

5.5.3 Threats to Validity

One limiting factor for the thematic analysis process is the number of reviewers recruited and their relationship to the research study. Both reviewers were members of the same university department and were experts in teaching programming. To secure researchers who were independent from the institution may have improved reliability of the review phase, as it is possible that the two reviewers share common views as a result of them working in the same institution. There were only two reviewers available for this project; more would have also been an improvement.

Another threat to the validity of the thematic analysis and sentiment analysis process is that the tags were considered in isolation from the source code to which they related to. The result of this is that each feedback tag and each instance of its use could have a different meaning dependent on which source code fragment it was associated with. The feedback tags' contexts were intentionally removed from the research in this thesis primarily to keep the investigation consistent between the automated tool and human participants.

That is, the NaCTeM tool would be unable to benefit from the source code context information and as such, this information was hidden from human users who may have been able to derive additional meaning from it.

Another threat to validity in this investigation is the suitability of the NaCTeM automated sentiment analysis engine. It has been noted in Section 5.4, that the NaCTeM tool is able to provide a reasonable indication as to the sentiment of feedback tags. The tools weakness in analysing subject-specific or technical terms may not give an entirely accurate interpretation of the data.

The results collected in this study have been collected from one particular usage of the system and the analysis technique. This has resulted in a very specific dataset that cannot, without further research, be applied outside the original research context.

5.5.4 Evaluation

The most frequently used tag within the 'Programming Standards' theme is that of "use constants" (23), closely followed by the "use generics" (12) tag, both of these were detected as being neutral within the theme. These tags demonstrate that some students had trouble using or had forgotten to use specific language features. The tag "use generics" refers to a new Java feature involving generic abstract typing. This tag was particularly frequent, which implies that students had trouble remembering to use the new feature. However, due to the neutral language used by the examiner, these tags were classified as being neutral and may have been obscured if the thematic sentiment analysis was used without closer inspection. The issues identified by these tags are of somewhat low importance in the assessment purpose and are easily addressed in lectures. As such, based on the results of this thematic analysis, lecturers are able to adjust their teaching to better support their students' learning needs.

The 'Programming Standards' theme is hugely weighted towards neutral feedback. Upon further investigation, it is clear this is because a majority of

the feedback is given to direct students to use alternative syntax or language features that are more appropriate. None of the feedback is particularly positive or negative, it is more advisory. For example, "use mouseadapter" and "use string.format" are both tags which have been detected as being neutral and have been selected as being apart of the 'Programming Standards' theme. They both suggest that the student should investigate specific features found in the programming language.

The theme 'Programming Standards' is a particularly general theme and could be split in to a variety of sub-themes including 'Java Specific Feedback' and 'General Programming Feedback'. This separation did not persist due to a difficulty encountered during the review phase. Both reviewers were confused because in some circumstances Java specific feedback and general programming feedback overlapped. This resulted in them having different analysis results, thus causing disagreements. The end result was that the themes were merged in to a larger more general theme.

One of the most important skills in learning programming is that of writing readable and maintainable code. When an examiner is assessing source code, readability and comprehensibility is at the forefront of their mind. This is simply because any aspects of students' work that are difficult to understand become very obvious to the examiner as mark it. As a result, 'Comprehension' is the next most prominent feedback theme, with over one third of all feedback tags being associated with it. This theme appears to have had a relatively high proportion of negative feedback associated with it, possibly due to how noticeable deficiencies in the comprehensibility of students' code can appear to examiners. Examples of tags associated with the 'Comprehension' theme include "needs refactoring", "good commenting" and "bad commenting". The tags referring to how well students have commented their code are quite clear, however "needs refactoring" is a tag that can cause confusion. This tag actually caused part of the initial disagreement in the review phase of thematic analysis as it could refer to the 'Comprehension' theme as well as the 'Programming Standards' theme.

The tag "needs refactoring" implies that something is organisationally or structurally out of place with the students' code. It may be that the external functionality provided by the system is adequate but the code could be improved structurally. This tag could also refer to cases where a student has named a method or field in such a way that it does not reflect its purpose. It could also simply mean that a method is too big and needs to be extracted in to a series of smaller ones. Often the problems highlighted by this feedback tag do not affect the software directly at runtime but can affect its maintainability.

Upon closer investigation, the difficulties indicated by the NaCTeM sentiment analysis engine within the 'Comprehension' theme do highlight a pattern of feedback. These are a variety of tags such as "bad commenting", "lacks documentation" and "needs comments" which have relatively high frequencies within the theme and all have been marked as being negative from sentiment analysis. This information could be used to reinforce the importance of comprehensibility of programs to software maintenance during lectures.

The 'Completeness' theme may appear as the most worrying as there is no positive feedback reported only high proportions of negative and neutral. After closer inspection it is clear that the automated system is indeed correct. It seems as though examiners did not tend to give positive comments about the completeness of the students work for this particular assignment. In fact according to the human sentiment analysis all of the tags within the 'Completeness' theme were perceived as being negative. However, it is important to recognise that the 'Completeness' theme has a very small frequency of tags associated with it so the overall effect of the graph appears to be somewhat exaggerated due to percentages being plotted instead of frequencies.

5.5.5 Section Overview

This section has focused exclusively on investigating *RQ5 Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?*.

The ability for this combination of sentiment and thematic analysis pro-

cesses to highlight patterns in feedback which may otherwise have remained hidden has been demonstrated by this preliminary investigation.

Using sentiment and thematic analysis on feedback tags can be used to compliment and inform 'Just in Time Teaching' as it supports lecturers in giving students the support they need. It also provides a high level overview of the types of feedback students are receiving for a particular assessment, allowing lecturers to better identify which general areas need further work in lecturers.

Thematic analysis complements feedback tagging as any specific issues that appear throughout students' feedback can be presented in individual tag feedback and the results from thematic analysis can help to identify the high level teaching or learning issues. Therefore, the unique combination of sentiment analysis, thematic analysis and feedback tagging allow for more interesting feedback data.

The purpose of including thematic analysis is to demonstrate the type of analysis that can be performed using feedback in the form of tags and to document the types of patterns that can be detected. RQ5 focuses on investigating the patterns that can be detected from the use of feedback tags. The patterns identified within this investigation primarily relate to the themes or categories that feedback delivered to students are related to. As such, within this investigation, the primary finding is that 'Programming Standards' and 'Comprehension' are the most common feedback themes in this instance of source code assessment. It is useful to note that 'Comprehension' has a high proportion of negative tags. This, combined with the knowledge that 'Comprehension' tags have the second highest frequency, leads to a clear need to reinforce the importance of code comprehensibility to students in this cohort.

5.5.5.1 Relevance to Research Questions

This preliminary investigation has provided an indication as to the answers RQ5:

RQ5: Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching? It is clear that through using this combination of analysis approaches new information has been uncovered about the feedback which may have otherwise remained hidden. The focus of examiners giving neutral comments to students about 'Programming Standards' and largely negative comments to do with 'Completeness' and 'Comprehension' are important findings which can be used to address cohort wide problems through lectures or practical remedial work. So far according to the findings of this investigation it is clear that these two analysis techniques can provide additional information which can benefit teaching.

5.6 Chapter Overview

This chapter has presented the first preliminary investigation in to the SWATT approach to feedback generation and dissemination. In this investigation a summative assessed group project was used to test the feedback approach. As a result, a number of recommendations can be made to improve the subsequent preliminary investigation and ultimately the final investigation.

The key findings of this chapter are summarised below.

- 42% of student groups opted in to the sharing program facilitated by the SWATT system.
- There is a mixed response from students as to how useful tags are as an approach to feedback for group assessment activities.
- Examiners and students can perceive sentiment of feedback tags in different and sometimes conflicting ways.
- The NaCTeM automated sentiment analysis tool provides a reasonable indication of how a human may perceive the sentiment of feedback delivered in the form of tags.

- The NaCTeM tool is less able to interpret the sentiment of feedback tags that use highly specialised technical vocabulary or references to high level domains specific concepts.

- On this occasion 'Comprehension' is the theme of feedback with the most negative tags as interpreted by the NaCTeM sentiment analysis tool. This perhaps is linked to the fact that the group project used was a large software project and therefore there will have been an increased overhead in comprehending it.

The recommendations from Chapter 5 are as follows:

- It is clear from this investigation that group projects are not particularly well suited for the SWATT approach of feedback delivery. As a result it has been decided to continue to investigate individually assessed projects only.

- In order to avoid complicated cross sample analysis it has been decided to use a single sample during sentiment analysis that all participants review, instead of having greater coverage and multiple samples.

- Strategies to improve questionnaire response rates should be considered and implemented for the final investigation.

Chapter 6

Preliminary Investigation Using an Individual Assessment

6.1 Introduction

In order to formulate and fine tune the research methods needed to address the research questions, another exploratory preliminary investigation was carried out and is described in this chapter.

The preliminary investigation outlined in this chapter uses feedback tags generated from individually assessed work. This is in contrast to the investigation presented in Chapter 5, which uses group based assessment. The work however, is formative in nature and does not contribute to students' final qualifications. A number of the recommendations from Chapter 5 have been applied in this investigation to help inform their suitability for use in the final investigation. The recommendations from Chapter 5 are summarised below:

- Participants indicated that the SWATT approach would be more useful if used with an individual piece of work and not a group exercise. As a result of this observation the following investigation uses an individual assessment activity.
- The sentiment analysis sampling is to be simplified in this investigation.

- Questionnaires will be shortened to increase the likelihood of student completion.

A similar series of investigations, as presented in Chapter 5, have been designed to explore the benefit of using the SWATT system in a small formative individual programming assignment.

6.2 Investigation Context

At the end of the first year undergraduate course it is often common practice to set a refresher exercise that involves students practicing their programming skills before they go away on their summer holidays. In academic year 2008/09, it was decided to provide formative feedback to students who completed this using the SWATT feedback tagging approach.

Out of the total 59 students registered for the course, 21 submitted their work to be formatively assessed; meaning only 36% of the cohort opted to complete the formative exercise. Each of the 21 students received feedback in the form of feedback tags annotated throughout their source code via the SWATT system. Students also received feedback in the form of a short comment focusing on how their source code performed at runtime. This enabled comparison of the feedback tags to traditional text-based comments.

For this investigation there were two examiners involved, one looked specifically at the source code and used the SWATT tagging approach to annotate and generate feedback. The other simply ran each of the student's source code projects and commented on the user experience and functionality of the submitted work.

After the assessment process, both the runtime and source code feedback were delivered to students using the SWATT web interface.

A total of 81 unique tags were generated as feedback to the students' work. Some of these tags were reused in more than one student's feedback. The number of tag annotations made was 446, which is an average of 21 feedback tags per student. All source code files submitted by students were tagged

using the SWATT approach. As such, no selection algorithms were needed for sampling files to be marked.

6.3 Investigating Sharable Feedback Tags

The results of this investigation focus on whether or not students opted into the sharing aspects of the SWATT system. This helps determine whether or not the same behaviour is exhibited by students in an individual assignment as in a group one.

The purpose of this investigation is to gather results to support the answering of the following research questions:

- *RQ1 Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?*
- *RQ2 Do students opt-in to share their code and associated feedback?*
- *RQ4 Do students perceive benefit from having access to other students' code and associated feedback tags?*

RQ3 cannot be addressed fully in this investigation due to the assignment being formative in nature. Formative assignments, by their nature, do not result in assessment marks being calculated. Since RQ3 focuses on investigating whether a student's attainment relates to their decision to share; this research question cannot be addressed in this investigation.

6.3.1 Investigation Method

The investigation method is identical to the investigation described in Section 5.3.1, with two exceptions. Firstly, students are no longer in groups and so it is each student's personal decision on whether or not to share their work and feedback. The second exception is that the exercise is formative in nature and as such no quantitative marks are associated with it. This means that

this investigation is unable to consider how student attainment relates to their decision to share their work.

6.3.2 Results

A total of 95% (20/21) of students logged into the system a total of 88 times over the investigation period. This figure represents a 37% increase on the number of individual users who logged in to view their feedback when compared to individuals who logged in to view group feedback tags as discussed in Section 5.3. The mean number of logins per user is 4.4, with a majority of users logging in more than once to view their feedback.

A total of 43% (9/21) of individuals opted to share their feedback and associated source code. The proportion of individuals in this cohort who opted to share their work is almost exactly the same as that of the groups who decided to share their work in Section 5.3, in the previous cohort.

The questionnaire response rate is again low with only 38% (8/21) of students completing the online questionnaire. This however is a higher proportion response rate than the previous preliminary investigation. The results of the questionnaire are summarised below:

- 73% (6/8) students thought that their feedback was either "Very Easy" or "Easy" to understand using the new feedback tagging approach. 2 students found it neither easy nor difficult.

- 50% of students suggested that the amount of feedback they received was "About Right". With 38% of students indicating that more feedback was needed.

- 100% of respondents reported that the quality of the feedback they received was either "Good" (38%) or "Average" (63%).

- 76% of students rated their ability to improve based on the feedback they received as being "Very Easy" or "Easy".

- 50% of students reported that they were able to notice patterns in their feedback as a result of seeing it as a tag cloud.
- 63% of all respondents to the questionnaire found that sharing their work was useful. The remaining 38% who filled out the questionnaire did not share their work.
- 100% of respondents reported that they did not use the discussion board facilities of SWATT, but 100% also stated they do like the idea of an online community where they can discuss their work and feedback.

There was a 50% divide between respondents who thought the SWATT approach was better than traditional approaches to feedback and those who stated that both approaches are useful in different ways.

100% of respondents reported that they would like to see the SWATT approach used again for giving feedback to programming work.

6.3.2.1 Investigating the Differences Between Sharers and Non-Sharers

As in the previous preliminary investigation, it was decided to conduct a variety of statistical tests to determine whether there are any significant differences between those students who reported that they had shared their work and feedback and those who did not.

Due to the particularly low questionnaire response rate and the fact that all respondents reported high satisfaction regardless of whether or not they had opted to share, none of the independent t-tests showed any statistically significant differences between the two groups. Table 6.1 presents the results of the independent sample t-tests carried out.

The questionnaire results for quantity in this study are reported in a slightly unusual scale. The scale is as follows:

1. Far too Much
2. A Little Less Needed

3. About Right

4. A Little More Needed

5. Far too Little

As such, the closer the value is to 3, the more satisfied the respondent is with the amount of feedback they received.

On this occasion, the students who opted not to share appear to be slightly more satisfied overall with their feedback than those who did opt to share. The average responses indicate a higher level of satisfaction for each metric. This includes satisfaction with the amount of feedback received. This implies that those who shared their feedback received slightly too much feedback on average, whereas those who did not share received slightly too little.

It is important to note that due to the very small sample of only 8 respondents, these results are only useful to test and refine the research methods for the final investigation.

	Understandability	Ability to Improve	Quality	Quantity
Sharers (S)				
Mean	3.80	2.80	3.40	3.80
SD	0.84	0.84	0.55	0.84
Non-Sharers (NS)				
Mean	4.00	3.00	3.33	2.67
SD	0.00	0.00	0.58	0.58
t-test				
Result	t(6)=0.40	t(6)=0.40	t(6)=-0.16	t(6)=-2.04
Significance	n.s.	n.s.	n.s.	n.s.

Table 6.1: Table showing statistical tests run on sharers vs non-sharers' questionnaire responses

6.3.3 Threats to Validity

Sharing statistics could still be under-represented as students may still be sharing informally without using the system. In order to detect this, subsequent investigations should include a question in the questionnaire asking respondents if they had shared informally, outside of the system or perhaps approach the topic in focus groups.

The response rate for the questionnaire is low again making the results from it less reliable and perhaps less representative of student opinions.

Another factor to be considered in this investigation is that the students' work was submitted before the summer holidays and the feedback was not delivered to them until their return to university, this represents a significant delay of three months. This delay is abnormal as the project deadline was, due to unforeseen circumstances, set at the end of a university academic year. This delay was caused by the module leader gaining alternative employement at short notice. The last minute staff changes made it difficult to release the feedback until the students returned from their holidays. This delay could negatively effect students' engagement with their feedback and may make the results less representative of a normal assessment and feedback process.

6.3.4 Evaluation

A possible reason that the sharing figure was still low in this investigation could be that students were still not fully aware that their feedback would be shared anonymously and as such were reluctant to share it. This should be addressed using clear messages within the SWATT system, or a sample screenshot of a shared piece of work, so that students are more clearly informed.

Another reason that sharing was below 50% may be due to the assignment being formative and being set in the previous academic year. These reasons may contribute to students perceiving it as having less of a relevance to their second year work.

The questionnaire response rate was very low again. A possible explanation

for this low response rate is the fact that it was released at the beginning of the academic year and as such students were particularly busy and had effectively completed the course to which the feedback related to.

This investigation has raised a number of salient findings including the fact that students do value the feedback they receive. One student commented on how they used the feedback to modify their code and modified their work according to the examiners feedback. "Good references to sections of code, so I knew where was being discussed very quickly - this helped in modifying the code".

Students did highlight one issue which is inherent with the use of tags and this is the issue of ambiguity. Inevitably using a form of metadata such a tag will not provide all of the information that a long comment would. A student suggested that the "... tags could be links to a definition about that tag ... ". One particular student suggested that they already knew where they had made mistakes and would have preferred to have more detailed comments on how to improve. "I generally already knew WHERE I went wrong I just didn't know HOW to correct it". In this case it is strange that if the student already knew where they could improve the code, why did they not seek specific guidance? However, this does highlight the fact some tags may, on their own, lack the detail to enable students to take immediate corrective action.

One student, who did not opt to share their work, reported in the questionnaire that they thought their feedback was "...specific to my code...". They also indicated that they thought it would not be useful to any other students as a reason why they elected not to share.

A clear 50/50 divide occurred in the questionnaire for this study. Half of all respondents reported that the SWATT system was better than the traditional feedback techniques the students had been previously exposed to, whilst the other half said that both traditional methods and the SWATT methods should be used together as they are both useful in different ways.

6.3.5 Section Overview

The Research Questions considered in this section are shown in Table 6.2. The remaining questions are considered in the subsequent sections.

RQ	Research Question	Considered
RQ1	Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?	✓
RQ2	Do students opt-in to share their code and associated feedback?	✓
RQ3	Which students tend to opt-in? E.g. weaker or stronger students?	*X*
RQ4	Do students perceive benefit from having access to other students' code and associated feedback tags?	✓
RQ5	Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?	
RQ6	How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?	

Table 6.2: Research questions considered in section 6.3

The results outlined by this investigation have shown that in a formative individual assignment students are more likely to login and view their feedback individually. However, the proportion of students who opted to share their work remained the same as in the preliminary investigation with group work. This indicates that sharing may not be a popular activity when it comes to formative feedback and programming work.

In future investigations of this nature, sharing should be explicitly described and perhaps a sample of shared feedback should be shown to ensure students know the level of anonymity they will receive. Perhaps more incentive to share their work could be given by allowing students to view high level statistical information about the entire cohort with the promise of more detailed information when they opt to share their work.

It is clear that an important modification to the questionnaire strategy is required to encourage more responses. The use of a prize draw as an incentive

and careful consideration of timing is required to encourage a higher response rate in the final investigation.

6.3.5.1 Relevance to Research Questions

This preliminary investigation has provided an indication as to the answers to the following research questions:

RQ1: Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software? The results for the use of the SWATT approach for individually assessed work are generally more positive than in the group work investigations as presented in Section 5.3.

The questionnaire results indicate that 76% of students thought that they could improve their work based on the feedback they received through the SWATT system. The perceived quality of the feedback students received has improved in this study with 38% of students reporting the quality as being good, with the remainder saying that it was average according to their expectations.

The quantity of feedback received, again as in the previous investigation, has a mixed result. In this investigation 50% of students said that the amount of feedback they received was "about right", while 38% would have liked to have had more.

Overall indications from this investigation show that students had perceived some benefit to their learning from the feedback being delivered in tag form. However, due to the low response rates it is important that more research is carried out.

RQ2: Do students opt-in to share their code and associated feedback? Roughly the same proportion of students, 43% opted to share their feedback and work in the individually assessed work as in the group project. This is particularly interesting as the cohorts involved were different as was the assignment project.

Early indications show that less than half of the student population are

interested in sharing their work and feedback. Rational for this should be gathered from focus groups, if the same result occurs in the final investigation.

RQ3: Which students tend to opt-in e.g. weaker or stronger students? In this investigation the assessment process did not provide quantitative data that could be used to measure student fulfilment of learning outcomes, so unfortunately no answer to this research question is possible.

RQ4: Do students perceive benefit from having access to other students' code and associated feedback tags? 100% of students who opted-in to the sharing scheme and who responded to the questionnaire, reported that they did find it useful to view other people's work and feedback.

One comment from students as to why they found viewing their peers feedback and code useful was "..Because I could see where they had written better code..". This clearly indicates that the student is interested in using other people's feedback to improve their own programming skills. However, only 40% of students who viewed other people's feedback said that it helped them to understand their own tags.

This investigation clearly indicates that those who used the sharing functionality perceived a benefit to their learning.

6.4 Sentiment Analysis of Feedback Tags

This study followed a similar procedure as those described in Section 5.4. The aim being to generate data to be used along with Thematic Analysis to gain additional information that could benefit learning or teaching, as well as investigating how the sentiment of tag based feedback is perceived by examiners and students for tags generated with individually assessed work.

The Research Questions to be focused on in this section are as follows:

- *RQ5 Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?*
- *RQ6 How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?*

6.4.1 Investigation Method

In order to investigate the sentiment of feedback tags a sample of 44 tags, corresponding to over 50% of the tags generated for this individually assessed assignment were analysed by 3 examiners and 3 students. The three student participants were selected as they are members of a committee in which the students had been elected by their peers to represent them in academic issues. This provided a convenient sample for this investigation. The 3 members of staff who participated were not involved in the original assessment activity but have experience in assessing students' programming work and giving feedback.

