

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

**Music education software: An HCI evaluation of existing software and prototype implementation of an ear training programme based on a proposed approach.**

A dissertation  
submitted to the  
Department of Computer Science,  
Faculty of Science,  
University of Cape Town  
in partial fulfilment of the requirements  
for the degree of  
Master of Information Technology

by

Stella Könik

20 August 2003

Supervised by

Prof. Gary Marsden

## **Abstract:**

This research paper uses Human Computer Interaction(HCI) evaluation methods, namely questionnaires and interviews, to obtain information about the kind of music software currently in common use in music departments, and the level of satisfaction of the students with this software. The evaluation results are then used to examine the possibilities for some improvement of this software.

The evaluation reveals that certain software, such as that used for the area of aural (ear) training in music study, may not be used to its fullest extent as a result of a lack of student interest in the software.

This research considers one possible reason for the lack of interest, namely that students find no relevance or meaning in the content of such software.

We then suggest that the contextualisation of information, in line with prevalent educational theories, may provide one mechanism for addressing students' lack of interest.

This suggestion is followed by a design for, and development of a piece of prototype software which delivers aural drills in much the same way as other such programs, but also includes a small contextualised section.

This software prototype is then tested and evaluated by an expert user, and the results discussed briefly.

This paper concludes that software design for this musical field of study may benefit from the incorporation of external factors such as contextualised content, in order to create software that has relevance and meaning for the people using it.

It is anticipated that this, in turn, may lead to increased interest in, and usage of, such software and ultimately, an enriched learning experience for the student.

## **Acknowledgements**

I would like to thank the following people for their contributions, directly or indirectly, to this research paper:

Dr. J. Kirakowski. Director of the Human Factors Research Group and lecturer in the Department of Applied Psychology, University College. Cork. Ireland. for suggestions regarding questionnaire design;

Music students at the Dublin Institute of Technology, Rathmines, Dublin, Ireland, for participating in the survey on music education software;

My parents, Mr and Mrs Konik (Port Elizabeth. South Africa) for their encouragement in all my endeavours, academic or otherwise;

My partner, Michael Manchip (Cape Town, South Africa and Dublin, Ireland) for his tremendous and unwavering support (not to mention numerous cups of tea!);

Prof Gary Marsden, my supervisor, for his excellent advice and assistance.

University of Cape Town

## Table of Contents

1. Introduction	1
2. Background	2
2.1 Education	2
2.2 Music Education	3
2.3 HCI and design guidelines	3
2.4 General information on music courses	4
2.5 Music software available as educational tools for music studies	4
2.6 Education theory: using software / computers in education.	5
2.7 HCI evaluation methodology	5
3. Research Question	7
3.1 Research process: a short outline of the rest of this paper	7
4. Evaluation of available music software	8
4.1 Questionnaire design	8
4.2 Questionnaire format	8
4.3 Interview design	10
4.4 Questionnaire and Interview process	10
4.5 Analysis of completed questionnaires	11
4.6 Analysis of interviews	12
5. Understanding and interpreting the results	13
5.1 Questionnaire results	13
5.2 Interview results	19
5.3 What do these results mean?	20
5.4 How to proceed with this information	21
6. Statement of problem to be solved	22
6.1 Analogy – the “Lottery Factor”	22
7. Solving the problem	24
7.1 Background to educational theory	24
7.2 The Theory of Constructivism	24
7.3 Contextualised Learning	26
8. A software design for an ear-training software prototype which incorporates simple contextualisation.	27
8.1 Prototype requirements specification	27
8.2 Pseudocode	28
8.3 Data Flow Diagram (Context)	29
8.4 Use Case Modelling	30
8.5 Software requirements	31
8.6 Software design definition ( Entity Relationship Diagram)	32
8.7 Software design definition (Data Flow Diagram – Level One)	33
8.8 Software design document (System Architecture and Detailed Design)	34

8.9 Software testing against product requirements	35
8.10 Screen shots of the prototype ear-training tool	36
9. Evaluating the prototype software	42
9.1 Evaluation results	43
9.2 Comment on the prototype evaluation	44
10. Conclusion	45
10.1 The features and benefits of the prototype ear-training software	46
10.2 Techniques used in this research	47
10.3 The results of this research paper's investigations	49
10.4 Implications of this research paper	49
11. References	50
12. Appendix A: Raw questionnaire data from initial music software evaluation	
13. Appendix B: Questionnaire analysis per question	
14. Appendix C: Questionnaire analysis per section	
15. Appendix D: Sample Questionnaire	
16. Appendix E: Software Detailed Design Document	
17. Appendix F: Prototype programme code	

University of Cape Town

## **1. Introduction**

This research paper seeks to explore some possibilities for the improvement of certain software used in music education. To do so, survey questionnaires and interviews will be constructed in an attempt to evaluate specific software from the user's point of view. The findings of this evaluation will be analysed to determine whether there is any room for improvement in any of this software. If so, an investigation into the nature and extent of these improvements will be carried out.

This paper will then explore the underlying educational theory called constructivism which would seem to support computer-assisted education in a number of ways.

Finally, a software prototype, built using input from these previous activities, will be evaluated by an expert user, and the results of this evaluation will be discussed.

The following section is intended to serve as a background discussion of the key areas for this research paper, namely music education and the use of software, and the evaluation of software from an HCI perspective.

Together, these fields form a cohesive starting point for the rest of this paper.

## 2. Background

### 2.1 Education:

In the journal article entitled "*Putting the Computer in Its Place: A Study of Teaching with Technology*", (1994), authors Larry Miller and John Olson contend that the promise of computers and other educational technology is overstated. [1]

Looking back over the past decade, it would seem that such an opinion may have been a little hasty.

In the field of education, software has transformed the way in which students and scholars are taught. For example, as far back as 1990, 18 successful approaches by experts and practitioners in using technology as a tool to improve learning, were combined in a collection of papers entitled "*Technology in Today's Schools*" [2]. Examples of these approaches include "*Using Computer-Assisted Instruction To Support Learners*", and "*Integrating Technologies To Enhance Learning in Science and Math*".

More recently, events such as the 19<sup>th</sup> *International Conference on Technology and Education*, held in Tallahassee, Florida, May 2-5, (2001) [4] featured the presentation of papers on current education techniques, such as "*Curriculum planning in the 21<sup>st</sup> century: Managing technology, diversity and constructivism to create appropriate learning environments for all students*".

Furthermore, case studies show examples of how technology is acting as a catalyst for reform – these include the use of technology to support learning beyond the school walls, and an example of how technology can support gifted and talented students.

(Somekh, et al, 2002) [3].

Technology has therefore been widely accepted both as a beneficial aid to teaching, and as an educational resource itself, in that students are now, from an early age, able to include computer skills in their growing list of abilities. As a result, the commercial education software market has become a huge industry, with software being produced to address every conceivable level of learning, from kindergarten to post-graduate, in almost all educational fields. The range and diversity of these products is evident in the display of commercial software featured in retail sites[5].

## **2.2 Music Education:**

In Music Education, an entirely new strand of study, namely music technology seems to have become a popular alternative academic strand to traditional music studies. The number of undergraduate and postgraduate music technology degrees and courses now offered by most universities and colleges [6] bears witness to the surge of interest in this field, which offers access to music studio technology and composition aids. Furthermore, organisations such as Association for Technology in Music Instruction [34] offer assistance, advice, information and support in this field.

However, traditional music studies (such as music performance or musicology) have also embraced technology (with varying levels of success) by incorporating software programs such as notation aids, aural training aids, etc, into these music courses. Such software is beneficial to both lecturer (by providing computer-assisted tuition) and student (by providing opportunities to hone musical skills such as aural ability, outside of the classroom environment).

## **2.3 HCI and design guidelines:**

From the 1980's, software programmes shifted from being non-interactive to involving increasing levels of direct interaction with the user. A new approach to the design of such software was needed, to embrace the growing demands of the user, and ultimately this led to a new strand of study: Human Computer Interaction (HCI).

The computer-human special interest group of the Association of Computing Machinery[7], attempts the following working definition of HCI:

*"Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them."*

HCI has become a major player in the field of software design. According to Ian Sommerville, *"Appropriate user interface design is critical for successful system operation"*. [8]

The evaluation of software from an HCI perspective has become an extremely useful method for determining whether software is designed in an optimal way for the end user. In this respect, HCI design guidelines have been developed to address certain key areas such as

usefulness and usability of software, and these guidelines can be used to inform software design processes.

With educational software products streaming into the software market, evaluation and improvement of this software would appear to be a valuable exercise in maintaining software quality. It also presents an opportunity to keep abreast of trends and new developments, and allows for the continuous improvement of this software wherever possible, so that education, in all fields, may avail of the best possible software tools.

#### ***2.4 General information on music courses:***

Although tertiary-level (i.e. university or college standard) music courses vary from one academic institution to the next, there are a number of similarities in terms of the subjects offered by these courses.

Subjects such as harmony, counterpoint, aural (ear) training, music history and musicology (analysis) generally form part of the traditional music course curriculum[9].

Of these subjects, perhaps the harmony (notation) and ear training courses are most open to augmentation by the use of software programs.

Music technology courses offer alternative subjects to those mentioned above, such as sound theory, sound engineering, and acoustics[6]. For the purposes of this study, however, the area of interest is the traditional music course, which is generally taken by performance, education or musicology students.

Naturally there are areas where these two strands of music study overlap.

#### ***2.5 Music software available as educational tools for music studies:***

It is therefore possible to use some music software packages in more than one subject, for more than one purpose.

Examples of software widely available for music education include[11]:

*Sibelius*: a notation programme which can also be used for ear training due to its playback features, and a harmony tuition tool.

*Cakewalk Home Studio*: primarily a powerful music production suite simulating a recording studio, but can be used as an orchestration training tool.

*Cubase*: a sequencing and recording programme which could also be used to teach reading music.

Other programmes are intended for one purpose but can be used by students of both strands to improve their skills (for example: *Earope*, and *Auralia* [11], ear training programmes, which both sets of students would use for this particular purpose)

Notwithstanding the purpose or use of this sort of software, the primary reason for its incorporation into the tertiary music course curriculum is the user – such software offers great assistance to the student in his/her quest to improve and hone musical skills. These tools are therefore seen to be essential elements of modern music education.

### **2.6 Education theory - using software / computers in education:**

In the last 10 years, studies on the use of software in education have revealed an increasing admiration for the number of benefits to both the student and the teacher/lecturer.

In the article entitled “*Technology in the Classroom: Computers and Instruction in Vermont's Schools*” [12], the authors outline the importance of technology skills for employability and citizenship, and highlight the use of technology to interest children in learning.

Furthermore, in the book entitled “*Changing Minds: Computers, Learning, and Literacy*” [13], the author Andrea A. diSessa suggests that the computer is not merely a tool for more efficient instruction, but rather that computers can be the basis for a new literacy that will change how people think and learn.

### **2.7 HCI evaluation methodology:**

HCI principles maintain that the design and evaluation of interactive computer systems should improve usefulness and usability. These two aspects can be defined as follows:

*“An interactive system is **useful** to its user if it is goal-centred, and therefore enables its users to attain their goals.”*

*“An interactive system is considered to have good **usability** if its design incorporates its users' capabilities and limitations. In this regard, the system should be learnable, flexible and robust” (as outlined by Dix et al, 1998)[14].*

These HCI principles and characteristics may be used to evaluate an interactive system in order to discover the extent to which it can be considered useful and usable to its users.

HCI evaluation methods suggest approaches such as user observation, and user feedback[15]. For the purposes of this research, it seems suitable to employ an approach that will elicit the maximum amount of information from the user, and so the evaluation process will consist of a combination of questionnaires and interviews.

University of Cape Town

### **3. Research Question**

**This research will investigate whether there is any room for improvement in music software supplied to tertiary-level music students, and if so, will further investigate possible reasons and suggestions for such an improvement.**

The research therefore aims to discover whether HCI design and functionality guidelines can be complemented by the inclusion of an additional element (such as contextualisation of content or information) which may result in a raised level of interest from the students, and therefore an increased level of usage of the software, and an associated improvement in the student's learning experience.

The success of this research may be measured by the results of a student evaluation of the prototype software developed using the information obtained from the initial software evaluation. The prototype evaluation will take the form of a semi-structured interview, the focus of which will be the contextualisation of genre-specific information contained in the prototype.

#### ***3.1 Research process - a short outline of the rest of this paper:***

This paper will now provide a brief background discussion on music software currently available. This is followed by an outline of the questionnaire design phase for the survey, and the results of the survey. One possible solution for the problem is presented, and incorporated into a software prototype design, which is then evaluated by an expert user in this domain. Finally the project conclusions are drawn and the resulting implications briefly outlined.

#### **4. Evaluation of available music software**

##### ***4.1 Questionnaire design:***

The aim of the questionnaire will be to evaluate specific music software from the user's point of view.

Further thoughts on this reveal that it might be wise to gather the required information using more than one mechanism, in order to maximise the results of this activity. Therefore the questionnaire will be combined with a brief interview, and both these activities will take place within the context of a personal meeting.

However, it would appear that such an approach significantly increases the amount of time required for the information-gathering phase of this project. This will unfortunately lead to a reduction in the intended size of the sample set of music students. Ultimately, it will only be possible for this researcher to work with a sample set of music students from one tertiary-level college.

In the design phase of the questionnaire, it will be necessary to ensure that the document remains generic enough to be used in a variety of music departments, and yet specific enough to elicit meaningful and relevant human-computer interaction information about the music software in use, which will then be helpful in the design of a piece of prototype software.

##### ***4.2 Questionnaire format:***

As suggested by existing, established questionnaires and questionnaire designers[16], my questionnaire presents questions as statements, each stating a particular point of view, as unambiguously as possible, so as to provide a clear viewpoint against which the user is able to form an opinion. Questionnaire statements are constructed as either positive or negative sentences, and a mixture of each of these is included in each section.

The questionnaire employs a closed-question format, in that it presents a limited number of possible answers to each question.

Each question is given five possible answers, in addition to a Not Applicable option. The answers are ranked according to a simple multi-point rating scale, from 1 – 5, where the answer at 1 represents strong disagreement with the question statement, while the answer at 5 represents strong agreement.

