University of Southern Queensland
Faculty of Engineering and Surveying

Prototype Development For a Procurement and Electronic Document Management System

A dissertation submitted by

Darryl James Evans

In fulfillment of the requirements of

Courses ENG4111 and ENG4112 Research Project

towards the degree of

Bachelor of Engineering (Civil)

Submitted: 27th October 2011

Abstract

The purpose of this project is to design a prototype for a system that will increase business efficiency through improved electronic document management, improved procurement management and automation of repetitive tasks. This prototype is functional and is used as the primary system at present, with a view to develop it into a dedicated piece of software. The only software that could be used for development was the Microsoft Office 2008 for Mac package or any standard software that exists as part of the Macintosh OSX snow leopard operating system

The system consists of three main parts

1. File management system
2. Procurement and vendor analysis system
3. Task modules

The file management system was developed using the Macintosh OSX file browser, 'Finder', as its backbone. The file management framework was developed using a template approach to promote consistency in the filing process.

The vendor analysis system was developed using Microsoft Excel. It enables the user to manage the contact details of all vendors, the status of all quotes and automatically perform simple analyses on quotes. In addition, it also performs reports statistics such as exposure, opportunity, adjustment and coverage.

The task modules are pieces of computer code written using the Applescript computer language. There are five modules:

1. New job. This automates the job registration process.
2. Close job. This automatically moves old jobs from the active job list while maintaining the functionality of all hyperlinks.
3. Update folders. This automatically creates folders within the file framework that match the package names in the vendor analysis system.

4. RFQ. This takes the contact information and bill of quantities from the vendor analysis system and automatically drafts a request for quote letter. It then automatically drafts an email and attaches the request for quote to it.
5. Return RFQ. This module reads the returned quote and automatically enters the prices into the vendor analysis system.

Opportunities to use different parts of the system have varied considerably. The file management system, new job and close job task modules have been used several times with promising results. Jobs are now registered, filed and stored in a consistent manner and may be done so by any staff member quickly.

The vendor analysis system has been used in practice twice. The system was very successful and proved to be a powerful tool for reporting on the procurement process. The system was designed to provide information at the end of the process, however, it was found to be extremely useful for progress reporting.

The RFQ and Return RFQ task modules have not been used in practice as yet. Alpha testing gave hopeful results as far as the speed and accuracy with which this information is handled and transferred.

It is too soon to realise the full impact of this system on business efficiency. More data will need to be collected over a number of jobs to confirm the preliminary results in this paper.

This project can be expanded further to

- Include a database for storing various vendor and job information
- Develop a method that will notify vendors when new jobs are available and allow them to retrieve documents and submit quotes online.

Chilli Project Services are investigating these ideas.

University of Southern Queensland

Faculty of Engineering and Surveying

ENG4111 Research Project Part 1 & ENG4112 Research Project Part 2

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Engineering and Surveying or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled "Research Project" is to contribute to the overall education within the student's chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Professor Frank Bullen
Dean
Faculty of Engineering and Surveying

Certification

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Darryl James Evans
Student Number: 0050043125

_____

Signature

_____

Date

Acknowledgements

I would like to thank my son Zachery, daughter Addison and especially my wife Rebecca whose love, support, encouragement and understanding has made this work possible.

I would also like to acknowledge Mr. Ben Casley for being the sounding board in my immediate workspace and his advice regarding computer code related issues. Even though Ben is not familiar with the Applescript language his advice was valuable on a number of occasions. Being able to explain what I was doing to someone who is not a civil engineer allowed me to clarify my own thoughts.

I would like to thank Mr. Shaun Goodwin and Mr. Toby Casley, the owners of Chilli Project Services, for providing me the opportunity to undertake this work and the resources required to carry it out.

I would like to acknowledge the help and advice given by my supervisors Dr. David Thorpe and Mr. Toby Casley.

Table of Contents

List of Figures

1.0 Introduction

Civil estimating for construction projects requires a procurement phase often handled by a dedicated procurement officer. This involves contacting potential suppliers and subcontractors in the market place for the purpose of receiving quotes for various works. For any one item, a number of quotes are required; often three or 4 as business practices dictate. These quotes must then be analysed in order to determine the price to use in the estimate. Each supplier or subcontractor will have his or her own inclusions, exclusions and conditions that need to be taken into account when comparing quotes. Other factors such as spread of prices, potential exposure and potential opportunity also need to be calculated for each quote package as a tool for further analysis.

The management of documents pertaining to the procurement process can be extremely time consuming. Furthermore, if the documents are not stored in a logical place they may become difficult to find, leading to a loss of productivity. A system that integrates and largely automates these procedures may lead to increased efficiencies and access to the most up to date information.

1.1 The Procurement Process

The vendor analysis sheet (or procurement sheet) lists and summarises the quotes from various vendors and suppliers. It is used as an information repository for contact information and costs; and assists the estimator in choosing suitable unit rates to use in the estimate. Cost comparisons can be made between vendors allowing potential exposures or opportunities to be evaluated using this information. All quantities with their corresponding unit rates are stored on the vendor sheet. Additional items such as mobilisation and demobilisation of equipment and various inclusions or exclusions are also given dollar values. The vendor analysis allows a comparison of vendors using consistent conditions to find the best overall value. It should be noted that this analysis is performed on a strictly price only basis. The assumption is that all vendors who have

submitted a quote are reputable and capable of performing the volume of work at the time required. Those that cannot meet these requirements should be removed prior to the analysis being performed.