It was decided to simplify this investigation and have only one sample of 44 tags that every participant analysed for the underlying sentiment. This reduces the complexity of having multiple samples as was employed in the previous preliminary investigation in Chapter 5.

In order to make the analysis process as convenient as possible, students were asked to complete an online questionnaire to return their analysis

responses. Examiners completed the same analysis via a paper based questionnaire.

6.4.2 Results

The average agreement between students and examiners perception of the feedback is 65.91%. This figure was calculated using the majority perception from the three examiners and three students and then calculating the average agreement between the students and examiners.

Table 6.3 shows the proportion of the sample tags students and examiners identified as being "Positive", "Neutral" and "Negative". The "No Agreement" category represents where there was no majority in participant responses. That is, each of the participants chose a different sentiment for a particular feedback tag.

Sentiment	Students	Examiners	NaCTeM Tool
Positive	18.18%	22.73%	29.55%
Negative	22.73%	27.27%	20.45%
Neutral	52.27%	50.00%	50.00%
No Agreement	6.82%	0.00%	0.00%

Table 6.3: Sentiment analysis: percentages of feedback tags in each sentiment category

The results in Table 6.3 show there is relatively little difference in the proportion of feedback tags students and examiners have indicated as being positive, neutral and negative in this investigation. However, it should be noted that there is only the 65.91% average agreement between students and examiners perceptions. The reason for this is that whilst examiners and students may have allocated similar proportions of feedback tags as being positive, neutral and negative, they have not always allocated the same tags as being the same sentiment. This means that on a per tag basis there was a 34.09% disagreement between students and examiners perception of feedback tags.

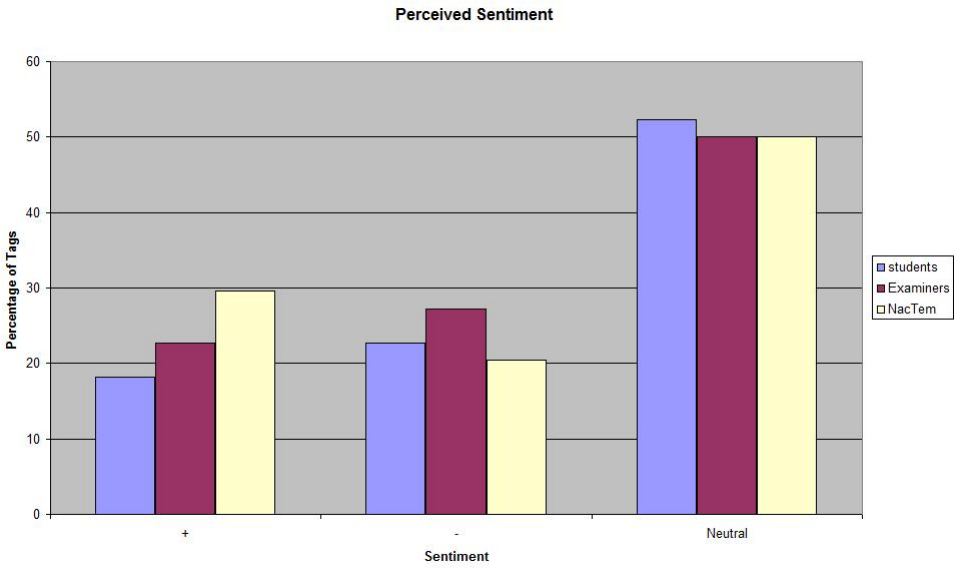

Figure 6.1: Graph showing perceived sentiment by respondent

Figure 6.1 shows that for this investigation the students perceived more neutral feedback than the examiners. The examiners and NaCTeM system reported the same proportion of tags as being neutral. These results contrast with the previous investigation where students were more likely to identify positive and negative tags and the examiners more likely to perceive feedback as being neutral. On this occasion the examiners perceived more feedback as being either positive or negative than students did.

6.4.3 Threats to Validity

The threats to validity outlined in the previous preliminary investigation, in Section 5.4, still apply to this study as the number of human participants remains low, again this is due to the time consuming nature of manual sentiment analysis.

An additional threat to validity in this investigation is that the two groups of participants used different media for returning their analysis results. Students used electronic questionnaires and examiners returned their analysis

on paper. There could be issues such as increased fatigue from using the computer monitor for the electronic method, which may have impacted the reliability of the results from student participants. There could also be an increase in the amount of fatigue experienced by both groups in this study as a larger sample of feedback tags were given for sentiment analysis.

6.4.4 Evaluation

Results in this preliminary investigation highlight further differences in how students and examiners perceive the underlying sentiment of feedback issued using tags. Whilst the proportions of tags perceived as positive, neutral and negative are not very different, the actual tags being perceived as a particular sentiment are not the same between students and examiners.

The three examiners seem to hold the similar perceptions of what is positive, negative and neutral, all three examiners had a 96.14% average agreement rate.

Students, however, have more diverse and sometimes conflicting perceptions of the sentiment of feedback. The average agreement between the students who participated is 72.32%. This is partially due to three occasions where there was complete disagreement between student participants. In these cases the three students between them selected all three possibilities (Positive, Neutral and Negative). The three feedback tags are 'move to method', 'inform the user' and 'duplicate code'. It is difficult, from an examiners perspective, to see how 'duplicate code' could be construed as being positive when students are told regularly about the dangers of duplicating code, however one student did indeed report it as being positive. This misconception from one of the student participants may be a motivation for readdressing in lectures the dangers of copy and paste coding.

The proportions of tags associated with each sentiment category do not follow the same trend as in the previous preliminary investigation. In this case, students have, on average, identified more neutral tags than either positive or negative, whereas examiners have been more likely to identify positive and

negative tags. This is clearly shown in Figure 6.1.

Unusually, in this investigation, a majority of the students perceived tags such as 'good GUI' and 'good Javadoc' as being neutral. However, examiners, the NaCTeM tool and at least one student perceived these as being positive. A possible explanation of this is that in this investigation the number of tags each participating student was asked to analyse was larger than in the previous one and therefore fatigue may have set in, or participants may have rushed through the electronic questionnaire to finish without considering each feedback tag carefully.

6.4.5 Section Overview

The research questions considered are shown in Table 6.4.

RQ	Research Question	Considered
RQ1	Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?	
RQ2	Do students opt-in to share their code and associated feedback?	
RQ3	Which students tend to opt-in? E.g. weaker or stronger students?	
RQ4	Do students perceive benefit from having access to other students' code and associated feedback tags?	
RQ5	Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?	✓
RQ6	How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?	✓

Table 6.4: Research questions considered in sections 6.4 and 6.5

This investigation has shown how different perceptions of the sentiment of feedback, especially in tag form, can be. It is for this reason it is recommended that inclusion of the intended sentiment along with a feedback tag should be investigated in future research. By including the intended sentiment along with each tag, their will be less chance of ambiguity.

6.4.5.1 Relevance to Research Questions

This preliminary investigation has provided an indication as to the answers to research question 5 and 6.

RQ5: Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching? Again, this investigation has highlighted the ability for analysis techniques to be used with feedback tags to determine how positive, negative or neutral a cohort's feedback is. This information can be used to determine which aspects of the course require additional support.

RQ6: How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment? Once again, as in the previous preliminary investigation, it is clear that there are important differences in how students and examiners perceive feedback delivered in the form of tags. On this occasion, examiners perceived more polarity, that is a greater number of tags as either being positive or negative, whilst the students had more neutral perceptions.

The automated sentiment analysis tool used has performed reasonably well. However, the NaCTeM tool has also confirmed the limitations exhibited in the previous preliminary investigation. It is clear that there is scope for the sentiment analysis tool to be used during tag creation to mitigate the ambiguity in the sentiment of feedback tags. Although, it may be necessary to require human verification of this data, especially if the feedback is for a largely technical subject like programming.

6.5 Extending Sentiment Analysis of Feedback Tags: Using Thematic Analysis

This section investigates the combination of sentiment and thematic analysis data to determine if any additional information can be derived. The cross analysis approach is the same as that described in Section 5.5.

This section focuses exclusively on:

- *RQ5 Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?.*

6.5.1 Investigation Method

The thematic analysis phase of this investigation was carried out following the same process as described in Chapter 4 and the more refined process outlined in Section 5.5.1.

All of the 81 feedback tags are thematically analysed, with the reviewers being given a 60% sample, comprised of the most frequently used tags. This sample has been doubled from the previous preliminary investigation to give a more reliable analysis.

The two reviewers selected are, once again, both experienced examiners who have both previously been involved in assessing programming code for undergraduate projects. However, neither is directly involved with the assessment process that led to the generation of the feedback tags being investigated.

The initial review phase highlighted an average agreement rate of 75% which is 5% below the desired agreement rate. This resulted in a need to discuss these disagreements with the reviewers to identify the cause.

A majority of the disagreements in this initial review phase were caused by misunderstandings in the definition of the themes. As a result, the definitions were clarified and the reviewers elected to change their themes according to

their revised interpretation of the definitions.

The second iteration of reviewing resulted in an average agreement rate of 88.48%, which exceeded the desired 80% agreement rate and so the analysis process was deemed to have completed.

6.5.2 Results

The same themes and definitions are used as are defined in Section 5.5.1. The distribution of themes for this assignment is reflected in Table 6.5.

Theme	Unique Tags	% of tag uses
Completeness	3	2.91%
Comprehension	21	41.70%
Design	5	2.02%
Miscellaneous	12	9.19%
Programming Standards	40	44.17%

Table 6.5: Distribution of feedback tags in to themes

In the final phase of analysis a total of 70% (35/50) of the tags received a unanimous 100% agreement from the researchers and the reviewers. In a majority of cases (93%) where there was disagreement between a single reviewer and the researcher, the other reviewer did agree with the researcher's original decision, hence an average percentage agreement was taken.

There was one case where there was complete disagreement between the researcher and both of the reviewers disagreed with each other as well. This disagreement was not reconcilable, even after the second review phase. The feedback tag in question was "consider extensibility". The researcher, in this case, labelled the tag as being in the theme 'Comprehension', due to the fact that extensible code is often characterised by its structure or organisation. The first reviewer labelled it as being 'Programming Standards', as making software easily extensible is recognised as good practice within software

engineering. The final reviewer thought that this tag was better placed within the 'Design' theme as extensible design should be considered when the system is being designed. All of these are valid reasons for disagreement and therefore this is an example of the irreconcilable disagreements that may occur throughout this analysis process.

Once again, the results of the extended sentiment analysis are presented in the same format as in Section 5.5. Table 6.6 includes the NaCTeM sentiment analysis results alongside the human participants' perception for reference purposes.

	% Positive		% Negative		% Neutral	
Theme	NaCTeM	Human	NaCTeM	Human	NaCTeM	Human
Completeness	0.00	0.00	76.92	76.92	23.08	23.08
Comprehension	34.95	26.88	1.08	22.58	63.98	50.54
Design	88.89	88.89	0.00	0.00	11.11	11.11
Miscellaneous	63.41	63.41	12.20	12.20	24.39	24.39
Programming Standards	10.15	11.17	9.64	14.72	80.20	74.11

Table 6.6: Sentiment analysis presented in context of thematic analysis data

On the whole, when looking at Figure 6.2, the only theme to have a huge proportion of negative tags is that of 'Completeness'. This mirrors the previous investigation's results. Once again, the automated tool is providing a reasonable analysis for this theme and it is just the case that examiners tend not to comment positively on the completeness of students' work.

From Figure 6.3 it is clear that the 'Comprehension' theme has a higher proportion of negative feedback than that reported by the NaCTeM automated analysis. This is largely because technical terms are once again being used to describe different approaches to improving program comprehension. For example, the tag "more comments required" clearly highlights an aspect lacking in the source code. On this occasion, all human respondents agreed this tag was negative but the NaCTeM tool reported it as being neutral. Table 6.6 confirms that the largest disagreement between the human respondent and the automated analysis tool is under the 'Comprehension' theme. Where

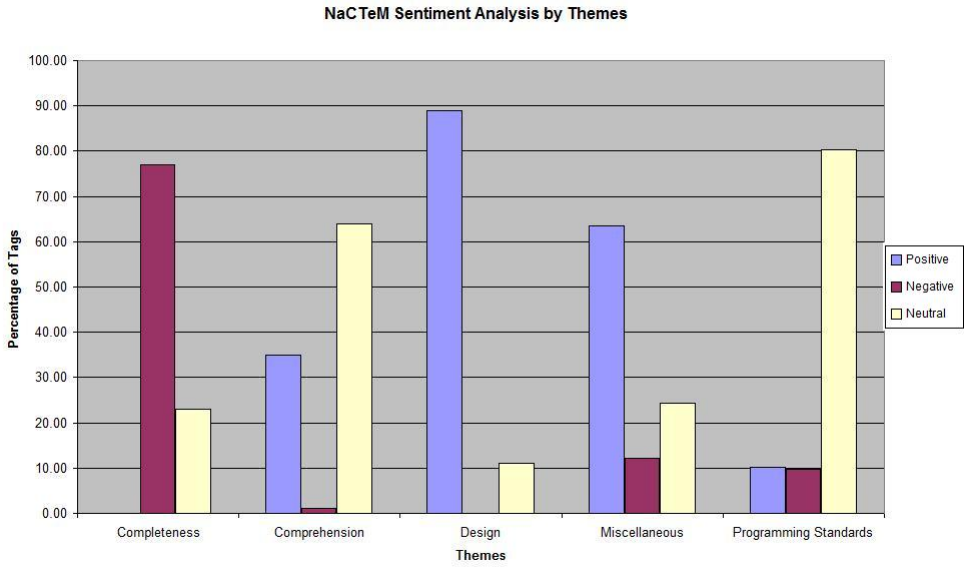

Figure 6.2: Graph presenting the NaCTeM sentiment analysis thematically

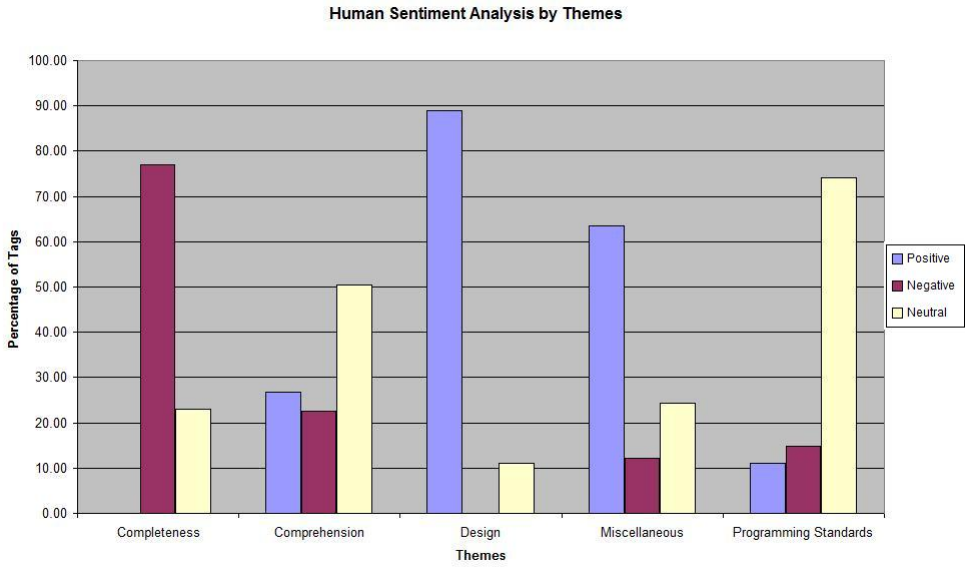

Figure 6.3: Graph presenting the human respondent's sentiment analysis thematically

the human respondent perceived 22.58% of the tags as being negative and the automated system only reported 1.08% as being negative.

6.5.3 Threats to Validity

Similar threats to validity affect this investigation as did the previous preliminary investigation discussed in Section 5.5.3.

6.5.4 Evaluation

It is clear from the results presented in Section 6.4 and both Figure 6.2 and Figure 6.3, that whilst the NaCTeM analysis tool provides a good indication of how the sentiment of feedback is distributed, it has noticeable limitations. In this case, it is its failure in detecting a significant proportion of the negative feedback in the 'Comprehension' theme. This was similar, to a lesser extent, in the previous preliminary investigation. However, the NaCTeM tool does have a relatively high agreement rate for the other themes and a majority of the values in Table 6.6 are consistent with the selected human responses for this dataset.

A majority of tags given in this dataset are once again within the 'Programming Standards' theme. However, on this occasion almost 10% more tags were in the 'Comprehension' theme and, according to the human participant, almost a quarter of these were negative. The top three feedback tags in the 'Comprehension' theme are: "use javadoc", "good commenting" and "more comments required". All of these are clearly relating to students usage of in-code documentation, to either highlight the lack of, or to praise sufficient usage of it.

The most commonly used feedback tags within the 'Programming Standards' theme are "create hash code method" and "create equals method". This occurred primarily as many students failed to override the inherited methods as required by the exercise. This could highlight a difficulty in understanding the concept of inheritance or the fact that use of certain data

structures that use these methods and should be reinforced. Once again, as in the previous preliminary investigation, a majority of feedback tags were interpreted as being neutral in this theme, due to the language chosen by examiners. This could result in some of the weaknesses in students' work being hidden in the high level sentiment analysis results if the associated feedback tags are perceived as being neutral. The tags "create hash code method" and "create equals method" were interpreted as being neutral by both the NaCTeM tool and examiners. However, these tags highlight aspects of the student's work that are missing and could quite easily have been considered negative feedback.

After closer investigation of the 'Completeness' theme, it is clear that the graph in Figure 6.2 is slightly misleading due to the low number of tags in this theme. Only 3 tags were allocated to be in this theme and all but one were negative. They were all very general tags, for example "incomplete", "shows incomplete work" and "check correctness" which are too general to be of any use when considered in isolation from their associated source code fragments. As such, without additional analysis of the tag-source code associations, none of these tags help discover what, if any, the underlying problems were. It could be speculated that these feedback tags are simply an indication that students struggled to complete the exercise in the given time or due to the formative nature of this assessment. Perhaps the students involved did not feel motivated to complete the implementation to the expected standard because it did not contribute to their end qualification result.

It is positive to note that the 'Design' theme has a very high proportion of positive comments which may indicate that students have successfully applied what they have learnt in terms of object orientated design within the assignment. An example of the most frequent design tags are: "good design" and "good choice of datastructure". However, once again this theme has a relatively low number of tags associated with it as shown in Table 6.5 and so the graphs may be misleading due to the use of percentages instead of tag frequencies.

This sentiment and thematic analysis data could prove useful if collected over a number of assignments for a given cohort and therefore any changes over time in a cohort's feedback could be captured to provide additional information.

6.5.5 Section Overview

This section has focused exclusively on investigating *RQ5 Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?*

The use of sentiment and thematic analysis on this occasion has clearly shown how general areas of strength or weakness in student learning can be identified and used as a starting point in investigating the specific conceptual difficulties in a cohort's learning which may be addressed in future teaching.

6.5.5.1 Relevance to Research Questions

This preliminary investigation has provided an indication as to the answers to RQ5.

After post-analysis interviews with examiners, a suggestion was raised which was to derive the themes for thematic analysis from a mark scheme, where appropriate. This idea will be investigated for the final investigation, since no appropriate mark schemes were available for this assessment or the previous preliminary investigation. This investigation did not have a formal mark scheme due to the formative nature of the assessment activity. The first preliminary investigations mark scheme was inappropriate for deriving themes and as such themes could not be derived retrospectively in this way.

RQ5: Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching? This investigation has demonstrated once again how feedback tags can be analysed to detect high level patterns, in this instance the topics to which the feedback is distributed in to for a cohort.

On this occasion, a higher proportion of the feedback was focusing on the comprehension of the students' code. This could be indicative of the difference in cohort learning throughout the program of study. For example, the previous investigation utilised second year students at the end of the academic year, whereas this investigation involved first year students at the end of the academic year. The difference in student levels and course learning objectives could be the cause for this difference.

6.6 Chapter Overview

This chapter has presented the final preliminary investigation in advance of the final investigation. In this investigation, a formative individual (non-group work) assessment was used to investigation the SWATT feedback delivery approach.

Before the recommendations for the final investigation are presented, a summary of the key findings are outlined.

- A total of 43% of students opted in to the sharing program facilitated by the SWATT System. This is essentially the same proportion of students to opt in as the proportion of groups who opted to share in the previous preliminary investigation.

- In this investigation there was a much more positive response from students as to how useful they thought tags were as an approach to feedback for programming work. This is characterised by the fact that 100% of students who completed the questionnaire reported they would like to receive feedback using the SWATT approach again in the future.

- Examiners and students have again perceived the sentiment of feedback tags in different ways, however on this occasion it is the students who have perceived the feedback as being more neutral and the examiners who have identified more positive and negative feedback.

- The NaCTeM tool again has demonstrated it can provide a reasonable indication of how a human may perceive the sentiment of feedback, however its inability to deal with domain specific terminology and references to complex high level concepts has been noted particularly in Section 6.5.

- On this occasion a combined thematic and sentiment analysis has revealed that 'Completeness' was the theme with the highest proportion of negative tags. This, may reflect a lack of engagement with the assignment from students, owing to the fact that the assessment was formative and may have been perceived as being less important from students perspective.

- The ability to combine the data collected from thematic and sentiment analysis of feedback tags has been used to demonstrate how high level feedback themes can be visualised according to sentiment to provide an immediate indication as to where potential remedial teaching could be used to support students.