A sample question from the questionnaire is included below, for clarity:

**“This software often ‘crashes’ for no apparent reason.”**

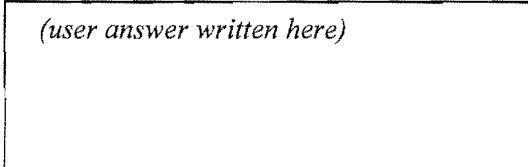
(strongly disagree) 1.  2.  3.  4.  5.  (strongly agree) N/A

In addition to these pre-formatted questions, a few open questions are included, in the form of a free text box for users to answer the question statements as they wish.

A sample open question from the questionnaire is included below:

**“The 3 best features of this software are:”**

*(user answer written here)*



The questionnaire contains five sections, each addressing a different HCI design guideline (Dix et al, [14]) for the purposes of this evaluation.

These five sections are:

- How does this software help you, the user, achieve your tasks and goals (*usability*)
- How easy was this software to learn (*learnability*)
- Are you, the user, able to customise this software (*customisation*)
- How does this software behave towards you (*robustness and stability*)
- Evaluating the visual appearance of this software (*the user interface*)

A blank version of the final questionnaire used for this evaluation is included in the Appendix to this research paper [App04].

#### **4.3 Interview design:**

The aim of the interview will be to capture any information additional to the “functional” data required by the questionnaire. It is hoped that the interview will lend a “human” element to the information gathering process and allow users to expand on their opinions expressed in the questionnaire.

This activity will therefore take the form of a guided interview (also known as a semi-structured interview), as defined by Dr. Rita S. Y. Berry in her paper presented at the British Educational Research Association Annual Conference, University of Sussex at Brighton, September 1999, entitled “*Collecting data by in-depth interviewing*” [17].

A basic checklist will be used to ensure that all relevant topics are covered. This will also leave the interviewer free to explore issues and ask questions while eliciting information about specific topics.

The main topics of interest will be restricted to:

- Student’s opinion of the use of technology in music, in general.
- Student’s opinion of the use of technology in music, in his/her music department
- Student’s opinion of the software made available to him/her for use, in the music department.
- Student’s level of interest in this software.

#### **4.4 Questionnaire and Interview process:**

The user will be asked to select from the range of software available to him/her, a particular piece of software that he/she will have in mind when answering the questionnaire. The user will then be given as much time as they need or want, in order to answer the questionnaire.

Each user will, if possible, be asked to give the interviewer a brief demonstration of their chosen piece of software, before the interview, therefore allowing the interviewer to employ a direct observation method in order to gain an overview of the manner in which the software is used by the students. Direct observation will probably be useful in this particular context because it will not require the interviewer to participate in using the software (which in turn would increase the amount of time required for these interviews). Therefore, using such a focussed activity, it will probably be possible for the researcher to gain valuable information about the student’s use of software, in a relatively short period of time. [18]

The interview will be held after the questionnaire is completed, so that the user's attention will already be focussed on the subject of this research. In each interview, the four questions outlined above will be raised as areas of interest for this research, but the students will not be pressed to answer these. Instead, it is hoped that, of these questions, those selected for elaboration by the student will provide the most insight into these areas.

The combined process contained in a single interview should take (on average) about 25 minutes per person.

#### ***4.5 Analysis of completed questionnaires:***

As mentioned previously, each answer option is assigned a particular value. The answer labelled "strongly disagree" is assigned a value of 1. From there, the value increments for each option until the answer labelled "strongly agree", which carries a value of 5.

In the analysis process, the value of these checkboxes is used to provide a numeric result, which is in turn useful for comparison purposes.

Firstly, the raw data is captured in a simple spreadsheet format, so that no data is lost in the analysis process. [App01]

Secondly, the questionnaire is analysed on a per question basis [App02] in the following way:

- The value of the checked box is captured, for each occurrence of the question, to obtain an **overall score** for that question based on the number of questionnaires answered.
- A **maximum possible score** for each question is established by multiplying the maximum possible score for each question (i.e. 5) with the number of questionnaires answered (3). In this case, the max possible score is 15.
- The overall score is then divided by the max possible score (15), and finally multiplied by the **max per question score** (5) to obtain the **final average result** for each question.

Thirdly, the per question analysis is collated to form a per-section view of the questionnaire results [App03].

#### ***4.6 Analysis of interviews:***

It would appear that most students, when presented with the four main questions for the interview section, pass over the first three, and settle on the 4<sup>th</sup> question (i.e. student's level of

interest in this software) as the main area for elaboration. Their comments are discussed in the next section of this paper.

University of Cape Town

## **5. Understanding and interpreting the results:**

### ***5.1 Questionnaire results:***

The students' choice of software for the purposes of this evaluation seems to be either Earope or Sibelius. Their software selection is noted on the first page of each questionnaire.

It is important to note that each question's score maps to a level of agreement in the scale 1 – 5. Therefore low score figures indicate stronger disagreement with the statement. However, this has to be seen in the context of the statement structure, in terms of its positive or negative overtones. For example, a score of 2 for a negatively structured question reveals disagreement with the negative connotations of that question, which in fact is a positive outcome.

The following section details an analysis of each questionnaire section, per question, and includes short comments on each.

University of Cape Town

*Section One: "How does this software help you, the user, achieve your tasks and goals?"*

Topic: Evaluation of usefulness

Question Number	Statement text	Average score for this question
1	This software helps me to do what I need to do, with good results	3.3
2	This software helps me to do what I need to do, efficiently	3.3
3	Some examples of the things I use this software for, are:	N/A
4	I feel I have become more productive using this software	2.6
5	This software allows me to complete what I need to do, quickly	2.6
6	I am satisfied with what this software can do	3.6
7	The software can do everything I need it to do	3.6
8	There are some tasks for which I cannot use this software	2.6
9	Some examples of these tasks are:	N/A
10	When using this software to do something, the steps I need to take are logical	3.6
11	There are times when I do get lost while using this software	2.6
12	The software responds quickly to my inputs	3.6
13	The logical flow of steps and options is very similar to my way of doing things	3
14	This software is satisfying to use	3
15	This software could be improved in some areas	2.3
16	I can suggest the following improvements:	N/A

Free text questions: No 3,9,16 (These answers relate to the type of software being evaluated.)

***Comment on analysis:***

It is interesting to note that no positively angled question receives a particularly high score (i.e. strong agreement with the statement). Further, some of these questions (e.g. 4,5) received scores indicating some level of disagreement (score below 3).

Of particular relevance are the following free-text questions:

No 9: students evaluating an ear-training programme include comments such as "Musical notation: computer vs. hand-written" and "Tasks outside of ear training".

No 16: Note that students evaluating an ear-training programme do not make any suggestions for improvements in this question.

*Section Two: "How easy was this software to learn?"*

Topic: Learnability

Question Number	Statement text	Average score for this question
1	Learning to use this software took a very long time	2.3
2	I am comfortable about using this software	3
3	I am confident about using this software	2.6
4	Using this software, I rate myself as: (Beginner/Intermediate user/Advanced user/Expert)	N/A
5	There is a lot of help available to me for this software (documentation, tutorials, etc)	3.3
6	This help material is simple to understand	3.3
7	I need another persons to help me learn something new using this software	1.3
8	I need another person to help me with the way in which I currently do things using this software	2
9	I am completely comfortable with all the terminology I come across when using this programme	2.6
10	There are still some areas or levels of the software programme that I find difficult to understand	3

Question No.4 requires the selection of one option from a group of four possible answers.

***Comment on analysis:***

Again, mid-point or below mid-point scores for levels of comfort and confidence with this software, even though all users have been using the software under evaluation for at least 5 months, as part of their full time studies. This would seem to indicate average attitudes towards the learnability of this software.

Note: the students evaluating the ear training programme all select the level of intermediate user for question 4.

*Section Three: "Are you, the user, able to customise this software?"*

Topic: Flexibility in the form of customisation options

Question Number	Statement text	Average score for this question
1	It is possible for me to change some aspects of this software's screen appearance, to suit my preferences (e.g. colours, layout, etc)	3.3
2	Examples of aspects I am able to change include:	N/A
3	I would like to have more freedom to choose my preferences for this software's appearance on screen	1
4	Examples of aspects I would like to be able to customise, include:	N/A

Free text questions No 2,4: Dependant on software being evaluated

***Comment on analysis:***

Although students seem to agree that some customisation options are possible for the software under evaluation, the resounding disagreement to question 3 is a little surprising, as it would be reasonable to expect at least some interest in further customisation options.

*Section Four: "How does this software behave towards you?"*

Topic: Robustness

Question Number	Statement text	Average score for this question
1	This software often "crashes" for no apparent reason	1
2	After a "crash", I can pick up where I left off without having to redo a large section of work	1
3	The software detects when my actions will cause an error, and informs me of this	1.3
4	The software provides me with opportunities to change my choices and options, before saving these	3.6
5	When an error occurs, the software gives me adequate information on the kind of error being experienced	1.6
6	When an error occurs, the software gives me adequate information on possible ways in which to correct the error	1.6
7	I can always "undo" my previous step, if necessary	4
8	My work is continuously saved by the software	2.6
9	Sometimes this software behaves in an unexpected way	2.6
10	Sometimes this software gives me unexpected results	2.6
11	On the whole, the behaviour of this software can be anticipated	4.3

***Comment on analysis:***

Strong disagreement with statements of questions 1 and 2 indicate that the software under evaluation rarely crashes. However, students register fairly strong dissatisfaction with the statements regarding the amount of info supplied to them by the program before an error, and info available in order to correct these errors (questions 3,5,6).

On the whole, the ability to "undo" a step(question 7) seems to be present in the software, and it is felt fairly strongly that the software's actions can be anticipated (question 11).

Section Five: "Evaluating the visual appearance of this software"

Topic: evaluation of the user interface

Question Number	Statement text	Average score for this question
1	Key items (such as menus, exit button, etc) are located in the same place throughout the software	4.3
2	Colours are well chosen and are pleasant on the eye	4.3
3	The menus are logically organised (i.e. similar and related actions are grouped together)	4
4	Items such as buttons and toolbars are placed in sensible obvious locations	4.3
5	It is easy for me to find my way around this software	4
6	I always know exactly where I am in this software	3.3
7	The software makes use of symbols and icons that are easy to recognise	4.3
8	I like the "Look and Feel" of this software	3.3
9	There are aspects of this software that are confusing because they don't make sense	2
10	There are aspects of this software that are annoying because they are not useful and yet keep cropping up	1.6
11	The 3 best features of this software are:	N/A
12	The 3 worst features of this software are:	N/A

Free text questions No 11,12.

**Comment on analysis:**

Key aspects of good Graphical User Interface (GUI) design include choice of colours, layout, menu's, navigational info, etc, and the above section would seem to indicate strong agreement that these aspects are present in the software under evaluation.

Note: Students who are evaluating the ear-training programme provide the following observations for Question 11:

"Allows you to test at different levels & immediately informs you if you've made a mistake"

"Geared towards students of all levels of musical experience"

"Simple", "Quick", "Practical"

## **5.2 Interview results:**

As mentioned previously, the main interview question topics are:

- Student's opinion of the use of technology in music, in general.
- Student's opinion of the use of technology in music, in his/her music department
- Student's opinion of the software made available to him/her for use, in the music department
- Student's level of interest in this software.

The interviews reveal the following:

- The software most prevalent in music departments is music technology software such as Q-Base or Sibelius, intended mainly for studio usage and scoring / composition.
- Other technology in use appears to be ear (aural) training programmes, such as Earope or Auralia, which is designed to assist students with aural skills by presenting them with a number of aural sequences and requesting recognition of specified elements in the sequence.
- Students who are registered for a music technology course have the most access to available technology in the music department (when compared to students studying more traditional music programmes, e.g. performance or musicology)
- Some of the technology currently available to music technology students in music departments could be useful to students studying more traditional fields in music, e.g. performance. An example of such software is the ear-training programme.

On the subject of the student's level of interest in the software available to him/her:

- Students appear to be eager to use the technology course software available to them, such as Sibelius.
- Students appear less eager to use the ear-training programmes.
- Sibelius and other such mixing/notation software are seen as creative and challenging, with opportunities to create and manipulate their own compositions. Interest level in this sort of software is high.
- Ear training software, however, is rated as mundane, functional and boring. Interest level in this software is medium to low.

This last point may be elaborated as follows:

During the interview, students seem to subconsciously compare other software at their disposal with the ear-training software, and find it lacking.

Further discussion reveals that they find the ear-training software to be functional in that it performs its task (exercising and honing aural skills) but that it is not interesting to use.

The students indicate that they would prefer to accomplish the same task (exercising and honing aural skills) by using the secondary features of other, more “heavy-weight” software available to them.

Reasons given for this preference include a more interesting way of learning aural skills, along with increased relevance and meaning to the students.

The ear-training software provides rote learning in a fixed format. It is seen as a single-faceted learning environment and therefore seems inferior in comparison to the other, multi-faceted software at their disposal.

### ***5.3 What do these results mean?***

The questionnaire analysis does not reveal startling or unexpected results in the evaluation of the music software in use by the students. There would appear to be some areas where slight improvements could be made, such as the amount of information presented to the user for error cases, and so on, but on the whole the questionnaire does not highlight any major problem areas.

However, the interview reveals another aspect of this evaluation. It is here that the user dissatisfaction with ear-training programmes becomes apparent, and their lack of interest in such programmes is strongly expressed.

Although the questionnaire used for this survey is constructed using accepted HCI design guidelines as points for evaluation in each section, and although the use of a questionnaire is an accepted form of evaluation in itself, it does not seem to afford the students an opportunity to articulate their lack of interest in ear-training software. This scenario perhaps serves to illustrate the importance of interviews as opportunities for free expression by the users, and seems to highlight the necessity for, whenever possible, combining a survey process with an interview process to obtain clear, unbiased results.

#### ***5.4 How to proceed with this information:***

This survey indicates that ear training software is considered a “light-weight” music education tool by music students. However, it also indicates that this is the tool most readily available to non-music technology students.

It is possible to use other software for ear-training purposes, even though this might be a secondary feature of such software, and not an intended primary use. However, it is also possible that music departments will continue to purchase separate software for ear training, as it would seem to be a logical decision – *“supplying students with software that does exactly what it says on the tin”*. There may also be cost implications for such a purchasing decision, as ear-training software may well be purchased at a lower price than more sophisticated music education software.