When estimating the cost of any project, there comes a time to ‘lock’ the estimate in order to finalise the bid. At this point, no further adjustments are made to the first principles estimate. However, supplier and subcontractor quotes may arrive or change after this date and it is desirable to include these changes in the final price. This is achieved by doing a ‘below the line adjustment’, which involves the addition or subtraction of a lump sum from the ‘locked price’. This lump sum is taken from the vendor analysis sheet as the sum of the affected item quantities multiplied by the difference in their respective unit rates.

Occasionally, vendors omit items from their quote making them seem more cost effective. In this case, an adjustment needs to be made to the quote in order to reliably compare quotes between companies. This gives the estimator a good basis for selecting the price to use in the estimate and provides a more realistic evaluation of the spread (highest price minus lowest price of quoted prices) received. The spread is used as an indication of the volatility of an industry and gives an idea of the pricing risk associated with it.

In addition to the vendor analysis, basic management of electronic documents needs to be performed and linked with the vendor analysis sheet. The vendor sheet should be stored within a particular job along with all other documents pertaining to that job. This change in business practice needs to improve quality and productivity without additional cost to the business and needs to be easily integrated into the business’ practices. Each package that is sent out to be quoted on consists of documents such as drawings, scope of works, specifications and bills of quantities, therefore these documents need to be stored where they can be retrieved most efficiently.

1.2 Applescript

Purchasing new software, and training staff on it's use, can be an expensive exercise for a small business. With all of the computers in our office running the Apple Macintosh OSX operating system and Microsoft Office Suite, it has been decided to develop a system that suits our purposes from these existing tools.

There are a number of vendor comparison tools and electronic document management systems on the market, such as Microsoft Sharepoint. The question is however, "Can we develop our own basic system using the Microsoft Office Suite and the Apple Macintosh operating system?"

Applescript is a standard Macintosh operating system computer language that allows the user to control various applications through software commands. This language makes it possible to automate processes and information sharing between applications. For example, the user can enter information into a dialog box, which is then automatically placed into a Microsoft Excel spreadsheet. Further information is automatically extracted from Excel and used to create folders on the system server, or perhaps sent to Microsoft word to automatically draft a letter.

Applescript is the key to allowing a functioning prototype to be developed. It ties all of the individual software packages together without compromising the functionality of any of them. Essentially, Applescript can be likened to a much more powerful version of visual basic macros, able to be used across the entire operating system rather than restricted to a single application. It may be for these reasons that the Microsoft Office 2008 for Mac suite no longer supports visual basic macros.

1.3 Project Objectives

The intent of this project is to develop a functioning prototype of a procurement software package. This will be used as a tool to demonstrate to software engineers the

kind of functions that are required, from a civil engineering point of view, in a dedicated software package. It will also serve as the primary procurement system at Chilli Project Services Pty Ltd until a dedicated software package is developed. The broad objectives of this project are listed below with their corresponding specific objectives.

1. Develop a vendor comparison system to
    a. streamline the procurement procedure and increase efficiency;
    b. store vendor contact information;
    c. compare vendor quotes;
    d. assess potential exposures, opportunities and spread of prices;
    e. calculate 'below the line' adjustments.

2. Increase efficiency through automation of business practices
    a. Register a job – Automatically setting up required filing framework and generating a unique job number.
    b. Automatically file returned quotes while complying with AS ISO 154891 (electronic records management).
    c. Generate request for quote (RFQ) documents automatically.
    d. Automatically update the vendor analysis sheet from the returned RFQs.
    e. Automate archiving of closed jobs while maintaining access to them.
    f. Link as much information as possible together through methods such as hyper linking.

3. Develop the basic framework for a document management system:
    a. Design a filing system to electronically store job information that complies with the relevant Australian Standards.
    b. Allow easy and intuitive access to documents required.
    c. Centralise the storage of information.

Document control is not a direct objective of this project, however, consideration must be made in this regard when generating new documents. Automatically created documents should check for the document version or revision and name the new file accordingly. The document management system is intended to be operated manually except for the automated business practices as outlined in point 2 above, therefore the user will need to exercise care when storing documents to avoid misfiling.

2.0 Literature Review

There are a number of factors to be considered when performing a literature review of this subject such as

- Document management vs document control vs records management
- The benefits of an electronic document management system
- Barriers to introduction of new business practices
- Australian standards quality requirements
- Currently available software

These are discussed throughout this chapter.

2.1 Document Management

Electronic documents are a fast and cost effective method for sharing information. A problem arises when we are required to store and retrieve them in an efficient and accurate manner. An electronic document management system is now a necessity for any company in the current business environment.

Document management is not equivalent to document control. Document management is concerned with the efficient storage and retrieval of documents from a filing system. Document control has the added concerns of version control, controlling access to documents, document approval and tracking who has which version of each document.

In his 2009 paper Robertson states, “The fundamental principle of document management is all information resides in one place.” There are a number of benefits to employing this principle and they will be discussed further in section 2.3 below.