6.7 Recommendations from Preliminary Investigations

To conclude the final preliminary investigation chapter, a discussion of the recommendations derived from both Chapter 5 and 6 is presented in this section.

Both preliminary investigations have highlighted a number of recommendations that shall be considered and used to direct the final investigation. These are summarised below:

- The SWATT approach should be used to best effect in individually assessed software projects and not group work. This is because individuals in groups already are provided with a forum to share their work and discuss feedback within the group itself.

- The SWATT system appears to be equally engaged with in both summative and formative assessment, however a greater range of analysis can be performed on summative assessments. As such, a summative assignment is preferable because it yields richer analysis for the final investigation.

- Anonymity appears to be a possible factor in determining whether students opt into the sharing functionality of the system. More visible notification that sharing does not reveal students identity should be made within the system.

- Since questionnaire response rates have been low in all preliminary studies, it is recommended to hold focus groups to supplement questionnaire results. This should help in exploring student perceptions of feedback tagging and sharing. Further to this, added incentives and careful investigation of the best timing for completion of the questionnaires should be considered.

- The recommendation for the sentiment analysis aspect of the investigation is to reduce the sample size of tags each human participant must analyse. The aim is to reduce the impact of fatigue on the investigation.

- The primary recommendation for the thematic analysis aspect of the investigations is to derive a secondary set of themes from the assessment criteria sheets so that, in the final investigation, a direct link to the assessment objectives can be made. In all preliminary investigations this was not possible and this technique should be applied if possible in the final investigation.

6.7.1 Revised Experimental Design

The final investigation will be operated using the recommendations made from this chapter.

- An individually assessed assignment will be used.

- The assignment will be Summative to enable comparison of student achievement and to answer of RQ3, see Table 1.1.
- Clear notices will be added to the SWATT system to make it clear shared work will be anonymous.
- Questionnaires will be released with an incentive, which will be entry into a prize draw to win gift vouchers; this will hopefully improve response rates.
- Focus Groups will be run to clarify any data gathered in the questionnaires and automated data collection.

Chapter 7

Final Investigation

7.1 Introduction

This chapter presents the final investigation results and uses these and the experiences gained from the preliminary investigations in Chapters 5 and 6 to answer the research questions posed in this thesis.

The key recommendations from Chapter 5 and 6 applied in this chapter are summarised below:

- An individually assessed software project should be used.
- A summative project should be used to enable answering of the RQ3.
- Questionnaire response rates have been consistently poor so an added incentive of a prize draw will be added. Timing the release of the questionnaires will also be carefully planned to minimize disruption.
- The sample size for the sentiment analysis aspects will be reduced to lower the fatigue experienced by participants.
- A secondary set of themes will be generated alongside the themes used in the preliminary investigations to determine the effect.

7.2 Investigation Context

The final investigation was designed to test the SWATT approach to feedback by using it along with the undergraduate first year "Introduction to Programming" module. As part of this module, a 10 week summative programming project was introduced as a way of continuous assessment for students in the first term and over the winter break. The purpose of this project was, not only to keep students familiar with programming techniques over the holidays, but to also support students in constructing a portfolio of programming work that can be shown to a prospective employer or summer internship interview.

Students were tasked with programming a BlackJack card game using the Java programming language and using as much of the material covered in the course as appropriate to design and implement a software solution.

A total of 49 students were enrolled on the course and 45 of them submitted a project to be assessed. Assessment was carried out using the SWATT approach to generate and deliver feedback to the students. A pre-designed proforma sheet was used to provide feedback on the specific learning outcomes and to deliver the student's final mark.

A total of 1531 feedback tag annotations were made for this project and these were comprised of 132 unique tags. This represents an average of 34.02 tags per student submission. This average number of tags is significantly larger than the average number of tags per submission in both of the preliminary investigations. This large average is perhaps due to a combination of the relatively large time scale of the project and the fact it is being assessed early in the program of study. All files submitted by students were annotated and marked using the SWATT prototype. This means there was no need for a formal selection algorithm to be used for determining the source code files that should be marked.

7.3 Investigating Students Use and Perception of Feedback Tags

7.3.1 Introduction

This section represents the primary focus of this thesis. The purpose of this investigation is to gather results to support the answering of the following research questions:

- *RQ1 Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?*
- *RQ2 Do students opt-in to share their code and associated feedback?*
- *RQ3 Which students tend to opt-in? E.g. weaker or stronger students?*
- *RQ4 Do students perceive benefit from having access to other students' code and associated feedback tags?*

7.3.2 Research Method

The methods used in this investigation are very similar to those presented in Sections 5.3 and 6.3, in that quantitative data, which is automatically collected from student usage of the SWATT feedback system, are analysed in combination with qualitative questionnaire and focus group data.

One important difference in the running of this investigation, when compared to all the preliminary investigations, is that students did not receive their summative marks at the same time as their feedback tags. It was decided to release their feedback tags in advance of their marks to encourage further engagement from students. As such, there was a delay of approximately 3 weeks between delivery of the students' feedback tags and their summative marks being released. This also means that students were forced to make a decision on whether they should opt-in to the sharing aspects of the system

using only their feedback tags as a basis for measuring how well they have done.

The questionnaire was administered electronically for convenience and remained open to new responses for one week. To encourage a greater response rate, respondents had the opportunity at the end of the questionnaire to enter a prize draw to win one of three gift vouchers. The questionnaire itself remains anonymous and respondents were informed of this before the online questionnaire began.

Two focus groups were conducted based on initial analysis of the questionnaire results in order to explore the results in greater depth. In both focus groups 5 students were invited and were selected based on the following criteria:

- Group 1 - A mixture of students who opted into sharing and who did not. 3 who did share and 2 who did not. (3/5 students attended, 1 of which did not share)
- Group 2 - Entirely composed of students who did not opt to participate in the sharing features provided by the system. (4/5 students attended)

A third focus group was planned as a result of the low attendence from the first two, however, no further students volunteered to participate and it had to be cancelled.

The purpose of Group 1 was to investigate in greater detail some of the feedback given in the questionnaire and to try and get a deeper understanding of how students perceived feedback, their opinion of feedback delivered in the form of tags, their opinion of sharing feedback and their overall expectations of what feedback for programming assignments should be like.

The purpose of Group 2 was to investigate exclusively the perception of sharing in the context of feedback delivered in the form of tags. To a certain extent, the issues explored in Group 1 are revisited to try and explore if sharers and non-sharers hold different opinions. The questionnaire results upon initial review held useful information as to why those who opted to

share did so, however they provided little insight into why those who did not share chosen not to. As a result it was decided to explore the opinions of those who did not share exclusively in a focus group.

7.3.3 Results

The results are discussed in a similar format to those presented in the first preliminary investigation involving summative assessment in Section 5.3.

7.3.3.1 System Usage Results

Every student who submitted a project logged in at least once to the SWATT system to view their own feedback. This is the highest level of individual student engagement in terms of feedback collection from all of the preliminary investigations. There were a total of 124 student logins which is on average 2.76 (SD: 2.96) logins per person. The median number of logins was 2 and the modal number was 1.

	Sharers		Non-Sharers	
	Frequency	No. Logins	Frequency	No. Logins
1st Class	6	43	4	4
Upper Second	7	16	9	16
Lower Second	3	11	8	15
3rd Class	1	4	2	5
Fail	2	6	3	4

Table 7.1: Table showing achievement frequencies alongside number of logins, by sharers and non-sharers.

Table 7.1 shows clearly that there are more people with the highest classification of marks who opted to share their feedback and work when compared to those who did not share.

Once again 42% (19/45) of the cohort opted into the sharing scheme. This is the same percentage of the cohort to opt into the sharing functionality in both of the preliminary investigations. It is important to note that the two

preliminary investigations involved different cohorts, different tasks, different examiners and different assessment criteria and yet the same proportion of students opted to share their work using the SWATT system on each occasion.

7.3.3.2 Questionnaire

The questionnaire response rate was much higher in this investigation when compared to all of the preliminary investigations. A total of 32 students out of a possible 45 completed the questionnaire, equating to a 71.11% response rate; 87.5% (28/32)of respondents were male and 12.5% (4/32) were female. The average age of respondents was 19 years 9 months. A total of 94% (30/32) were domestic students, whilst the remaining 6% were international students.

The summary of questionnaire responses are as follows:

- 100% of respondents reported as having at least looked at their own feedback using the SWATT system. This has been confirmed by the system's usage data.
- 94% of students thought that the tag based feedback was "Easy" or "Very Easy" to understand.
- 56% of students indicated that the feedback they received was "About Right", 41% reported that it was "Not Quite Enough" and finally one respondent (3%) reported that the feedback they received was "Far From Enough".
- 72% of students indicated that the feedback they received was of a "Very Good" or "Good" quality. 22% of respondents reported that it was of an average quality and the remaining 6% indicated that they perceived it as being poor in quality.
- 81% of students reported that receiving this type of feedback was "Very Useful" or "Useful" in helping them to improve their work. 13% indicated a neutral response and the remaining 6% reported it as being "Not Very Useful".

- 56% of respondents to the questionnaire shared their feedback, with the remaining 14% opting not to share their work and feedback. 73% of which said that it was useful to be able to view other peoples' work and feedback. Only 38% of people who shared their work thought that, in viewing shared work and feedback, they were better able to understand their own feedback.

- 100% of all respondents reported that they did not use the discussion board facility provided by the SWATT system. However, 81% of respondents indicated that they liked the idea of an online community which they could use to discuss their work / feedback in.

- 59% of students indicated that the SWATT approach should be used in conjunction with traditional feedback, whilst 34% reported that the SWATT approach on its own was better for source code assessment. Only a single respondent said that traditional approaches to feedback were better than the SWATT approach.

- 100% of respondents want to see the SWATT system used again in future programming assignments.

A number of students indicated that their favourite feature was that they were able to immediately get an indication of how well they had done and identify areas that need improvement by looking at the tag cloud on their feedback summary. This ability for the SWATT system to enable students to see a high level overview of their feedback and then allow them to explore the specific issues presented within their original submission, was regarded as one of the most important features. One student's comment summarises this aspect of the system well: "It is a really really weird way to get feedback, but I liked it after the initial moment of getting used to it. You can see a general theme to how you've done instantly but then, drill into certain areas to get more information. Really good though, I hope more stuff is like this in future."

Students gave a range of potential improvements to the prototype implementation of the SWATT system when asked if there were anything that they did not like about the system. However, a common theme in this question was that students sometimes found it difficult to understand the feedback tags due to their inherent lack of detail. Many of these students suggested that definitions and examples should be included along within the tags profile so that more information about the tag could be provided. Students also noted that in some cases feedback tags are very good at highlighting problems in their work but do not always directly tell the student how to fix them. One student's comment in particular summarises this point: "It would be more helpful if the 'page' for a specific tag contained more information about the general reasons for the tag and (if appropriate) links to more elegant methods." This was a response to receiving the feedback tag "consider more elegant approach" which does not inherently inform students of where to start looking for better way of implementing their algorithm.

7.3.3.3 Focus Groups

This section summarises the results of the focus groups run during this investigation. Table 7.2 shows the breakdown of the focus group participants, the target number of 5 participants per group was not reached however, interesting results were gathered nonetheless.

Focus Group	Male	Female	Total
Focus Group 1 - Mixture	3	0	3
Focus Group 2 - Non-Sharers	3	1	4

Table 7.2: Table showing breakdown of focus group attendees.

The focus groups' discussions were composed of a number of topics, some of which were pre-planned by the researcher, whereas others were brought up by the participants. These topics are summarised below.

- **Opinions of General Feedback and Feedback Tags** - This focuses on exploring participants' perceptions of feedback what they expect

from good quality feedback.

- **Quantity or Amount of feedback** - This topic explores what students perceive as being sufficient feedback for programming exercises.
- **Sharing of Feedback and Work** - This topic investigates the participants' motivation for or against sharing their feedback or work.
- **Engagement with Feedback** - It is of interest whether the process and method of feedback delivery was perceived as being useful by participants.
- **Different Approaches to Student Engagement with SWATT** - This topic was specifically included to investigate whether there were clear differences in how groups of students used the system to supplement their learning activities.

A key finding from both focus groups was that the participants' key criteria for good feedback is that it should help them to improve their future work in some way. When asked about tag based feedback, all participants from both focus groups reported that the SWATT approach was very good at giving feedback on issues localised within their own work. However, some participants would have preferred more personalised feedback that was specific to their own work and less focused on reusability. One particular disadvantage of the SWATT approach to feedback in its current state was, that when feedback tags highlighted problems in students work, they tended not to provide solutions to those problems. Participants commented that adding additional information in the tag profile page, that may help to point students in the right direction, would be beneficial.

It was clear from both focus groups that participants were satisfied with the amount of feedback they received each from the feedback tag system. In Focus Group 1, however, it was noted that the quality of feedback was more important to the participants than the quantity. One participant even

suggested that if he had too much feedback he would be overwhelmed and be unlikely to read any of it.

On the topic of sharing, it was clear that there are a large number of reasons participants did or did not share their work and feedback. Of these one was particularly prominent and that was that some of the participants did not feel comfortable sharing their work with strangers and would have been more inclined to participate in the sharing aspects of the system if they could select specifically which of their peers would be able to view their work. Students cited a 'facebook', social networking style of sharing and privacy as being desirable. Out of the group composed entirely of non-sharers, 40% of them informed the researcher that they had in fact intended to share their work but had forgotten to go back online and do it. The remaining participants who did not share reported they would have, had they been able to select individuals to share with instead of having to share with the whole cohort.

Both focus groups reported consensus that the process of feedback being delivered in advance of the summative marks had a profound impact on how they engaged with the system and therefore their feedback. In both groups it became clear that a majority of participants engaged in a process of attempting to estimate their summative marks by investigating the meaning of their feedback tags and how they linked to aspects of their work. When asked if the students had been given their marks and feedback at the same time whether they still would have engaged with their feedback in the same level of detail, it was unanimous that they would not have. This finding highlights the importance of timing in the feedback process. The feedback tags were released early whilst the marks then had time to be verified and released at a later date.

Based on a combination of focus groups and questionnaire results, as well as the automated data collected from the system, it is clear that different types of users exist who used the system. These are described in detail in the evaluation section as data from all investigation methods were used to

discover these behavioural groups.

The results of the focus group have helped to clarify some of the positive and negative comments raised through the questionnaires and, on the whole, participants were positive about the SWATT approach to feedback. The primary criticism raised was that of feedback tags being unable to provide as much inherent meaning as full length comments. This criticism was also raised as a positive point during the same focus group, on the grounds that it forced students to actively engage with investigating the meaning of the tags and thus may have helped students remember the reasons for the feedback in their work.

7.3.3.4 Investigating Differences Between Sharers and Non-Sharers

Table 7.1 clearly shows that there are some patterns in sharers' and non-sharers' usage of the feedback. This section investigates these and presents statistical tests on the two groups: Sharers (S) and Non-Sharers (NS) to identify any differences both in academic achievement and responses to the questionnaires.

The mean percentage mark of group S is 62.05% (SD=14.12), the median is 62% and for group NS the mean is 58.15% (SD=10.44) and the median is 61%. An independent sample T-test was run to determine whether or not there is a statistically significiant difference in marks between those students who opted to share and those who did not. The statistical analysis was hampered once again by the small population of interest. Unsurprisingly, from the similar mean values of the two groups, there was no statistically significant difference found in the marks between those who shared and those who did not; $t(43)=-1.07$, $p=0.29$.

However, Figure 7.1 shows clearly that out of those awarded the highest classification for this assignment there are slightly more who shared their work than who did not. The largest number of students who opted to share their feedback and work were awarded a upper second class mark for the assignment, however this is also the most frequently awarded grade in this

assignment overall. It is also interesting to note that the grade classification with the lowest proportion of sharers was those awarded a Lower Second Class mark. From Figure 7.1, it is clear that the other grade classifications only have a difference of one or two students between those opting to share and those not, whereas those awarded a lower second class mark had a difference of five.

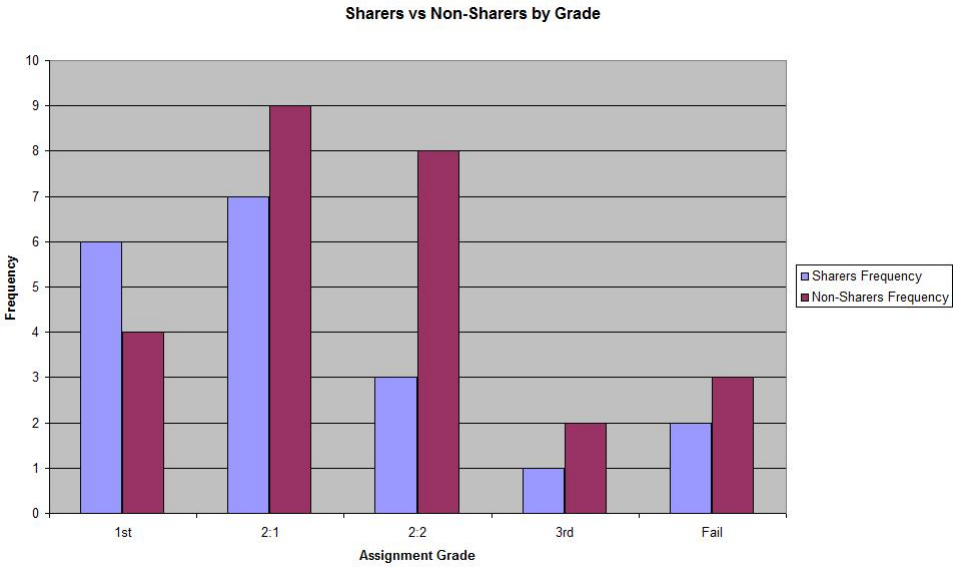

Figure 7.1: Graph showing Sharers vs Non-sharers and the frequency of students who achieved each grade for the assignment.

In order to determine whether or not there was a link between how many times students logged into the SWATT system and their summative marks a Pearson Correlation analysis was run. It was found that there was no statistically significant correlation between the two data sets. The number of logins does not accurately represent the amount of time the system was used. This is because a student could have logged in for 1 minute or 1 hour and no distinction would be made.

A number of independent sample T-tests were run in order to analyse the statistical significance of the questionnaire results of both groups S and NS. The results of these tests are presented in Table 7.3.

	Understandability	Ability to Improve	Quality	Quantity
Sharers (S)				
Mean	4.28	4.06	3.94	3.44
SD	0.46	0.80	0.80	0.62
Non-Sharers (NS)				
Mean	4.14	3.86	3.64	3.50
SD	0.66	0.77	0.75	0.52
T-test				
Result	t(30)=-0.68	t(30)=-0.71	t(30)=-1.09	t(30)=0.27
Significance	n.s.	n.s.	n.s.	n.s.

Table 7.3: Table showing statistical tests run on sharers vs non-sharers questionnaire responses

It is clear from Table 7.3 that the two groups show no statistically significant differences in their responses. On the whole, the responses were all positive irrespective of whether the students opted to share or not. It should be noted once again that the results in Table 7.3 are all based on Likert scales between 1-5 with 1 being the most negative response and 5 being the most positive. This is with the exception of the amount or quantity of feedback, in which the closer the response was to 3 is the more satisfied the student was with the amount of feedback they received. See the preliminary investigation in Section 6.3.2.1 for a more detailed explanation of the scale for this question.

7.3.4 Threats to Validity

One threat to validity, that could impact the qualitative data gathered from questionnaires and focus groups, is that some of the students had regular contact with the primary researcher in a teaching capacity. This could mean that students could have been overly positive or negative based on their personal opinion of the researcher. This makes it difficult to be sure that the results are not biased in anyway.

7.3.5 Evaluation

This section evaluates the primary foci of this thesis which is ultimately to determine whether or not the SWATT system of feedback delivery for programming assessment provides any benefit to students. The research questions addressed are shown in shown in Table 7.4.

RQ	Research Question	Considered
RQ1	Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?	✓
RQ2	Do students opt-in to share their code and associated feedback?	✓
RQ3	Which students tend to opt-in? E.g. weaker or stronger students?	✓
RQ4	Do students perceive benefit from having access to other students' code and associated feedback tags?	✓
RQ5	Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?	
RQ6	How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?	

Table 7.4: Research questions considered in section 7.3

7.3.5.1 Evaluating the SWATT approach - Student Perceptions

This investigation has presented the most positive student engagement with SWATT generated feedback of all previous preliminary investigations. However, a number of positive and negative issues were noted by students and are discussed in this evaluation.

One of the most prominent issues, which received a great deal of positive comments from students, was the ability for the SWATT approach to provide both a high level overview in the form of the tag cloud, as well as allowing deeper exploration of where in the student's original work a feedback tag was associated. Both the questionnaire responses and focus groups have made

reference to this feature. Another related and equally important feature noticed by students was the fact they could see their original programming work with the feedback alongside. With reference to tag based feedback a student commented in a focus group that "I very much like being able to see how it refers to specific methods, like in specific code blocks...". This positive response could be because students are used to receiving feedback that is separated from their original work and therefore receiving feedback embedded within their original programming work is novel to them. These responses confirm that feedback presented in the context of students' original submitted work is perceived as being beneficial, as is implied by the literature in Section 2.4.

The medium of delivery was also commented on as being particularly useful. It was noted, especially in the focus groups, that participants preferred feedback delivered online rather than via paper. A majority of participants in the focus groups reported they would be less likely to look at paper based feedback as opposed to electronically delivered feedback. This benefit of using electronic feedback was confirmed by the 100% feedback access rate recorded by the system. It was noted by one participant, that whilst they would collect paper based feedback, they are most likely to look at it once and then to file it away in a folder and never use it again. "I would want to see what it said but without all of the feedback tags, I would just look and think oh, ok. Chuck it in my folder and be done with it." The participants also reported they liked the SWATT system because it "made you interact with it to find out more".