Ear training software is acknowledged by students to be a simple and practical opportunity to test and improve one’s aural skills, even though such a functional system appears boring to these students when compared to other software at their disposal.

Given all these valid points, it would now seem worthwhile to address the main issue raised by this survey, (i.e. student’s lack of interest in ear-training software), by exploring the possibilities for improvements to such software which could remedy this attitude by going some way towards keeping the users (i.e. the students) interested, amused, entertained or stimulated.

## 6. Statement of problem to be solved

The survey discovery, with regard to ear training software, can be summarised as follows:

- The software would appear to be well-designed, and the evaluation of this, carried out in the survey, does not point out any major design flaws, or areas where the software fails outright in terms of usefulness and usability.
- Similarly, the software's user interface would appear to be well-designed, and the evaluation of this does not indicate any area requiring major improvement.

BUT

- The software's scope appears to be limited to ear-training drills, and does not offer opportunities for expansion.
- The software's content is presented in an isolated way, and is not delivered in any broader musical context.
- The software fails to address the educational need for acquiring, and sustaining, student interest. Therefore, students display a lack of interest in using this software.

The problem, then, would seem to centre on the fact that such ear-training programs are limited in both scope and content. While scope is usually a restriction enforced by the functionality of the software, the content of such software may be the area most open to improvement.

### **6.1 Analogy – the “Lottery Factor”:**

At this point I would like to draw an analogy with a common daily activity: buying a newspaper, or checking a news web site.

People do not read newspapers or news web sites for the well-designed layout, or the provision of navigation aids (page numbers etc), or the pleasing choice of colours, or the fact that the site always seems to be available for browsing and never crashes. These are secondary aspects which nevertheless serve an important purpose – they facilitate the easy use of that media form, they are certainly helpful, and therefore are concerned with the usability of the media form.

However, these secondary aspects do not address the concept of user interest (in this case, reader interest).

People buy newspapers and check news web sites for their content, or more specifically, the continuous change in content (meaning that you never know what you might learn), and even more specifically, because this content is presented in a way that has meaning and relevance to the reader (meaning that I might learn something new that affects some aspect of my life).

And while buying a newspaper every morning may constitute a mundane or boring activity, the painless and fairly enjoyable act of acquiring knowledge in this way is in itself fair motivation to continue doing that activity.

Conversely, if this content is not stimulating, interesting or at least relevant to the reader, the reader may well cease to buy that particular newspaper or check that particular news site, and might seek this content elsewhere.

This is something that I have termed “The Lottery Factor” to illustrate that people seem to enjoy learning about things they don’t know, as long as this learning action occurs in a relatively pleasant and fairly non-taxing way.

Certainly from the above analogy, it would seem that content might have a significant role to play in attracting and maintaining user interest. Possibly, it may also be of use to the particular problem at hand, namely student lack of interest in an ear-training software programme.

Therefore, it would be a worthwhile exercise to revisit the design of an ear-training software programme, with the aim of increasing student interest in (and usage of) the programme by formatting some part of the content so that it attracts the interest of the student, providing relevance and meaning in the otherwise barren context of a drill programme.

Taking this thought process one step further, it would seem necessary then, that in order to present content in a meaningful and relevant manner to the student, that content must be **contextualised**. An improved interface design for an ear-training programme would present a broader context to the user by combining the functionality of ear training with a variable musical context. Such an interface design could provide a richer, and far more interesting experience in terms of ear training exercises.

## 7. Solving the problem

Further research into more recent education theory reveals that the problem of students' lack of interest, along with a number of other known educational problems, may already have been acknowledged and discussed by means of revised teaching methodology and the incorporation of computer-assisted instruction.

### **7.1 Background to educational theory:**

Historically, educational theory is informed by sociology, psychology, and philosophy. At this time, it would seem to be centred on the philosophy of Constructivism as a pedagogic model. Present day educational theory and practice would appear to support the constructivist viewpoint that learners build their own knowledge in response to sensory inputs from authentic experiences[19]. In the classroom, this model is manifested in the concepts of apprenticeships, reciprocal teaching, problem-based learning, and collaborative teams.

### **7.2 The Theory of Constructivism:**

In his paper entitled "*Teaching for Understanding: Educating Students for Performance*", Ken Kickbusch [20] begins by stating that teaching (with the goal to improve student performance) necessitates a belief in "*the capacity of students to create, to construct knowledge and to assign meaning to what they have learned and experienced*", and that "*Constructivist theory provides a framework through which the emergent ideas about teaching, learning, and assessment can be unified*".

Kickbusch goes on to quote Talbert, McLaughlin and Rowan (1993) [21] as saying that "*The constructivist view of effective classroom instruction is often called 'teaching for understanding'*".

Kickbusch[20] concludes by saying that "*The idea of constructivism presents a major conceptual challenge in that it requires educators to rethink both their assumptions about the nature of knowledge and about learning.*"

The early work of Berger and Luckmann (1966) [22] in sociology argues that each human being must inevitably develop or construct meaning. That is, each of us must "*make meaning or make sense of our own social world. Knowledge, then, is the result of the individual construction or "sense-making" of reality*".

Constructivism may therefore be defined as “*meaning making*” (Bruner, 1990)[23], rooted in the context of the situation (J. Brown, Collins, & Duguid, 1989)[24], whereby individuals construct their knowledge of, and give meaning to, the external world (Duffy & Jonassen, 1992)[25] (Jonassen, Peck, & Wilson, 1999)[26], (Schunk, 2000)[27] as a product “*shaped by traditions and by a culture’s toolkit of ways of thought*” (Bruner, 1996)[28].

*“Consistent with this view of knowledge, learning must be situated in a rich context, reflective of real-world contexts for this constructive process to occur and transfer to environments beyond the school or training classroom.”* (Bednar, Cunningham, Duffy, & Perry, 1992, p. 22).[29]

Kincheloe and Steinberg (1993, p.301)[30] put the social construction of reality into educational terms:

*“The frontier where the information of the discipline intersects with the understandings and experiences that individuals carry with them to school is the point where knowledge is created (constructed). The . . . teacher facilitates this interaction, helping students to reinterpret their lives and uncover new talents as a result of their encounter with school knowledge”*. According to Kincheloe and Steinberg (1993)[30], the features of a curriculum that cultivate this post-formal, postmodernism thinking include:

***epistemology*** – *the study of theories of knowledge or ways of knowing, particularly in the context of the limits or validity of the various ways of knowing. (the exploration of the forces that produce what the culture validates as knowledge )*

***pattern*** – *the understanding of the connecting patterns and relationships contained in the world*

***process*** – *the cultivation of new ways to read the world that attempt to make sense of both ourselves and contemporary society, and*

***contextualisation*** – *the appreciation that knowledge can never stand alone or be complete in and of itself*

### **7.3 Contextualised Learning:**

Contextualised learning has emerged as one of many elements which, together, are seen to constitute an improved approach to education, based on constructivist principles of learning. These principles are set out by Grabinger and Dunlap in their theory of the REAL - Rich Environment for Active Learning (Grabinger et al, 1997)[32].

The paper entitled "*Rich environments for active learning: a definition*" (Grabinger, R. S. and J. C. Dunlap, 1995)[31] and presentation entitled "*Rich environments for active learning and distributed learning*" (Grabinger, S., Batty, M., & Richardson, K., 1999)[32], as presented to the American Educational Research Association, outline the attributes for these REAL instructional systems, and their associated implementational complexities. According to Grabinger, an important aspect of a REAL system is the Authentic Learning Context, and in this regard, Grabinger[33] recommends that teachers should "***Contextualise learning using cases, problems, realistic projects and simulations***".

University of Cape Town

## 8. A software design for an ear-training software prototype which incorporates simple contextualisation.

### 8.1 *Prototype requirements specification:*

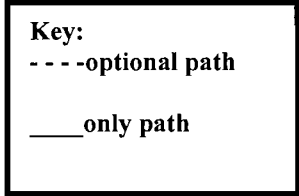
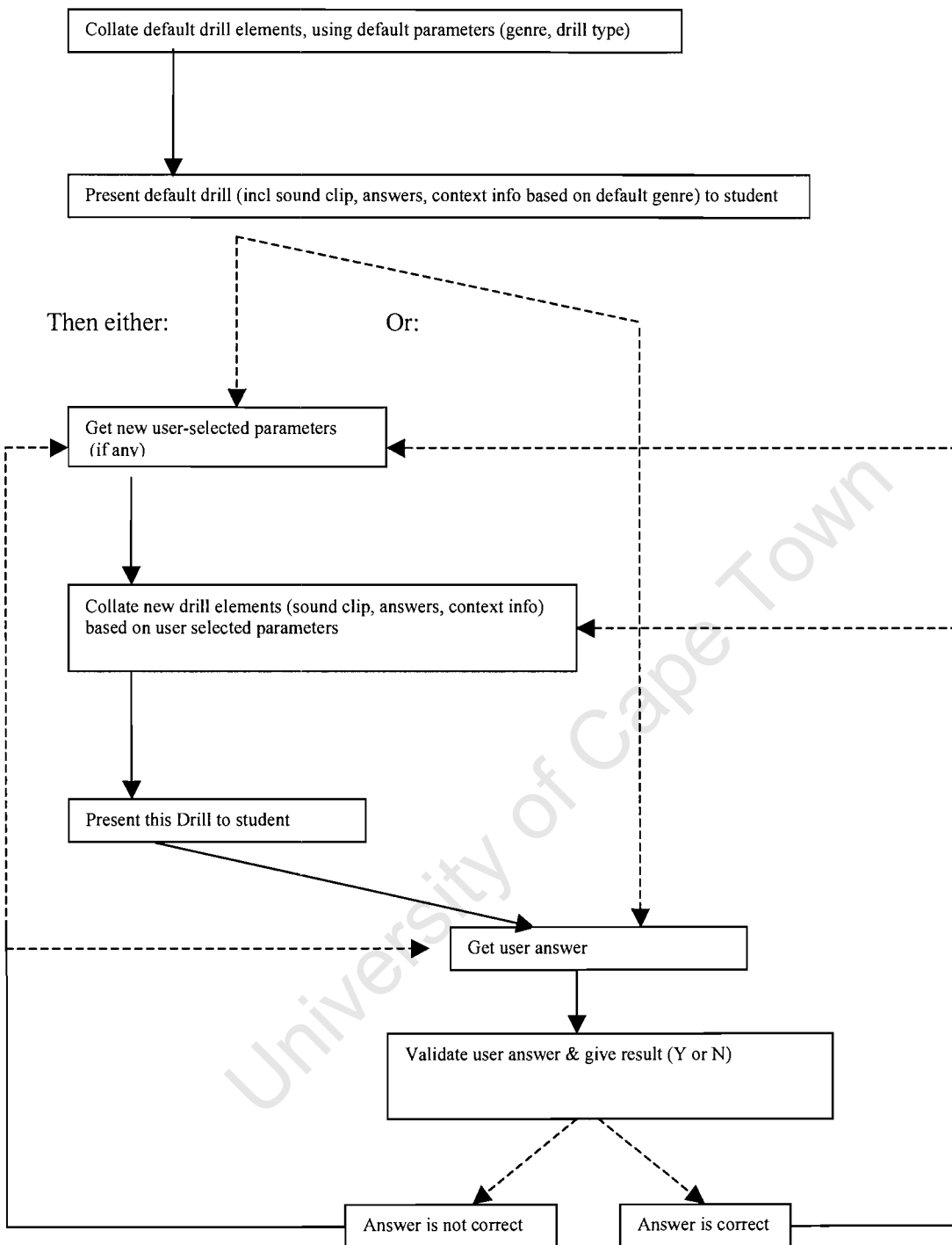
The software will need to be a tool for testing and improving one's aural ability, i.e. the ability to hear, understand, recognise and identify certain musical aspects such as chord sequences, rhythmic patterns in context, pitches and intervals between notes, and so on.

Functional overview of prototype requirements: (**contextualisation** elements are in bold)

- A student should be able to **specify the genre** (style) of music in which the ear training exercise (drill) should take place (e.g. classical or jazz or rock). This option will assist in contextualising the aural training programme, by making it more meaningful and relevant to the student.
- A student should be able to **choose the kind of drill** he/she would like to attempt (such as notation, recognition of rhythms, identification of scales, etc). Again, this will allow a user to focus on his/her choice of skill requiring improvement.
- The programme should present a suitable drill to the student, based on the parameters supplied. The musical sample should be presented as a sound clip.
- The student should be able to play the sound clip and identify aspects of the clip as required for the drill.
- Each drill should present five possible answers to the user, who should be expected to select one of these as the correct answer.
- The student's answer should then be validated by the programme. If the answer is correct, the programme should load and present the next drill that matches the student's parameters. If not, the same drill should inform the user that the previous answer was incorrect, and present the same drill to the student for a second (or further) attempt at answering it correctly.
- The student should be able to change his selection of parameters at any time, or exit at any time.
- In addition to the above, the programme should present **information snippets** to the student, based on the student's selected genre. These snippets should be updated in the GUI for every sound clip played. This sort of information can, for example, include notes on aspects of a composer's life, social and political trends of a certain musical period, examples of some composers works, reviews of first performances of a work, and so on.

