A software application to enable the assessment of inhaler technique through sound recognition

Software Development Plan

Version 1.3

Kieron Allsop 6/7/15

Problem specification

Vitalograph Ireland have been exploring the possibility of using sound data to assess a patient's inhaler technique. This is facilitated by recording the sound made during inhaler use by a small module attached to the inhaler. The technique currently employed uses two proprietary pieces of software, SpeechSpectra and QASR, which require a knowledgeable operator to use due to the complexity of set up. SpeechSpectra uses a series of steps to calculate Mel Frequency Cepstral Coefficients (MFCCs) and both Delta and Delta^2 features from a wave file input. A features file is generated which is passed to QASR which uses Hidden Markov Models (HMMs) to determine if the 'words' match the trained vocabulary with a confidence score being presented to the user.

As research [1] in this area has proven successful, Vitalograph Ireland wish to move to a clinical trials stage, but with a new application which will allow their Trials Team to assess patients with minimal set up. Ultimately the software will be used by Doctors to critically assess their patients' inhaler technique with the goal of improving their care and reducing hospital admissions.

Background

Both SpeechSpectra and QASR were written in Borland C++ Studio. They are Windows applications with a large number of input settings. The goal of this project is to develop a Windows application which will allow the user to make use of both programs without the need to input all the settings. Previously a student at QUB [2] removed the GUI from both SpeechSpectra and QASR in Microsoft Visual Studio, the source files of which have been supplied. Vitalograph Ireland have supplied both training data and test data for use in the project.

Proposed Solution

Methodology

An agile methodology has been adopted for this project. There shall be two week iterations to achieve realistic milestones. Within these iterations there will be a planning stage before work begins and brief retrospective before the next iteration begins. Story points have been assigned to all User Stories to help with estimation of work. The volume of work achievable during an iteration will become more accurate as the project progresses.

The following User Stories have been written, with more detailed Uses Cases being written before each iteration:

#	User Story	Story Points
2	As a RegisteredUser I can login to the system	3
9	As a DataTechnician or a DiagnosingDoctor I can select files for recognition	5
10	As a DataTechnician or a DiagnosingDoctor I can submit files for recognition	13
1	As an UnregisteredUser I can create a new account	8
3	As a RegisteredUser I can update my login credentials if required	3
6	As a DataTechnician or a DiagnosingDoctor I can register a new patient	8
7	As a DataTechnician or a DiagnosingDoctor I can retrieve a patient record	3
8	As a DataTechnician or a DiagnosingDoctor I can amend a patient record	3
15	As a DataTechnician or a DiagnosingDoctor I can change a patients inhaler type	
11	As a DataTechnician or a DiagnosingDoctor I can view result set	2
5	As a SystemAdministrator I can delete redundant patient data	2
12	As a DataTechnician or a DiagnosingDoctor I can retrieve individual .wav files	5
14	As a DataTechnician or a DiagnosingDoctor I can view feature data for a .wav file	5
13	As a DataTechnician or a DiagnosingDoctor I can play individual .wav files	5
16	As a DataTechnician or a DiagnosingDoctor I can manually recognise features and over ride the system	13
17	As a DataTechnician or a DiagnosingDoctor I can decide what data is added to the training set	8
20	As a ProjectSupervisor I can view a project plan	2
21	As a ProjectSupervisor I can review a dissertation describing the product	40

There may be more User Stories written around training data and the re-recognising of wave files when it becomes clearer what exactly is of required in these two areas. User stories which detail the look and feel of the system may also emerge.

An example of a Use Case:

Use Case	**A RegisteredUser should be able to login**
User Story	#2 (As a RegisteredUser I can login to the system)
Goal in context	Registered users can log in to the system
Scope	System
Level	User-goal
Primary Actor	Registered users
Stakeholders & Interests	Technicians, Doctors, System Administrators
Precondition	User is already registered
Minimal Guarantees	User gets a response indicating success or failure
Success Guarantees	User is logged in
Trigger	User needs to login

Main Success Scenario

- **1.** User enters username.
- **2.** User enters password.
- **3.** User submits credentials.
- **4.** User is logging into landing page/screen.
    - **4a.** Technicians and Doctors are directed to the main patient administraion page
    - **4b.** System Administrators are directed to the system maintenance page.