2.2 EDMS vs. ERMS

First we need to differentiate between an Electronic Document Management System (EDMS) and an Electronic Records Management System (ERMS).

> "An electronic records management system (ERMS), as the term stands, could be an electronic system for managing records on any media. An electronic system for managing paper records in a records centre or registry would be an ERMS. An ERMS could also be a system for managing electronic records, i.e. computerised records."(Gary P. Johnston 2005)

An EDMS is an electronic filing cabinet. Documents are deposited, retrieved and stored here within some kind of framework that facilitates easy location of the documents. An example of this is the situation where a company stores files on a central server and browses it with Windows Explorer. According to Gary P. Johnston (2005), many of the commercially available software packages (e.g. TRIM® from Tower Software, Wisdom® form Diagonal Solutions) are combined EDMS and ERMS and are therefore called EDRMS or Electronic Data and Records Management System.

An EDMS in the construction industry is of a specialised nature and commercially available software may not be suitable for our needs.

> "Construction EDM presents a very different picture from expert systems or building product modelling, where commercial development was very rare before the last years of the 1990's, but researchers developed numerous prototypes"(Bjork 2003).

(Ryan 2005) notes that many of the systems are little more than an extension of the operating system file browser and Microsoft Office but with limited integration with those systems. This suggests that developing a functioning prototype of a system using these components is achievable. In a recent article, Chimes (2009) describes the evolution of the web based EDMS (e.g. MS Sharepoint or DIRAQ). This allows the

user to access documents anywhere at any time leading to improvements in customer service and the ability for the business to become mobile i.e. there is no need to be in the office to access documents required in the field. This is another consideration for this project.

2.3 Benefits of EDMS

AS ISO 154891.1 (Standards Australia 2002) lists many benefits of good records management (ERM) in section 4. These include the ability to conduct business efficiently, delivery of services in a consistent and equitable manner and facilitation of the effective performance of activities throughout the organisation to name a few. Johnston (2005) lists the benefits with respect to EDMs in particular as follows.

1. "For individual users:
    a. information available when required;
    b. greater quality, efficiency and effectiveness at work (processes are better, easier and quicker);
    c. less blame and dissention when looking for lost information; and
    d. evidence is available for what they were asked to do and what they did.

2. For the organisation:
    a. work is done more quickly;
    b. completing a task requires less effort;
    c. quality of processes and their outcomes is improved:
    d. cash flow is improved; and
    e. compliance with laws and regulations is achieved and demonstrated.

3. For society as a whole:
    a. organisational processes are open and can be understood and monitored;
    b. organisations comply with laws and regulations;

c. quality of life is improved; and
d. the historical record is accessible and reliable."

Want (2009) found that 48% of office workers considered that at least half of the information created in their business was not handled efficiently. This suggests that 25% of all information is inefficiently handled. This is where the above benefits are realised. She goes on to say that the automation of business processes can greatly save time. The benefits are often difficult to measure directly since we lack the suitable metrics to do so according to Bjork (2003), however his view was based on a single project at a time whereas business processes may be measured by time savings to complete a certain task.

This project aims to gain the benefits associated with the EDMS from the above list and the design process will be geared towards achieving this. The primary advantages of this list can be broadly classified as gains in efficiency and reduction in costs. Advantages associated with the tracking of documents are related to the ERMS and are outside the scope of this project.

Want (2009), identifies the function of the EDMS and how it achieves time related benefits through giving the business control over its documents.

> "DMS enables a variety of documents, from Word and Excel to image files and emails to be saved in the same way and in the same place. Everything related to a single project is held together rather than being scattered across a network."(Want 2009)

2.4 Barriers to Introduction

While the above-mentioned articles outline the benefits of implementing an EDMS we must be mindful of the disadvantages that the system may impose. Purchase and training costs, as well as implementation of the system are a concern. A case study by

Maguire (2005) noted that the degree to which the introduction of an EDMS is successful, is measured by how well the staff use the system.

> "In spite of extensive training, most staff never got to grips with the system. This was partially due to three things: (1) staff found the system user unfriendly; (2) the way parts of the system were set up; and (3) having to create complicated business rules to direct staff where the system could not."

This supports the earlier work of Bjork (2003), where he states that

> "The general lesson coming from this type of research is that the simplicity to learn and use the system is a major factor to be considered. The capability to search documents should also as far as possible resemble current practice, thus relatively simple hierarchical folder structures are popular with end users, despite the more advanced features which can be offered by meta data based search mechanisms."

Having used Microsoft Sharepoint myself, I observed reluctance from staff to use it. This was due to the software being difficult to set up and too complicated to use immediately. It tries to cater generically to all types of document management situations and has many Meta search functions that staff were not comfortable using. In the end, Sharepoint was abandoned and a common drive used again for the management of project documents.

These findings suggest that advanced features, which are not necessarily required by the business, may be detrimental to efficiency rather than beneficial. Considering this, using the current Macintosh OSX operating system file browser called 'Finder' (equivalent to Windows Explorer) as the backbone of the EDMS is a reasonable suggestion. This will keep training costs low and ensure a degree of familiarity with the system. The use of the Microsoft Office for Mac Suite is also justified on these grounds.

