

SOFTWARE ENGINEERING PROCESSES AND MANAGEMENT



MEET THE CASE STUDIES

INTRODUCTION TO THE CASE STUDIES

The aim of the case-studies is to provide concrete, real-world examples of how software engineering principles can be applied. That is to say that the aim of studying cases is to help you develop the *practice of software engineering* by studying what others have done.

Throughout this subject we will be using two case studies. The first is a Virtual Reality (VR) simulator for training surgeons to perform temporal bone and general mastoid surgery. This will largely be used for assessment items throughout the semester. The second is a web-based framework that allows users to share language data, language resources and application modules across the web. This will be used in lectures and workshops to demonstrate concepts.

These two cases studies represent two different kinds of systems, each requiring different approaches, tools and techniques. The point of software engineering is to *design* processes that build in the most appropriate approaches, tools and techniques so that each type of project will meet its goals, and in turn be a success.

The virtual reality simulation requires a software system that will integrate a number of devices as well as simulate a real world task. It combines computer graphics and embedded systems and must:

- be reliable¹;
- meet strict performance criteria²; and
- be usable.

The types of processes, tools and techniques needed here are those pertinent to measuring and evaluating software reliability and performance³ and those pertinent to assessing the usability of the Human Machine Interface (HMI).

On the other hand the web framework has different characteristics for example, it must:

- be extensible;
- be usable; and
- maintain the integrity of its data across different formats.

The types of processes, tools and techniques needed here are those pertinent to evaluating distributed systems, networks and network latencies. Note that there is also a need to choose processes, tools and techniques pertinent to the usability of the interface, but it is a different interface to the one for the VR simulation. The reason is that the type of tasks being undertaken are more centred around the interchange of knowledge and applications while the VR simulations are centred around enacting a real-world task.

Even when two characteristics are apparently similar we still find that we require different approaches or different tools and techniques.

¹We will encounter the term *reliability* quite frequently, however, in software engineering it means something specific — it is the probability of the system *not* failing for a given time.

²In this project, the term *performance* is taken to mean the actual execution time of the system running on a machine. We must distinguish carefully between terms like *performance*, *efficiency* (which is a more general term), and *algorithmic complexity*.

³We will meet these types of processes, tools and techniques in the subject *Software Engineering Methods*.

CASE STUDY 1 — VIRTUAL TEMPORAL BONE SURGERY

INTRODUCTION

The problem in the first case study is to develop a simulator for temporal bone surgery. Temporal bones are found at the side of the head, as shown in Figure 1.

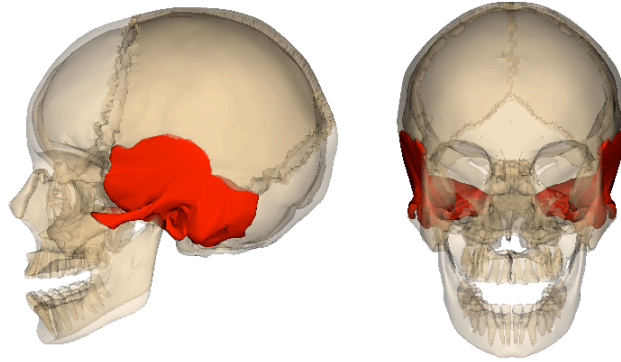


Figure 1: An image of the skull with the temporal bone shaded.

There is a drive in surgical training towards giving trainee surgeons more practice at standard procedures and also to expose trainees to a more diverse caseload that gives them some experience of non-standard procedures. The traditional apprenticeship model of training surgeons has been criticised for not delivering the kinds of in-depth focussed training nor the diversity in the operative caseload that helps trainee surgeons become more competent at their craft.

There are a number of methods for giving trainee surgeons the extra practice that they need, but few for giving them the *diversity* that they need. One of the most promising methods for providing that with diverse experience is *virtual reality* (VR) computer-based simulations.

The problem in the first case study is the development of exactly such a virtual reality simulator. The simulator will be for:

1. the development of surgical skills in drilling the temporal bone; and
2. learning surgical decision making by being exposed to different cases of temporal bone surgery.

The VR simulator will require several 3-D visual models of the temporal bone, especially the internal anatomy of the temporal bone including the facial nerve, the sigmoid sinus, the porous bone of the temporal bone, the major aircells in the temporal bone, the incus, and the middle ear. It will also require a force model for the amount of force required to drill bone.

All this makes the VR simulation project interesting for a number of reasons.

