CS310 Project Specification

IT in the contemporary Church

James Williams
Computer Science,
University of Warwick,
`J.Williams.6@warwick.ac.uk`

October 11, 2007

Abstract

This document specifies the details, such as the title, problems, methods and objectives, of James Williams' project for module CS310. Having been assessed for feasibility, the components of the project are split between core and extended parts. Also included is a proposed timetable and a discussion of the required resources and any issues such as ethics involved in the project. This specification is subject to constant review.

1   Introduction

This specification seeks to define the current understanding of the project, laying foundations for working through the components listed. The project is motivated by a desire to look into the problems that IT, or specifically software, could help solve in a church context. Traditionally, churches can be slow to adapt to new ideas, but increasingly many churches are embracing newer technologies. IT has integrated into other areas of life, so this project looks at how it is being used, or can be used, to help a church.

2   Project title

IT in the contemporary Church
This is the current title that best reflects the concept for the project. The starting point of the project is an investigation of how and why IT is used, or not used, in the contemporary Church. Problems may be uncovered that software could be created to solve. A piece of software will be created as an aid for leading songs in times of corporate worship.

3   Context and terminology

The Church refers to the wider collective group of Christians and their institutions, as opposed to *a church* which refers to a specific group of Christians. This project will focus on a small subset of the Church in Britain, by looking at a few different types of churches and other Christian institutions. This document shall hereafter include the other Christian institutions under the term *churches* for simplicity unless specified otherwise.

In many modern churches, particularly those with a young or less conservative membership, a *time of corporate worship* will involve a period of time where several songs are sung by a group of people (a *congregation*) in succession. A band usually accompanies the songs. The *lead worshipper* leads the singing, perhaps with an instrument such as a guitar, and chooses which songs or other activities are done

as part of the time of worship. However, other people may also start singing a song spontaneously for everyone to join in with.

Choosing a selection of songs (a *setlist*) to sing, is like building a playlist. Songs are chosen according to mood, theme, and musical suitability. A progression between themes is desirable throughout a setlist, while ensuring each song succeeds the previous one in a smooth manner. To choose a song, the lead worshipper must try to recall or search in a songbook for a song that best fulfils the criteria. There may be several songbooks, each containing 50-200 songs organised in different ways, often with an index. This project shall focus on a lead worshipper that only requires chord letters for a musical instrument (for example: 'A', 'Em7') beside the song lyrics, as this is the most common configuration. See figure 1 for an example of a song.

Figure 1: An example of a song from a songbook.

A *capo* is a tool that can be used to change the way a song sounds or is played on a guitar.

4 Identified problems

The following problems shall be addressed by the aims and objectives in Section 5. The project can be broken into two main components: An investigation into the use of IT in churches; and the creation of a program to aid in leading a time of worship.

4.1 Problems relating to the investigation

- What role(s) does IT currently play in churches?
  - What is the extent of the various applications of IT across different churches?
  - How well are the current applications of IT in churches implemented and used?
- What role(s) could IT potentially have in churches?
  - What are the issues facing the use of IT in the church setting?
- What problems exist in a church setting that could be solved using IT, or more specifically, software?
  - How could the problems found in the above question be solved using software?

Section 4.2 covers one such problem that software could be created to solve.

4.2 Problems relating to the software aid for leading times of worship

- The system of songbooks is often difficult to organise. Annotations and added songsheets make organisation even more difficult. Existing theme groupings and indexes are not sufficient.

- Often a song may be wanted in a different key to the one in the songbook. Transposing either involves more annotations, or mentally transposing while playing, but mistakes are easily made by either method. If using a guitar a capo greatly helps but this can still result in confusion.

- Others in a meeting might start a song, however they often start in an unrecognised key. The key must be known for instruments to accompany correctly, but it is difficult to ascertain which it is.

- Finding songs that fit desired criteria is difficult if there are many to choose from.

- Often only part of a song's lyrics or chord progression can be recalled, but the rest of the song needs to be known to sing it.

- There is usually no organised record of songs that have previously been played together. This information would be very helpful for recommending songs to play in a setlist.

- Communication between the lead worshipper and others in a meeting (for example, the operator of the overhead projector used for projecting lyrics for the congregation to see) can be physically difficult whilst singing and/or playing an instrument.

5 Aims and objectives

Again, these shall be split between the investigation and the software aid for leading times of worship. Aims (numbered below) here are general goals, whereas objectives (below the aims) are more specific and technical which build towards the aims.