2.5 Quality Requirements

Robertson (2004) states that documentation managed by a single system leads to quality improvements. He also states that many quality issues stem from incorrect or superseded information being used due to the difficulty in finding the most recent documents. This leads to time delays, reworks and returns, all at a cost to the company.

The requirements under ISO 9001 section 4.2.4 'Control of Records' are that "The organisation shall establish a documented procedure to define the controls needed for the identification, storage, protection, retrieval, retention, and disposal of records." (Standards Australia & Standards New Zealand 2008). This project will address the management of electronic documents only.

Document control procedures are outside the scope of this project as far as the following requirements of ISO 9001(Standards Australia & Standards New Zealand 2008) are concerned:

a. To approve documents for adequacy prior to use
b. To review and update as necessary and re-approve documents
c. To ensure that changes and the current revision of the document are identified
d. To ensure that relevant versions of applicable documents are available at point of use
e. To ensure that documents remain legible and readily identifiable
f. To ensure that the documents of external origin determined by the organisation to be necessary for the planning and operation of the quality management system are identified and their distribution controlled
g. To prevent unintended use of obsolete documents, and to apply suitable identification to them if they are retained for any reason

These procedures refer to controlling documents rather than managing them. Automated document control is an area that this project may expand into in the future, but it is outside the scope of the current project. Documents produced by the designed management system, however, should abide by these standards, particularly parts c, d and e above to aid the control procedures that are already in place within the company.

AS ISO 154891.1 (Standards Australia 2002) sets out the requirements of a document management system along with guidelines advising how to design and integrate such a system into the business practices. This project will follow these standards as far as they are applicable.

2.6 Vendor Analysis

An Internet search revealed a number of Microsoft Excel templates for comparing vendors. Most of these templates were simplistic in their design and did little in the way of the analysis required by a civil estimating company. They are stand-alone spreadsheets and require additional software (i.e. through Applescript) to be integrated into a business system. Therefore the functionality of these available templates is not suitable. The idea that a spreadsheet can be used for this type of function is one that I have adopted.

Available commercial software focuses mainly on purchasing orders or decision matrices. These are based on previous vendor performance and aid in choosing a vendor from a non-pricing point of view. There are a number of reports available from these packages but they are largely that of a historical trail e.g. keeping track of when payments or orders were received. They offered little in the way of the required functionality.

3.0 Methodology

Chilli Project Services uses the Apple Macintosh platform. It follows that all components of the developed system are required to be either standard parts of the Macintosh OSX operating system (e.g. Mail or Applescript) or part of the Microsoft Office for Mac Suite (e.g. Word for Mac or Excel for Mac). Following the literature review, it was concluded that the use of these familiar software tools increases the chances of successful system implementation (Bjork, 2003). It is worth noting at this point that the version of Microsoft Office for Mac installed on the Chilli Project Services computers does **not** contain a database such as MS Access.

3.1 Identify Requirements

The requirements of the system were identified by collecting information regarding Chilli Project Services' needs via interview, own experience, examination of past work and desired outputs. Requirements to comply with Australian standards were also researched through AS ISO 154891:2008 – Records Management.

In order to design a system, the business practices that it handles and their relationships to one another need to be clearly identified. This is a crucial step for the success of this project and correctly completing this task is extremely important.

3.2 Design Filing System Framework

The file management framework will be designed to suit the identified requirements of the organisation i.e. to handle business practices with the functionality required. Findings from the literature review will be used to assist in making general design decisions. An audit of previous jobs will be conducted to determine the business' requirements for file management.

3.3 Design the Vendor Analysis System

The vendor management system will be designed to operate within the previously designed EDM framework. The features of this system will be largely based on the organisations specific requirements. This system will be developed using Microsoft Excel 2008 for Mac.

3.4 Develop Task Modules

Individual task based modules will be developed using Applescript to control the actions of appropriate software packages e.g. MS Word, MS Excel. The purpose of the task modules is to reduce the time taken to perform repetitive tasks and reduce errors due to user input. These modules will be based on the identified requirements of the project and may include, but are not limited to, items such as:

- Registration of a new job
- Generating a Request For Quote (RFQ)
- Archiving old jobs

Visual Basic Macros are not available in the Office for Mac suite; therefore the Applescript scripting language will be used to run automated processes. This has the added advantage that multiple applications can be run within a single script allowing the modules to be linked together and integrated into the EDM framework.

3.5 Test Modules Locally

In order to minimise disruption to other staff and maintain the security of company data, a dummy server and dummy files will be set up on my laptop for the purpose of developing and testing the software. The system will be tested for both accuracy and functionality. Testing will be initially undertaken concurrently with the system

development. Upon completion of the system, it will be trialled by other users on my laptop. This will reduce the chances of corrupting company data.

Comments and feedback will be invited from other staff regarding problems, potential improvements and user friendliness.

3.6 Implement and Monitor The System

The implementation of the system consists of a number of steps

- Move the system to the company server. This consists of altering the Applescript codes to work with different file pathnames and installing the required software onto the other staff laptop computers.
- Train employees to use the system. There are only two other staff at Chilli Project Services who will be using this system. Each staff member will require training to operate the system correctly and an orientation through the file framework.
- Monitor the performance of the system. This may take many months in order to gather enough meaningful data and will not form part of this project unless time permits.

