

CS310 Project: Progress Report

IT in the contemporary Church

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

November 26, 2007

Abstract

This is a formal progress report for James Williams' project for module CS310. Progress is assessed referring to the project specification. The development and current state of both the research and software development components to the project are critically reviewed and analysed. This includes analysis of key decisions and design work. The plans for the project's continuation are laid out.

1 Introduction

This report primarily seeks to evaluate the management of the project, but will also include analysis of the technical developments, over the six weeks that have passed since the project specification was submitted.

Please refer to the timetable set out in the specification. This report shall look at the extent to which the objectives scheduled have been accomplished and how much progress has been made in each area. In simple terms, the plan has been adhered to, but the following sections shall analyse just how true this broad statement is.

2 Development of the project concept and management

The project specification summarizes the project concept as follows: "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." This remains the foundation of the project. The program, now known as PSALM - Personal Software Aid for Leading Music, and investigation form the two halves of the project.

The focus of the software development half of the project has changed over time somewhat though. Originally, key recognition was to be an important part. But it became clear that this would be a very difficult challenge beyond the level of a third year project such as this. My thanks go to Roland Wilson, Roger Packwood and Steve Russ for their advice on this issue.

Instead, part of the project shall be to investigate what current solutions exist for the key recognition problem, and what the issues are in making this problem so difficult. Instead, theme recognition and setlist recommendation have become the most important technical challenges.

Theme recognition involves the automatic detection of themes found in lyrics, based on lyrical content of a song. Songs could be clustered in themes depending on keywords set as certain themes and common words.

Setlist recommendation would be based on recommending songs for a playlist according to a mix of musical suitability (e.g. key), history (e.g. songs that have been played together in a set previously) and theme.

It has proved to be difficult to manage the timing of work on PSALM. Although the first increment was scheduled to be designed before week 7, design and implementation have actually progressed alongside each other, with the development of a prototype. Although this has meant some of the design behind the prototype has not been as well planned, using a prototype allowed different designs to be actually tested. This model may be followed for further increments, subject to evaluation once the first increment is completed. Design is an important phase so must not be merged with implementation too much, or the quality of designs may be compromised.

3 Chronology and decision analysis

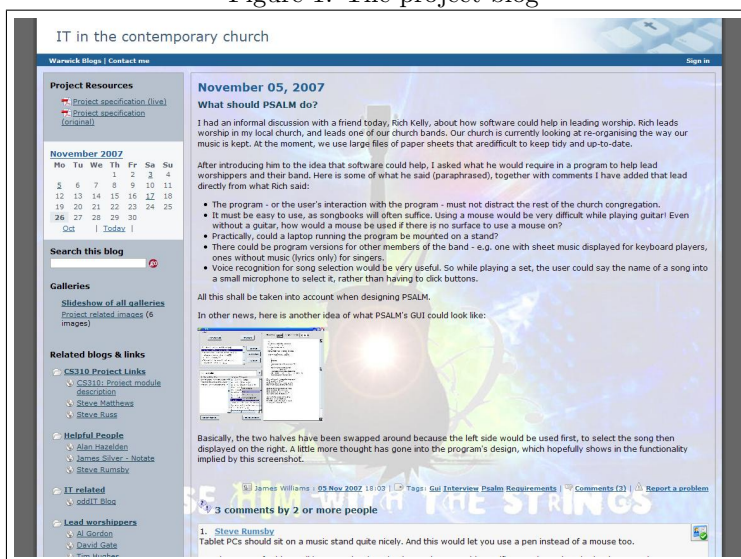
3.1 Week 2:

- The project specification was submitted.

3.2 Week 3:

- A project blog was created at <http://blogs.warwick.ac.uk/church-it/>. See figure 1. This blog has allowed interaction with other people that are interested in the project, which has provided some helpful feedback and good ideas. It has been used to distribute project resources (documentation, software) and track progress. It has also been used as a resource that can be accessed from anywhere with an internet connection, should I need to need to store private notes or files on a computer.
- A search and review of online literature was begun. See the bibliography at the end of this report for more information.

Figure 1: The project blog



3.3 Week 4:

- Four books were ordered: *Beyond the OHP*, Jackie Sheppard; *Wired for Ministry*, John P Jewell; *Help, There's a Computer in My Church!*, David Trammel; and *High-tech Worship?*, Quentin J. Schultze. See the bibliography for more information.
- A SVN repository server was set up to store and manage project files, hosted at <http://www.draknek.org/>, courtesy of Alan Hazelden.