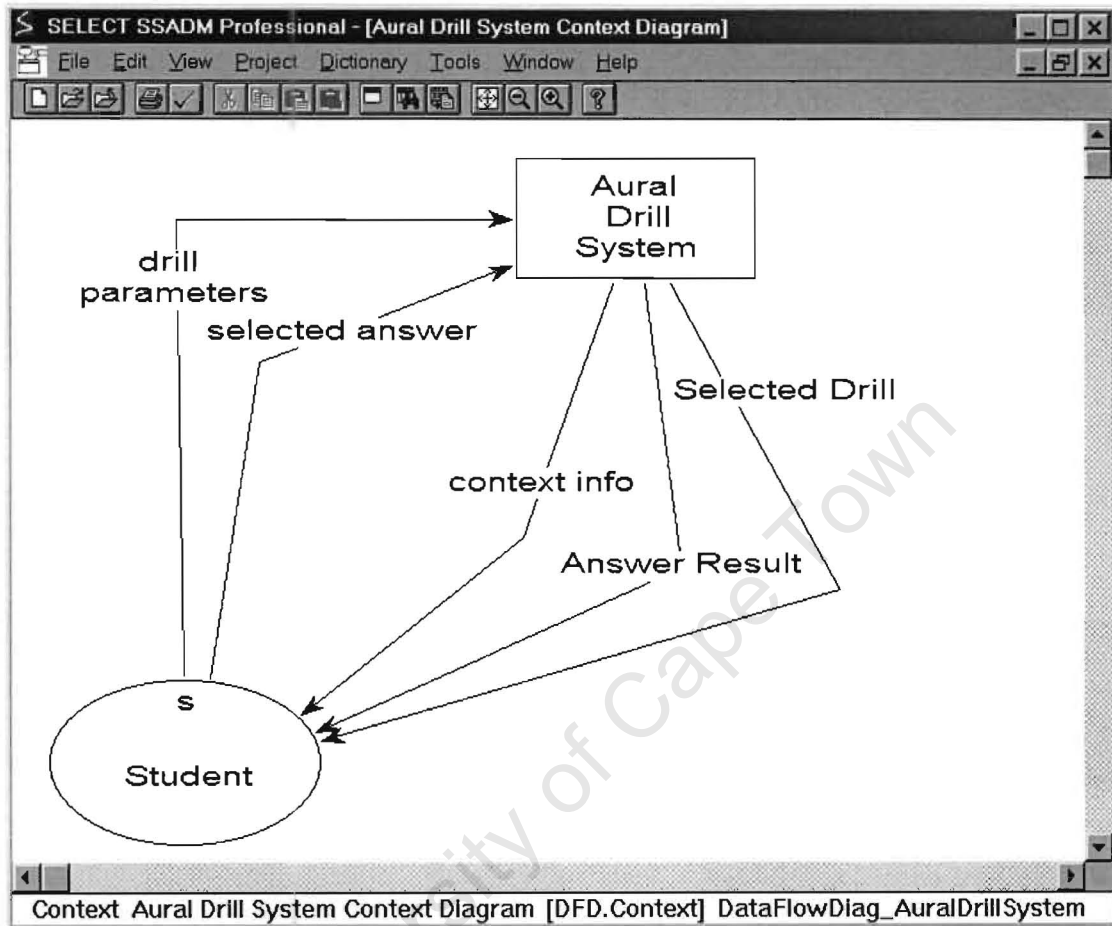
### 8.2 *Pseudocode:*

Default, on start up:

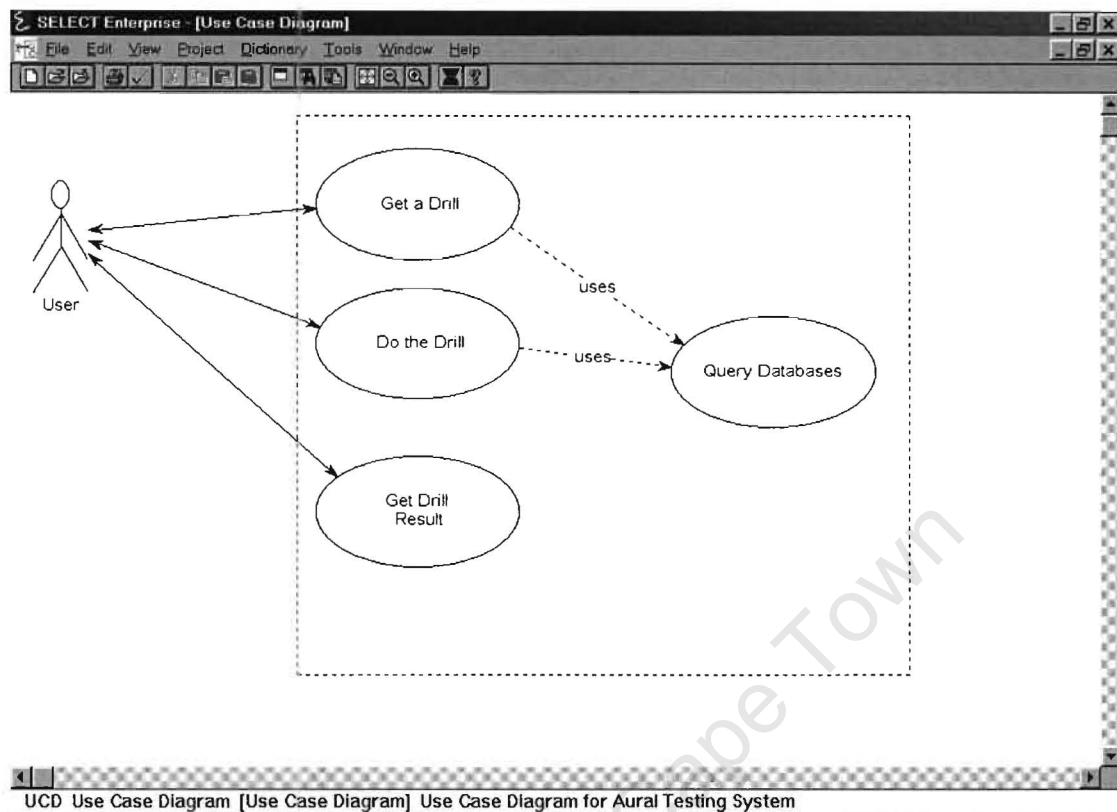


**8.3 Data Flow Diagram (Context):**

In order to assist with the modelling of the prototype system, **Select SSADM Professional** and **Select Enterprise** software was used to create the following diagrams:



## 8.4 Use Case Modelling:



**Note:** although the above diagram indicates that the programme should query a database, the prototype to be developed for the purposes of this paper will simply read sample files for the required information, to enable the prototype to remain light-weight and portable.

### **8.5 Software requirements:**

For the purposes of this prototype, it is apparent that:

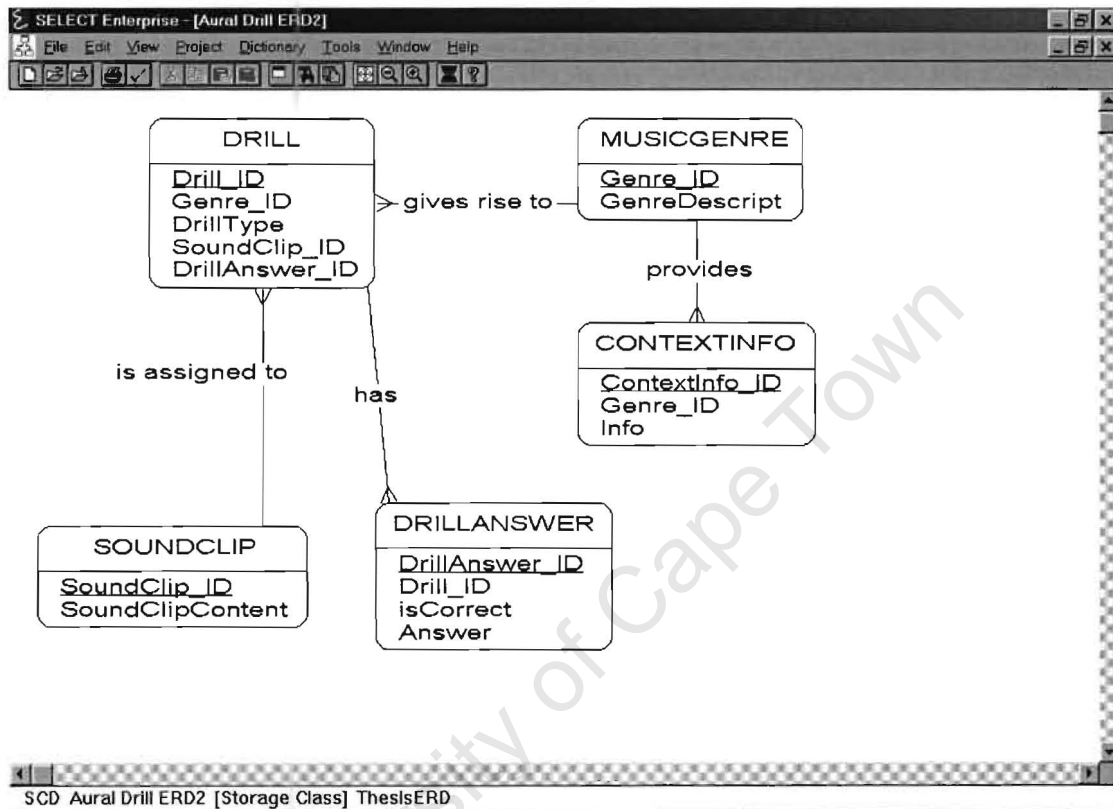
- there is no need to store session information on user.
- there is a need to store, and access information on drills, music genres, context info, sound clips and drill answers.
- the prototype should be written in a language that is portable and robust as this system may be required to run on a variety of operating systems.
- sound clips are to be played to the user, and as this is an integral feature of this system, it must be incorporated into the prototype.
- the prototype must also be capable of displaying a GUI for the drill elements, as well as being able to format and present contextualised information.

As a result of the above requirements, the following software requirements definition will apply to this prototype:

- The prototype will be written exclusively in the Java programming language, specifically Java version 1.3.
- The prototype will employ Swing technology for the graphical elements.
- The prototype will make use of the AudioClip class in java.awt.Applet, in order to play sound clips. This functionality may need to be extended over a number of classes, but must, in its entirety, be incorporated into the prototype GUI.
- A fully implemented version of this software would almost certainly make use of a database and a number of tables. However, for simplicity, this prototype will instead read from a number of text files.
- The prototype will demonstrate a small, fixed number of examples in order to illustrate its main focus, i.e. contextualisation of information.

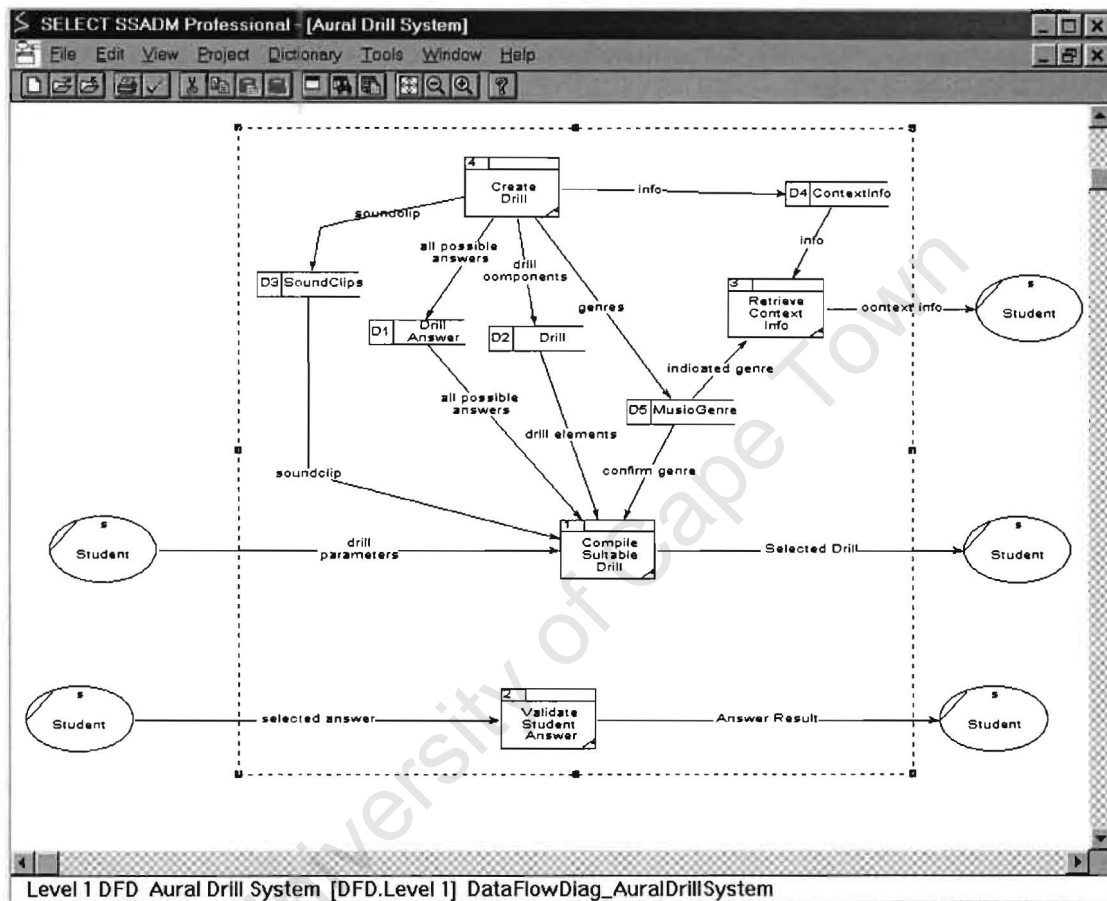
### 8.6 Software design definition: (Entity Relationship Diagram)

The Entity Relationship Diagram (also known as Logical Data Structure) for this prototype will take the following format:



### 8.7 Software design definition (Data Flow Diagram – Level One):

Following on from the Context Diagram included previously in this paper, the diagram at level 1 (below) takes a closer look at the requirements for interaction between the entities external to the system under development (i.e. the user) and the prototype itself:



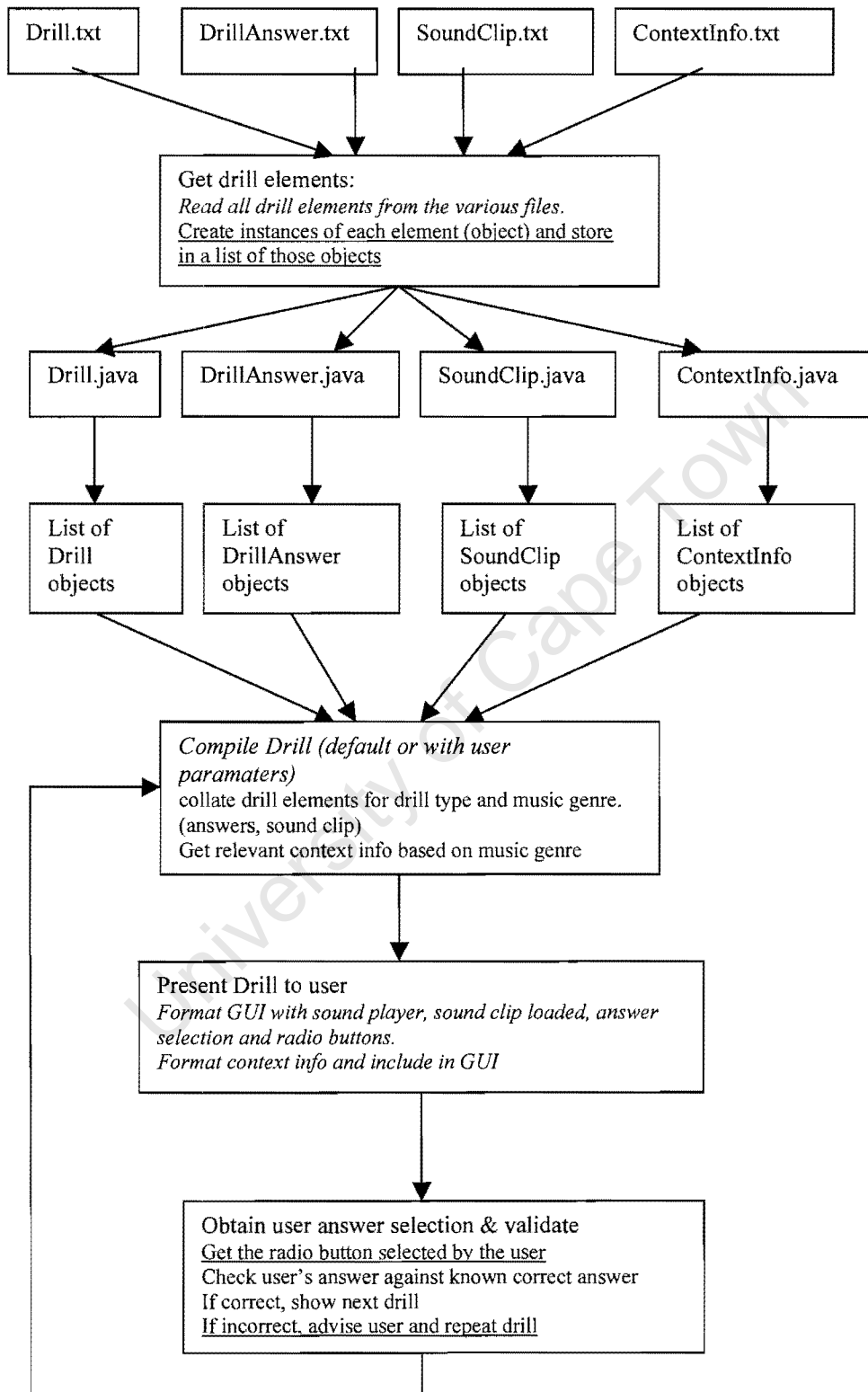
**Note:** there is no input in the above diagram for the process called Create Drill.

This is because the functionality associated with such an input is not part of the system being modelled.

For the purposes of this prototype, the drills are pre-formatted as Strings, and read by the system from text files. A full development of the prototype would probably allow a system administrator to create drills as needed, in which case this activity would form the input to the Create Drill process in the above model.

### 8.8 Software design document (System Architecture and Detailed Design):

The architecture of the prototype aural training system may be illustrated as follows:



The Software Detailed Design Document [App05] is included in the Appendix. This document provides a detailed outline of the classes built for the prototype, including a list of attributes and methods (with parameters and return types) for each class.

### 8.9 Software testing against product requirements:

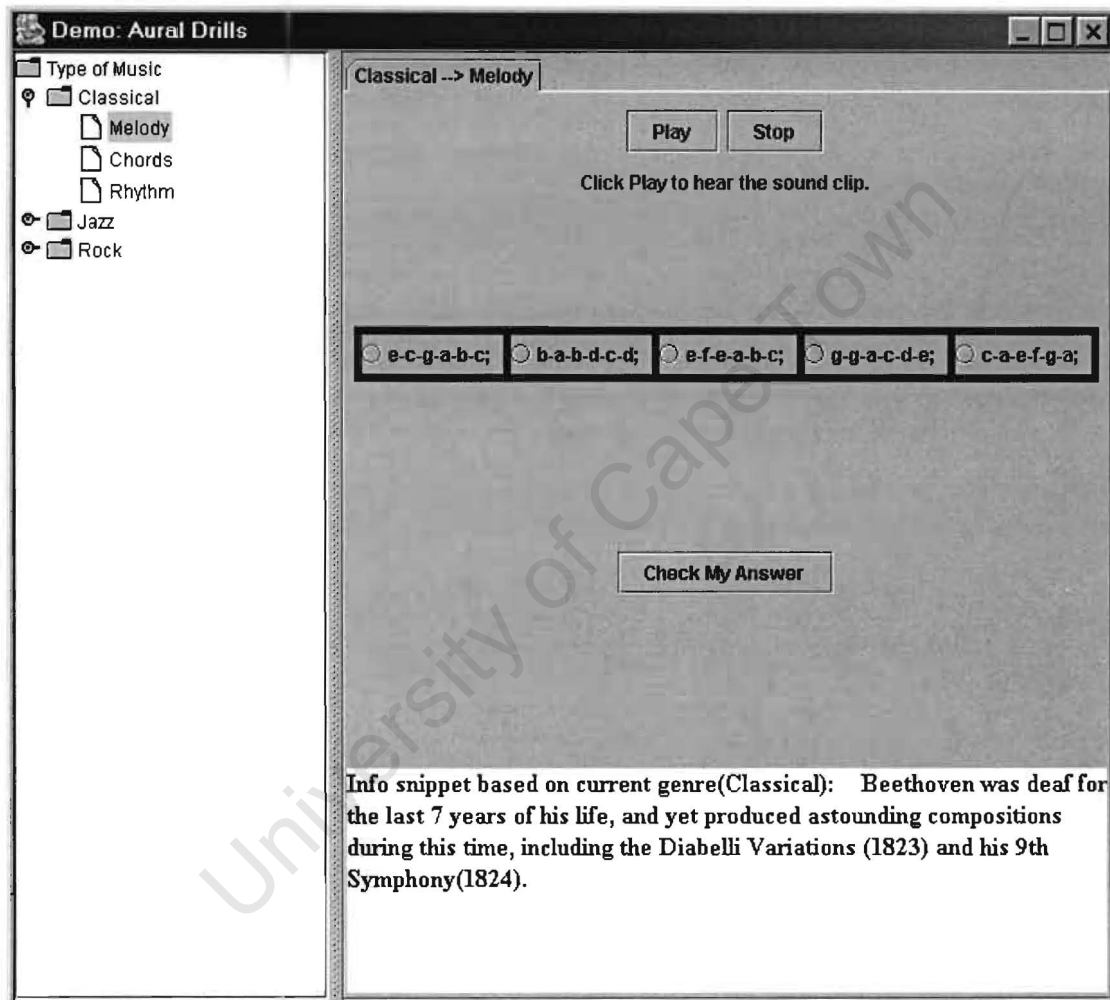
The following table is included to show that the completed prototype meets the product requirements outlined earlier in this paper.

<b>Requirement</b>	<b>Matching Prototype feature/ability</b>
<i>No storage of session data</i>	<i>No storage of session data</i>
<i>Store info on drills, music genres, context info, sound clips and drill answers</i>	<i>Yes, this info is stored in, and read from text files on start of programme</i>
<i>Programming language to be portable and robust</i>	<i>Yes – prototype written in Java</i>
<i>Sound clips to be played to the user</i>	<i>Yes – achieved using java.awt.AudioClip contained within the prototype GUI</i>
<i>GUI display, formatting and presentation of context information</i>	<i>Yes – achieved using Swing components of Java, and by reading context information from files with associated formatting</i>
<i>Small, fixed number of samples for demonstration</i>	<i>Yes – prototype features 3 drills for each drill type (e.g. melody, rhythm) in each genre (e.g. classical, jazz, rock)</i>
<i>Written in Java</i>	<i>Yes, Java version 1.3</i>
<i>Swing technology for graphical elements</i>	<i>Yes – this is included in the main class Demo.java</i>
<i>Use of the AudioClip class in java.awt.Applet, in order to play sound clips.</i>	<i>Yes – this is spread over 3 classes, namely Player.java, PlayListLoader.java and PlayList.java</i>
<i>No database</i>	<i>Yes – prototype reads from a number of files for the purposes of demonstrating the software in a light-weight version</i>

### 8.10 Screen shots of the prototype ear-training tool:

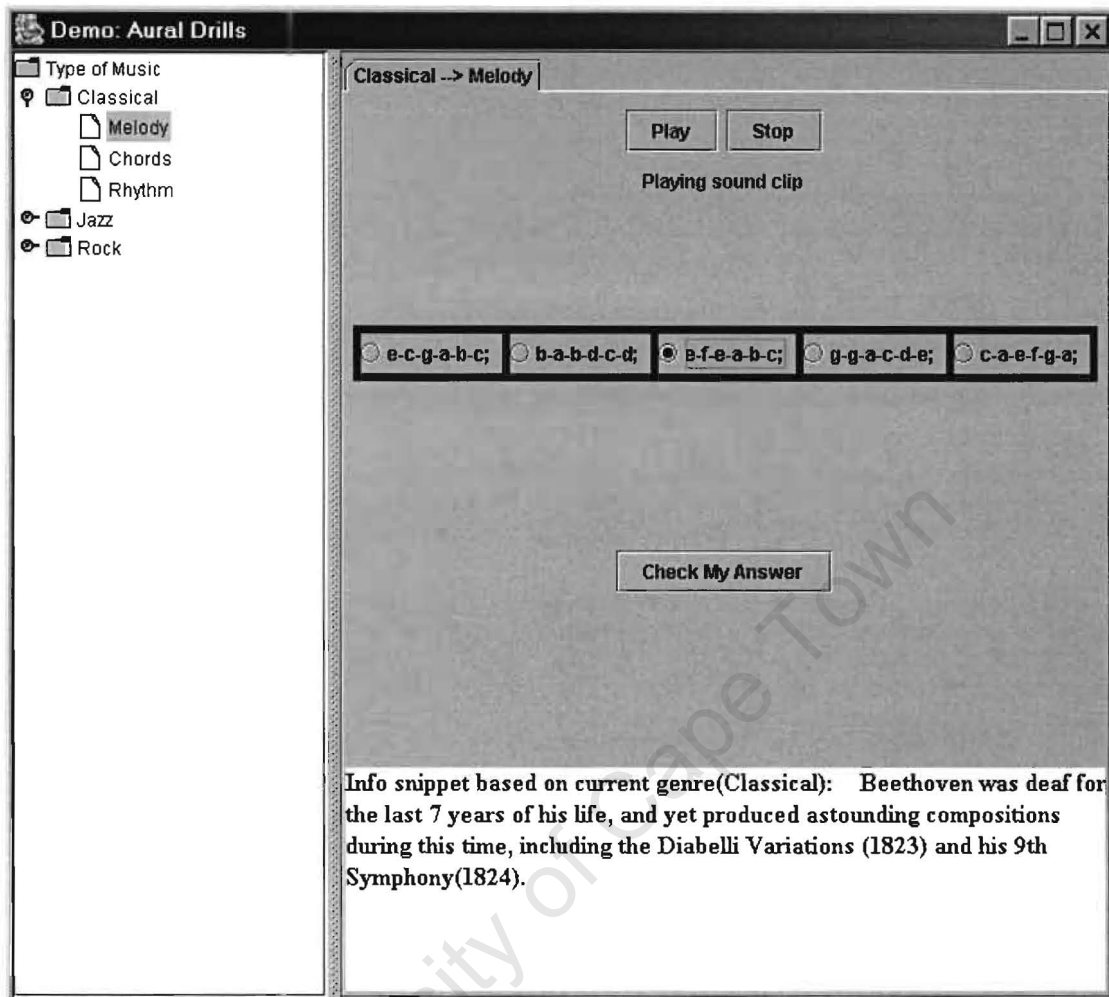
The prototype below will be used for all remaining sections of this paper, in which it is evaluated in terms of its usability and, most importantly, its presentation of contextualisation elements. The prototype code is included in the Appendix[App06].

Screen shot 1: The default Graphical User Interface



This is the default start-up GUI, and includes the default drill type (melody drills) and default genre (classical music). The information snippet presented along with this drill is contextualised in order to be relevant to the classical music genre.

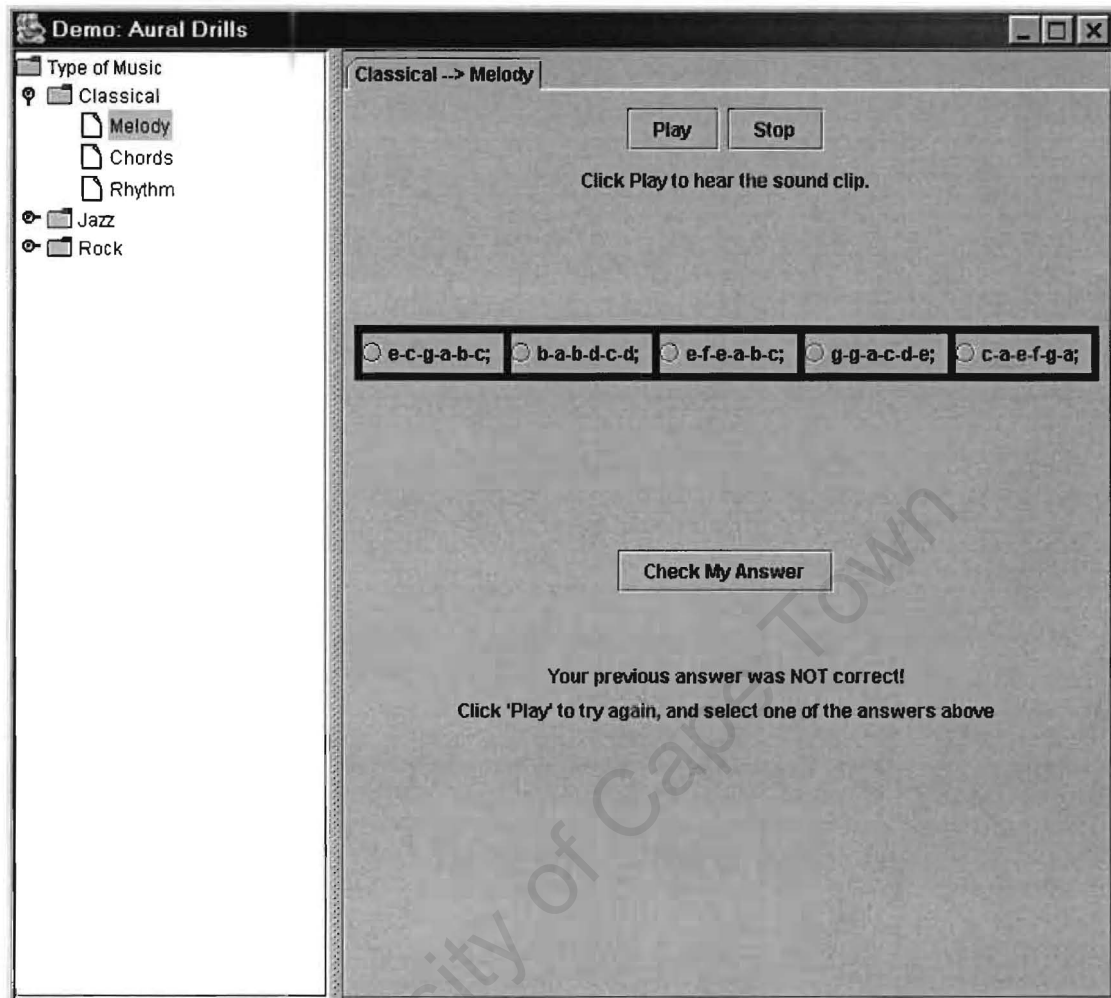
Screen shot 2: Playing the sound clip and selecting an answer



As the drill type objective is Melody, the answers present the notes/pitches of possible melodies in the form of their alphabetic equivalents, using the letters ABCDEFG.