1. The software must be able to reproduce a real-world task — the simulation — but how close can we get? The point here is that we will need to do some non-standard systems analysis to get to a system that will be *fit for purpose*.

2. The software will need to make several new and interesting pieces of hardware work together *reliably* and according to strict *performance constraints* that will make this software system usable.
3. The final system will be used by trainee surgeons, not computing experts. It must, therefore, be intuitive to use and easy to learn.
4. It will require some interesting algorithms to compute the forces, changes to model and responses of the system.

THE PROBLEM STATEMENT

The problem of training otolaryngology surgeons to be safe and competent in the insertion of cochlea implants is becoming critical because of the shortage of quality cadavers on which to practice.

The problem affects the training of surgeons, the quality of care provided to patients, and the medical indemnifiers, because of the risk of poorly trained surgeons undertake procedures. It also affects the manufacturers of implants because of the perceived risk among the population.

A computer aided virtual surgery environment would benefit not only the public by having surgeons who are more practised with the procedure, but also the companies selling implants, indemnifiers and surgeons by lowering risk.

KEY FEATURES OF THE SOLUTION: A VR TEMPORAL BONE SURGERY SIMULATOR

The aim of the project is to build a virtual reality haptic training simulation for temporal bone and general mastoid surgery. Note the word *haptic* relates to the sense of touch and the perception of manipulating objects through the sense of touch. The sense of touch is delivered through haptic input/output devices such as those in Figure 2. When a trainee touches a piece of bone, the haptic device applies resistance against the trainee's hand to simulate touching real bone.



Figure 2: Force input/output devices by Sensable technologies.

Assume for now that the key features of the solution that will make it *fit for purpose* are the following.

1. A *load* feature that allows the surgeon to load on of several temporal bone models to work on.
2. A *3D stereoscopic rendering* feature where the model can be viewed in 3D.
3. A *3D viewing feature* that allows the user to rotate the model about various axes, view the model from different viewpoints, to zoom in and out, and to view the model as a surgeon would through a microscope
4. A *select surgical procedure* feature that does the following:
 - (a) allows the user to select a specific type of surgical procedure, such as cochlea implant, or mastoidectomy;
 - (b) then to provide the user with an interface that shows at least the surgical tools, a model, and a set of parameters for the procedure; and
 - (c) allows the user to configure the haptic IO device for the procedure.
5. A *perform surgical procedure* feature that allows a user of the system to simulate a surgical procedure by:
 - (a) interacting with the model via the haptic device and other IO devices; and
 - (b) having the model updated to reflect the impact of the user's actions.
6. A *save and restore* feature that allows to the surgeon to save the current state of the simulation and restore it late.
7. A *replay and pause* feature that allows the surgeon to replay their interaction with the simulation and to see and feel all of their actions as well as pausing and rewinding the simulation.

AN INITIAL LIST OF STAKEHOLDERS

The initial group of stakeholders represent the collection of people and organisations relevant to the project at the project concept phase — the phase of the project at the very start before any real development activity has taken place. The stakeholder analysis may be expanded to take in other stakeholders as we uncover more about the project.

1. *The clients* are two otolaryngology surgeons from the Royal Victorian Eye and Ear hospital (RVEEH) engaged in both surgical practice and training of surgeons in temporal bone and mastoid surgery.

It is expected that the surgeons will be able to provide more comprehensive training for their trainee surgeons overcoming some of the problems in today's training methods. It is also expected that the two surgeons will enhance their own standing by undertaking cutting edge projects.
2. *The development team* is a group of 12 relatively inexperienced software engineers. The team has experience in Java programming, but not an extensive experience in the Java environment, for example, they have only a limited knowledge of the Java libraries for image processing and graphics (perhaps you may want to think of them as 4th year software engineering students).
3. A *management team* consisting of two experienced people from the developer's organisation (perhaps a good way to think of these two people is as the supervisors and coordinators from the University of Melbourne).

4. *Trainee otolaryngology surgeons* who are learning the procedures pertaining to the temporal bone. Interestingly these trainee surgeons are not complete novices because in order to get into otolaryngology they must have at least a general background in surgery.
5. *The IT team at the RVEEH* who will need to maintain the system and take it through their hardware and software upgrades. The main concern of most IT people is that they can manage the major changes to their systems as effortlessly as possible. They certainly don't want the software to break and cause problems that they can't fix.