Out of those students who opted into sharing, the most common positive aspect to be reported in the questionnaires was that they liked being able to compare their work and feedback to others in the cohort. "It allows you to view the comments given to other students!". Similar responses were given to a number of questionnaires when asking about the respondents favourite aspect of the SWATT sharing facilities.

The primary criticism, which has been noted throughout the preliminary

investigations and again in this investigation, is the limitations associated with using shorter feedback. Focus group results confirm that participants preferred feedback that was very specific to their work and focused on specific issues. They mentioned how some of the feedback they had received in their tags was too general and they would have preferred tags that were more specific to their work and less generic or reusable. Once again the suggestion of providing the tags with additional metadata to help students understand how to improve if they receive a given tag was mentioned as a potential solution. The questionnaire results also identify the inability of feedback tags on their own to enable students to take corrective action without further investigation being required.

Further exploration of the inability for feedback tags to provide in depth feedback within the focus group led to a discussion of the feedback tags acting as a starting point for students to be able to find out more. During the group interview the researcher asked "Do you think that the tags you have received were easy to understand?...". The group's consensus was that for a majority of tags they were able to use a search engine or the Java API to find additional information. One member of the group mentioned that they were actually able to understand the meaning of their own tags by looking at how the same tags were used in the feedback of their peers. This student had opted to share their feedback. It was also noted by two participants, one in each focus group, that having to investigate the meaning of the feedback tags was a positive point, as they were then more likely to remember the corrective action if they had to do some form of investigation using external sources. It is important to note this behaviour was not reported by all participants, at least one participant did not think to try any form of additional investigation on the feedback tags at the time of receipt.

One of the problems identified in the literature section of this thesis is that the vocabulary of experts often differs from that of students. This problem has been seen once again in this investigation, however future use of peer assessment activities may help to mitigate this. In the meantime the inclusion

of additional metadata to support student understanding of the feedback could act as an intermediate solution. It is valuable to introduce students to the expert vocabulary so that students eventually do become familiar with it. However, care is needed to ensure that students are not overwhelmed with feedback which they cannot interpret or worse could interpret incorrectly.

Another limitation, as mentioned by one student in Focus Group 2, was that the SWATT system provides qualitative feedback but is unable to provide quantitative feedback. That is, that whilst feedback tags can highlight areas of strength and weakness, they are unable, in their current form, to tell you how exactly strong or weak an aspect of a student's work is. An example given by participants was, if the tag "good Javadoc" was received, it is unclear quantitatively how good the student's java documentation is. A proforma sheet often provides a scale which allows students to identify from the summary, quantitatively, how well they have performed on a given high level aspect of their work. This appears to be one area which traditional methods of feedback are superior to tag based feedback in its current form.

When asked how students would compare the SWATT approach to traditional approaches they have experienced, a majority of students (56%) indicated that the SWATT approach is useful but should be used in combination with other methods. Over one third of respondents however, said that the SWATT approach was better and a possible replacement for 'traditional approaches' for programming assessment. In a focus group one student reported that they felt that traditional paper based feedback "... comes sort of detached ... and it has no bearing, no reflection on any future thought processes...". This comment suggests two things, firstly is that feedback delivered in isolation from the student's work is "detached" and secondly that it is less likely to be used to further the student's learning.

A total of 100% of students responded that they would like to see the SWATT approach used in future assessments. This indicates that despite the reported lack of detail in tag based feedback students still value it as a form of feedback. This finding was also reported from the focus groups where

general positive comments were given about the system.

RQ1: Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?

The primary method of evaluating whether or not the SWATT approach was successful was to measure four attributes: the students' perceived ability to improve, their ability to understand the feedback, the quality of the feedback and finally how satisfied they are with the amount of the feedback they received. These were measured using questionnaire results; the overall finding was as follows:

- 81% found their feedback tags "Useful" or "Very Useful" for improving their work.
- 94% found their feedback was "Easy" or "Very Easy" to understand.
- 72% found that their feedback was either of a "Good" or "Very Good" quality.
- 56% found that their feedback was "About Right" in terms of quantity.

It is clear that, for the first three metrics, a majority of students were satisfied with the SWATT approach to feedback. These high percentages indicate that, on the whole, students perceived a significant benefit from receiving tag based feedback for their programming assignment. The comments from the questionnaires and the focus groups both report that students had the need to perform additional research in order to understand some of the more complex, domain specific feedback tags. This, however, has not reflected in the questionnaire result for understandability in which 94% rated their feedback as being "Easy" or "Very Easy" to understand.

The final metric of investigating how sufficient the quantity or amount of feedback has yielded a mixed reaction. It is clear from statistical analysis that there is no significant difference between those who shared and those who did not and their response to this topic. However, it should be noted that for students who did opt to share, there is a slightly higher frequency

who were satisfied with the amount of feedback they received. This mixture of responses may be due to the number of tags given to each student varying. The division of opinion on the quantity of feedback received could suggest that, in this particular investigation, the use of feedback tags have not significantly improved student perceptions of how much feedback they have received. It should however, be noted that during the focus groups, participants came to a consensus that the quantity of the feedback is less important than the quality.

As previously mentioned, one student commented that they liked the fact that it presented a "general theme [as] to how you've done instantly but then [allows you to] drill into certain areas to get more information". This is a positive finding as this comment directly refers to one of the fundamental intended benefits of the SWATT approach. It is this benefit which, when compared to traditional mechanisms of feedback delivery, the SWATT approach appears to be superior according to focus group participants and questionnaire responses. Traditional feedback such as proforma sheets are able to provide summary feedback but are less able to highlight specific issues within students work due to the mode of delivery being isolated from the students' original work.

7.3.5.2 The Importance of Timing

An important aspect of this investigation, when compared with the previous investigations, was the schedule of feedback release. In the preliminary investigations the feedback tags were released slowly, and in one case, after the summative marks were given to the students. This had led to students not engaging with the feedback because they had already been given their final marks. For this experiment, the feedback tags were released rapidly within 1.5 weeks after submission and the marks a few weeks later. This rapid release of feedback seemed to encourage much more engagement with the feedback and as a result a higher level of satisfaction from students.

It is clear from the literature, as discussed in Chapter 2, that swift release

of feedback is important to student learning, but the order of release seems to have been crucially important in this investigation. That is, the feedback tags were released significantly before the final summative marks. One side effect of delivering the feedback tags in advance of the summative marks was mentioned in one of the focus groups. One student commented that they liked being able to use the tag cloud of their feedback in "...totting up the good comments against the bad comments and if there were more good than bad... I realised that I didn't do half bad". This two staged release of feedback and marks seemed to have resulted in a richer engagement with the feedback than would have happened had the marks been released alongside or before the feedback tags. This theme was discussed in both focus groups and the unanimous response was that the participants had all engaged with their feedback much more as a result of the timing of its release and the release of the summative marks. One student summarised how they engaged with their feedback using the SWATT system, "It is good because it forces you to go back and look at your code again, because if you just get a sheet of paper, you go yeah right fantastic, next. Whereas, having the comments next to the code, with the comments being not massively detailed you have to look at your code and you have to work it out for yourself."

The process of exploring one's own feedback for the purpose of trying to estimate the summative results is particularly important. Perhaps it was this process that led to a higher student engagement in their feedback on this occasion. This phenomena and effect of changing the order of delivery of marks and feedback tags has been explored and is discussed in Section 2.4.2. It is clear that the results presented in this section have confirmed the behaviour identified in (Black and Wiliam, 1998) concerning the timing of feedback.

7.3.5.3 Investigating Sharing

This thesis has presented three versions of similar investigations using different cohorts and different projects. However, despite this, the proportion of

students who opted to share their feedback and associated programming source code has remained remarkably constant. In two of the three cases 42% opted to share and in the other case it was 43%. It could be that this percentage of students is the typical proportion which are comfortable with sharing their work and feedback. Statistically there is very little difference between the responses and behaviour of those who did or did not opt into the sharing functionality.

RQ2: Do students opt-in to share their code and associated feedback?

Throughout the questionnaires, detailed reasons were given as to why students opted into the sharing scheme. These include the following key motivations:

- **Checking up on examiners** - Students, especially in preliminary investigations, reported they wanted to see what the examiner was commenting on in other peoples' work and to check for examiner consistency.

- **Competition** - Some students desired to see how well they had done in comparison to others.

- **Confidence** - Some students opted to share their work for no apparent benefit to themselves. They reported that they did not actually look at any other students' work but felt as though they wanted to help other people by sharing theirs. Automatic collection data confirmed that some students did share and did not look at anyone else's work or feedback.

- **Curiosity** - Some students reported that they were just curious as to how their peers had approached the same problem using different solutions.

- **Learn From Others Mistakes** - Some students reported that they had a desire to learn from other peoples mistakes and ensure they did not make them in future.

- **Understand Feedback Better** - At least one student, in the questionnaire, reported they had shared so that they could see how other students feedback was similar to their own, for the purpose of better understanding their own feedback.

These motivations seem reasonable and were reflected consistently through the preliminary investigations as well as this final investigation in questionnaires and focus groups.

Due to the relatively small number of questionnaire comments providing motivations for students not wanting to share, the need for Focus Group 2 was determined. The focus group did provide an insight into why some students would not want to share their work. These motivations are summarised below.

- **Distrust of Anonymity** - Some students reported that they did not trust that their peers would be unable to identify them through their code.

- **Fear of Being Discontent** - At least one student was concerned that they would realise their work was significantly inferior to that of their peers, if they could see other peoples' work and as such did not want to know how well others had done.

- **Forgetfulness** - At least two of the participants selected in Focus Group 2 reported that they had actually intended to share their work but had forgotten to login and do it.

- **Lack of Confidence** - Both in questionnaires and focus groups, at least one student reported that they did not think their work was good enough to share and were worried about the standard of their source code. Some participants suggested that they thought no one could possibly gain benefit from seeing their work and so decided to not share.

- **Lack of Interest** - One participant reported that they were not interested in other peoples' work or feedback as they could not see how it would help them in their learning.

- **Paranoia** - There was a concern expressed that a student could share their work and if it was regarded as being inferior by a group of peers and this inferiority was discussed in public, the owner of the work could possibly overhear and would feel victimised personally even if the peers did not know who it was, the owner would.

- **Social / Informal Sharers** - At least two participants in the focus groups confirmed that they had shared their programming source code informally and outside of the SWATT system. They said they preferred discussing face-to-face their feedback and work with their peers and in some cases simply logged into the system at the same time as a friend to look through each others feedback and work.

The most surprising finding was that of paranoia as described above. This fear was reported via questionnaire and was completely unexpected by the researchers. In most cases a majority of these fears would be alleviated if students were able to select exactly who was able to view their feedback. During the focus group, held with exclusively non-sharer participants, it was suggested that a 'facebook' style sharing approach would encourage more sharing between individuals. All participants who did not intend to share originally agreed that they would have selected individuals to share their work with and were largely apprehensive of blanket sharing across the whole cohort.

A total of 19% of questionnaire respondents stated that they would not share their work or feedback no matter what, when asked if anonymity made a difference to their decision to share / not to share. However, a majority of respondents reported that they would still have shared their work irrespective of whether it was anonymous or not. This could indicate a desire to learn from each other or perhaps it could allude to students being proud of their work, which is understandable due to the length and nature of the project. A total of 28% of respondents confirmed that they would not have shared at all had the system not provided some degree of anonymity.

As previously mentioned, 42% of a cohort appears to be a common proportion across all investigations presented in this thesis. It is clear from this that, on average, less than half of all students in a cohort have shared their feedback and programming work with their peers. However, based on questionnaire data and focus groups the comments from those who did opt to share their work were largely positive.

RQ3: Which students tend to opt-in e.g. weaker or stronger students? According to statistical analysis, there are no apparent statistical differences between those who opt to share and those who do not. This includes perceptions of tag based feedback, in which all responses were largely positive, as well as in terms of academic achievement. However, when considering the distributions of marks presented in Section 5.3 and those presented in this investigation, it is clear that a higher proportion of students at the lower and higher extremes of marks tend to opt to share their feedback and work, unlike those who achieved a mid range mark. It is particularly interesting that this pattern holds in this investigation because the students were unaware of their summative marks when they chose whether to opt into the sharing scheme or not.

It can be hypothesised that the mid ranged students are less interested in improving their future work based on their feedback or perhaps they are satisfied with their given marks and have no desire to improve on them. There could potentially be a link between surface and deep approaches to learning and how students use tag based feedback. All of these claims however, would require much more experimental work and are outside the scope of this investigatory thesis.

Perhaps the weaker students who receive critical feedback intend to improve their work by viewing the feedback of their peers or perhaps they intend to see how their work compares to the others'. It has become clear from one of the focus groups that some stronger students decided to share their work purely to help other people. However, others did so to allow them to investigate other programming approaches used by their peers.

Largely the answer to this research question is that within the localised scope of the investigations presented in this thesis, and ignoring statistical significance tests, both the stronger and the weaker students tend to share, with the mid ranged students opting not to engage with the sharing activities.

RQ4: Do students perceive benefit from having access to other students' code and associated feedback tags?
Questionnaire results show, that out of all respondents who opted into the sharing scheme, 73% of them found it useful being able to see the feedback and source code of their peers. Only 1 respondent reported that it was not useful seeing their peers' work. The remaining 24% of respondents, stated that they had shared their work but, at the time of completing the questionnaire, they had not looked at their peers work. This could be due to that at the time the respondent filled out the questionnaire, few people had shared and there may not have been much to look at. This problem was mentioned in a few of the early freeform comments in the questionnaire. Since the questionnaire was started the same time that the feedback was released, the early respondents who shared may have done so before anyone else had. This would mean that they could only see their own work until someone else opted into the sharing aspects of the system. Another possibility is that some students were happy for everyone to see their work and feedback, without having the desire to view anyone elses'. This possibility was confirmed by one focus group respondent who reported that he had shared his work for other peoples benefit and did not actually look at any other shared work.

One of the benefits, which was commented on numerous times by respondents who opted into the sharing scheme, was that they felt particular benefit in being able to compare the different ways of designing or implementing the project with their own way. Another commonly used explanation for why it was useful to see other peoples work and feedback was that individuals wanted to determine whether anyone else had "made the same mistake..." as them. This is an indication that the system may have helped dispel the idea that an individual's mistakes are entirely unique, thereby improving student

confidence.

One student commented that they wanted to share in order to view other peoples' work so that they can compare the standard of their own work to that of their peers. This indicates that the student wanted to try and position themselves into a ranking based on their peers' work. This idea may have spawned from the fact that the summative marks were not released until 2 weeks after the feedback tags were, thus students could have been trying to guess their marks.

There was a mixed response when students were asked whether they thought viewing other peoples' work and feedback helped them understand their own. A majority of respondents, (61%), who had shared their work reported that it did not help them in understanding their own work any better. A total of 38% suggested it did in someway help them to understand their own feedback. A majority of students, who thought that it did help them to understand their own work and feedback better, cited that it was useful to "add perspective" and see how other people had implemented comparable designs. A majority of students who said that viewing other peoples' feedback did not help them understand their own any better stated that this was because they felt they had already sufficiently understood their individual feedback tags.

One student stated that they found it very difficult to explore other peoples' work as it was unfamiliar and difficult to navigate. This is perhaps the case for some of the larger, more advanced, implementations that used numerous classes and separate packages. However, as mentioned in Chapter 2, being able to comprehend other peoples' code is an important skill in itself that must be taught along with programming. Perhaps the SWATT approach of facilitating sharing may support development of code comprehension skills in student users, especially if used with a peer assessment activity. That being said, it is clear for those who did share, that a majority of them found some form of benefit in doing so. Whether it was being given access to a variety of approaches to solving the same problem, or simply providing them

with the ability to compare their design to other peoples'.

7.3.5.4 Discussion of Different Behaviour Exhibited by Students

Throughout this investigation and all of the preliminary investigations, it has become clear that students used and interacted with the SWATT approach in numerous different ways. Some of these were noted through analysis of the automatically collected data; others were reported by participants in focus groups or from the questionnaire data.

The process for analysing the automatically collected data involved manually reviewing each students electronic usage logs and identifying patterns in their behaviour. This was done using the snapshots collected by the SWATT system.

Since some of the group behaviours were discovered through the focus groups, it is unclear as to exactly the frequency distribution of each, therefore little quantitative data is available to inform how many of the cohort fall into each group. A summarised description of the different types of typical student usages or groups is listed below:

- **Explorers** - This group of students appeared to repeatedly login to the system over a wide spread of dates and times and on each occasion they explored one of the projects shared by their peers. This group of students viewed both feedback tags and the associated source code.

- **Informal Sharers** - These students decided not to share their work using the SWATT environment; instead they informally discussed and shared their work and feedback with their friends face-to-face.

- **Librarians** - Some students, who did not share their work, reported that they had used the SWATT system as a personal library of source code that they could reuse or look at to make improvements to their future work.

- **One-off Viewers (Non-Sharer)** - Students in this group logged in

once and explored their own feedback and explored it in the context of their source code but did not use the system more than once.

- **One-off Viewers (Sharer)** - Students in this group logged in once and explored their own feedback and other peoples' feedback and source code but did not use the system more than once. In this case, it was clear students were more interested in viewing the feedback tags of their peers not necessarily the associated source code.

- **Surface Users** - Students in this group simply logged in once, looked at their feedback tag cloud and did not at any point explore the system or view their tags alongside their own work.

The group 'Explorers' was detected by analysing the automatic data logs collected from the system. Students within this group shared immediately. They then appeared to, over the course of the month, systematically explore the feedback that was shared by their peers. It also appeared that they took interest in reviewing the source code submitted by other students. This group was equally small with only two apparent cases where this type of prolonged usage of the system occurred.

'Informal Sharing' was a threat to validity mentioned in both preliminary investigations presented in this thesis. This process of informally sharing was quite common and was confirmed through the results from the focus groups. It was mentioned that students preferred showing their friends and discussing their work and feedback in a face-to-face environment. Students in this cohort have admitted to using the SWATT system to share their work and feedback, but the sharing occurred by simply showing their friends the screen instead of using the sharing functionality provided by the system. This type of sharing was unmonitored and would not have been detected using automated data collection methods.

It became apparent from both data collected from the usage of the system and the focus groups, that some students used the SWATT system purely to view their own code on a regular basis. After further investigation, it appeared

that the students were using the system as a central point where they could access their code to be reused in different programming work. These students have been labelled 'Librarians' as they seem to have used the system to keep a personal library of their work and feedback. Thus the students had adapted the system to suit their own purposes as a central source code repository of their work and attached feedback. Two participants in the focus groups reported that they had used the SWATT system on multiple occasions in order to make sure they were not making the same mistakes again in their current programming work.

The two most commonly noticed groups of students are the 'One-off Viewers' (Non-Sharer) and (Sharer) groups. Students who have been classified as being a part of these two groups used the system once to view their feedback and/or the feedback of their peers. It is clear that there is a subset of students within this group who viewed their own feedback and then opted to share their work but at no point viewed any other students' work in exchange. One such student was a participant in a focus group and simply stated he was happy for other people to see how he had approached the problem but had no need to explore other peoples' work. However, upon further discussion it became apparent that the student had in fact shared their programming work informally with their friends. This group of users appeared to explore their own and/or others' feedback in detail but they only did so once.

The 'Surface Users' group is a small group of students detected through reviewing the automatic data collected from the systems usage. Two students out of the cohort appeared to login and view their feedback tag cloud and summary page but did not perform any other interactions with their feedback. This includes not opting to share their feedback. Due to the logging being anonymous it is unclear as to what other factors may have influenced this behaviour. The term 'Surface Users' has been borrowed from educational literature, specifically that of Deep and Surface learning discussed in Section 2.2.5, as it implies students in this group have only glimpsed the surface of their feedback and have not fully explored the meaning of it.

It is clear from this investigation that students adapt and use the SWATT system in different ways to try and effectively augment their personal learning. For some of the students it is clear that they treated the tag based feedback as any other type of feedback and looked at it once and never looked at it again. However, for a majority of students whilst they only used the system on one occasion, they did interact with their feedback and explored it thoroughly during that one session.

7.4 Sentiment Analysis of Feedback Tags

7.4.1 Introduction

The secondary investigation presented in this section explores whether or not feedback tags, when considered in isolation, can communicate sentiment information between examiners and students without the need of additional sentiment data. Therefore, this investigation is primarily focused on answering research question 6.

The research questions to be focused on in this section are as follows:

- *RQ5 Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?*
- *RQ6 How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?*

7.4.2 Research Method

This investigation is closely modelled on the sentiment analysis aspects of the previous preliminary investigations. See Sections 5.4 and 6.4 for detailed explanation of the research methods employed.

For this investigation, the same process was used for human respondents to evaluate the sentiment of a subset of the feedback tags issued, as in the investigation presented in Section 6.4. The 20 most frequently used tags were selected for this sample. The sample was limited to 20 tags to reduce the risk of fatigue affecting the respondents' ability to engage in careful sentiment analysis.

This investigation was the same as the preliminary investigations in that three examiners were recruited as participants and the same version of the NaCTeM automated sentiment analysis tool was used. The main difference in this investigation, when compared to the one outlined in Section 6.4, is the number of student participants involved and how their analysis results are to be contrasted against the examiners' and the automated tool.

It was decided to try and recruit a larger sample of students from the cohort instead of simply having the same relatively small number of student participants as there are examiners. For this investigation, an electronic questionnaire was setup that provided a mechanism for students to analyse the sentiment of the twenty feedback tags by providing a response that was either "positive", "negative" or "neutral" for each one.

The electronic questionnaire was open for four days only and a prize draw for a gift voucher was given as incentive for completion of the questionnaire.