When the user clicks on the “Play” button, a melodic sound sample (of duration 2 – 4 seconds) is played by the embedded sound player. The user is expected to select one of the five possible answers displayed, and click the button labelled “Check My Answer”.

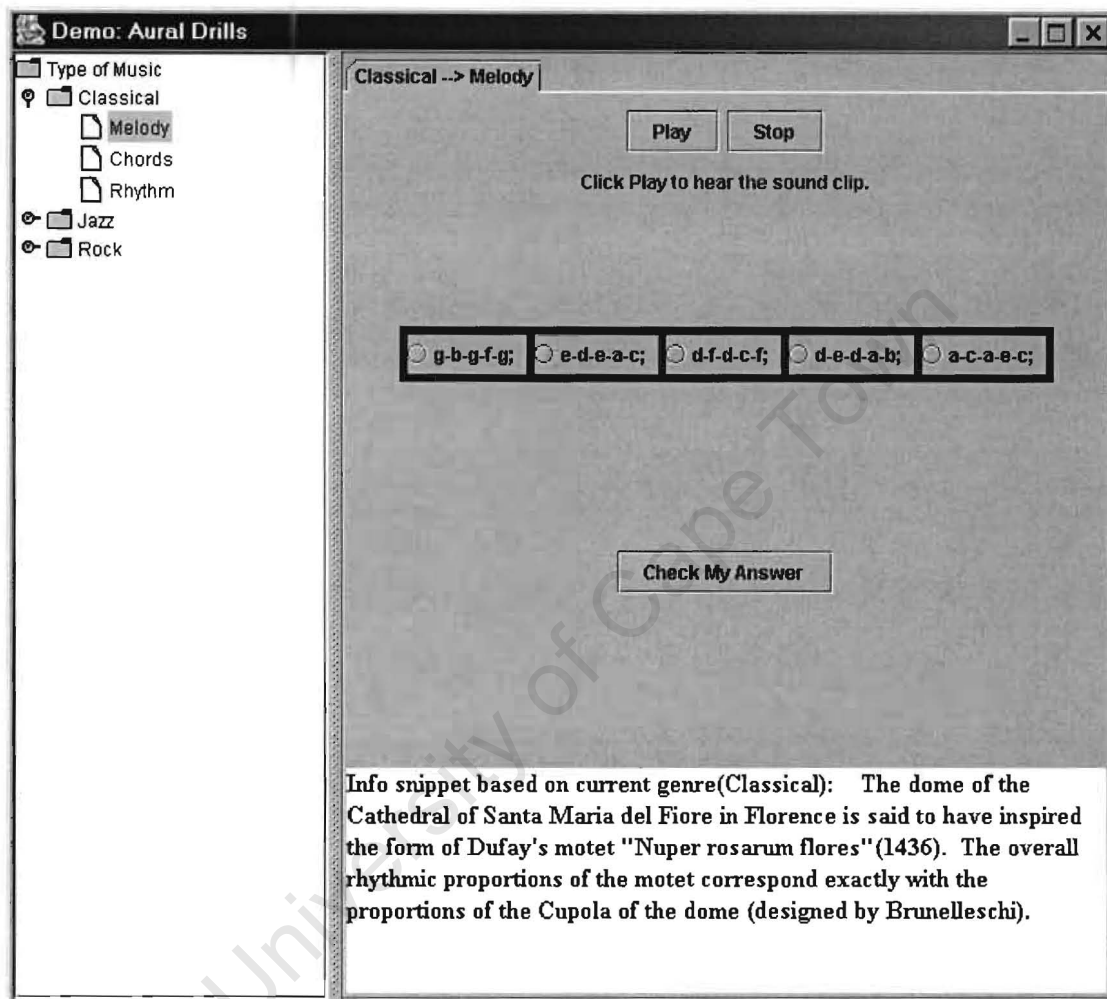
Screen shot 3: Validating the user's answer (when incorrect)



In the above instance, the user's answer was incorrect. In order to make second and further attempts at this drill, the user is asked to click the "Play" button once again (to hear the same sound clip) and to make another selection from the available answers.

Screen shot 4: Validating the user answer (when correct):

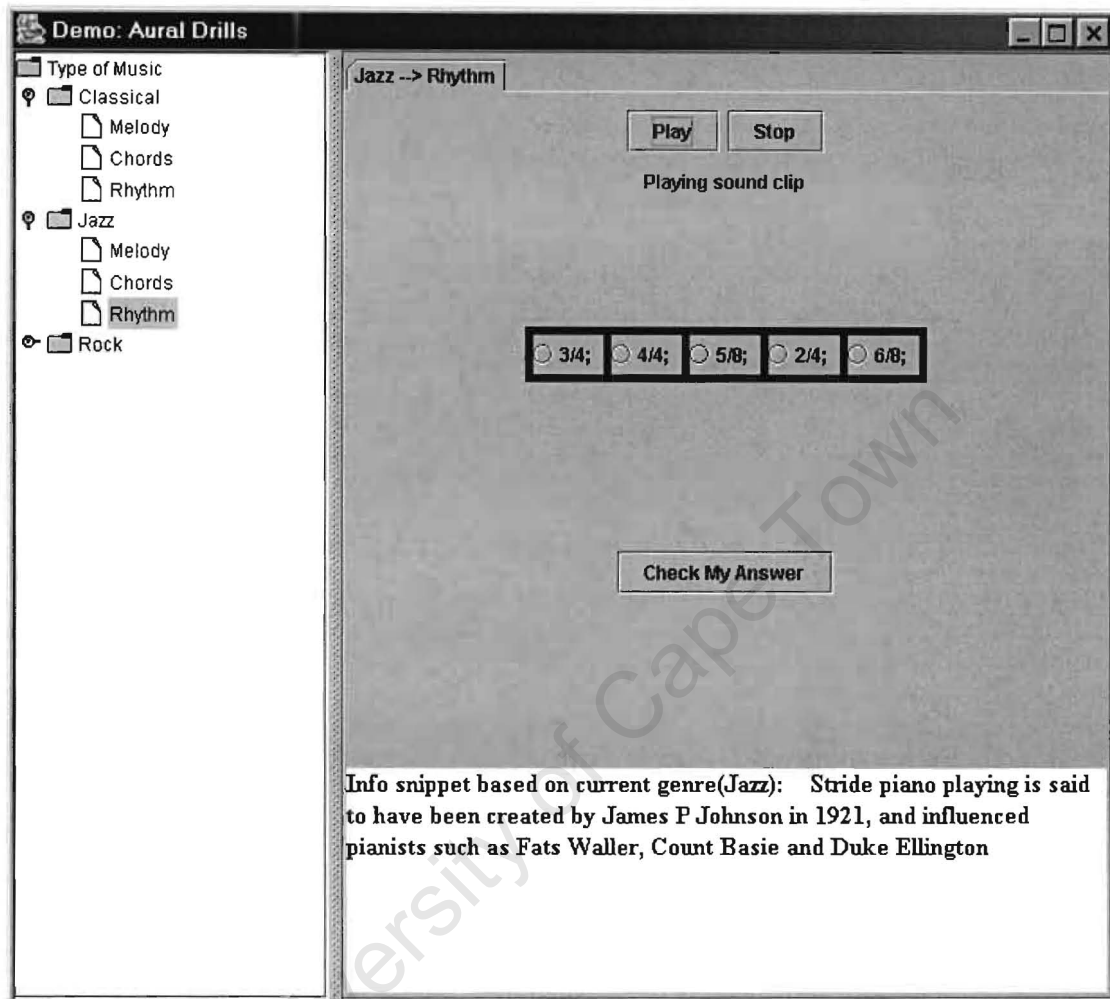
When the user selects the correct answer, the drill programme simply presents the next drill (for the same genre and drill type) along with the next contextualised information snippet, as below:



The prototype drill programme does not inform the user when his/her previous answer is correct (e.g. by means of a pop-up message or success screen). Rather, it simply presents the next drill and its possible answers.

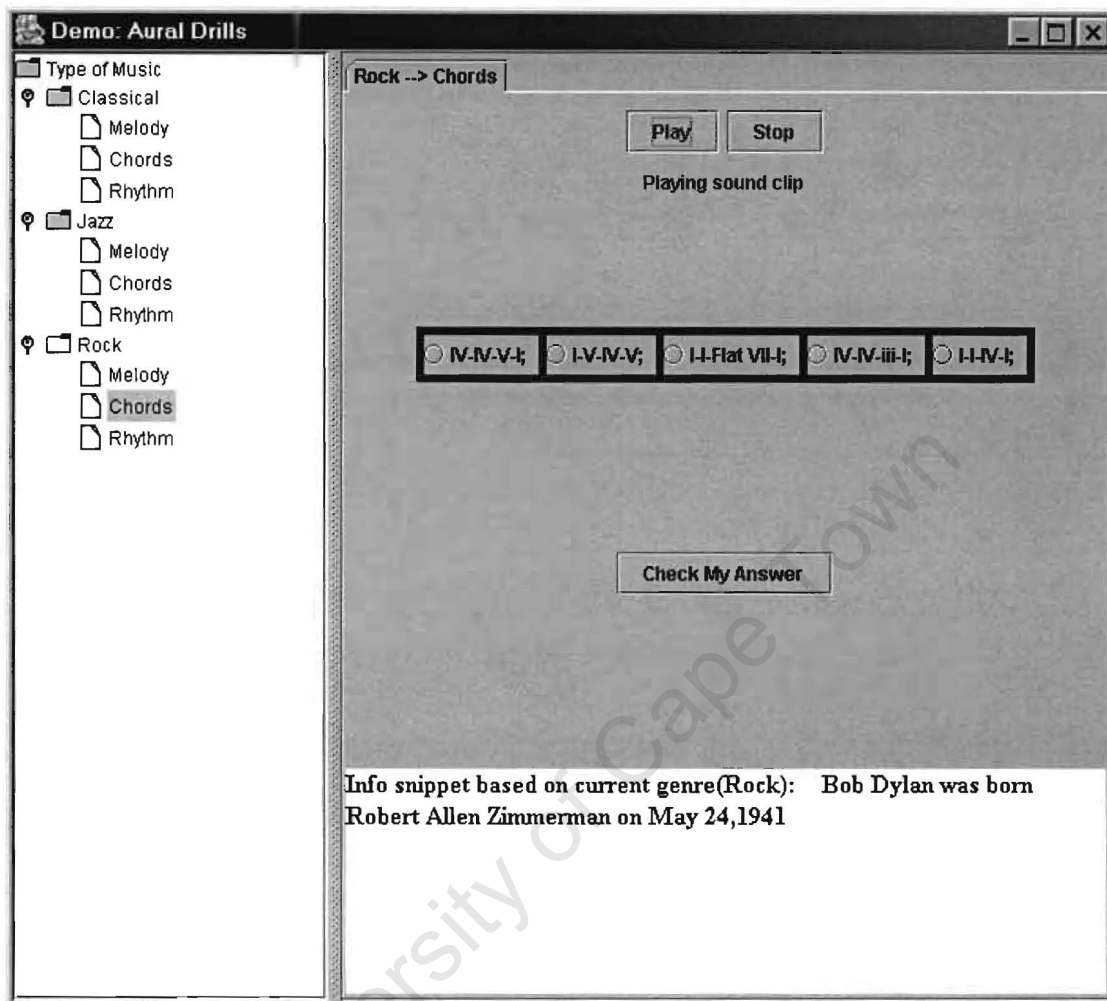
This is because the prototype programme (like most standard drill programmes such as Earope and Auralia) is designed so that the drills may be run in quick succession. As the user becomes more skilled in recognising and identifying the required musical elements, the drills are undertaken at increasing speed. Inserting success messages or indicators may become a hindrance in this process, not least because of the drill's repetitive nature.

Screen shot 5: The user changes selection of music genre and drill type from default values to Jazz genre and Rhythm drill type:



The information snippet now presents contextualised information relating to this genre (i.e. Jazz). The drill type has been changed to rhythmic drills by the user, therefore the sound clip now played is an exercise in recognising jazz rhythm in the form of beats per bar, while the answers reflect the possible time-signatures for this sound clip.

Screen shot 6: The user changes selection of music genre and drill type to Rock (genre) and Chords (drill type):



In this drill, the objective is to recognise the chord progressions of the sound clip. Possible answers are therefore presented in standard harmonic notation (i.e. roman numerals) with capitals indicating major chords (e.g. “IV” represents the chord built on the 4<sup>th</sup> step of the relevant tonic scale, as a major chord), and lower-case indicating minor chords (e.g. “vi” represents the chord built on the 6<sup>th</sup> step of the relevant tonic scale, as a minor chord).