It is important with any project to put control measures in place so that the degree to which it is successful can be measure (Davidson & Griffin 2006). I plan to measure the time taken to perform tasks as a measure of efficiency, and collect user satisfaction feedback. The results of these measurements will be used to modify the design if time permits.

4.0 System Requirements and Limitations

The system has two distinct components being:

a. The file management system
b. The procurement management system

While these systems are autonomous, they are required to operate concomitantly as shown diagrammatically in figure 1. For example, the vendor analysis system is capable of working alone, but it is required to store its information in the filing structure that is created by the file management system. This requirement is responsible for the use of the Applescript scripting language in this project. Applescript allows actions to be performed between different software applications and will be discussed in later sections.

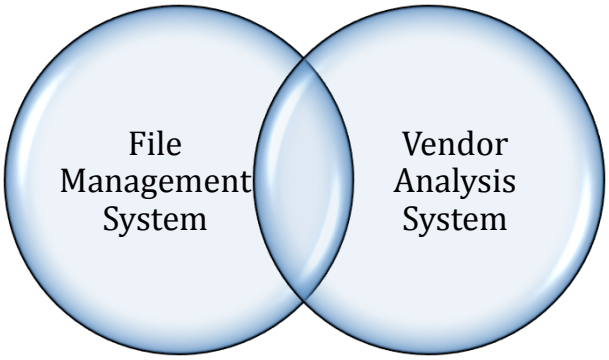


Figure 1 - System Relationship

4.1 File management System Requirements

Examination of the business practices revealed a number of functions that are essential to a suitable file management system. There must be:

a. a method of recording and registering each job in a manner that allows efficient identification and retrieval of documents,
b. creation of a file structure to house all documents and
c. a method to close and archive completed jobs in order to manage the active job list.

The registration process must capture the following information:

- A unique job identification number
- The client for whom the job is being performed
- The staff member or members allocated to the job
- The geographical location of a job
- The year in which the job was started
- The type of job e.g. tender bid, cost plan etc
- The content of the job e.g. road upgrade, waste water treatment plant etc

In addition to this, the process must be largely automated in order to increase business efficiency. It must also be self-directing enough so that any staff member is capable of registering a job and capturing all of the above essential information. Capturing information in a consistent format is desired since this aids processes such as searching, sorting and filtering of the information.

Upon successfully registering the job the filing framework for the job needs to be created. This structure is required to comply with the following business practices:

- It must be housed on the company server
- Created and run within the Macintosh OSX operating system
- Able to be accessed and used remotely
- Use a company standard naming convention so that documents may be located using the standard operating system file browser called ‘Finder’ (equivalent to Windows Explorer). This requirement ensures that if the system fails for any reason, files can still be located manually.

- Flexible and organic i.e. it must be able to grow and allow folders to be added within the framework as required.

The creation of the file framework for each job is to be fully automated. The purpose of this requirement is threefold.

1. To reduce the chances of making an error or omission of data when transcribing information from the registration process into the file management system.
2. To reduce the time taken to create the file structure for a new job.
3. To allow any staff member to use the system and have the correct structure in place.

The file management system must also differentiate between current and finished jobs. Therefore a mechanism by which a job can be 'closed' must exist. Closed jobs must contain all the documents and files that it did when it was active and all hyperlinks must remain functional.

The requirements under ISO 9001 section 4.2.4 'Control of Records' are that "The organisation shall establish a documented procedure to define the controls needed for the identification, storage, protection, retrieval, retention, and disposal of records."(Standards Australia & Standards New Zealand 2008). All of these requirements have been identified through the examination of business practices above and are central to the design of the file management system.

4.2 Vendor Analysis Requirements

The vendor analysis component is the major part of the system and is housed within the file management system. Therefore compatibility with the file management system is essential. Each job must have its own vendor analysis spreadsheet, which is automatically named and filed by the file management system. An analysis of Chilli Project Services' current business practices revealed the following information must be held by the Vendor Analysis system:

- Job identification number.
- Client name.
- Vendor company names.
- Vendor contact person name.
- Vendor contact telephone number.
- Vendor contact email address.
- Vendor unit price quotes.

The company's needs and desired outputs for use of the above information were analysed and the vendor analysis system requirements were identified as follows:

- The ability to house up to 50 subcontractor packages and 50 supplier packages with a minimum of four (4) vendors for each package.
- The ability to house up to 100 items in each package bill of quantities.
- The ability to quickly compare quotes and perform a simple analysis. This must be flexible enough to allow non-standard analyses to be conducted manually if required.
- A method by which a vendor may be selected by the user as a proposed company to award a sub contract to. This may not necessarily be the cheapest quote therefore the flexibility to select any vendor for a particular package is required.
- A method for tracking the status of the vendors quotes i.e. waiting for the quote, quote complete, declined to quote or no response.
- Allow a comparison between quotes and first principles estimate values.
- Provide simple statistics such as
    a. The spread of prices on each package.
    b. The potential exposure due to the spread of prices in each package.
    c. The potential opportunity due to spread of prices in each package.
    d. The total potential exposure of the project due to spread.
    e. The total potential opportunity of the project due to spread.
    f. The supplier coverage for the project.

g. The subcontractor coverage for the project.
h. The relative value of each package within the project.
i. Calculate the 'below the line' adjustment value due to vendor selection changes.