For this investigation, it was decided to handle disagreements between the students and examiners responses by taking the majority response as the sentiment opinion for that group. This decision was necessary since the number of student participants was significantly higher than the number of examiners and it is deemed more useful to be able to compare the modal responses between the groups.

7.4.3 Results

A total of 25/45 students participated in the online sentiment analysis questionnaire representing 55.55% of the cohort. There were 2 feedback tags where there was no majority between student participants' responses. These two

tags were "specific to blackjack" and "move to different class", both of which had a 48% divide between students perceiving them as either Negative and Neutral and 4% thought these tags were positive. The "specific to blackjack" tag highlights an aspect of the student's work which could be made reusable but instead the student has made the feature or class very specialised to the project. Whereas, "move to different class" highlights an aspect of a student's source code, which could make more sense if it were reorganised to be a member of a different class. The examiners agreed unanimously that "move to different class" was a neutral tag. However, there was one examiner who thought "specific to blackjack" was negative, whilst the other two reported it as being neutral. It appears there is not unanimous agreement between examiners on this particular feedback tag.

In total, there were 11 disagreements between student majorities and staff majorities for sentiment analysis of this particular sample. This represents a 55% disagreement rate between students and examiners. Table 7.5 and Figure 7.2 show the distribution of tags by participant type. It is clear from this table that once again, as in the preliminary investigation as discussed in Section 5.4, examiners are more often identifying feedback tags as being neutral, whereas students are more often identifying tags as having a polar sentiment. In this particular investigation the students have identified a majority of tags as being negative.

Sentiment	Students	Examiners	NaCTeM Tool
Positive	20%	20%	25%
Negative	55%	20%	5%
Neutral	15%	60%	70%
No Agreement	10%	0%	0%

Table 7.5: Sentiment analysis: percentages of feedback tags in each sentiment category

The NaCTeM automated tool is once again more aligned with the examiners opinion in this sample. However, it has failed to identify as many negative

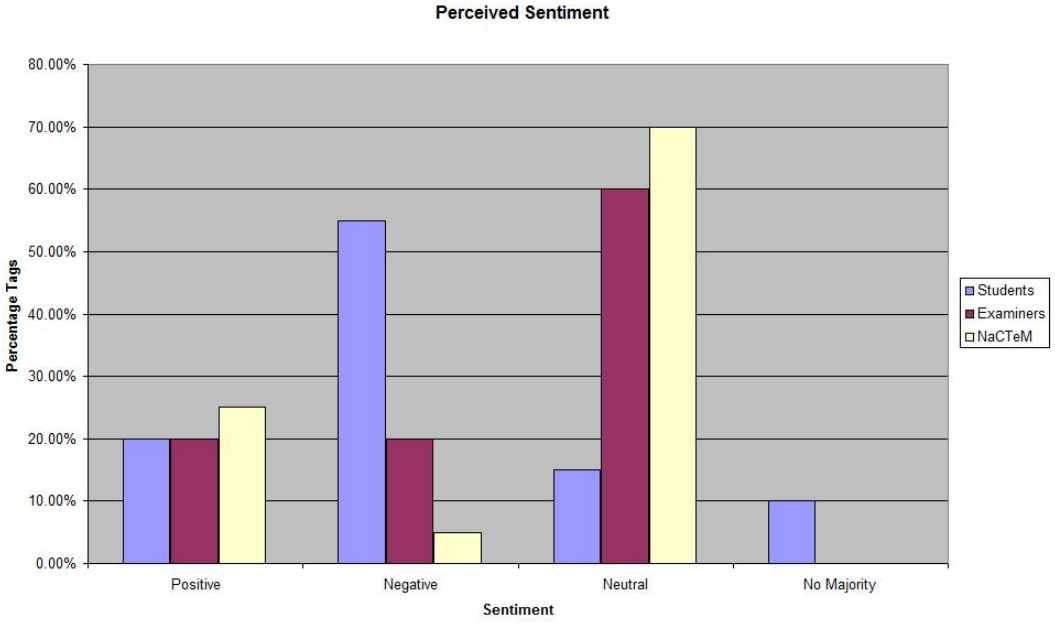

Figure 7.2: Graph showing distribution of perceived sentiment

tags as both the examiners and students have. This suggests that on this occasion the NaCTeM tool was too conservative about what it interpreted as being negative feedback. It was also particularly generous about one tag, which it classified as being positive and the majority of human participants perceived it as being negative; this tag was "consider a more elegant approach". Both examiners and students modal analysis result was that this tag was Negative. However, amongst the students this was not an overwhelming majority. A total of 12% of students thought this feedback tag was positive, 56% thought it was negative and 32% reported it as being neutral.

After reviewing the students' analysis results, there were very few who had a unanimous agreement between all 25 participants. There were, in fact, only 2/20 tags that had unanimous agreement as to the sentiment within the students analysis. These were "good error handling" and "good javadoc", both noted as being positive tags. Based entirely on student sentiment analysis, there were 75% (15/20) tags that had at least one participant providing each

of the possible responses. That is some students thought a given tag was positive, whilst others thought the same tag was negative or neutral. This occurred for 75% of the sample, however despite this, in most cases there was a clear majority response from student participants.

7.4.4 Threats to Validity

Once again the threats to validity in this study could be influenced by the fact that human participants may have become fatigued from the activity of manual sentiment analysis. However, this effect has been mitigated by reducing the number of feedback tags in the sample to be analysed and increasing the number of student respondents.

7.4.5 Evaluation

The research questions considered in this section are shown in Table 7.6.

RQ	Research Question	Considered
RQ1	Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?	
RQ2	Do students opt-in to share their code and associated feedback?	
RQ3	Which students tend to opt-in? E.g. weaker or stronger students?	
RQ4	Do students perceive benefit from having access to other students' code and associated feedback tags?	
RQ5	Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?	✓
RQ6	How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?	✓

Table 7.6: Research questions considered in sections 7.4 and 7.5

The results from this investigation demonstrate how varied human perceptions of sentiment are, especially when considering feedback delivered in

the form of tags. Students provided the most diverse range of opinions for the feedback tags given with only 2 tags receiving complete agreement, and 3 tags receiving entirely positive and neutral responses or negative and neutral. The remaining 15 tags received the full range of sentiments. This indicates that despite the fact most of the tags did have a majority verdict as to the perceived sentiment, there is substantial difficulty for students in interpreting and agreeing on the underlying sentiment of feedback delivered in tag form.

Examiners have been noted as having a higher agreement when compared to students, however this is not necessarily representative due to the smaller sample of examiner participants involved. It has been noted, however, that there are only 3 occasions where there was not unanimous agreement between examiners on the sentiment of a feedback tag. On each occasion it is only one examiner that disagrees with the other two. This could allude to the possibility of examiners having a shared vocabulary or understanding of the complex terminology from which the feedback tags were composed and to which the students may not yet have access to.

With regards to how student participants compared with examiners perceptions of sentiment, it is clear that once again substantial differences between the two groups exist. On this occasion, as in the first preliminary investigation, examiners have perceived a majority of the feedback as being neutral, whilst students perceived more negative tags. This is particularly concerning as should an incident occur whereby an examiner makes a neutral suggestion for improvement but a student misinterprets this as being negative; the student could become demotivated. This situation could also cause an imbalance in the so called feedback sandwich as introduced in Chapter 2. It should be noted that on no occasion has an examiner perceived a feedback tag as being positive and the student majority has not agreed with examiner perceptions. It seems positive feedback is clearly differentiated and agreed upon by human participants in this sample of feedback tags.

It should be noted once again, that this investigation is somewhat restricted artificially as the feedback tags were considered independently of their

associated source code fragments. This means that the human respondents were required to interpret the feedback tags outside the context to which the tag was created. This is a disadvantage because tags are heavily reliant on their context; in fact it is their context that gives them applied meaning. This decision was taken to keep the investigation fair, since the NaCTeM tool is unable to benefit from this context information, it was decided to impose the same restriction on human participants to enable comparison between human analysis and automated analysis.

7.4.5.1 RQ6: How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?

It is clear, from the diversity of student perceptions of the sentiment of feedback tags, that feedback in this form does not clearly embody the intended sentiment. Supplementary research should be conducted to allow comparison between other forms of feedback and tag based feedback to determine which one is empirically better in this respect. However, as identified through focus groups presented in Section 7.3.3.3 and by the preliminary investigations, this lack of clarity could be remedied by using additional metadata attached to the feedback tags. This metadata could be displayed by colour coding feedback according to the intended sentiment. This additional sentiment data could either be encoded into the feedback tags using automated tools such as the NaCTeM tool or by the examiner upon tag creation, or perhaps a hybrid of the two approaches.

As previously, mentioned there are problems with the automated tool's ability to cope with highly technical subject specific terminology or feedback referring to high level concepts. This would mean that examiners would have to verify the sentiment selected by the automated engine before the feedback was released. However, the NaCTeM tool provides a reasonable approximation of how a human could perceive feedback delivered in tag form and this has

been confirmed by both preliminary investigations. Unfortunately, on this occasion, the feedback tags used had a large amount of technical terminology and made references to high level concepts which occurred especially with the negative feedback. On this occasion the NaCTeM tool was unable to identify the same amount of negative tags as either of the human participant groups.

The final investigation result is that there is a 55% difference in how two different groups of human participants interpret the sentiment of feedback tags in given sample of 20 tags. This is actually the lowest agreement rate when compared to the two previous preliminary investigations of 88% and 65%. However this investigation did have a smaller sample of feedback tags for analysis. It is clear that feedback tags on their own do not sufficiently communicate the sentiment of feedback, however it is difficult to contrast feedback tags to other feedback approaches as there is little literature or previous investigations which allow such comparison.

7.5 Extending Sentiment Analysis of Feedback Tags: Using Thematic Analysis

This section investigates the combination of sentiment and thematic analysis data to determine if any additional information can be derived about cohort learning. The cross analysis approach is very similar to that described in Section 5.5 and 6.5.

Once again the process for thematic analysis was followed, as described in Chapter 4, and the investigation was run in the same way as in the previous preliminary investigations in Section 5.5.1 and 6.5.1. The only difference is that an additional thematic analysis process was run using new tags derived from the mark scheme provided with the module. The results of both sets of analysis are presented in this section along with the sentiment analysis data for the respective themes.

This section focuses exclusively on addressing:

- *RQ5 Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?.*

7.5.1 Investigation Method

Thematic analysis, using the original themes that were used throughout all preliminary investigations, are used in this final investigation. In addition to this, an analysis with a set of themes derived exclusively from the mark scheme for the student's project, is included. This additional set of themes was included in this investigation primarily because, in the preliminary investigations, there was little documentation available that detailed assessment criteria which could easily translated into themes, whereas this particular project did have such documentation available.

The new set of themes is described as follows and were derived from the project assessment criteria sheet.

- **Object Structure and Encapsulation** - This theme contains feedback tags that focus on how students have applied object orientated design techniques in the solving of the task. It included issues such as: data hiding, modularisation, duplication of code, use of exceptions, reusability of code, scope of variables.

- **Miscellaneous** - This theme is used for tags which do not relate to any of the other themes or are too vague to classify such as "good".

- **Selection of Data Types and Structures** - This theme holds all feedback tags that comment upon the selection of appropriate data types or structures, for example, if the examiner has identified an appropriate or inappropriate use of a data structure.

- **Style** - This theme would encompass all feedback that focuses on how well documented the code is or how easy it is to read and understand. Therefore commenting and Javadoc feedback would commonly be found in this theme. Feedback relating to creative or efficient approaches to problem solving may be included in this theme.

- **Use of Data Types and Structures** - This category represents how well the students have used data structures or the Java Application Programming Interface (API).

As a consequence of introducing the additional set of themes, the process of blind review used in the preliminary investigations to achieve a thematic analysis agreement rate of at least 80% must be done for both the original set of tags and the new sets of tags.

A total of 132 tags were analysed by the primary researcher, using both the original themes and the new themes as derived from the assessment mark sheet. A sample of 40 of the most frequently used tags was given to two reviewers, this represents 30% of the tags used to assess this assignment. The sample size was reduced from the 60% from the previous investigation due to reviewer comments suggesting it was too large.

The initial blind review process resulted in an average agreement rate of 48% and was less than the acceptance threshold for both reviewers in both sets of themes. As a result, the themes were verbally clarified and the reviewers elected to go through and modify their analysis results accordingly. The resulting reviews from both participants on the second round did meet the 80% agreement threshold. Therefore the thematic analysis process was considered as being complete.

The final review stage yielded an average agreement result of 82.5% for the original themes as were used in the preliminary investigations. The average agreement rate for the new themes derived from the mark scheme was 91.25%, significantly higher than that of the original themes. This is possibly due to the newer themes being more specific to the assessment activity, whereas the original themes are general and intended to cover a wide range of programming and software engineering assessments.

7.5.2 Results

7.5.2.1 Original Themes

Table 7.7 presents the distribution of tags according to the original themes. As with all previous preliminary investigations, the most frequently occurring theme within the feedback is 'Programming Standards'.

The sentiment analysis data taken from the NaCTeM automatic sentiment analysis tool was combined with the thematic analysis data collected using the original themes, as shown in Figure 7.3. It is important to consider the finding presented in Section 7.4, which was that the NaCTeM tool performed poorly at identifying negative feedback within the dataset. This was especially true for the feedback tags that made reference to high level subject specific concepts. Once again a human participant's results have been included along with the NaCTeM results in Tables 7.8 and 7.10 to add perspective. Due to the difficulties of the NaCTeM tool being able to identify negative feedback tags in this dataset, the human sentiment analysis data is the focus of the

analysis.

Theme	Unique Tags	% tag uses
Completeness	3	0.39%
Comprehension	15	17.90%
Design	38	25.34%
Miscellaneous	17	12.21%
Programming Standards	59	44.15%

Table 7.7: Distribution of feedback tags in to the original themes.

	% Positive		% Negative		% Neutral	
Theme	NaCTeM	Human	NaCTeM	Human	NaCTeM	Human
Completeness	0.00	0.00	83.33	83.33	16.67	16.67
Comprehension	64.60	44.16	0.36	36.50	35.04	19.34
Design	26.55	12.89	2.06	65.72	71.39	21.39
Miscellaneous	42.25	41.71	16.04	52.94	41.71	5.35
Programming Standards	22.19	24.11	5.03	8.88	72.78	67.01

Table 7.8: Sentiment analysis presented in context of thematic analysis data (original themes)

It is clear that the NaCTeM results in Figure 7.3 show a different perspective when compared to the human examiners results in Figure 7.4. In the preliminary investigations, the amount of discrepancy between the NaCTeM analysis and human analysis was low enough for the automated results to be used without much reference to the human generated results, however on this occasion the NaCTeM approach does not provide a sufficient match. This is the motivation for focusing on the human analysis results for this final investigation.

Figure 7.4 shows that the 'Completeness' theme has a high proportion of negative feedback associated with it and indeed this is agreed on in both the Human and NaCTeM analysis results. This however, may be misleading as the number of tags within this theme is very small and perhaps may suggest that there is a widespread problem where in fact there is not. There were

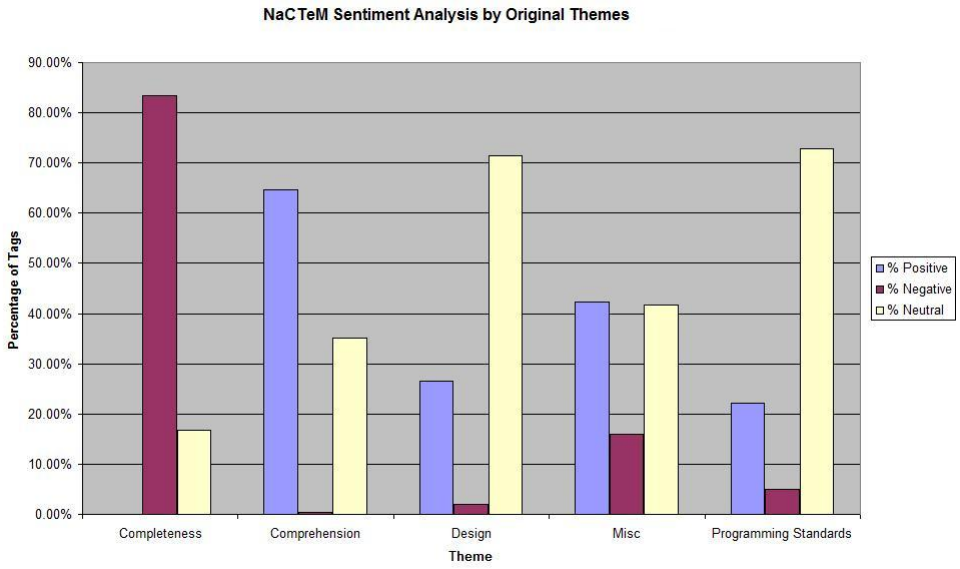

Figure 7.3: NaCTeM sentiment analysis by the original themes

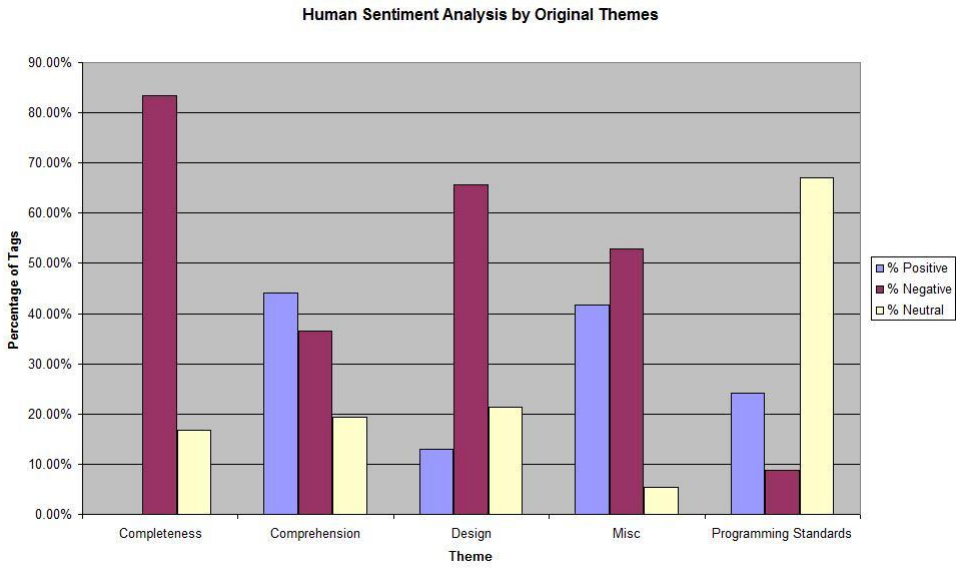

Figure 7.4: Human Sentiment analysis by original themes

only 6 usages of feedback tags in this theme and 5 of these were identified as being negative, the last one was neutral. The small number of feedback tags in this theme indicate that whilst the feedback is largely negative this is mostly a localised problem with relatively few students affected.

The 'Comprehension' theme in the human analysis shows that there is roughly a balance between positive and negative feedback. There was, in fact, a high amount of positive feedback about the cohort's usage of Javadoc, the in built Java documentation markup language. However, many students equally received feedback to indicate that there was a lack of appropriate comments and/or Javadoc in their work; hence a relatively balanced result.

The 'Design' theme appears to also have a high amount of negative feedback associated with it. This theme represents the assessments focus for this assignment, as practice in object orientated design was a key outcome from this assessment. The most frequently occurring negative tag in the 'Design' theme is "specific to blackjack". This tag refers to a component or class that is not reusable outside of the student's project and could have been made to be more generic. This however, is not a piece of feedback that is crucial to the learning objectives and whilst it has been recognised as a negative tag, it does not constitute a fatal flaw in the student's design.

The 'Miscellaneous' theme also has a high amount of negative feedback associated with it. Upon closer inspection it appears that this is caused from tags such as "never used", which questions the student's decision to included some object, field or variable that is never used within their program. While this tag highlights an aspect of the student's work which could cause additional unnecessary maintenance burden, it does not highlight a crucial deficiency in the student's work. Positive tags within this theme largely provide general feedback such as "good approach", which when considered in isolation does not identify what is actually good, but does help the student identify strengths within their work. These tags may have been intended by the examiner to increase student confidence in certain aspects of their work and may have a greater meaning if considered with the associated source

code.

The 'Programming Standards' theme has very similar sentiment results between human and NaCTeM analysis and has a high amount of neutral feedback associated with it. An example of a neutral piece of feedback within this theme is "include access modifier". This tag refers to an instance where a student is relying on undefined or ambiguous behaviour and not providing fields with specific access modifiers, which is good practice for information hiding.

7.5.2.2 New Themes

The results of using the thematic analysis technique with the themes derived from the mark scheme are shown in Table 7.9. It is clear that the tags distributed with the new themes are more balanced when compared to the original thematic analysis distribution as shown in Table 7.7.

Theme	Unique Tags	% tag uses
Miscellaneous	38	28.79%
Object Structure & Encapsulation	45	34.09%
Selection of Data Types & Structures	16	12.12%
Style	25	18.94%
Use of Data Types & Structures	8	6.06%

Table 7.9: Distribution of feedback tags in the newly derived themes.

Since the dataset being used in this section is the same as that presented in Section 7.5.2.1 the same limitations persist. As such, the data is presented in the same way as in Section 7.5.2.1 but with the new themes being used for the thematic analysis instead of the original ones.

It is interesting how using a different set of themes for feedback distribution has changed the overall outlook of the same feedback data. When comparing Figure 7.4 and 7.6, it is clear that there appears to be substantially more negative feedback when using the original themes instead of the new themes.