- The name PSALM was given to the software to be made as part of the project. An icon (see below) was also made, based on the lyre referred to in the Psalms in the Bible.



- Java was confirmed as the language to be used for PSALM development, together with the NetBeans development environment for GUI creation. Visual Basic had been proposed as an alternative, but the backup machine could not be used. Java, used in DCS, has the advantage of being cross-platform, which will be useful because my church, which is a potential user of my software, may want to run it on a Mac computer.
- MySQL was chosen as the database management software (DBMS) to be used.

3.4 Week 5:

- MySQL was dropped as the DBMS. Installation and configuration of MySQL was more complicated than expected, and security issues meant that it became unusable. It had previously been assumed that SQL and a DBMS would be necessary for the database used by PSALM.

However, XML was chosen instead, without the use of a DBMS. The assumption that SQL would be best was made because of its popularity and my knowledge of SQL. But PSALM needs to be cross-platform and suitable for novice users. The installation and configuration of a DBMS or SQL server would not be possible for most potential users of PSALM.

The structure of the song organisation system naturally lends itself to a hierarchical database representation, so XML is a good candidate. The plain text storage method of XML would also allow easier integration with other programs currently used in the church context, such as SongPro.

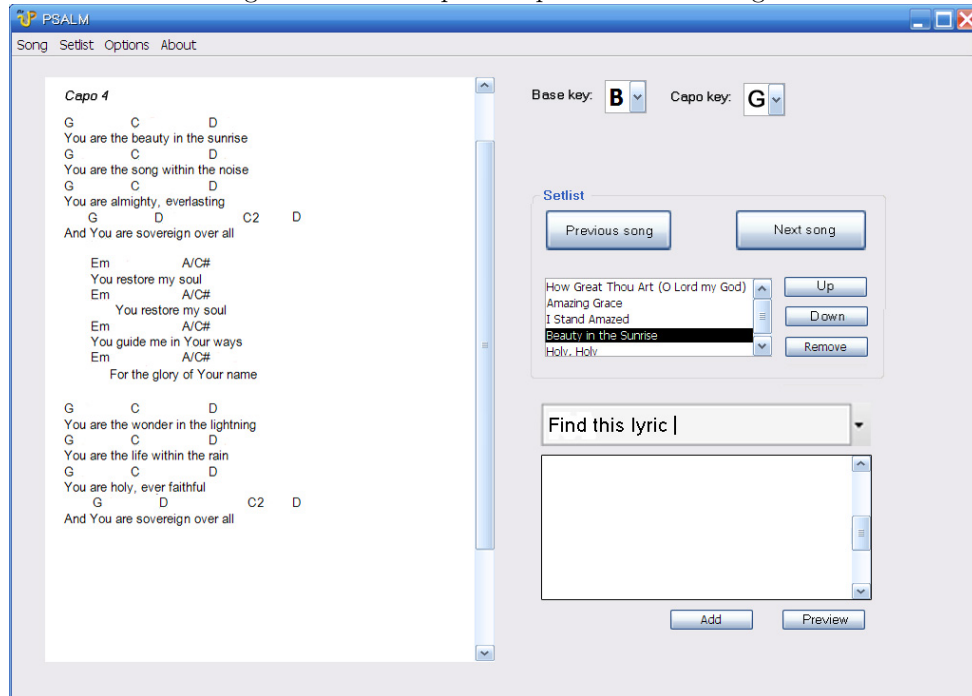
It is clear that assumptions like this about the best methods should not be made, but rather informed decisions should be made.

- Some existing programs for churches were investigated. While some of the programs found at this time would share similarities with PSALM, none seek to make setlist recommendations. This is now clearly the most important part of PSALM.
- Requirements for the first increment of PSALM are set.

3.5 Week 6:

- Some preliminary designs for the GUI done. See figure 2 for an example. A layout with the setlist and search items on the left side and the song on the right was preferred to this example because it follows a more logical structure for the way the program would be used. The project blog has alternative designs. However, there has not been much work looking into the HCI issues with this layout yet. It does seem rather cluttered and may be difficult to use when operated from a stage where a mouse cannot be used easily.
- An informal interview with Richard Kelly, a lead worshipper at Jubilee Church was conducted. The possible requirements and practical problems for a program like PSALM were discussed.