- Provide a summary of the information held such that the entire project may be viewed 'at a glance'
- Have a method to alert the user to that action needs to be taken e.g. if two vendors have been selected for the same package then the user should be alerted.

One of the main objectives of this project was to increase business efficiency. This objective prompted an analysis of the current procurement procedure to identify where potential inefficiencies existed and formed the final requirement of the vendor analysis system. The current business practice is as follows:

1. Create a new vendor analysis spreadsheet
2. Enter the required packages into the spreadsheet
3. Create folders within the file management system for each of the packages.
4. Find potential vendors and enter their details into the system
5. Construct a bill of quantities for each package
6. Compile the documents for each package consisting of
    a. A request for quote cover letter for each vendor
    b. All relevant drawings
    c. All relevant specifications
    d. A bill of quantities
    e. Any other relevant documents i.e. subcontract or supply agreements
7. Email the package to each vendor. This can be a series of up to 6 or more emails to each vendor due to the 10MB email size limitation of most Internet service providers.
8. Receive quotes and enter them back into the vendor analysis spreadsheet.
9. Perform a manual analysis on the quotes
10. Manually produce all of the statistics listed in points a-i above

The final result is a lengthy process that involved a substantial proportion of time to be dedicated to data entry data analysis and other clerical type duties. The aim is to reduce the time taken to complete these tasks.

4.3 Limitations

The main limitations for this project were the type of software packages that I was required to use. When using Microsoft Excel, for example, I was limited to using the functions, formatting and graphics that are available within the package. This was similar for Microsoft Word and Applescript.

'Automator' is an application that allows the user to select tasks from a predefined Applescript library and apply them in a sequential order. I found that these tasks were usually either inadequate for my purposes or did not function in the expected manner. It was also not flexible enough in its structure to allow me the required degree of functionality. This meant that I needed to use the Applescript programming language to write my own tasks. Applescript was a new concept to me and with no real prior software programming experience I had to learn the language before I could move forward with automating the system. This was one of the keys to increasing efficiency.

The opportunity to field test the system was another limitation. It is not often that the entire procurement process is handed from a client to Chilli Project Services, however two concurrent jobs in August afforded me with the chance to undertake limited field-testing.

5.0 Design and Solution

Given the requirements and limitations as described in chapter 4, the data flow diagram in figure 2 was constructed. This shows all of the identified requirements with their associated flow of data and forms the basis of the system design. The circles represent activities or processes while information repositories are shown with parallel solid lines surrounding their names. External entities, such as a user, are drawn in boxes. Arrows with descriptive labels indicate the flow of data. The 'Wiki Package' activity is a future development that I hope to include in the system, but it is outside the scope of this project. The data such as username and password etc required by the 'Wiki Package' activity is still produced by the system but not used as yet.

It can be seen that the right hand side of this diagram represents those activities mainly associated with the vendor analysis system while the top left section is the file management system. The data flow diagram clearly shows the interaction between the two systems and identifies where storage of information is required. This aids in the design of the file management system since we now know how many information repositories are required and what processes need access to them. Two of the repositories namely, 'inbox' and 'sent items' are products of using an email application and were not created by the system. All other repositories have been designed into the system and are required to be created.

Furthermore, we can see the relationships between information and activities, which indicates where Applescript may be used to automate these interactions. For example all activities in figure 2 can be completed by Applescript code. This is the source of most of the potential efficiency gains.

It was decided that a set of templates would be created to maintain the consistency of appearance, naming convention and structure that were required. These templates are copied and renamed upon registering a new job allowing these objectives to be achieved.