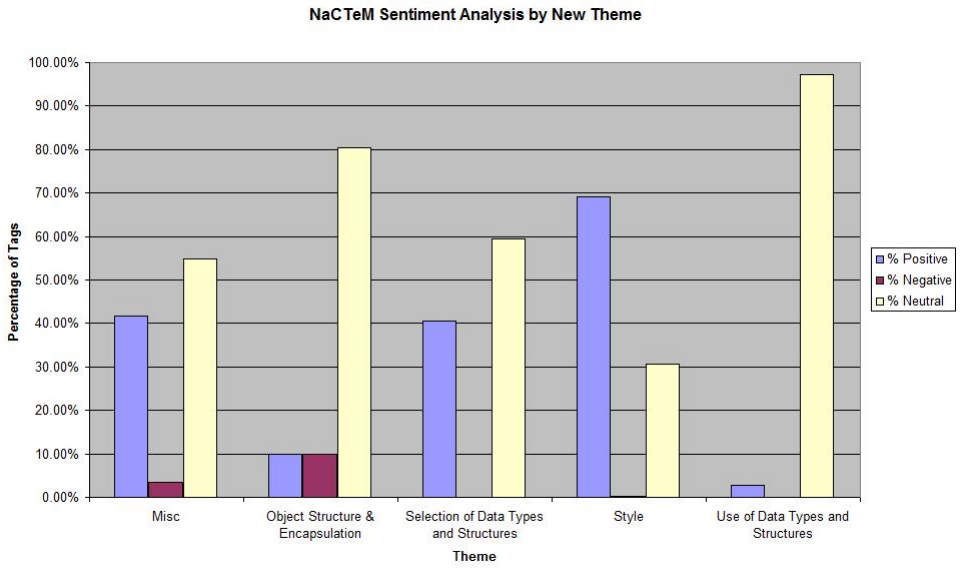

Figure 7.5: NaCTeM sentiment analysis by new themes

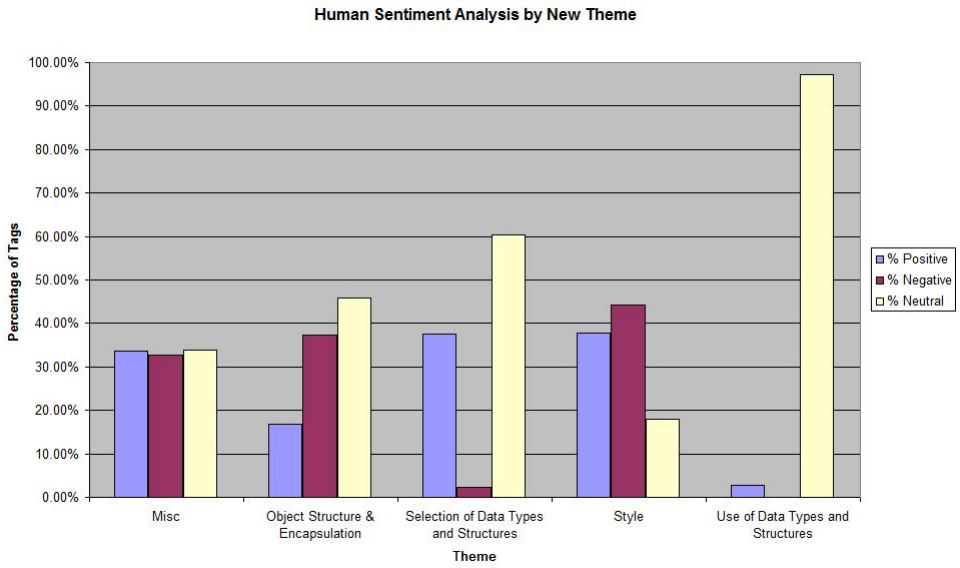

Figure 7.6: Human sentiment analysis by new themes

This is partially due to the way the graphs were generated, using percentages instead of frequencies, to show the proportion of feedback received for each theme. This, combined with the fact the original themes did not yield a very balanced distribution of feedback, meant that some of the smaller themes appeared to have extremely high levels of negative feedback because they had fewer tags overall.

It appears in Figure 7.6 that the themes 'Miscellaneous' and 'Style' have almost balanced proportions of positive and negative feedback within their respective themes. Once again the 'Miscellaneous' theme is difficult to analyse due to its very vague and general nature. The 'Style' theme, however, does have a more specific set of tags associated with it. The most frequently used tags in this theme are very similar to the 'Comprehension' theme, for example the most commonly used tag is that of "good Javadoc" which is also the case in the 'Comprehension' theme, as discussed in Section 7.5.2.1. The 'Style' theme does have slightly more negative feedback tags than any other sentiment which is concerning. Upon closer inspection, it appears that the two most commonly used negative tags were those of "consider a more elegant approach" and "more comments required". The first refers more to design decisions or elegance of the implemented solution. For example, has the student submitted a concise solution for the given programming problem or one that is convoluted and difficult to understand? "more comments required" indicates that some students may be unable to decide when it is necessary to include in-line comments within their programming work and that the

	% Positive		% Negative		% Neutral	
Theme	NaCTeM	Human	NaCTeM	Human	NaCTeM	Human
Miscellaneous	41.69	33.53	3.50	32.65	54.81	33.82
Object Structure & Encapsulation	9.83	16.79	9.83	37.37	80.33	45.84
Selection of Data Types & Structures	40.44	37.50	0.00	2.21	59.56	60.29
Style	69.01	37.75	0.28	44.23	30.70	18.03
Use of Data Types & Structures	2.78	2.78	0.00	0.00	97.22	97.22

Table 7.10: Sentiment analysis presented in context of thematic analysis data (new themes)

examiner felt comments would have been a worthwhile addition to a section of the student's work.

The 'Object Structure & Encapsulation' theme appears to have a significant proportion of negative feedback associated with it in Figure 7.6. This is for the same reason that the 'Design' theme in the original thematic analysis had a high proportion of negative tags, and is related to the high occurrence of the tags "specific to blackjack" and "duplicate code". However, this theme is fundamentally different to the 'Design' theme in the original analysis in that use of language features specific to Java are included. For example, one particular tag used in this theme is "make private", instructing students to utilise information hiding principles in their implementations. It appears that the positive tags within this theme mainly correspond to expressions of approval of how students have considered error handling and encapsulation techniques.

The theme with highest amount of positive feedback when compared to negative is that of 'Selection of Data Types & Structures'. While this may indicate that the students have largely selected appropriate types and data structures within their work, upon closer inspection a large amount of neutral feedback in this category is actually composed of suggestions for alternative choices of data structures. For example, "consider using a stack" has a neutral sentiment as it is simply a suggestion from the examiner and does not highlight an aspect of the student's work which is entirely incorrect.

The "Use of Data Types & Structures" theme, which one might expect to have negative feedback, appears to be almost entirely neutral. Primarily, this is due to the theme having a relatively small number of tags associated with it. Apart from this, the tags associated appear to be worded in a neutral way, for example one of the most frequently occurring tags is "use generics". This implies the student has forgotten to include generic type information within their source code. Whilst this could be considered a negative comment because the student has forgotten something, the way the feedback is worded is not inherently negative and as such both the human examiner and the

automated tool perceived it as being neutral. Another example of this use of neutral language within this theme is the tag "could use foreach". This tag advises the use of a simpler syntax for iterative loops but does not in itself have negative connotations. It is these type of neutral tags which appear to have been clustered in the "Use of Data Types and Structures" theme, hence the high proportion of neutral feedback.

7.5.3 Threats to Validity

The threats to validity are identical to those presented in Section 6.5.3. Since the NaCTeM tool is being relied on less in this study, the threats to validity discussed in Section 7.4 involving the use of the automated sentiment analysis tool apply to a lesser extent to this analysis.

7.5.4 Evaluation

It is clear from Section 7.5.2.1 that due to the imbalance in the allocation of tags to the original themes, which is caused by the high frequency of "Programming Standards" and the particularly low frequency of the "Completeness" theme, the results from the analysis are partially distorted. As such, the focus of the evaluation will be on the new themes derived specifically from the associated assessments mark scheme.

It is worth noting that on a number of occasions the results have shown negative feedback with relatively low importance, but which has occurred frequently within the cohort, and as such these had a significant impact on the levels of negative feedback shown in the graphs. In one sense this is desirable as it alerts the examiner that there is a widespread problem, however, it obscures the fact that the problem is of perhaps a lower importance in the context of the assessment function. One potential remedy is to incorporate a wider scale for recording the sentiment of feedback instead of simply the three options of positive, negative and neutral. Perhaps allowing the human participants to indicate roughly how negative or positive a feedback tag is,

could allow a more detailed analysis. The NaCTeM system, in fact, does provide this finer degree of analysis by using a numeric value to represent the sentiment of analysed text. The greater positive or negative the number is indicates how positive or negative a texts sentiment has been interpreted as being, with 0 being neutral. Use of this information could enable the examiner to better distinguish between highly important feedback and feedback with lower importance, but which may have been identified throughout the entire cohort.

One particular aspect of using sentiment analysis with feedback only really came to light during this in-depth combined analysis process. This is the notion of how the neutrality of the language selected by examiners during feedback tag creation can obscure problems in the feedback analysis. In Section 7.5.2.2 it was noted that many tags in the 'Use of Data Types & Structures' theme, had mainly neutral feedback associated with it and that most of these could have been easily worded to be negative. For example, "consider using a stack" could easily have been "should use a stack" or simply just "use a stack", both of these have slightly more indication that the student could have improved their work. However, the examiners decision to use neutral or less forceful language in their tag has led to some tags being interpreted as being less important by the sentiment analysis. The motivation for providing neutral feedback instead of negative to bolster student morale is an important consideration but as demonstrated in this investigation, it can impact the usefulness of automated analysis. Using neutral language when the examiner intended to identify problems or suggestions for improvement can obfuscate them from high level analysis. As such, it is important for examiners not to be too neutral in their feedback and where possible, stress important issues using either positive or negative feedback tags.

Using Figure 7.4, it is possible to immediately identify that there is a reccurring design problem in the cohort's assignments. As mentioned in Section 7.5.2.1, this is largely to do with students' failure to consider making aspects of their programs future proof or reusable outside of the particular

assignment context. This could inspire the lecturer to target reusable design strategies in their lectures, to better inform students, so that they can put these ideas in to practice in their future work.

The use of this combined approach to feedback analysis can provide the examiner with an at-a-glance view of how a cohort has preformed according to general high level themes. This can be used to narrow down the feedback tags and identify the specific issues within a theme that students have strengths or weaknesses in. The relative high cost of manual thematic analysis, in terms of researcher and reviewer time, could outweigh the benefits of being able to see how the sentiment of feedback is distributed throughout themes. This is especially true if a similar effect could be achieved through simply visualising the sentiment of feedback tags within a tag cloud and allowing the examiner to pick out the most frequently occurring negative tags from the visualisation. This process of visualising the sentiment of feedback tags in the tag cloud could be semi-automated by using a sentiment analysis engine such as NaCTeM and simply requiring the examiner to review the data generated for correctness. This extension is a likely candidate for future work using the SWATT approach but at this stage is outside the scope of this thesis.

7.5.4.1 RQ5: Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?

The use of sentiment analyse provides additional information for understanding the collective outcome of a programming assessment for a cohort. It enables the analyst to quickly, after analysis, identify both isolated and widespread areas of strength and weakness in students' learning and allow them to act upon this information accordingly. It is important for lecturers to be aware of how their students, both as individuals and as a cohort, are progressing through the course and this usage of sentiment analysis helps to facilitate this.

Using the results presented, it is apparent that sentiment analysis, when

combined with thematic analysis, provides a high level overview of students' feedback which can be used as a foundation for exploring the body of feedback to help understand the general level of the cohort's programming ability.

Thematic analysis provides a structured approach to analysis of feedback and allows for high level representation of the feedback tags. However, the benefits of having this high level overview may not be outweighed by the overall cost in man power required to conduct the analysis on the full dataset, let alone the review phases. Perhaps a less formal and rigorous review process could be applied if the technique was to be used regularly to reduce the time overheads experienced by staff.

Additional information has been collected through the use of both sentiment and thematic analysis and potentially this could be used by lecturers to adjust their teaching to the benefit of their students. The ability for lecturers to identify cohort wide strengths and weaknesses and act to address them is in line with the concept of just-in-time teaching as discussed in Chapter 2.

7.6 Chapter Overview

This chapter has used the experience gained from the preliminary investigations to conduct a final investigation in to how students use feedback tags through the SWATT prototype and the type of analysis that can be conducted on feedback delivered in this form. The conclusion for this chapter is delivered in the next and final chapter of this thesis.

Chapter 8

Conclusion and Future Work

8.1 Introduction

This chapter describes the overall conclusions based on the results and experiences gained from the preliminary investigations and the final investigation. The answers to the core research questions, as introduced in Chapter 1, are outlined with reference to the results presented in the final investigation. Throughout this chapter, the contributions to computer programming and education are outlined and the implications to existing literature are discussed with references specifically to the literature described in Chapter 2. Finally, this chapter will propose possible extension projects that could be conducted in the future.

8.2 Research Contributions

This thesis has contributed to the interdisciplinary domain of teaching introductory programming, which spans the fields of both computer science and education. It does this by investigating a new strategy of feedback delivery that operates by applying a technique found in Web 2.0 information management systems, namely that of tagging. The usage of this new feedback technique has been investigated with a view of determining whether it is

beneficial to students who are learning how to program. Exploring whether additional feedback analysis techniques are available for either students or examiners through usage of feedback tags was also investigated. In order to evaluate the ability for feedback tags to effectively communicate the underlying sentiment information contained in examiner comments to students, a process of sentiment analysis was also conducted. This involves determining how positive, negative or neutral feedback tags appear to human participants. In addition to investigating feedback tags, this thesis examined how students react to being given the ability to share their feedback tags and associated source code with their peers.

The SWATT approach appears to have provided feedback which is used by students in different ways to suit their individual style of learning, most of which have had a positive impact according to learners' perceptions. The different behaviours observed through analysis of the usage data captured by the SWATT system have been discussed in Section 7.3.5.4 and are summarised in this Chapter.

8.2.1 Answers to Research Questions

The answers to each research question is discussed in this section and summarised in Table 8.1.

8.2.1.1 RQ1: Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?

Largely, students in the final investigation were satisfied with their feedback when it was delivered as sharable tags annotated throughout their original work.

The primary weakness highlighted from the questionnaire results is the amount or quantity of feedback received. A total of 56% of students were completely satisfied, however during focus groups participants reported that the quantity of feedback was the least important out of all of the benefit

metrics defined in this investigation. The quantity of feedback is largely determined on an individual examiner basis and whether the student opted into the sharing scheme to be given access to more student feedback. The other metrics considered are the students perceived ability to improve, their perceived ability to understand the feedback, the overall perception of quality of the feedback.

The primary benefit of the SWATT approach to feedback tagging, as reported by students, was the ability for the SWATT system to present a high level overview of the student's feedback in the form of a tag cloud. In addition to this, the ability of the SWATT system to facilitate focused exploration of the feedback or to facilitate "zooming in" on specific feedback from the tag cloud, as well as allowing students to see the feedback in context were noted benefits.

The primary disadvantage, as reported by students, was the inability for some tags to convey feedback without additional metadata or external research being required by the student. A few students complained that they could not take immediate corrective action because they had to research the meaning of a feedback tag. However, it was noted in the focus group that half of the participants found the activity of researching the technical terms used in the feedback tags as being constructive to their learning and increasing their overall engagement with the feedback.

It was clear from the preliminary investigations that students prefer this type of feedback when delivered for summative individually assessed projects in contrast to group projects or formative assignments. The reason there is more engagement in summative projects is that there is a higher perception of importance associated with it from students. This is despite formative assessment and the resulting feedback being crucial for improving learning. It is expected the reason that the SWATT approach was less successful in the summative group project investigated in Chapter 5 is due, in part, to the timing of release of the feedback tags being after the summative feedback. This reinforces the findings highlighted in the literature (Black and Wiliam,

1998; Winter and Dye, 2004) concerning the order and timing of feedback.

The most important reason for a lower engagement is speculated as being the fact that group projects already have the mechanisms for informal sharing of work and feedback inherently available within the groups. That is, students are able to discuss within their teams the meaning of feedback in a face-to-face environment and as such have less need for the SWATT system to act as a conduit.

The ability for students to improve their learning in some way from the feedback is the most important criteria for success as reported from students in focus groups and a significant portion of the literature (Higgins et al., 2001) discussed in Section 2.4. Overall, it can be concluded that students surveyed, especially in the final investigation, did perceive a significant benefit in receiving feedback in the form of tags allocated throughout their source code, this is especially true as 81% reported they were able to improve their work using the feedback.

8.2.1.2 RQ2: Do students opt-in to share their code and associated feedback?

As discussed in Section 7.3.5, there has been three separate investigations, using completely separate cohorts at different stages of their respective courses and in every case the share ratio was 42% or 43%. This implies that this percentage is the normal proportion of Durham undergraduate computer science students who opt to share their assessment feedback tags and associated programming work. It is unclear whether this would be consistent across Higher Education institutions, further research would be required in order to evaluate this claim.

There are a variety of reasons for and against sharing feedback to assessed work as put forward by students in focus groups and questionnaire responses. These are categorised and discussed in Section 7.3.5.3.

8.2.1.3 RQ3: Which students tend to opt-in e.g. weaker or stronger students?

RQ3 only can be addressed using one of the preliminary investigations, Section 5.3 and the final investigation, Section 7.3. The second preliminary investigation was unable to contribute to this research question due to it involving a formative assessment and because no summative marks were generated that could be used for judging academic performance. It appears that the results are consistent between the summative preliminary and final investigations. This is despite the preliminary investigation involving a group based assessment activity and the final investigation using an individually assessed project.

It appears that based on all of the evidence from the investigations conducted, there is a tendency for the weakest and the strongest students to share their feedback and work using the SWATT system. This therefore, means that those students with mid ranged marks do not opt into the sharing functionality provided by the SWATT approach to feedback. What is more interesting is that in the final investigation students did not know their summative assessment marks until after they had elected whether or not to share. Perhaps there is some factor which separates the weakest and strongest students from those that achieve mid ranged marks.

Numerous possible explanations may explain this behaviour. One of which may be that the strongest students know they are strong and want to help their peers or are simply proud of their work, whereas the weaker students maybe simply looking for any possible way to improve. One of the more unusual explanations which surfaced from the first preliminary investigation was that students wanted to verify the consistency of the examiners' marks by comparing their work with other students'. This behaviour may have been more prevalent in the first preliminary investigation because the summative marks were released in advance of the feedback tags generated from the SWATT system.

An unexpected finding from the final investigation is the identification of

a number of groups of students who exhibit different behaviours and ways of interacting with their feedback. A detailed discussion of the types of students and how they appeared to engage with the SWATT system and sharing functionality is discussed in Section 7.3.5.4. Further research could be conducted to investigate the link between academic achievement and the behaviour exhibited by students when they interacted with the SWATT generated feedback.

8.2.1.4 RQ4: Do students perceive benefit from having access to other students' code and associated feedback tags?

In the final investigation, 73% of students who shared their work reported that they did find benefit in seeing the feedback and work of their peers. The remaining 27% stated that they did not find benefit because whilst they had shared they had not at the time of completing the questionnaire looked at anyone else's work and so were unable to comment. These results suggest that on the whole those who did share their work and looked at their peers work and feedback did find some benefit in doing so.

As discussed in Section 7.3.5.4, a number of distinct groups of students appeared who used the shared feedback functionality for different purposes and in different ways.

8.2.1.5 RQ5: Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?

All of the investigations presented in this thesis have demonstrated that the additional information collected using sentiment analysis and / or thematic analysis can be used to help lecturers direct their teaching to cater for the needs of the specific cohort. It is important to note that whilst some of the information gained from thematic analysis could be derived from ad hoc frequency analysis, the formal analysis approach does enable clearer and more structured results. In Sections 5.5, 6.5 and 7.5 it has been shown that

additional information can be generated through the process of sentiment analysis and thematic analysis which can be used to support remedial teaching.

The primary difficulty with using thematic analysis as a means of categorising the tag and sentiment data is the amount of time required for coding and validating the themes. This process can be simplified, at the cost of reliability, by removing the review phase of the thematic analysis. However, the results may be less representative. A simpler, yet less formal analysis technique is to visualise the sentiment information within the cohort tag cloud to allow tags with larger frequencies and negative sentiments to be identified by the examiner. The problem with this approach is it only uses a subset of the data and it may be more difficult for the examiner to identify the high level themes that could occur in the feedback tags.

8.2.1.6 RQ6: How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?

In Sections 5.4, 6.4 and 7.4 the results show that the sentiment of tags, when considered in isolation from their associated source code and without any additional metadata, do not receive a consistently high agreement rate between examiners and students. This is possibly the case for other types of feedback, not just tag based feedback however, further research is required in order to investigation this claim. It is clear that some form of additional metadata should be included along with the tag based feedback to allow disambiguation of the tags intended sentiment. The NaCTeM automated sentiment analysis tool has been demonstrated to, in most cases, provide a reasonable approximation of how a human could perceive the sentiment of feedback tags. However, the limitations described in Section 7.4 of its inability to interpret high level technical terminology means that the examiner would have to use it in a semi-automated way and review the selections made by the tool before release. It is expected that other sentiment analysis tools would

perform in a similar way to the NaCTeM tool due to the technical vocabulary being so specific to the domain of programming.

Students and examiners appear to have different perceptions of what constitutes positive, negative and neutral feedback and it is important to reconcile these differences so effective communication between examiners and students can occur. It is useful to note that students often have different perceptions between each other on the sentiment of particular feedback tags, whilst examiners do to a much lesser extent.

8.2.1.7 Summary of Research Questions

The answers to the research questions are summarised in Table 8.1 and have been discussed in this section, with a short summary of each research question and its answer discussed in the following subsections.