Figure 2: An example of a potential GUI design



3.6 Week 7:

- Work done on designing PSALM. As a summary, see figure 4 represents the modules involved in PSALM. This diagram is not in any standard notation, which is not ideal. Future designs will be made in UML.
- The format of the XML song database was designed. It will be used for increment one, but some changes will be necessary. Fields like writer/author are currently attributes that can only take one value, but they should probably be independent tags that may take multiple values. An example is attached as an appendix.

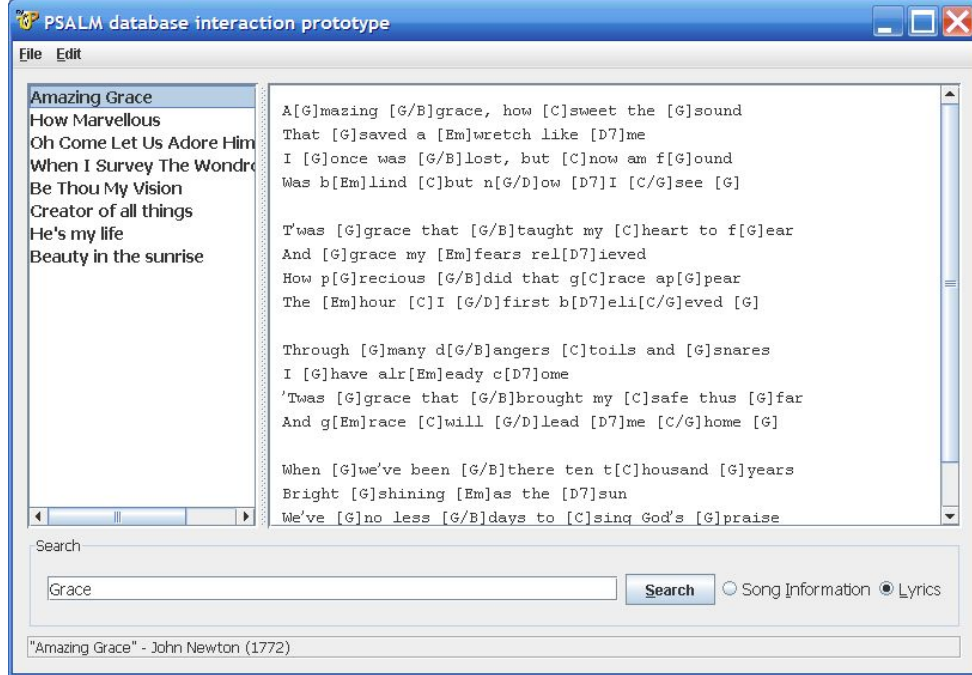
3.7 Week 8:

- Development work on the prototype and first increment of PSALM in progress (See figure 3, and section 3.9).
- Testing of PSALM on different platforms begun.
- This progress report written.

3.8 Changes to the original specification

- One of the core aims for the software in this project was to 'Make a tool to recognise what key spontaneous singing may be in', this has been changed to 'Investigate solutions that use IT to recognise what key spontaneous singing may be in' to reflect the change in focus.
- In the resources section, the backup options to be used were dependant on which programming language would be used. It has been decided that Java will be used, so the DCS Linux machines can be used as backup, which is now reflected in the updated specification.

Figure 3: The prototype for PSALM, built to establish basic database and GUI functionality.