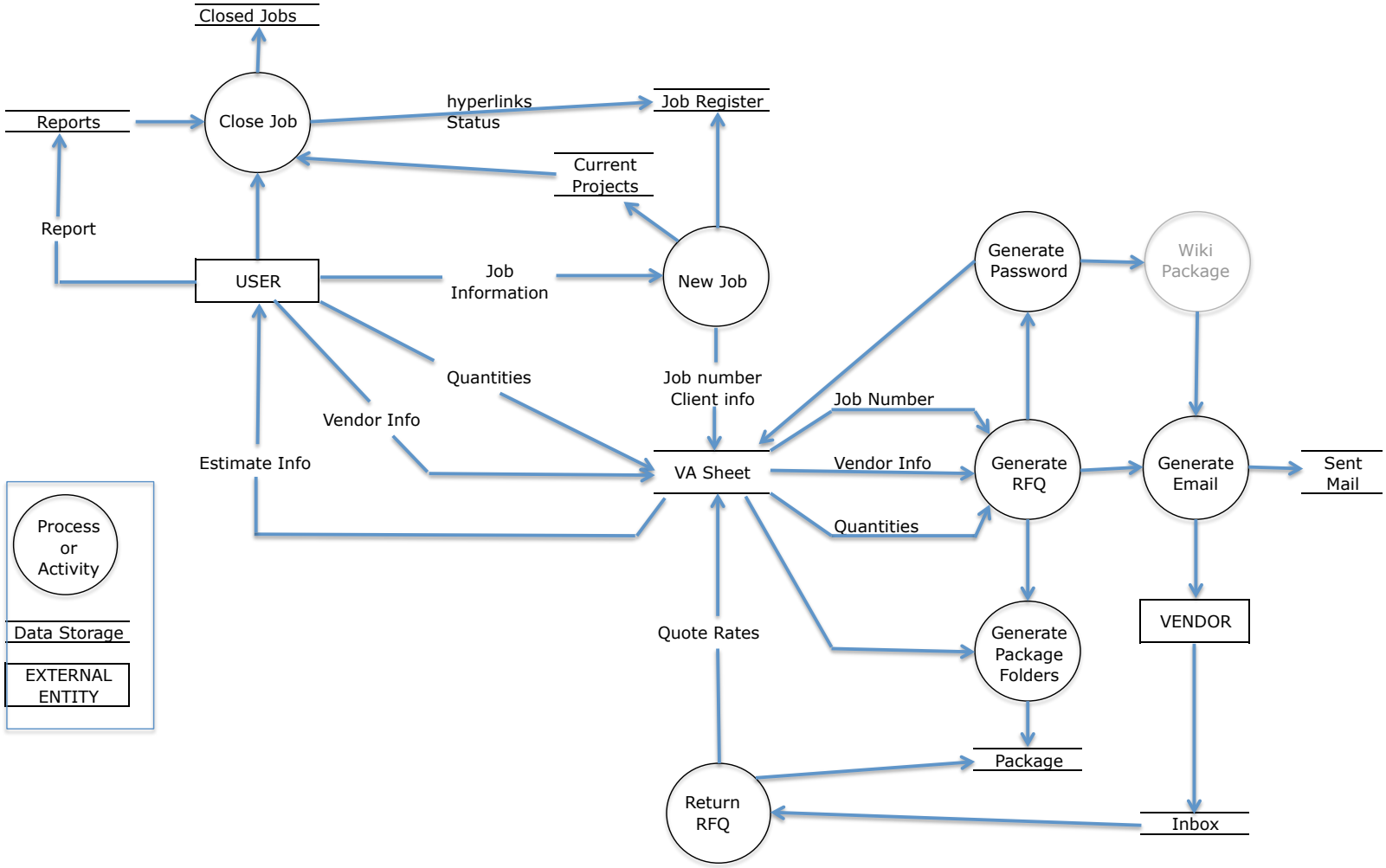


Figure 2 - System Data Flow Diagram

5.1 File Management System Solution

The file management system consists of the registration of new jobs and closing old jobs within a file framework. The registration of new jobs includes creating all folders that are necessary for storing documents and the generation of a unique job identification number. Closing jobs involves moving these files to a closed job folder in order to keep the current job list at a manageable size.

The requirement for the system to be able to be used remotely is satisfied by the use of a Virtual Private Network (VPN). Since all of the required information is stored on the company server, the system will operate over the VPN that is already in place.

5.1.1 File Framework

There were two options considered when deciding on the file framework.

1. Create the folders as necessary
2. Create a template of folders that cover most situations

Through the audit of past jobs, I found that Chilli project Services had been creating folders as required in previous years. Information was difficult to find due to inconsistencies in the naming conventions that different staff members used. It was decided that a template approach should be taken; even if it meant that some folders were unnecessary for certain jobs. The exceptions to the template method are the folders created for storing quotes, which are automatically generated as necessary by the vendor analysis system. If additional folders become necessary elsewhere in the system, they may be added manually to a specific job or to the template if they are likely to be required for future jobs.

Figure 3 shows the file framework template that has been adopted. All boxed terms on the diagram represent folders, whereas unboxed terms represent files. All folders labeled ‘Job Number’ are in practice, the required unique identification number of the

job. Each of these folders contains all of the folders and files housed in the ‘New Job Number’ template folder and are not shown for clarity.