Note: All notation used in this prototype is based on standard musical notation forms, and will be familiar to music students.

## 9. Evaluating the prototype software

The main focus of this evaluation is **to determine whether ear-training software, when combined with an element of contextualised information**, and presented in prototype format, **will have any impact on student interest levels in the software**. Therefore, the evaluation process will focus on the new element (i.e. contextualisation) introduced in the prototype, rather than the general usability of the prototype.

The interview in the initial, preliminary survey reveals that user opinion of non-contextualised ear training software is, generally speaking, quite low. This observation is supported by student comments such as “mundane” and “boring”. As this opinion is only revealed in the initial interview and not the initial survey, it would seem appropriate to use the semi-structured interview technique once again, when evaluating the prototype ear training programme.

Therefore, for the prototype programme, the evaluation process will proceed as follows: The student will be provided with the prototype software. The student will be given an opportunity to perform a number of drills using the prototype, in order to familiarise himself (or herself) with the new system. It is anticipated that this familiarisation should not take very long, as the prototype is a fairly simple programme.

The student will then be asked about certain aspects of the prototype, as part of a semi-structured interview. The featured topics for the interview will centre of features provided by the prototype, including:

- The option to select a genre for a drill (e.g. classical, jazz, rock)
- The option to select the types of drill (e.g. melody, chords, rhythm)
- The provision of contextualised information

Regarding the inclusion of contextualised information in this prototype, specific questions will be asked as follows:

- Does this place the drill in a more musical context?
- Does this give the drill some relevance or context?
- Do the info snippets add to, or detract from, doing the drills?
- Are these info snippets interesting on any level?
- Do these info snippets make the programme more meaningful in any sense?
- Would you consider this feature to be any kind of improvement on previous ear-training software you have used?

### 9.1 Evaluation results:

When the prototype is presented to the students, they are almost immediately comfortable with using the programme, as it is designed to run in much the same way as other ear training software and therefore seems quite familiar to them. The students are given approximately 5 minutes to perform a number of the drills using the prototype, after which the interview process begins.

In the interview, it becomes apparent that the student considers the option to select a genre for a drill as a new feature, not seen in ear-training programmes previously used by the student. The immediate comment is that this makes the drill seem more interesting.

The student describes the categorisation of drills into types (e.g. melody, chords, or rhythm) as a fairly standard feature of ear-training programmes. This is a valid observation, and indicates that the prototype design is comparable on this level with other drill programmes.

Answers to the specific questions regarding the contextualised information are as follows:

Question 1: *Does this place the drill in a more musical context?*

Answer: Yes, in that the context is broadened to include other information aside from the drill.

Question 2: *Does this give the drill some relevance or context?*

Answer: The drill is no longer as mindless. The ability to select genres along with drill types means that one is able to do the drills in a relevant format – jazz students can now do the drills using the jazz genre, which is more relevant to their interests and area of study. A suggestion is to expand this software to feature drills played by instruments other than the piano. This will be a great attraction to music students whose instrument is not the piano but who currently have to practice drills in a piano format. Customising the type of instrument would therefore make the drills even more relevant.

Question 3: *Do the info snippets add to, or detract from, doing the drills?*

Answer: The info snippets add greatly to the drill. Even though they may be slightly distracting, it's a "good distraction".

Question 4: *Are these info snippets interesting on any level?*

Answer: The info snippets are interesting. A suggestion is to match them to the drills being done, rather than the genre type currently selected.

Question 5: *Do these info snippets make the programme more meaningful in any sense?*

Answer: They (the info snippets) add meaning to the drill, in that one learns something else whilst doing the primary exercise.

Question 6: *Would you consider this feature to be any kind of improvement on previous ear-training software you have used?*

Answer: Yes, this kind of drill programme is more interesting.

### **9.2 Comment on the prototype evaluation:**

The evaluation results of the prototype would therefore seem to support the theory and suggestions regarding the possible impact of contextualisation, as expressed in this research paper.

University of Cape Town

## 10. Conclusion

A brief summary of the main aims and activities of this research paper may be helpful at this point:

This research paper began with an investigation into the type of music software available to, and used by, tertiary level music students, as well as students' attitudes to this software. The research intended to discover whether any room for improvement could be found in the design of music software available to students.

HCI evaluation methods determined that a problem existed regarding the ear-training programme. Students expressed dissatisfaction with this software, and their interest in this programme was rated as very low. Therefore, the design of ear-training software became the focus for the remainder of the research paper.

In an attempt to solve this problem, possible reasons for student's lack of interest were examined and investigated. This led to a proposal that the design process for such ear-training software did not seem to address an important user (in this case, student) requirement, i.e. that the software should have meaning and relevance for the students.

This research paper then proposed that widely accepted educational approaches such as contextualisation, and pedagogic models such as Constructivism, should be introduced as key elements into the software design process, in an attempt to address this requirement.

In order to test this proposal, a prototype ear-training programme was created. The design for the prototype incorporated the suggested approach by including contextualised features, along with standard ear-training programming features.

The prototype was then evaluated (by means of an interview with the students) to gauge whether student interest levels would be affected in any way by the inclusion of these contextualisation elements in an otherwise standard ear-training programme.

The prototype evaluation interview revealed promising results in the form of increased interest in the software as a direct result of the contextualised features.

### ***10.1 The features and benefits of the prototype ear-training software:***

In general, the prototype design (in terms of both the user interface and the way in which the programme runs) is based on a typical design found in most ear-training software programmes.

However, the prototype does employ a number of **contextualised features**, which sets it apart from other software packages in this field. It is these features that go some way towards increasing students' interest levels in the software.

Firstly, the prototype ear-training programme offers a **choice of musical genres** (categories or styles of music) in which the ear-training drills can be attempted. This represents a departure from other ear-training software, where users are usually restricted to attempting the drills in a single genre (e.g. classical music).

The sample genres on offer in the prototype are classical music, jazz and rock music. These genre possibilities allow students to undertake drills using a style of music that they particularly enjoy, or with which they are particularly comfortable or familiar. The inclusion of rock music and jazz is therefore of great benefit to those students who are either not interested in, or not studying, classical music. In this way, the musical style for the ear-training drills is made more meaningful and relevant to these students.

The prototype therefore offers a new approach by allowing the ear-training programme to be incorporated into the students' immediate realm of musical experience, so that he/she is presented with a more relevant opportunity for learning.

Secondly, the prototype presents **information snippets** to the user for each drill attempted. These snippets offer small amounts of interesting information, based on the user's current selection of musical genre. The information contained in these snippets relates to a broad range of fields, such as a commentary on aspects of society at that point in history, the relationship between music and architecture in early classical music, interesting facts about composers of the genre, political issues at the time, etc.

The prototype therefore presents the user with a subtle distraction to the potentially mundane exercise of attempting these ear-training drills. The information snippets contextualise the musical genre by providing a glimpse of the "bigger" picture of that period of time, while simultaneously affording the student an opportunity to learn interesting facts in a non-taxing way.

## **10.2 Techniques used in this research:**

This research made use of certain Human-Computer Interaction and Software Engineering techniques, as follows:

### ***HCI evaluation techniques: Questionnaire and interviews***

During the course of this research, fairly extensive use was made of HCI evaluation techniques, namely questionnaires and interviews.

A questionnaire was designed and used for the initial evaluation of music software, where it was deemed necessary to collate and compare user ratings for various aspects of the software under evaluation. The questionnaire proved to be a very useful tool for gathering qualitative information (i.e. the opinions of the participants) in a quantitative way (i.e. expressed as scores out of 5).

The interview process followed a semi-structured format, and provided an excellent opportunity for obtaining the survey participants' thoughts and opinions, expressed freely, on the selected topics.

As mentioned previously, the questionnaire did not reveal the problem ultimately addressed by this research paper (i.e. student's lack of interest in ear training programmes). This problem only came to light during the course of the interviews held in conjunction with the survey. This would seem to suggest that, whenever possible, both evaluation methods should be employed as complementary approaches in order to obtain the most accurate results.

Finally, the design and compilation of the questionnaire, along with the analysis of the completed questionnaires, required an extraordinary amount of time and attention. In comparison, the interview required a far shorter compilation time.

### ***Software Engineering techniques: Modelling tools and documentation***

A number of software engineering techniques were used in the design and development of the prototype ear-training programme.

Firstly, pseudocode was employed to illustrate a general overview of the way in which the prototype should perform.

Thereafter, the prototype design made use of modelling techniques in order to examine the proposed system from a number of angles. Modelling tools such as Select Enterprise and Select SSADM Professional proved invaluable in this part of the design phase.

Using these tools, a Context diagram helped to establish the initial context in which the prototype would function. This was clarified by the construction of a Use Case model for the system. An Entity Relationship Diagram (also known as a Logical Data Structure) was then created to illustrate the information storage requirements for such a system – however, it was decided that, for the purpose of the prototype, no database would be used. Lastly, a Data Flow Diagram (level 1) was compiled to show the interaction between entities external to the system, and the system itself.

Further software engineering techniques in the form of documentation were used in the design and development of the prototype. These included a Requirements Specification (to ensure that the prototype system would contain certain required functionality), a System Architecture document which addressed the architectural design requirements for the prototype, and a Software Detailed Design Document in which programme code decisions were recorded.

Once the prototype system's coding was complete, it was subject to two testing processes. The first test phase ensured that the prototype's functionality matched the requirements set out in the requirements specification. The second test phase concerned testing the functionality of the prototype code.

These software engineering techniques were extremely helpful in maintaining clarity and consistency, in the course of the prototype design. Furthermore, they assisted in the planning and construction of a work schedule, and facilitated a steady, logical approach to the implementation of the prototype design.

### ***10.3 The results of this research paper's investigations:***

This research paper set out to investigate the possibilities for improvement in music software. It has accomplished this task by establishing that, with regard to the design of ear-training programmes, improvements can be made in terms of addressing user interest levels in the software.

The favourable student reaction to the prototype points to contextualisation as an important means of capturing and sustaining user interest.

### ***10.4 Implications of this research paper:***

This research paper implies that the existing software design process for music education software (in this case, ear training software), could be complemented by incorporating prevailing pedagogic models and education methodology “best practices” (of which one example is contextualised content).

Further investigation, examining the scope and limitations of such a design process, may be of significant benefit to future music students as a result of the potential for improved music education software.

University of Cape Town

## 11. References (by section)

### Background

[1] Miller, Larry; Olson, John. (1994). "Putting the Computer in Its Place: A Study of Teaching with Technology"

Journal of Curriculum Studies; v26 n2 p121-41 Mar-Apr 1994

(REPORT\_NO: ISSN-0022-0272, ERIC number EJ493884)

[2] Warger, Cynthia, Ed. (1990)

"Technology in Today's Schools" (report)

Association for Supervision and Curriculum Development (ASCD),

ISBN-0-87120-169-0

ERIC\_NO: ED323961

[3] Somekh, Bridget; Mavers, Diane; Lewin, Cathy (2002)

"Broadening Access to the Curriculum through Using Technology To Link Home and School: A Critical Analysis of Reforms To Improve Educational Attainment for All K-12 Students. "

Paper presented at the Annual Meeting of the American Educational Research Association (New Orleans, LA, April 1-5, 2002).

ERIC\_NO: ED467195

[4] Polka, Walter S; Mattai, P Rudy (2001)

"Curriculum planning in the 21<sup>st</sup> century: Managing technology, diversity and constructivism to create appropriate learning environments for all students"

Paper presented at the 19<sup>th</sup> International Conference on Technology and Education

Proceedings (19th, Tallahassee, Florida, May 2-5, 2001)

[http://www.ictc.org/T01\\_Library/T01\\_123.PDF](http://www.ictc.org/T01_Library/T01_123.PDF)

(and further info from [www.ictc.org](http://www.ictc.org))

[5] Sample education software retail sites indicating range and diversity:

<http://www.gradware.com/>

<http://www.journeved.com/>

<http://www.ccvsoftware.com/>

<http://www.edu-software.com/>

**[6] Music technology courses (and subjects) offered by universities and colleges:**

New York University

<http://www.nyu.edu/education/music/mtech/>

Radford University Centre for Music Technology

<http://www.radford.edu/~cmt-web/>

University of Kent

<http://www.kent.ac.uk/studying/undergrad/medway/subjects/music.html>

University of Nottingham

<http://www.nottingham.ac.uk/music/UGinfo.html#course>

University of Natal, South Africa

<http://www.und.ac.za/und/music/mtech.html>

Rhodes University, South Africa

<http://www.rhodes.ac.za/academic/departments/music/fofs.html>

**Human Computer Interaction:**

[7] Association for Computing Machinery

[www.acm.org](http://www.acm.org) (special interest group SIGCHI)

[8] Sommerville, Ian,(1995). Section entitled “Human Factors”,

“Software Engineering”, 5<sup>th</sup> ed., pg38. Addison Wesley Longman Limited

**[9] Music Department curriculum subjects:**

South African College of Music, University of Cape Town

<http://web.uct.ac.za/depts/sacm/>

University of Natal, Music Department

<http://www.nu.ac.za/handbooks/introduction.asp?nav=040118>

Rhodes University Music Department

<http://www.ru.ac.za/academic/calendar2003/music.html>

Trinity College Dublin School of Music

<http://www2.tcd.ie/Music/courses.html>

Conservatory of Music and Drama, Dublin Institute of Technology

<http://www.dit.ie/DIT/study/undergraduate/courses/ft601.html>

[11] **Music software programmes:**

Sibelius and Auralia

<http://www.sibelius.com/products/>

Cakewalk

<http://www.cakewalk.com/Products/HomeStudio/default.asp>

Cubase

<http://www.steinberg.net/en/index.php>

Earope

<http://www.cope.dk/>

**Using software in education:**

[12] Hyjek, Phil; Gilbert, Allen; Graham, Christine; Marsters, David; Repos, Judi; Romond, Bill; Soule, Chuck; Tuscany, Bonnie. (1998)

“Technology in the Classroom: Computers and Instruction in Vermont's Schools”

Vermont Inst. for Science, Math and Technology, Burlington

ERIC\_NO: ED439665

[13] diSessa, Andrea A (2000)

“Changing Minds: Computers, Learning, and Literacy”

MIT Press, Massachusetts Institute of Technology, Cambridge

ISBN-0-262-04180-4

ERIC\_NO: ED441394

[14] Alan Dix et al (1998). Human Computer Interaction (2nd Edition), Chapter (4,5).