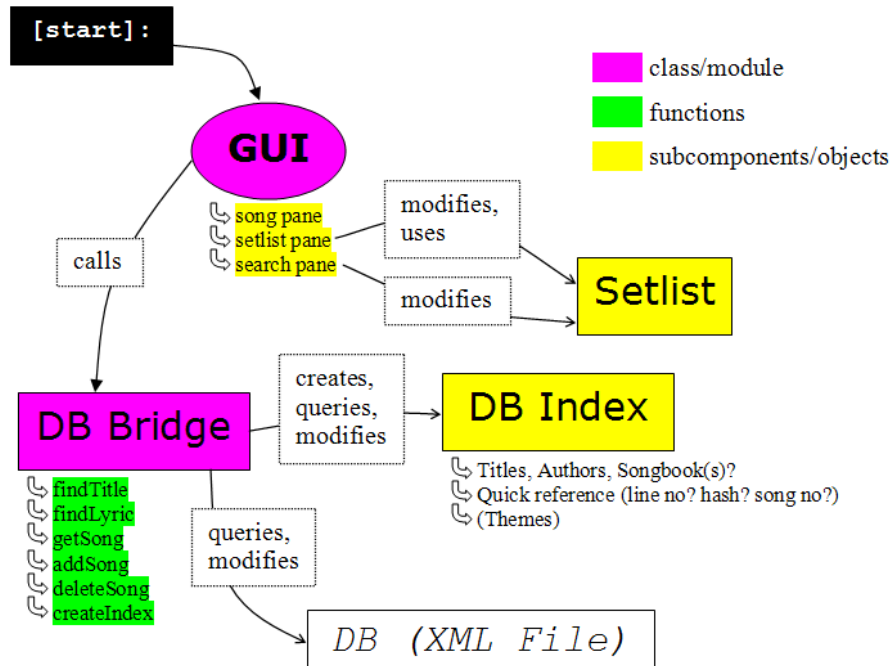


- As explained above, the database will be kept in XML format. This will not require any database management software so does not need to be included as a resource. When MySQL/JDBC was proposed for use, it was mentioned in an interim version of the specification, but this has now been removed. The use of NetBeans is now referred to in this section too.
- The methods section now has an expanded description of how the investigation shall be conducted. Questionnaires and surveys were never intended as the sole means of the investigation. Now the literature search and investigation into existing software solutions are specified as part of the investigation.

Although not part of the project specification, separate requirements analysis documentation will be drawn up for each increment of the PSALM program, that will be used as basic specifications for the respective increments. The requirements for the first increment are:

- Store songs as lyrics together with chords.
 - The database should be capable of holding up to at least 300 songs.
 - The database must allow songs to be added, modified and retrieved for display.
 - The database should be searchable on song titles and lyrics.
 - Searches should not last more than a few seconds.
 - Chords must be marked specially to enable transposition functionality in future.
- Display songs (retrieved from the database) as lyrics together with chords
 - Chords should be displayed in the correct place as in a songbook. Chord letters may be placed above a word, above a gap before a word or above a gap after a word.
 - Text must be at least as clear and readable as the text in a songbook.

Figure 4: The components of PSALM



3.9 Current state of the project

On the software side, the analysis and design scheduled have been completed. A Java prototype with basic database functionality in XML including the ability to search lyrics has been created. This has been implemented to match the design laid out in figure 4. This design seems to be appropriate, although no work has been done with integrating setlists yet. A history of setlists has been created. The first increment is on course to be completed by the end of term, as scheduled.

The search functionality of the prototype is currently based on regular expressions, which will need replacing as this does not work when chord tags are in the middle of a word. The parsing of the XML database file is done using the `StreamTokenizer` from Java's class library. The `Bridge.java` class has been created to parse the XML as required by PSALM rather than use a general purpose XML parser. The standard Java library for interacting with XML is currently incomplete so unsuitable. The song index is implemented as a `LinkedList` containing `Song` objects. The `Song` objects currently contain four fields: `title`, `writer`, `year` and `content`; together with methods to get and set the values for each field.

Some research has been done for key recognition on what software and algorithms exist, but this has become less of a priority as work on the core of PSALM has proceeded.

The progress made on the investigation is unsatisfactory. Many online sources have been reviewed, the best of which is included in the bibliography, but the books that have been acquired are yet to be fully reviewed. More importantly though is the lack of work done on the questionnaires. The results from the questionnaires should feed directly back into the work being done on PSALM, and needs to be done with plenty of time for responses to be received. But aside from some ideas for questions and topics to cover, questionnaires are behind schedule. This shall be a priority for the remaining weeks of term.

3.10 Project continuation

Increment 2 shall focus on implementing theme recognition and debugging the search facility, while also completing the database functionality and further developing the GUI.

The third increment will include a review of the GUI using HCI principles, key transposition functionality and allow setlist editing.

The final increment will build setlist recommendation ability into PSALM.

See the timetable laid out in table 1. There is some space towards the end left deliberately to allow some time for unforeseen delays.

Table 1: Provisional timetable

<i>Dates</i>	<i>Task</i>
Term 1, Monday, week 9 Before end of term 1	Submit progress report Complete first increment; Design and distribute questionnaires
Between the end of term and Christmas	Increment 2: Requirements and design; begin building and testing; prepare interviews
First week of January	Complete increment 2; complete review of the books already obtained
Term 2, week 3	Complete increment 3
By end of 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

4 Conclusions

The goals laid out in the specification are still a way from being complete, but progress is being made in accomplishing them. The two halves of this project need evaluating separately but in the context of each other. The software component is progressing satisfactorily. The planned analysis and design has been completed, and the first increment should be ready by the end of term.