RQ	Research Question	Summary Answer
RQ1	Do individual students perceive benefit from receiving feedback in the form of tags that are annotated throughout their software?	Yes, 75.75% average perceived benefit recorded in the final investigation.
RQ2	Do students opt-in to share their code and associated feedback?	42% or 43% opted to share in all investigations.
RQ3	Which students tend to opt-in e.g. weaker or stronger students?	Weaker and stronger students share - Mid ranged students do not.
RQ4	Do students perceive benefit from having access to other students' code and associated feedback tags?	73% of shares perceived a benefit
RQ5	Can Sentiment Analysis or Thematic Analysis of feedback tags generate additional information that benefits either Learning or Teaching?	Yes, the information generated can be used to support direction of future remedial teaching
RQ6	How well do tags communicate the intended sentiment of feedback between examiners and students when considered in isolation from their associated source code fragment?	Poorly, without additional information it is possible to have misunderstandings between examiners and students.

Table 8.1: Research questions and summary answers

It is clear that the answers to the research questions have, for the most part, shown the SWATT approach as being a positive technique for managing programming assessment feedback. The one negative aspect was for RQ6, which questions the ability for feedback tags to communicate the underlying

sentiment between examiners and students. It appears that in all investigations run there is always some inconsistency between examiners and students' perceptions. This has led to the hypothesis that this level of ambiguity could be reduced by including additional metadata on the intended sentiment of a feedback tag when it is delivered to students, either using automated or semi-automated strategies for sentiment analysis.

The number of students who opted in to the sharing functionality is consistently around 42% in all investigations and many of these perceived benefit in doing so. It is suggested that even if only two students shared (at least two students would need to share before any new information became available to sharers) and if they gained a significant benefit to their learning from doing so, the entire process can be considered as having a positive outcome. Evidence from the investigations presented suggests however, that significantly more than two students found benefit in sharing their feedback as 73% of students who shared in the final investigation reported that they found a benefit to seeing their peers work and feedback.

8.2.2 Relevance and Contribution

This section discusses the findings presented in this thesis with relation to existing literature.

8.2.2.1 The Importance of Timing

The results presented from this research confirm the importance of rapid release of feedback introduced in the literature (Black and Wiliam, 1998; Winter and Dye, 2004), see Chapter 2. The level of feedback collection in the case where the feedback was released within 1.5 weeks of submission was 100%. However, for all preliminary investigations where the release time of the feedback was significantly longer than 1.5 weeks, the amount of students who collected and subsequently engaged with their feedback was noticeably lower.

It has been noted, in this thesis, that by releasing the feedback tags in advance of the summative marks, students have engaged in a process of attempting to estimate how well they have done by thoroughly investigating the meaning of their feedback. Since the feedback is given in tag form and because some of the tags utilise high level terminology, students have had to carefully explore the meaning of their feedback and how it relates to their work. It has been reported by some students that this process has significantly changed how they used the feedback and a number of participants suggested that it has led to an improvement in their understanding of the material. It appears that, for assessment which intends to provide both formative and summative feedback, the order or release of the feedback is crucially important in determining how students engage with it. This finding is consistent with the findings of Black and Wiliam (1998).

There is a potential danger of weaker students being unable to understand their feedback tags and therefore opting to ignoring them. This risk appears to be mitigated by exploiting the students desire to estimate what their summative marks would be based on the feedback tags given to them. It is expected, based on the information received from focus groups and questionnaires, that a majority of students engage in this process of estimating their marks and, as such, this acts as a motivator for understanding their feedback tags. Students are also able to use the discussion board anonymously to ask questions of students and examiners if they are unable to understand a particular feedback tag. This facility is open to all students irrespective of whether they have opted to share. However, evidence from the investigations conducted, indicate that whilst students like the idea of this discussion facility, very few of them actually used it.

8.2.2.2 Investigating Sharable Feedback Tags

Providing a system that implements the SWATT approach of feedback has resulted in students interacting with their feedback in different ways. These are summarised below and discussed in greater detail in Section 7.3.5.4.

- Explorers
- Informal Sharers
- Librarians
- One-off Viewers (Non-Sharer)
- One-off Viewers (Sharer)
- Surface Users

This thesis has extended research in to how students use feedback, more specifically feedback delivered in tag form. This supplements the literature discussed in Section 2.4.2. It is interesting how one student in Carless' study reported that they went back to re-read their old assignments for the purpose of improving their confidence (Carless, 2006). Whilst this behaviour has not been detected with the same motivation in this thesis, the notion of using the feedback and associated programming work as a reference tool has been recorded in the group of students who used the SWATT feedback referred to as 'Librarians'.

This thesis has investigated the effects of providing an electronic facility to support sharing of students' feedback given in the form of tags. As a result a number of different reasons for and against sharing have been uncovered from the participating students. These are discussed in detail in Section 7.3.5.3 and summarised in Table 8.2.

Motivation For	**Motivation Against**
Checking up on Examiners	Distrust of Anonymity
Competition	Fear of Being Discontent
Confident	Forgetfulness
Curiosity	Lack of Confidence
Learn From Others Mistakes	Lack of Interest
Understand Own Feedback Better	Paranoia
	Social / Informal Sharers

Table 8.2: Summary of different motivations for and against sharing

These findings provide a foundation to future research in the area of sharable feedback as, to date, very little literature discusses the effects of providing feedback in a sharable form, let alone investigating how student engage with said feedback. The use of feedback tags as a medium for communication and how students use these is also a novel finding that provides a starting point for future experimental research.

8.2.2.3 The Importance of Context

A majority of students, in both questionnaire responses and focus group interviews, have reported that being able to see their feedback both in summary form via the tag cloud and in detail alongside their original submission is one of the most important benefits of the SWATT system. This reinforces the work of (Sweller, 1994; Plimmer and Mason, 2006), described in the literature review in Section 2.6, which highlights how delivery of feedback that has been physically isolated from the students' original work adds a cognitive overhead to interpreting it. This thesis has also proven, in the final investigation, that it is possible through electronic dissemination of feedback to achieve a 100% collection rate from students. This is reportedly less likely with paper based forms of feedback (Winter and Dye, 2004).

The importance of keeping feedback in its original context is crucial particularly for delivering programming feedback. Software projects can contain a huge number of lines of code and attempting to pinpoint areas that require particular attention, using paper feedback or even electronic feedback that is isolated from the original work, can add a cognitive overhead (Sweller, 1994). This can prevent students from identifying exactly which aspects of their work needs to be improved and where. The SWATT approach enables feedback to be focused around the students' originally submitted work ensuring that the students know exactly what aspects of their source code an examiner is commenting on.

8.2.2.4 Application to Existing Learning Theories

The SWATT approach to feedback aligns to the theory of constructivism as discussed in Section 2.2.2. In order for students to fully understand the meaning of some of their more complex feedback tags, it appears as though some have had to engage in a process of semantic exploration. Using feedback tags that have been positioned throughout originally submitted programming work, students have had to actually construct meaning from them in order to estimate how well they had done in the project. This is a consequence of releasing the feedback tags in advance of the marks and has led to some students reporting a perceived improvement to their learning from using SWATT feedback.

Following this link to constructivism, it is also likely that using feedback delivered in this way has allowed students to access the higher levels of Blooms taxonomy (Bloom et al., 1956; Krathwohl, 2002) of learning as introduced in Section 2.2.4. Students who have had to construct meaning from the feedback tags based on their location within their work and external sources, such as internet searches, have had to be able to both analyse and create new knowledge based on their feedback. They have done this to help understand their feedback so that they could estimate how well they had done in their overall assessment. This means that students could have potentially accessed level 4 (Analyse) and 6 (Create) of Blooms extended taxonomy. This is in addition to the early levels, 1 (Remember) and 2 (Understand), which would have also needed to be accessed by students in order to fully understand their feedback tags. Had a peer review exercise been included in the investigation level 5 (Evaluate) of the extended taxonomy may have also been accessible by students all from the same feedback activity.

Using the results presented in this thesis it can be speculated that there are links between how students have used their feedback and their affiliation to a particular learning approach. It is likely that students who were labelled as being 'explorers' in Section 7.3.5.4, were deep learners (Marton and Säljö, 1976b,a) who were trying to understand not only their own work and feedback

but also that of their peers over an extended amount of time. Their behaviour implies that they were trying to gain a deep understanding of alternative designs and implementations and examiner feedback on these. It is possible students who only briefly glanced and did not really interact with their feedback much are more aligned to a surface approach of learning. In order to fully investigate these relationships additional research would be required, which is outside of the focus of this thesis.

It appears as though the notion of communities of practice (Wenger et al., 2002), which is discussed in Section 2.2.6, has been hampered in this implementation of the SWATT system. Students were unwilling to engage in discussion around the feedback in the current system and would have been more interested in doing so if the system was more aligned to a social networking style of communication. It was unanimous that students like the idea of being able to join an online community that is based around programming work and feedback but there was no evidence of student discussion using the SWATT systems discussion board facility. This may be because the discussion board was focused around individual feedback tags and not around students' work - tag associations. Students in the focus group have suggested that discussion should take place at the point the feedback tag was assigned in work and not at the general tag profile level and that this could lead to more focused discussion.

The results in this thesis have shown that formative feedback can be given in a summatively assessed project and still be considered useful from students' perspective, providing the order of release and timing is carefully controlled. This thesis therefore presents evidence to support Wiliam and Black's assertion, as discussed in Section 2.3.1, that an assessment can indeed fulfil a formative and summative purpose at the same time (Wiliam and Black, 1996). Therefore, the evidence disputes Harlen and James' assertion (Harlen and James, 1997) that a distinction between assessment purposes should be maintained.

8.2.2.5 Contribution to the Sentiment Analysis of Feedback

Investigating the possibility of automatically detecting the sentiment of feedback tags forms part of the novel contribution of this thesis. As such, there is very little literature surrounding this topic, apart from the work of the CAFEX2 project (Gillam et al., 2009), as discussed in Section 2.4.3.

This thesis has explored the differences in perceptions held by examiners and students as to the underlying sentiment contained in feedback tags. It is accepted that the use of feedback tags is a more restrictive form of feedback due to the typically shorter length and that this in some ways impacts the clarity of comments delivered in this form. All of investigations presented in this thesis have highlighted that students and examiners can perceive the sentiment of feedback delivered in the form of tags in radically different ways. Therefore, it is clear that some additional information is required to ensure feedback tags can provide clear feedback from student to examiner. Owing to the lack of literature investigating the sentiment of traditional feedback, it is unclear as to how feedback tags perform in comparison. However, it is expected that due to the length being typically longer than feedback delivered in tag form, the longer feedback may have a clearer sentiment associated with it. This hypothesis requires full investigation which is outside of the scope of this thesis.

The findings of this thesis have led to the recommendation that the intended sentiment of feedback tags be recorded upon tag creation to reduce ambiguity for students when feedback is delivered. As a result, an automatic sentiment analysis tool was included in the investigations, to determine whether it could be used to support the generation of this additional sentiment metadata. The conclusion is that the NaCTeM sentiment analysis tool provides a reasonable indication of how a human could perceive feedback tags. However, it is unable to provide appropriate sentiment analysis data for feedback that contains programming specific terminology or language referring to high level programming concepts. Due to this, a recommendation of this thesis is that, automated approaches should be used as part of a

semi-automated sentiment analysis process to ensure students are not given incorrect sentiment information along with their feedback which could cause confusion.

8.2.2.6 Differences to Existing Approaches

The SWATT system could be classified as a semi-automated assessment tool since it supports the examiner by facilitating the annotation of short comments in the form of tags throughout the student's source code submission. The system does not intend to replace the examiner, instead it operates under the premise that the examiner will have to attempt to comprehend the key portions of the student's submission during the assessment process. Therefore the system enables the examiner to exploit the time that they need to spend comprehending the student's code by tagging it as they go.

The SWATT approach is such that any issues which are important to the student assessment can be commented on in a quick and flexible way without a need to pre-program the assessment criteria or comments in to the system. The flexibility of the SWATT system is derived from the ideas behind the Web 2.0 family of collaborative tagging applications which enable users to annotate online resources in a very simple and un-restrictive way. There is also the added benefit that examiners are not writing their comments by hand so students are not required to decipher handwritten comments. Since the comments are typed, it is possible for the tags to be analysed using a number of computational and manual methods including: co-occurrence of tags, frequency analysis and, as demonstrated by this thesis, sentiment analysis.

The focus on creating reusable feedback tags, which have been captured in the context of a student's work, allows scope for creation of a library of reusable feedback objects. These feedback tags can be used between cohorts or even to track changes in a particular cohort's progress. This is a feature that is rarely found in existing solutions to feedback delivery on programming work.

8.3 Limitations

The SWATT approach to feedback generation is not without its limitations and as such the investigations discussed in this thesis all have threats to validity sections which discuss the limitations of the results presented.

It is particularly important to note that all of the research presented has been conducted in the same Higher Education institution and in the same modular degree programme. Therefore, it is impossible to determine whether the results presented are applicable to other institutions or other programmes of study, without an expanded research exercise. However, the results presented do provide a foundation for further research and as such provide access to numerous avenues of extended research; these are discussed in the following section.

8.4 Further Work

This thesis whilst, successfully answering the research questions posed, has, due to its investigative nature, raised more detailed research questions and research possibilities. There are two categories of future work presented in this section, Research Activities, and Technical Improvements. These are introduced in this section.

8.4.1 Further Research Activities

Answering the research questions posed has led to a variety of follow up questions or experiments which could be conducted. Due to the large amount of possible extensions only a selection of these are presented in a summarised form.

- Comparison of perceived benefit between traditional approaches to feedback and tag based feedback

- Comparison of the ability of traditional feedback and tag based feedback to communicate sentiment information from examiners to students
- Investigation of the connection between feedback interaction with deep and surface approaches to learning
- Investigation of student perceptions of tag based feedback in peer assessment activities
- Investigation of the benefits of incorporating automated approaches of source code assessment with tag based feedback
- Determining if visualisation of the sentiment of feedback tags is beneficial to students' learning
- Verification, using a controlled experiment, to determine whether providing tag based feedback before, after and at the same time as summative marks has a significant impact in student engagement with their feedback.
- Research to further investigate the different behavioural groups that have been identified from student interaction with feedback tags
- Experiment to find out if the sentiment of feedback tags influences students' decision on whether to share their feedback or not
- Investigate the impact of providing students with details of the themes detected from thematic analysis of their own feedback

8.4.2 Technical Improvements

Initially, the SWATT system was designed to facilitate peer feedback tagging similar to folksonomy based systems. However, in order to test this as an approach to feedback, it was initially decided to use a simplified a one way feedback process. Therefore, the SWATT system acts as a one way assessment and feedback tool, facilitating communication from examiners to students and

then allowing students to share that expert information. Further research in to how this approach operates in a peer assessment situation is a likely candidate for future studies. However, this thesis aimed to focus exclusively on examiner to student communications and sharing of this information in order to determine the usefulness of tag based feedback. The success of the tag based feedback in the investigations has confirmed that peer assessment is an interesting avenue to further explore this research area.

A variety of helpful automated functions could be added to the SWATT system to improve its usability in assessment of programming exercises. For example, plagiarism detection (Luck and Joy, 1999; Daly and Horgan, 2004), automated test case assessment (Joy et al., 2005) as well as the aforementioned peer assessment functionality. These were not added to this version of the prototype as the primary focus of this investigation was the use of sharable feedback tags.

Since the SWATT system was developed as a prototype a number of improvements could be made based on the research conducted, some of these are summarised below.

- Extension of sentiment analysis tools to be able to identify the sentiment of high level programming terminology or concepts.
- Modify the SWATT approach to feedback sharing to determine if a 'social networking' style of sharing on an individual basis encourages students to share more.
- Investigate different ways of visualising feedback tags and potentially associated sentiment information.
- Create a web based tagging extension for SWATT to easily facilitate online peer assessment without the requirement of using Eclipse.
- Extend the SWATT system to offer more automated assessment to be displayed in conjunction with examiner feedback tags.

8.5 Conclusion

This thesis has presented three investigations which have evaluated the use of sharable feedback tags in terms of students' perceived benefit, the ability for feedback tags to communicate intended sentiment information and the possible high level analysis that can be done using the resulting feedback. It is clear that feedback tags require additional metadata to enable clear communication of sentiment information between examiners and students, however, as demonstrated by the use of the NaCTeM automated sentiment analysis tool, it is possible to make this a semi-automated process.

The ability for students to share their feedback and associated source code was consistently used by about 42% of the cohort in all three experiments, despite them involving different students. A number of different reasons for and against sharing assessment feedback in this way have been recorded and described in the final investigation presented in this thesis; some of these were unexpected and provide an interesting insight in to how students perceive their assessed work and feedback.

The SWATT system has been used by a number of students and many of these have adapted the system to suit their own learning needs by interacting with feedback tags in different ways. These different approaches to interacting with the SWATT system have been investigated and described in this thesis. The different interaction groups that have been discovered show a range of behaviours and ways of interacting with tag based feedback which were unexpected and warrant further experimental research.

This thesis has provided a significant foundation for further research in to tag based feedback as well as sentiment analysis of feedback. In addition to this, results have been presented to help classify student approaches to interacting with tag based feedback. Exploration of students' perceptions of tag based feedback has led this thesis to conclude that this feedback strategy is a new and exciting approach for delivering feedback to individual programming assignments.

Appendix A

Sample Questionnaire

This questionnaire was used in the final investigation. For the purposes of this thesis the electronic questionnaire has been converted to a text based equivalent.

A.1 Questionnaire (Final Investigation)

1. Please select your gender

[ ] Male

[ ] Female

2a. How easy was it for you to understand your feedback tags using the new feedback system?

[ ] 1. Very Easy

[ ] 2. Easy

[ ] 3. Neither Easy nor Difficult

[ ] 4. Difficult

[ ] 5. Very Difficult

2b. Please rate the amount of feedback tags you received using the new system.

[ ] 1. Far Too Much

[ ] 2. A Little Less Needed
[ ] 3. About Right
[ ] 4. A Little More Needed
[ ] 5. Far Too Little

2c. Please rate the quality of feedback you received using the SWATT system.
[ ] 1. Very Good
[ ] 2. Good
[ ] 3. Average
[ ] 4. Poor
[ ] 5. Very Poor

2d. Please rate your ability to improve based on the feedback provided using the new system.
[ ] 1. Very Easy
[ ] 2. Easy
[ ] 3. Difficult
[ ] 4. Very Difficult

3a. What did you think was good about the SWATT approach to feedback on source code?

3b. What did you think was bad about the SWATT approach to feedback on source code?

3c. Did you notice any patterns in your feedback cloud?
[ ] Yes
[ ] No
[ ] I don't know

Please explain any patterns you noticed in your feedback, if you noticed any.

4a. Did you choose to share your feedback tags?

[ ] Yes

[ ] No

Why did / didn't you share your feedback?

4b Sharing work was anonymous in this study, did this influence your decision to share?

[ ] Yes, I would not have shared otherwise.

[ ] No, I would have shared anyway.

[ ] I still would not share my work or feedback no matter what.

[ ] I didn't realise that it was anonymous sharing

4c: Did you find it useful viewing other peoples work?

[ ] Yes

[ ] No

[ ] Didn't Share

Why was / wasn't it useful to view other peoples work?

4d. Do you think you understood your own feedback any better after looking at other peoples'?

[ ] Yes

[ ] No

[ ] Didn't Share

5a: Did you use the discussion board facility to discuss the meaning of tags / feedback?

[ ] Yes
[ ] No

5b: Do you like the idea of an online community where you can discuss your work / feedback in?
[ ] Yes
[ ] No

5c: Please explain why you would or wouldn't like to use an online community to discuss your feedback / work?

6a: How would you compare the SWATT method of giving feedback to other approaches, such as the written comments given in assessors comments box.
[ ] 1. SWATT method is better
[ ] 2. Traditional Written Approaches are better (Proformas, summary sheets, assessors comments)
[ ] 3. Both are useful in different ways
[ ] 4. I don't look at the feedback either way.

6b: Would you like to see the SWATT system used for giving feedback to programming assignments in the future?
[ ] Yes, for everything
[ ] Just for Individual assignments
[ ] Just for Group Work
[ ] No, Never

6c. Any other comments or suggestions?

Bibliography

Agarwal, A. (2005). Sentiment analysis: A new approach for effective use of linguistic knowledge and exploiting similarities in a set of documents to be classified. In *4th International Conference on Natural Language Processing*, IIT Kanpur, India.

Al-Khalifa, H. S. and Davis, H. C. (2006). Folksannotation: A semantic metadata tool for annotating learning resources using folksonomies and domain ontologies. In *The Second International IEEE Conference on Innovations in Information Technology*, Dubai, UAE.

Al-Khalifa, H. S. and Davis, H. C. (2007). Exploring the value of folksonomies for creating semantic metadata. *International Journal on Semantic Web and Information Systems*, 3(1):12–38.

Angeletou, S., Sabou, M., Specia, L., and Motta, E. (2007). Bridging the gap between folksonomies and the semantic web: An experience report. In *4th European Semantic Web Conference (ESWC 2007)*, Innsbruck, Austria.

Au Yeung, C. M., Gibbins, N., and Shadbolt, N. (2007). Understanding the semantics of ambiguous tags in folksonomies. In *The International Workshop on Emergent Semantics and Ontology Evolution (ESOE2007) at ISWC/ASWC 2007*, Busan, South Korea.

Bailey, T. and Forbes, J. (2005). Just-in-time teaching for CS0. *ACM SIGCSE Bulletin*, 37(1):366–370.

Baillie-de Byl, P. (2004). An online assistant for remote, distributed critiquing of electronically submitted assessment. *Educational Technology and Society*, 7(1):29–41.