5.1 Investigation

- Aims

  1. Investigate the extent of the use of IT in the Church and the attitudes towards it.
  2. Investigate how IT could have potential further use in the Church.

- Objectives

  - Conduct a literature search and review on the subject of IT in the Church.
  - Strategically survey people in a diverse range of churches that use IT, or those that do not when they could.
  - Design a software solution to at least one problem highlighted by the investigation.

5.2 Software aid for leading music

- Aims

  1. Develop a new song organisation system to replace the songbook system.
     Core objectives related to aim 1. :
     - Create an organised, well structured database system that can be efficiently searched. It must perform well enough to be used in the middle of a time of worship without causing unnecessary delays or distractions.
     - Develop a GUI that is at least as clear as the songbooks. HCI issues must be carefully addressed.
     - Build a function to recognise the theme(s) of a song automatically from the lyrics.

     Further objectives related to aim 1. :

– Enable song sheets to be printed.
– Include a means of storing musical parts to songs in *tab notation*, which could be manipulated and searched for in a similar manner to the system for chords.

2. Enable easy key transposition
   Core objectives related to aim 2. :
   – Ability to choose and display transposed chords and/or chords for a capo.
   Further objectives related to aim 2. :
   – Store and learn preferred key or capo configurations for songs.

3. Make a tool to recognise what key spontaneous singing may be in.
   Core objectives related to aim 3. :
   – Investigate as far as is possible how to recognise a key from a recording or live feed of sound from an instrument or human voice. This aim and its related objectives are not expected to be completed, but an understanding of the difficulties involved and some possible solutions are desired. Time and difficulty will be the constraints on how far this can be investigated.

4. Include a function to recommend songs for a setlist.
   Core objectives related to aim 4. :
   – Design and implement such a function.
   Further objectives related to aim 4. :
   – Ability to generate whole setlists, based on theme and musical suitability (e.g. a setlist that flows from one theme to another, without key changes).

- Additional Optional Aims:

  1. Add features uncovered by the investigation that would naturally form part of this program.
  2. Develop a means of sharing or distributing users' own songs, such as a web site.
  3. Enable some functionality for a handheld device such as a PDA or mobile phone.
  4. Design a tool using IT, having consulted with some musicians that play in a church band, to enable better communication amongst people involved in managing a time of worship.
  5. Expand the scope of this project to look at IT in other religious contexts, or find other applications for the software produced in the project.

6   Methods

For the investigation part of this project, I shall mostly be surveying with questionnaires and interviews.

The nature of the software part lends itself to an incremental development model, since the requirements should be fairly clear and the objectives are mostly separate deliverables that can be built on top of one another. Each increment will add functionality, so once the core objectives are completed any additional ones completed can be added. I also have regular contact with potential users of the system at church so any issues that arise during the process of creating the software can be dealt with quickly.

The following features shall be added with each increment:

1. Song organisation system

2. Theme recognition and search facility

3. Transposition and capo features

4. Setlist/song recommendation tool

Please refer to figure 2 for a diagrammatical representation of the tasks involved in producing the software part. The key recognition component can be worked on in parallel to the whole program. Remaining components not listed above can be added at any point after the first increment, but those listed above will be prioritised. New additions will then probably be added in subsequent increments.