However the investigation component is behind schedule. Questionnaires must be designed and distributed and the literature review should be underway with the books already obtained reviewed. But with all other coursework for the term finished and these clear goals set for completion by Christmas, the investigation can be put back on track. It was expected that some delays would occur. The investigation will be made a higher priority from this point because its results may affect the progress of the software development component.

While it may be on track, PSALM development has not been as efficient as hoped. By working through the Christmas holiday on the second increment (focusing on theme recognition), with less other work running in parallel, this may be improved. Time was wasted on configuring MySQL. This demonstrates the importance of considering decisions properly rather than making assumptions.

5 Bibliography with brief notes

5.1 Books

J. Sheppard. (2002) *Beyond the OHP: A practical guide to using technology in worship*. Carlisle, UK. Spring Harvest Publishing and Authentic Lifestyle.

Makes the case for the use of technology (including IT) in worship, with case studies on specific issues. Beginners guide to what and how technology can be used in a church. Spring Harvest is a leading resource for British churches. The Spring Harvest festivals are one of the most influential institutions in the British Church today.

J. P. Jewell. (2004) *Wired for Ministry: How the Internet, Visual Media, and Other New Technologies Can Serve Your Church*. Grand Rapids, MI, USA. Brazos Press.

Evaluates technologies for use in a church. Helpfully, the first chapter is 'Be aware of the pitfalls'. This book is more objective as to whether technology should be used. Jewell has written a few books

on the subject of IT in the contemporary church. He is director of instructional technology and distance learning at the University of Dubuque Theological Seminary.

Q. Schultze. (2004) *High-Tech Worship? Using Presentational Technologies Wisely*. 2nd ed, 2005. Grand Rapids, MI, USA. Brazos Press.

The book claims to help the reader to ‘Learn when to adopt, reject or adapt technologies to help worshippers encounter God’. This is another objective look at whether technology should be used, and how. Includes ‘A Snapshot of Technology in Churches’, taken from a survey which can be found at <http://www.calvin.edu/worship/>, reported in Steve Koster, *Visual Media Technology in Christian Worship* (masters thesis, Michigan State university, East Lansing, 2003).

D. Trammel. (1990) *Help, There’s A Computer In My Church: Practical Advice for Using Computers in Ministry*. Nashville, TN, USA. Broadman Press.

A much older look at whether technology should be used in a church. Rather outdated but interesting as an early attempt at persuading churches to use technology.

5.2 Other published material

Booklet: Methodist ICT Project Group, (2005). *Using Technology in Worship and Mission*. London, UK. The Methodist Church. Available from http://www.methodist.org.uk/downloads/ca_technology_1005.pdf [Last accessed 25/11/07]

A guide to using technology written by one of the largest church groups in the UK. Makes recommendations for Methodist churches (that can be applied to other churches too).

5.3 Online material

Articles from *Christian Century*, posted at <http://www.religion-online.org> [Last accessed 25/11/07]:

Nancy S. Armstrong, Aaron Spiegel & John Wimmer. (2001). *Information Technology in Congregations*.

Brilliant summary of sectors where IT is used in church, except it is a little out of date.

Raymond B. Williams. (2001). *Information Technology in Seminaries*.

Evaluating the use of IT in Bible colleges (Seminaries). Lists possible advantages and disadvantages.

D. Lochhead. (1988). *A Software World*. URL: <http://www.religion-research.org/irtc/software.htm> [Last accessed 25/11/07]

David Lochhead, founder of the Institute for Religion, Technology and Culture and author of *Theology in a Digital World*, has written several articles and books on this subject. This article investigates the church attitude to computers, or their equivalents before personal computing, throughout history.

Debra Dean Murphy. (2006). *PowerPointless: Video screens in worship*. URL: <http://www.christiancentury.org/article.lasso?id=2215> [Last accessed 25/11/07]

Highlights the disadvantages of using presentations in worship. Presents a very powerful case.

Pew internet & American Life project. (2004). *Faith Online*. URL: http://www.pewinternet.org/pdfs/PIP_Faith_Online_2004.pdf [Last accessed 25/11/07]

Results of a survey, summarising how the internet is used for religious purposes on and around page ten.

Please note that only the most useful sources have been included in this bibliography.