Extensions

- **1a.** User cannot remember username (→ username retrieval / register new account?)
- **2a.** User cannot remember password → password retrieval.
- **3a.** User has not completed username or password so is not allowed to submit
- **4a.** Username and/or password is not recognised.
    - **4a1.** User is prompted to try again. Return to 1.

Technology and Data Variations List : n/a

Comments : n/a

Version Control

Version control is vitaly important in any software development project, therefore a private git repository has been set up. Not only will git be used as a repository for code, but also for all files associated with the project. This will be managed using a combination of Git Cola, git from the command line and the Github website.

Timetable

Github is being used as an agile management tool. All User Stories and their associated Use Cases have been created as 'Issues' which means that all work for a particular story can be tracked. Milestones and iterations have been created to allow work to be organised into manageable chunks. The aim is to develop vertical slices of the project so that at the end of each iteration there is working software which can be demonstrated and built upon during the following iteration.

Milestone	Start	Earliest	Latest	Story numbers
1. User can import and process inhaler sound data	6/7/15	22/7/15	28/7/15	#1, #2, #3, #9. #10
2. User can associate inhaler sound data with a patient	24/7/15	3/8/15	7/8/15	#5, #6, #7, #8, #11, #15
3' Users can register and maintain their accounts	5/8/15	23/8/15	31/8/15	#12, #13, #14, #16, #17
4. Deliver a market ready product and dissertation	26/8/15	16/9/15	23/9/15	#20, #21

Milestones will be delivered over the course of the following iterations

Interation	Dates	Story numbers
1	4th Jul – 18th Jul	#20, #2, #9, #10
2	19th Jul – 1st Aug	TBC
3	2nd Aug – 15th Aug	TBC
4	16th Aug – 29th Aug	TBC
5	30th Aug – 12th Sep	TBC
6	13th Sep – 23rd Sep	TBC

Testing Strategy

Test cases shall be written for each component as they are being developed.

Potential Problems

One of the main problem areas could be around the ability to choose which parts of a wave file to play based on the data obtained from feature extraction. Another issue could be my lack of experience using C++.

Language & Software

The first and most important area to be addressed is choice of language and environment. After some debate C++ has been chosen as the development language, due mainly to both SpeechSpectra and QASR already being written in it, albeit in Visual Studio. The next big consideration was choice of

IDE/tool to develop the application in. My preferred choice for C++ development is a Linux environment using Qt5 as both an IDE and GUI creation tool. Therefore some time was spent ensuring that both SpeechSpectra and QASR would build, and run from the command line Kubuntu.

The next step was to decide upon which database would be used. My experience thus far has been using MySQL in Microsoft Windows, however, whilst a good teaching database it is seldom used in a production environment. The most widely used open-source database in industry is PostgreSQL, so it will be used in the project.

The other main software that will be used include:

- SCons – an open source software construction tool—that is, a next-generation build tool. It is an improved substitute for the classic Make utility. Allows the use of two compilers at build time which can be very helpful in debugging; they may show different errors or warnings.
- cuppa – an extensible build system for use with SCons which enables the user to always use the most recent version of certain specified Libraries (Boost, Quince, Qt5, Asio and spdlog), it achieves this by checking for the latest versions at build time, then only *makes* the specific dependencies stated in the code, therefore greatly reducing overall setup time. It also means that even though development will be carried out in Qt5 that SCons can still be used as a build tool.
- gcc and clang – C++ compilers.

There may also be occasional use of other pieces of software, such as:

- Audacity – manipulation of sound files.
- LibreOffice – to write up the dissertation.
- PgAdminIII – to allow initial setup of database username and password. Once this has been carried out, Quince, which is an open source object relational mapping (ORM) library, enables the creation and population of the PostgreSQL database schema from within C++ code.
- gedit/Kwrite – as text editors.
- meld – code comparison application to lllow two versions of the same file to viewed side by side with all changes highlighted.

References

1. PMG 5 Graduate Presentation – Terence Taylor 2015
2. Automatic Recognition of Coughs Using Keyword Spotting with Hidden Markov Models – Fiachra Murray 12/05/2105