SOFTWARE ENVIRONMENT AND TECHNOLOGY

The software and technology environment may cause some difficult problems in this project. The original idea for the project brings together:

1. programming in C++;
2. programming 3D models of the human temporal bone;
3. rendering models for 3D stereoscopic viewing using 3D glasses;
4. rendering force models through the haptic device; and
5. maintaining the state of a surgical procedure.

For the development team involved there are quite a number of challenges: (1) the development team is largely unfamiliar with C++; (2) the development team is unfamiliar with modelling and rendering 3D stereoscopic models; and (3) the team has had little experience in haptics and programming for haptic feedback devices.

In the concept phase the original idea was to use C++ because that was more acceptable to the IT department at the RVEEH. It is not an absolute requirement though and can still be negotiated.

For this project, technology may become a major factor and so your project team needs to think about technology very carefully...!

CASE STUDY 2 — A WEB FRAMEWORK FOR LANGUAGE RESEARCH

INTRODUCTION

Research in human languages, ranging from the preservation of endangered dialects through to the use of strategies in collaborative language learning, is often stalled because it is time consuming, idiosyncratic and prone to error. The use of computers for data recording, archiving and as a medium for interaction promises to revolutionise how investigations are conducted and, indeed, offer venues for research into the application of computers in second language acquisition.

The problem with the few computer-based systems in existence are that they are not applicable beyond the limited contexts in which they were built and often must be used within institutions and within institutional aims. The issue is complicated by the fact that the majority of efforts tend to lack sound requirements, mature software engineering practice, clear documentation, and wide interoperability.

This project aims to deliver a web-based framework that can overcome the problems noted above, meet the complex demands of language technology research and education across the globe and be extensible.

This project is interesting because there are the following problems that are related to it being a distributed system:

1. There is the problem of data integration from a number of distributed resources in a number of different formats. One of the features of the system must be to take care of the differences between local — and idiosyncratic — data formats.
2. There is the problem of sharing processing modules in a distributed environment. Should modules be stored local and access remotely — for example by a remote method invocation — or should multiple copies of applications be spread around the network?
3. There is the problem of control. Since this is a distributed system and is intended to be peer-to-peer there is no obvious server that can act as a single point of control.

PROBLEM STATEMENT

The problem of sharing language resources is to allow researchers to share their language resources and processing modules on a global scale.

The problem affects the progress of research, the limiting of language research results to within local communities, the duplication of research and consequently funding and the progress of the discipline.

A web-based system would benefit research community by providing a key point of contact for resource sharing including data and analysis modules as well as a key focal point for the dissemination of the latest work.

KEY FEATURES OF THE SOLUTION: WEB-BASED LANGUAGE RESEARCH FRAMEWORK

The framework will need to provide the following features.

1. A *data integrity* feature that assures the integrity of the applications and data shared within the system and its community.
2. A *user groups* feature that enables certain data and applications to only be shared among a certain subset of users. The system must be able to create user groups, handle membership and assure that data and applications that are registered to a specific user group are used only by users within that group.
3. A *registration of user data and applications* feature that allows registered system users to add their own data and applications to the system for use by the community.
4. A *searching* feature that allows users to setup criteria for searching and to retrieve data and applications via searching.
5. A *transfer* feature that facilitates the transfer of data or applications registered at one location to another location.
6. A *data format translation* feature that supports the conversion of data between certain data formats and internal system formats. The framework should know, from a profile, what data formats are required, and should perform data conversion automatically.

AN INITIAL LIST OF STAKEHOLDERS

As before we have an initial list of stakeholders for the system. We can again expand this list or refine the outcomes for each stakeholder once we know more about the project.

1. *The client* who is a language and technology researcher at the University of Melbourne.
2. *The development team* is a group of 12 relatively inexperienced software engineers. The team has experience in Java programming, but not an extensive experience in the Java environment, for example, they have only a limited knowledge of the Java libraries for image processing and graphics (perhaps you may want to think of them as 4th-year software engineering students).
3. A *management team* consisting of two experienced people from the developer's organisation (perhaps a good way to think of these two people is as the supervisors and coordinators from the University of Melbourne).
4. *The users* of the system will be academics, researchers or language learners who will use the system to share and acquire applications and data resources. They are assumed to be capable of using a PC and to be able to competently use a web browser, register and share resources, and perform search and download activities relevant resources.
5. *The administrators* of the system are unknown at this point. How the system will be administered needs to be decided as requirements. Due to the distributed nature of the system, there will need to be a way for an administrator on one node to fix problems at another node.