Bancroft, P. and Roe, P. (2006). Program annotations: feedback for students learning to program. In *Proceedings of the 8th Austalian Conference on Computing Education - Volume 52*, Hobart, Australia. Australian Computer Society, Inc.

Bateman, S., Brooks, C., and McCalla, G. (2006). Collaborative tagging approaches for ontological metadata in adaptive elearning. In *The 4th International Workshop on Applications of Semantic Web Technologies for E-Learning*, pages 3–12, Dublin, Ireland. International Workshop on Applications of Semantic Web technologies for E-Learning (SW-EL).

Benford, S., Burke, E. K., Foxley, E., and Higgins, C. A. (1995). The Ceilidh system for the automatic grading of students on programming courses. In *Proceedings of the 33rd Annual Southeast Regional Conference*, Clemson, South Carolina. ACM.

Berners-Lee, T. (1999). *Weaving the web*. The Orion Publishing Group ltd.

Berry, R. E. and Meekings, B. A. E. (1985). A style analysis of C programs. *Communications of the ACM*, 28(1):80–88.

Biggs, J. (1979). Individual differences in study processes and the quality of learning outcomes. *Higher Education*, 8(4):381–394.

Biggs, J. (2003). *Teaching for Quality Learning at University*. Open University Press, second edition.

Bishop-Clark, C. (1995). Cognitive style, personality, and computer programming. *Computers in Human Behavior*, 11(2):241–260.

Black, P. and Wiliam, D. (1998). Assessment and classroom learning. *Assessment in Education: Principles, Policy & Practice*, 5(1):7 – 74.

Bloom, B. E., Engelhart, M., Furst, E., Hill, W., and Krathwohl, D. (1956). *Taxonomy of educational objectives : the classification of educational goals. Handbook 1, Cognitive Domain*. Longmans, London.

Braun, V. and Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101.

Brickley, D. and Guha, R. V. (2004). RDF vocabulary description language 1.0: RDF schema. Recommendation document, World Wide Web Consortium.

Brown, E. and Glover, C. (2006). Evaluating written feedback. In Bryan, C. and Clegg, K., editors, *Innovative Assessment in Higher Education*, pages 81–91. Routledge.

Brown, G. (1997). *Assessing student learning in higher education*. Routledge, London ; New York.

Burn, A. (2008). Thematic analysis of group software project change-logs. In *The 9th Annual Higher Education Academy Conference, Liverpool Hope University*.

Butler, M. and Morgan, M. (2007). Learning challenges faced by novice programming students studying high level and low feedback concepts. In *Australasian Society for Computers in Learning in Tertiary Education (ascilite)*, Singapore.

Carless, D. (2006). Differing perceptions in the feedback process. *Studies in Higher Education*, 31(2):219–233.

Carlson, P. and Berry, F. (2007). A web-based tool for implementing peer review. In *American Society for Engineering Education*, Honolulu Hawaii.

Chapman, A. and Busch, J. (2009). Improving student feedback using technology. In *The 10th Annual Higher Education Academy Information and Computer Sciences Conference*, University of Kent at Canterbury.

Cooper, S., Dann, W., and Pausch, R. (2003). Teaching objects-first in introductory computer science. In *The 34th SIGCSE Technical Symposium on Computer Science Education*, Reno, Navada, USA. ACM, New York, NY, USA.

Cummins, S. (2008). TR-TEL-08-05: Changing programming feedback using web 2.0 technologies. *Technical Reports, Durham University*.

Daly, C. and Horgan, J. M. (2004). An automated learning system for java programming. *IEEE Transactions on Education*, 47(1):10–17.

Deek, F. P. and McHugh, J. A. (1998). A survey and critical analysis of tools for learning programming. *Computer Science Education*, 8(2):130 – 178.

Delaney, J. D., Mitchell, G., and Delaney, S. (2003). Software engineering meets problem-based learning. *The Engineers Journal*, 57(6).

DES/WO (1988). National curriculum task group on assessment and testing - a report. Technical report, Department of Education and Science and the Welsh Office.

Diederich, J. and Iofciu, T. (2006). Finding communities of practice from user profiles based on folksonomies. In *1st International Workshop on Building Technology Enhanced Learning Solutions for Communities of Practice (TEL-CoPs'06)*, Crete, Greece.

Dijkstra, E. W. (1989). On the cruelty of really teaching computing science. *Communications of the ACM*, 32:1398–1404.

Dochy, F., Segers, M., and Sluijsmans, D. (1999). The use of self-, peer and co-assessment in higher education: a review. *Studies in Higher Education*, 24(3):331–350.

Dochy, F. J. R. C. and McDowell, L. (1997). Assessment as a tool for learning. *Studies In Educational Evaluation*, 23(4):279–298.

Doolittle, P. (1999). Constructivism and online education. In *Online Conference on Teaching Online in Higher Education*, Fort Wayne, USA.

Drummond, S. and Jamieson, S. (2005). The threshold concept: Helping students towards mastery. In *The 6th Annual Conference of the Information and Computer Sciences Higher Education Academy*, University of York, United Kingdom. HEA.

Dubinko, M., Kumar, R., Magnani, J., Novak, J., Raghavan, P., and Tomkins, A. (2006). Visualizing tags over time. In *Proceedings of the 15th International Conference on World Wide Web*, Edinburgh, Scotland. ACM.

DuBoulay, B. (1989). Some difficulties of learning to program. In Soloway, E. and Spohrer, J. C., editors, *Studying the Novice Programmer*, pages 283–299. Lawrence Erlbaum Associates Inc. Publishers, Sussex, England.

Duffy, T. and Cunningham, D. (1996). *Constructivism: Implications for the design and delivery of instruction.* Handbook of Research for Educational Communications and Technology. Simon and Schuster, New York.

Echarte, F., Astrain, J. J., Cordoba, A., and Villadangos, J. (2007). Ontology of folksonomy: A new modelling method. In *4th International Conference on Knowledge Capture (KCap 2007)*, Whistler, British Columbia, Canada.

Eckerdal, A., McCartney, R., Mostr, J. E., Ratcliffe, M., Sanders, K., and Zander, C. (2006a). Putting threshold concepts into context in computer science education. *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, 38(3):103–107.

Eckerdal, A., McCartney, R., Mostr, J. E., Ratcliffe, M., and Zander, C. (2006b). Can graduating students design software systems? *ACM SIGCSE Bulletin*, 38(1):403–407.

Elliott, B. (2007). SQA - what is web 2.0: Assessment 2.0. Technical report, Scottish Qualifications Authority (SQA).

Elliott, B. (2008). Assessment 2.0: Modernising assessment in the age of web 2.0. Technical report, Scottish Qualifications Authority.

Entwistle, N. (2001). Styles of learning and approaches to studying in higher education. *Kybernetes*, 30(5-6):593–602.

Felder, R. M. and Silverman, L. K. (1988). Learning and teaching styles: In engineering education. *Engineering Education*, 78(7):674 – 681.

Flick, U. (2006). *An Introduction to Qualitative Research*. Sage Publications Ltd, 3 edition.

Gee, T. C. (1972). Students responses to teacher comments. *Research in the Teaching of English*, 6(2):212–221.

Gehringer, E. F., Ehresman, L. M., and Skrien, D. J. (2006). Expertiza: students helping to write an ood text. In *Companion to the 21st ACM SIGPLAN Conference on Object-Oriented programming Systems, Languages, and Applications*, Portland, Oregon, USA. ACM.

Geldart, J. and Cummins, S. (2008). The automatic integration of folksonomies and taxonomies using non-axiomatic logic. In *17th International Conference on Information Systems Development (ISD2008)*, Paphos, Cyprus. Springer.

Geldart, J., Cummins, S., and Song, W. (2008). A web of active knowledge. In *Proceedings of the 4th International Conference on Semantics, Knowledge and Grid*. IEEE Computer Society.

Gibbs, G. and Simpson, C. (2004). Conditions under which assessment supports students' learning. *Learning and Teaching in Higher Education*, (1).

Gillam, L., Qin, G., Bush, D., and Newbold, N. (2009). Automating feedback: The CAFEX2 project. In *The 10th Annual Higher Education Academy, Sub-*

ject Centre for Information and Computer Sciences Conference, University of Kent at Canterbury.

Goldman, K. J. (2004). An interactive environment for beginning java programmers. *Science of Computer Programming*, 53(1):3–24.

Gruber, T. (2007). Ontology of folksonomy: A mash-up of apples and oranges. *International Journal on Semantic Web and Information Systems*, 3(1):1–11.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220.

Guy, M. and Tonkin, E. (2006). Folksonomies: Tidying up tags. *D-Lib Magazine*, 12(1).

Haines, C. (2004). *Assessing students' written work: marking essays and reports*. Routledge.

Harlen, W. and James, M. (1997). Assessment and learning: differences and relationships between formative and summative assessment. *Assessment in Education: Principles, Policy and Practice*, 4(3):365–379.

Heinström, J. (2000). The impact of personality and approaches to learning on information behaviour. *Information Research: An International Electronic Journal*, 5(3).

Higgins, Colin, A., Gray, G., Symeonidis, P., and Tsintsifas, A. (2005). Automated assessment and experiences of teaching programming. *Journal on Educational Resources in Computing (JERIC)*, 5(3):5.

Higgins, C. A., Symeonidis, P., and Tsintsifas, A. (2002). The marking system for coursemaster. In *Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education*, Aarhus, Denmark. ACM.

Higgins, R., Hartley, P., and Skelton, A. (2001). Getting the message across: the problem of communicating assessment feedback. *Teaching in Higher Education*, 6:269–274.

Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn? *Educational Psychology Review*, 16(3):235–266.

Horrocks, I., Patel-Schneider, P. F., and van Harmelen, F. (2003). From SHIQ and RDF to OWL: the making of a Web Ontology Language. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(1):7–26.

Hotho, A., Jaschke, R., Schmitz, C., and Stumme, C. (2006). Information retrieval in folksonomies: Search and ranking. *Semantic Web: Research and Applications, Proceedings*, 4011:411–426.

Hung, S.-L., Kwok, I.-F., and Chan, R. (1993). Automatic programming assessment. *Computers & Education*, 20(2):183–190.

Hyland, F. and Hyland, K. (2001). Sugaring the pill: Praise and criticism in written feedback. *Journal of Second Language Writing*, 10(3):185–212.

Jackson, D. and Usher, M. (1997). Grading student programs using ASSYST. *ACM SIGCSE Bulletin*, 29(1):335–339.

Jackson, M. W. (1995). Skimming the surface or going deep? *PS: Political Science and Politics*, 28(3):512–514.

Jenkins, T. (2002). On the difficulty of learning to program. In *LTSN Centre for Information and Computer Sciences*, Loughborough University.

Johnson, C. M. (2001). A survey of current research on online communities of practice. *The Internet and Higher Education*, 4(1):45–60.

Joint Task Force on Computing Curricula (2001). Computing curricula 2001 computer science. *Journal of Educational Resources in Computing (JERIC)*, 1(3).

Joy, M., Griffiths, N., and Boyatt, R. (2005). The BOSS online submission and assessment system. *Journal on Educational Resources in Computing (JERIC)*, 5(3):2.

Joy, M. and Luck, M. (1996). Software standards in undergraduate computing courses. *Journal of Computer Assisted Learning*, 12(2):103–113.

Joy, M. and Luck, M. (1998). Effective electronic marking for on-line assessment. In *Proceedings of the 6th Annual Conference on the Teaching of Computing and the 3rd Annual Conference on Integrating Technology into Computer Science Education: Changing the Delivery of Computer Science Education*, Dublin City Univ., Ireland. ACM.

Kalton, G., Roberts, J., and Holt, D. (1980). The effects of offering a middle response option with opinion questions. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 29(1):65–78.

Kelleher, C. and Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, 37(2):83–137.

Kelleher, C. and Pausch, R. (2007). Using storytelling to motivate programming. *Communications of the ACM*, 50(7):58–64.

Kember, D., Jamieson, Q. W., Pomfret, M., and Wong, E. T. T. (1995). Learning approaches, study time and academic performance. *Higher Education*, 29(3):329–343.

Knight, P. T. (2002). Summative assessment in higher education: practices in disarray. *Studies in Higher Education*, 27(3):275–286.

Knowles, M. S., Holton III, E. F., and Swanson, R. A. (2005). *The Adult Learner: The Definitive Classic in Adult Education and Human Resource Development*. Elsevier, 6 edition.

Koenemann, J. and Robertson, S. P. (1991). Expert problem solving strategies for program comprehension. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology*, New Orleans, Louisiana, United States. ACM.

Kölling, M. (1999). The problem of teaching object-oriented programming, part 1: Languages. *Journal of Object-Oriented Programming*, 11(8):8–15.

Kölling, M., Quig, B., Patterson, A., and Rosenberg, J. (2003). The BlueJ system and its pedagogy. *Journal of Computer Science Education, Special Issue on Learning and Teaching Object Technology*, 13(4):249 – 268.

Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory into Practice*, 41(4):212–218.

Lahtinen, E., Ala-Mutka, K., and Jarvinen, H.-M. (2005). A study of the difficulties of novice programmers. In *Innovation and Technology in Computer Science Education (ITiCSE2005)*, Monte de Caparica, Portugal. ACM.

Laniado, D., Eynard, D., and Colombetti, M. (2007). Using wordnet to turn a folksonomy into a hierarchy of concepts. In *Semantic Web Applications and Perspectives (SWAP)*, Universita' degli Studi di Bari, Italy.

Laurillard, D. (1993). *Rethinking university teaching : a framework for the effective use of educational technology.* Routledge, London.

Lefoe, G. (1998). Creating constructivist learning environments on the web: The challenge in higher education. In *Australasian Society for Computers in Learning in Tertiary Education*, University of Wollongong. ASCILITE.

Littman, D. C., Pinto, J., Letovsky, S., and Soloway, E. (1987). Mental models and software maintenance. *Journal of Systems and Software*, 7(4):341–355.

Luck, M. and Joy, M. (1999). Secure on-line submission system. *Software - Practice and Experience*, 29(8):721–740.

Macgregor, G. and McCulloch, E. (2006). Collaborative tagging as a knowledge organisation and resource discovery tool. *Library Review*, 55(5):291–300.

Malmi, L., Karavirta, V., Korhonen, A., and Nikander, J. (2005). Experiences on automatically assessed algorithm simulation exercises with different resubmission policies.

Marton, F. and Säljö, R. (1976a). Qualitative differences in learning: 1 - outcome and process. *British Journal of Educational Psychology*, 46(Feb):4–11.

Marton, F. and Säljö, R. (1976b). Qualitative differences in learning: 2 - outcome as a function of learners conception of task. *British Journal of Educational Psychology*, 46(Jun):115–127.

McGuinness, D. L. and Van Harmelen, F. (2004). OWL Web Ontology Language Overview. Recommendation document, World Wide Web Consortium.

Meyer, J. H. and Land, R. (2003). Threshold concepts and troublesome knowledge (1): linkages to ways of thinking and practising within the disciplines. In *Improving Student Learning. Improving Student Learning Theory and Practice - 10 years on.*, pages 412–424. C. Rust Oxford: OCSLD.

Miller, G., Fellbaum, C., Tengi, R., Wakefield, P., Poddar, R., Langone, H., and Haskell, B. (2006). WordNet 3.0. Technical Report 24/01/2008, Princeton University.

Miller, J. (2002). Examining the interplay between constructivism and different learning styles. In *The Sixth International Conference on Teaching Statistics (ICOTS)*, Cape Town, South Africa. The International Association for Statistical Education (IASE).

Morgan, D. L. (1988). *Focus Groups as Qualitative Research.* SAGE Publications Inc.

Novak, J., Gavrin, A., Christian, W., and Patterson, E. (1999). *Just-in-Time Teaching: Blending Active Learning with Web Technology*. Addison Wesley.

NSS2009 (2009). National Student Survey. Technical report, Higher Education Funding Council for England (hefce).

Orrell, J. (2006). Feedback on learning achievement: rhetoric and reality. *Teaching in Higher Education*, 11:441–456.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Informational Retrieval*, 2(1-2):1–135.

Passant, A. (2007). Using ontologies to strengthen folksonomies and enrich information retrieval in weblogs: Theoretical background and corporate use-case. In *International Conference on Weblogs and Social Media*, Boulder, Colorado, U.S.A. ICWSM.

Piaget, J. (1947). *The psychology of intelligence*. Routledge and Kegan Paul.

Piao, S., Tsuruoka, Y., and Ananiadou, S. (2009). Sentiment analysis with knowledge resource and NLP tools. *International Journal of Interdisciplinary Social Sciences*, 4(5):17–28.

Plimmer, B. and Mason, P. (2006). A pen-based paperless environment for annotating and marking student assignments. In *Proceedings of the 7th Australasian User Interface Conference - Volume 50*, Hobart, Australia. Australian Computer Society, Inc.

Pressman, R. (2004). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Higher Education.

Rist, R. S. (1996). Teaching Eiffel as a first language. *Journal of Object-Oriented Programming*, 9:30–41.

Robins, A., Rountree, J., and Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2):137–72.

Roth, W.-M. (2000). Authentic school science: Intellectual traditions. In McCormick, R. and Paechter, C., editors, *Learning & Knowledge*, pages 6–20. Paul Chapman Publishing, London, UK.

Rowe, A. D. and Wood, L. N. (2007). What feedback do students want? In *The Australian Association for Research in Education (AARE) International Educational Research Conference*, Fremantle, Austrailia.

Rowe, A. D. and Wood, L. N. (2008). Student perceptions and preferences for feedback. *Asian Social Science*, 4(3):78–88.

Rowntree, D. (1987). *Assessing Students: How Shall We Know Them?* Nichols Pub Co.

Ryoo, J., Fonseca, F., and Janzen, D. (2008). Teaching object-oriented software engineering through problem-based learning in the context of game design. In *21st Conference IEEE Software Engineering Education and Training, 2008. CSEET '08*, Charleston, SC. IEEE Xplore.

Saikkonen, R., Malmi, L., and Korhonen, A. (2001). Fully automatic assessment of programming exercises. *ACM SIGCSE Bulletin*, 33(3):133–136.

Sinclair, J. and Cardew-Hall, M. (2008). The folksonomy tag cloud: when is it useful? *Journal of Information Science*, 34(1):15.

Sitthiworachart, J. and Joy, M. (2008). Computer support of effective peer assessment in an undergraduate programming class. *Journal of Computer Assisted Learning*, 24(3):217–231.

Sommerville, I. (2004). *Software Engineering, 7th Edition.* Addison Wesley.

Specia, L. and Motta, E. (2007). *Integrating Folksonomies with the Semantic Web.* Springer.

Spyns, P., de Moor, A., Vandenbussche, J., and Meersman, R. (2006). From folksologies to ontologies: How the twain meet. *On the Move to Meaningful*

Internet Systems 2006: Coopis, Doa, Gada, and Odbas, Pt 1, Proceedings, 4275:738–755.

Sun, L. and Williams, S. (2004). An instructional design model for constructivist learning. In *The World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Lugano, Switzerland. Association for the Advancement of Computing in Education (AACE).

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction*, 4(4):295–312.

Tijerino, Y., Masaki, H., and Igaki, N. (2006). Academix juice- a hybrid web 2.0/semantic web platform for exchange of academic knowledge. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. IEEE Computer Society.

Van Damme, C., Hepp, M., and Siorpaes, K. (2007). Folksontology: An integrated approach for turning folksonomies into ontologies. In *Proceedings of the 4th European Semantic Web Conference (ESWC 2007): Workshop Bridging the Gap between Semantic Web and Web 2.0*, Innsbruck, Austria.

Vander Wal, T. (2005). Explaining and showing broad and narrow folksonomies. http://personalinfocloud.com/2005/02/explaining_and_.html, Accessed on: 05/10/2008.

Vander Wal, T. (2007). Wikipedia folksonomy is a mess with collaborative misunderstanding. http://www.vanderwal.net/random/entrysel.php?blog=1949, Accessed on: 17/01/2008.

Venables, A. and Haywood, L. (2003). Programming students need instant feedback! In *Proceedings of the 5th Australasian Conference on Computing Education: Research and Practice in Information Technology*, Adelaide, Australia. Australian Computer Society, Inc.

von Glasersfeld, E. (1989a). Cognition, construction of knowledge, and teaching. *Synthese*, 80(1):121–140.

von Glasersfeld, E. (1989b). *Constructivism in Education*, volume 1. Oxford/New York: Pergamon Press.

Watson, J. B. (1997). *Behaviorism*. Transaction Publishers.

Weaver, M. R. (2006). Do students value feedback? student perceptions of tutors written responses. *Assessment & Evaluation in Higher Education*, 31(3):379–394.

Wenger, E. (2000). Communities of practice and social learning systems. *Organization*, 7(2):225–246.

Wenger, E., McDermott, R., and Snyder, W. (2002). *Cultivating Communities of Practice: A Guide to Managing Knowledge*. Harvard Business School Press.

Wiedenbeck, S., Ramalingam, V., Sarasamma, S., and Corritore, C. L. (1999). A comparison of the comprehension of object-oriented and procedural programs by novice programmers. *Interacting with Computers*, 11(3):255–282.

Wiliam, D. and Black, P. (1996). Meanings and consequences: A basis for distinguishing formative and summative functions of assessment? *British Educational Research Journal*, 22(5):537–548.

Winslow, L. E. (1996). Programming pedagogy: a psychological overview. *ACM SIGCSE Bulletin*, 28(3):17–22.

Winter, C. and Dye, V. L. (2004). An investigation into the reasons why students do not collect marked assignments and the accompanying feedback. In *CELT Learning and Teaching Projects*. University of Wolverhampton.