Figure 2: The dependencies and relationships between the key features of the software part of the project. This shows what tasks can be done in parallel and in what order, with tasks at the top needing to be done first.
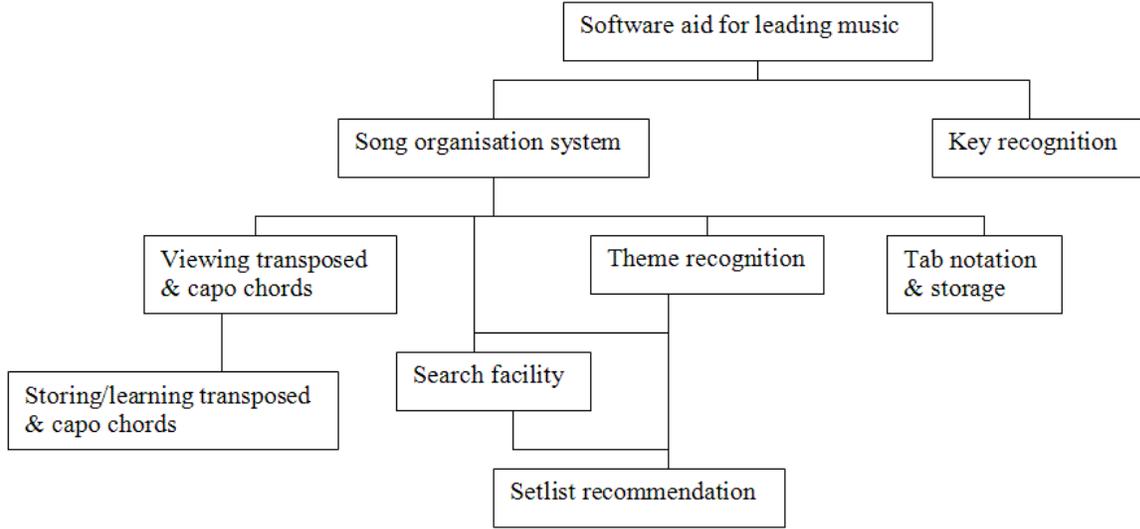
7 Timetable

At this early stage, the timetable can only serve as a rough guide. The literature search and review as well as the key recognition component will continue in parallel throughout the progress of the project. Documentation that may form the basis of the final report will also be written throughout the year. Space has been left to allow for delays or unforeseen circumstances. Please refer to table 1.

Table 1: Provisional timetable

Dates	Task
Term 1, Week 2	**Submit Specification**
Week 3	**Begin reviewing literature**; Create project web site
Before week 6	Increment 1: Requirements analysis and design
Monday, week 7	Other coursework deadline
Before week 9	Begin **key recognition** component; begin building and testing of first increment
Monday, week 9	**Submit progress report**
Before end of term 1	**Complete first increment**
Christmas holidays	Increment 2: Requirements and design; begin building and testing
First weeks of term 2	**Complete increments 2 and 3**
Before week 7	**Complete increment 4**
Week 9 or 10	**Deliver presentation**
End of Term 2	Other coursework deadline
Easter holidays	Complete project report
Term 3, Thursday, week 2	**Submit final report**

8 Resources

Most of my work shall be done on my own Windows based computer. It is absolutely critical that backup equipment is available and ready at any time in case of my own computer failing or becoming unusable.

Depending on what language I choose to program in, I can use either the ITS machines across campus that have Visual Basic (amongst other tools) available from the 'Novell Application Launcher', or the Linux computers in the Department of Computer Science. For either option I will ensure that I use compatible versions of software. I shall check every few weeks that the backup computers are still available and ready.

Several levels of backup will be made regularly. I shall save my work after at least every half an hour, and use SVN to store all working files. Updates to the repository will be made after every session of work. After each week, or sooner if major progress has been made, physical backups shall be made to CD(s) and removable media.

9 Other Issues

9.1 Legal issues

The author of any code or algorithms used must be properly acknowledged, and only used if permission is given.

Copyright must not be breached. So copyrighted songs shall not be used when distributing the software system. Songs unprotected by copyright or those in the public domain (e.g. 'I Stand Amazed', see figure 1) will be used when presenting or distributing the software. All the songsheets and songbooks (including the digital versions) that I will use for testing are legally owned by myself.

9.2 Social issues

Some people may be opposed to the use of technology in a church, so due care must be taken. Sensitivity must be shown when surveying users, or those that do not use, technology in church. The human element of religion must not be interfered with.

9.3 Professional issues

As a member of a local church myself, I must conduct this project in a professional manner rather than just as a personal interest. My interest will however help me to stay motivated, but issues must be investigated objectively.

9.4 Ethical issues

When conducting surveys, the University of Warwick's ethical standards must be adhered to. They can be found at `http://www2.warwick.ac.uk/services/rss/services/ethics/statement/guidance/`.

Because religion and technology can be a sensitive subject, objectiveness and sensitivity must be shown in equal measure. This applies to both those involved in the technology side (e.g. those supervising or helping on technical issues), and the religious side (e.g. lead worshippers, those being surveyed).

It is imperative that the technology **aids** the activities of a church, and does not replace the tasks that only a human should do. For example, the final choice of songs in a time of worship must rest with the lead worshipper, not a computer program. The program can however help in identifying possible songs quickly. Users must not come to rely on IT systems for worship-related activities, because otherwise the personal experience central to worship could be invalidated.