[15] **HCI evaluation methods:**

Usability Net

<http://www.usabilitynet.org>

Special Interest Group in Computer Human Interaction

[http://sigchi.org/cdg/cdg2.html#2\\_1](http://sigchi.org/cdg/cdg2.html#2_1)

Association for Computing Machinery

<http://www.acm.org/sigchi/>

[16] University College Cork - Human Factors Research Group and Dr Kirakowski, Dept of Applied Psychology

<http://www.ucc.ie/hfrg>

[17] Dr. Rita S. Y. Berry. Paper presented at the British Educational Research Association Annual Conference, University of Sussex at Brighton, September 2 - 5 1999, entitled "Collecting data by in-depth interviewing"

<http://www.leeds.ac.uk/educol/documents/000001172.htm>

[http://www.slais.ubc.ca/resources/research\\_methods/interviews.htm#semi](http://www.slais.ubc.ca/resources/research_methods/interviews.htm#semi)

<http://www.audienceialogue.org/kya10.html>

[18] "Qualitative Methods"

Research Methods Knowledge Base

William M. Trochim, Cornell University

<http://trochim.human.cornell.edu/kb/qualmeth.htm>

**Background to educational theory:**

[19] Poole, D. M. (2000). An email activity: Preservice teachers' perception of authenticity.

[20] Paper: Ken Kickbusch, WEAC research consultant.

Wisconsin Education Association Council, June 1996

<http://www.weac.org/resource/june96/under.htm>

[21] Talbert, Joan E., McLaughlin, Milbrey W., and Rowan, Brian. "Understanding Context Effects on Secondary School Teaching." TEACHERS COLLEGE RECORD (Fall 1993), p45-68.

**Theory of Constructivism:**

[22] Berger, Peter L. and Luckmann, Thomas (1966). The Social Construction of Reality. Garden City, New York: Anchor, 1966.

[23] Bruner, J. (1990). Acts of meaning. Cambridge, MA: Harvard University Press

[24] Brown, J.S., Collins, A., & Duguid, P. (1989) Situated cognition and the culture of learning. Educational Researcher, p32-42

[25] Duffy, T. M., & Jonassen, D. H. (1992). Constructivism and the technology of instruction: A conversation. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

[26] Jonassen, D. H., Peck, K. L., & Wilson, B. G. (1999). Learning with technology: A constructivist perspective. Upper Saddle River, NJ: Merrill.

[27] Schunk, D. H. (2000). Learning theories: An educational perspective. (3<sup>rd</sup> ed.). Upper Saddle River, NJ: Prentice-Hall, Inc.

[28] Bruner, J. (1996). The culture of education. Cambridge, MA; Harvard University Press

[29] Bednar, A.K., Cunningham, D., Duffy, T.M, & Perry, J. D. (1992). Theory into practice: How do we link? In T. M. Duffy & D. J. Jonassen (Eds.), Constructivism and the technology of instruction: A conversation. (pp 17-34). Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

[30] Kincheloe, Joe L. and Steinberg, Shirley R (1993). "Tentative Description of Post-Formal Thinking: The Critical Confrontation with Cognitive Theory." HARVARD EDUCATIONAL REVIEW p296-320.

### **Contextualised Learning:**

[31] Grabinger, R. S. and Dunlap, J. C. (1995). 'Rich environments for active learning: a definition' in ALT-J 3(2): p5-34.

[32] Grabinger, R. S., Dunlap, J. C. and Duffield, J. A, (1997). 'Rich environments for active learning in action: problem-based learning' p3-17.

[33] Scott Grabinger, Associate Professor and Director of the Technology and Learning Team, University of Colorado, Denver

[http://ceo.cudenver.edu/~scott\\_grabinger/downloads/Files/FIPSEREA.pdf](http://ceo.cudenver.edu/~scott_grabinger/downloads/Files/FIPSEREA.pdf)

[34] Association for Technology in Music Instruction

[www.music.org/atmi](http://www.music.org/atmi)

### **Other resources not directly referenced in the research paper:**

Journal of Technology and Teacher Education, 8, (1), p13-28.

<http://www.aace.org/pubs/jtate/>

ERIC – Education Resources database and specialized periodical database

<http://www.askeric.org/>

### **Appendix references:**

[App01] Appendix A: RawDataCapture.xls

[App02] Appendix B: PerQuestionAnalysis.xls

[App03] Appendix C: AnalysisBySection.xls

[App04] Appendix D: Sample Questionnaire

[App05] Appendix E: Software Detailed Design Document

[App06] Appendix F: Prototype programme code

**Questionnaire (version 1.3)**

**Background on user and software programme used.**

Your name: \_\_\_\_\_

Your age: (please circle) 17-19      20-23      24-26      27+

**This questionnaire aims to gather information about your experience of a piece of software. This will be a software programme that you have used for purposes related to your musical studies. Please provide details of the software programme you are about to rate, as follows:**

**Name of software programme:** \_\_\_\_\_

**Version:** \_\_\_\_\_

**Field of usage:** \_\_\_\_\_

**How long have you been using this software? (in months)**\_\_\_\_\_

**Note:**

*This questionnaire aims to capture the USER EXPERIENCE of using a particular piece of software. All the questions are therefore phrased to elicit the user's feelings and opinions about the software he/she uses.*

*The information gathered in this questionnaire is for research purposes only.*

**Section One: How does this software help you, the user, achieve your tasks and goals?**

In this section, we are interested in the benefits you have experienced through using this software programme, your opinions about the software's abilities (from a user's point of view) and your suggestions for improvements.

*Grading system: (please tick ONE of the following choices for each question)*

(strongly disagree) 1.  2.  3.  4.  5.  (strongly agree) N/A

**1. This software helps me to do what I need to do, with good results.** 1.  2.  3.  4.  5.  N/A

**2. This software helps me to do what I need to do, efficiently.** 1.  2.  3.  4.  5.  N/A

**3. Some examples of the things I use this software for, are:**

**4. I feel I have become more productive by using this software.** 1.  2.  3.  4.  5.  N/A

**5. This software allows me to complete what I need to do, quickly** 1.  2.  3.  4.  5.  N/A

**6. I am satisfied with what this software can do.** 1.  2.  3.  4.  5.  N/A

**7. The software can do everything I need it to do.** 1.  2.  3.  4.  5.  N/A

**8. There are some tasks for which I cannot use this software** 1.  2.  3.  4.  5.  N/A

**9. Some examples of these tasks are:**

**10. When using this software to do something, the steps I need to take are logical.** 1.  2.  3.  4.  5.  N/A

**11. There are times when I do get lost while using this software.**

1.  2.  3.  4.  5.

N/A

**12. The software responds quickly to my inputs**

1.  2.  3.  4.  5.

N/A

**13. The logical flow of steps and options is very similar to my way of doing things**

1.  2.  3.  4.  5.

N/A

**14. This software is satisfying to use.**

1.  2.  3.  4.  5.

N/A

**15. This software could be improved in some areas**

1.  2.  3.  4.  5.

N/A

**16. I can suggest the following improvements**

University of Cape Town

## Section Two: How easy was this software to learn?

In this section we are interested in your learning experience for this software, the amount of help available to you, and your level of comfort and satisfaction.

Grading system: (please tick ONE of the following choices for each question)

(strongly disagree) 1.  2.  3.  4.  5.  (strongly agree) N/A

1. Learning to use this software took a very long time 1.  2.  3.  4.  5.  N/A

2. I am comfortable about using this software 1.  2.  3.  4.  5.  N/A

3. I am confident about using this software 1.  2.  3.  4.  5.  N/A

4. Using this software, I rate myself as :  
 a beginner  
 an intermediate user  
 an advanced user  
 an expert

5. There is a lot of help available to me for this software (documentation, tutorials, etc) 1.  2.  3.  4.  5.  N/A

6. This help material is simple to understand. 1.  2.  3.  4.  5.  N/A

7. I need another person to help me learn something new using this software 1.  2.  3.  4.  5.  N/A

8. I need another person to help me with the way in which I currently do things using this software 1.  2.  3.  4.  5.  N/A

9. I am completely comfortable with all the terminology I come across when using this programme. 1.  2.  3.  4.  5.  N/A

10. There are still some areas or levels of the software programme that I find difficult to understand 1.  2.  3.  4.  5.  N/A

**Section Three: Are you, the user, able to customise this software?**

In this section, we are interested in elements of the software interface, visible on screen, that you are able to change to suit your preferences (i.e. customise).

*Grading system: (please tick ONE of the following choices for each question)*

(strongly disagree) 1.  2.  3.  4.  5.  (strongly agree) N/A

**1. It is possible for me to change some aspects of this software's screen appearance, to suit my preferences. (eg: colors, layout, etc)**

1.  2.  3.  4.  5.  N/A

**2. Examples of aspects I am able to change, include:**

**3. I would like to have more freedom to choose my preferences for this software's appearance on screen.**

1.  2.  3.  4.  5.  N/A

**4. Examples of aspects I would like to be able to customise, include:**

University of Cape Town

#### Section Four: How does this software behave towards you?

In this section, we are interested in how the software treats you, the user! Please tell us how you feel about the general operation of the software, and the kind of information which may be provided by the software.

Grading system: (please tick ONE of the following choices for each question)

(strongly disagree) 1.  2.  3.  4.  5.  (strongly agree) N/A

1. This software often "crashes" for no apparent reason 1.  2.  3.  4.  5.  N/A
2. After a "crash", I can pick up where I left off without having to redo a large section of work 1.  2.  3.  4.  5.  N/A
3. The software detects when my actions will cause an error, and informs me of this. 1.  2.  3.  4.  5.  N/A
4. The software provides me with opportunities to change my choices and options, before saving these. 1.  2.  3.  4.  5.  N/A
5. When an error occurs, the software gives me adequate information on the kind of error being experienced. 1.  2.  3.  4.  5.  N/A
6. When an error occurs, the software gives me adequate information on possible ways in which to correct the error 1.  2.  3.  4.  5.  N/A
7. I can always "undo" my previous step, if necessary. 1.  2.  3.  4.  5.  N/A
8. My work is continuously saved by the software. 1.  2.  3.  4.  5.  N/A
9. Sometimes this software behaves in an unexpected way 1.  2.  3.  4.  5.  N/A
10. Sometimes this software gives me unexpected results 1.  2.  3.  4.  5.  N/A
11. On the whole, the behaviour of this software can be anticipated. 1.  2.  3.  4.  5.  N/A

## Section Five: Evaluating the visual appearance of this software

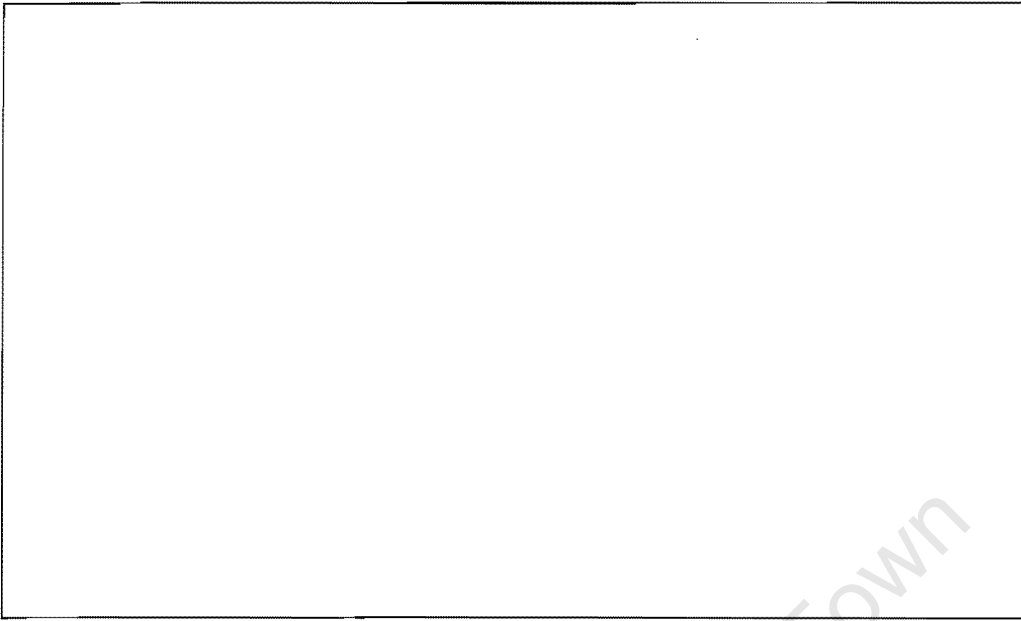
In this section, please tell us about how the software looks on screen, as well as what you like and what you hate about it!

*Grading system: (please tick ONE of the following choices for each question)*


(strongly disagree) 1.  2.  3.  4.  5.  (strongly agree) N/A

1. Key items (such as menus, exit button, etc) are located in the same place throughout the software. 1.  2.  3.  4.  5.  N/A
2. Colors are well chosen and are pleasant on the eye. 1.  2.  3.  4.  5.  N/A
3. The menus are logically organised (i.e. similar and related actions are grouped together) 1.  2.  3.  4.  5.  N/A
4. Items such as buttons and toolbars are placed in sensible, obvious locations. 1.  2.  3.  4.  5.  N/A
5. It is easy for me to find my way around this software. 1.  2.  3.  4.  5.  N/A
6. I always know exactly where I am in this software. 1.  2.  3.  4.  5.  N/A
7. The software makes use of symbols and icons that are easy to recognise. 1.  2.  3.  4.  5.  N/A
8. I like the "Look and Feel" of this software. 1.  2.  3.  4.  5.  N/A
9. There are aspects of this software that are confusing because they don't make sense. 1.  2.  3.  4.  5.  N/A
10. There are aspects of this software that are annoying because they are not useful and yet keep cropping up. 1.  2.  3.  4.  5.  N/A

**11. The 3 best features of this software are:**



**12. The 3 worst features of this software are:**



**THE END. Thank you for your time!**