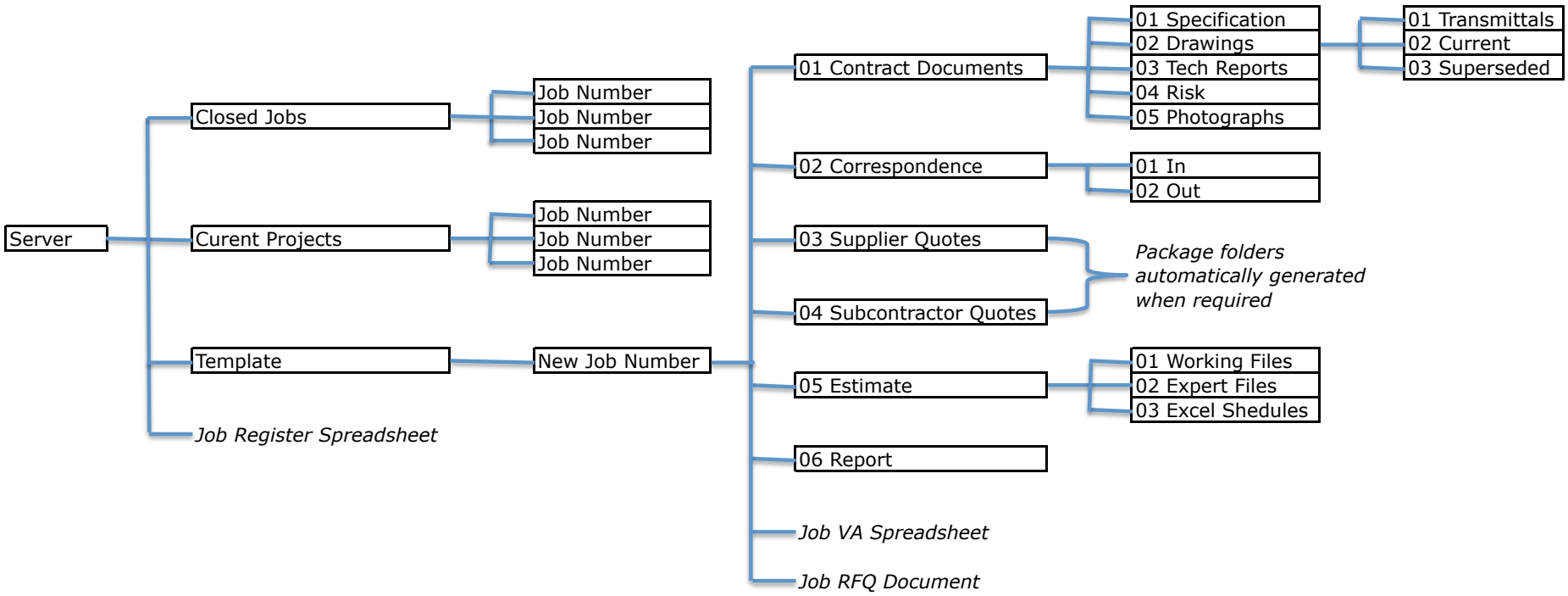


Figure 3 - File Framework

5.1.2 Job Register

The job register is a simple excel spreadsheet, as shown in figure 4 below, with auto filtering enabled for all of the fields. This allows easy searching and accumulation of information e.g. how many jobs has a particular staff member been allocated to or where to find a previous road upgrade project in Far North Queensland so that estimate resources or vendors contact details may be used for a similar new project.

Chilli PROJECT SERVICES

Job Register

Job Number	Client	Region	Location	Project	Job Type	Staff	Report		Status
11-TEST-001	TEST	NSW	dubbo	zoo	Tender Bid	DE			
11-TEST-002	TEST	VIC	melbourne	mcg	Design	DE			
11-TEST-003	TEST	Far North Qld			Budget	DE	PDF	Editable	Closed
11-USQ-004	University of Southern Que	SE Qld	Toowoomba	Car park upgrade	Cost Plan	DE			
11-USQ-005	University of Southern Que	SE Qld	Springfield	Car Park	Tender Bid	DE			
11-TEST-006	TEST	Far North Qld	Townville	stuff	Budget	DE			
11-USQ-007	University of Southern Que	SE Qld	Toowoomba	Presentation Example	Design	DE			

Figure 4 - Job Register

The basic layout of the sheet is intended to show all of the captured information about a project in one viewing screen. The alternating line colours are intended to increase the readability of the sheet and reduce potential mistakes by accidentally reading the wrong line. This is an idea that is carried throughout the design of the entire system. The job numbers in column A are hyperlinked to their corresponding vendor analysis sheets and the report column links a job to its report in both .pdf and editable formats.

The requirement to have a unique job identifier led to discussions about how this would be achieved. It was decided that the job number should indicate when the job was undertaken and the client that it was undertaken for. This simplifies browsing through the system manually to find a specific job. The following format is now standard business practice at Chilli Project Services Pty Ltd.

XX-XXX-XX which represents a 2 digit number representing the year - a 3 letter abbreviation of the client name - 3 digit number showing the number of jobs performed in the calendar year.

For example, if there is a new job for the University of Southern Queensland in 2011 and it is the 4th job that has been registered for the year, the job identification number will be 11-USQ-004. Since this is a standard company practice, the generation of the job number can be automated and is handled via a combination of Applescript code and a data sheet in the job register spreadsheet. Figure 5 shows the data sheet.

	A	B	C	D	E
1	20/10/11	2011	11	USQ	008
2	.	.	.	.	.
3	.	.	.	.	.
4	.	.	Chosen Client	University of Southern Queensland	
5	.	.	.	.	.
6	.	.	.	.	.
7	.	.	.	.	.
8	.	.	.	.	.
9	**Client List**	**Code**	**Region**	**Job Type**	**Estimator**
10	**NEW ITEM**	**NEW ITEM*	**NEW ITEM*	**NEW ITEM**	**NEW ITEM*
11	TEST	TEST	SE Qld	P90	TC
12	University of Southern Queenslar	USQ	Central Qld	P50	SG
13			Far North Qld	Design	DE
14			NSW	Option Analysis	TC/SG
15			WA	Tender Bid	TC/DE
16			VIC	Cost Plan	SG/DE
17			TAS	Budget	TC/SG/DE
18			SA		
19			NT		
20			ACT		

Figure 5 - Data Sheet

The Job number is generated via the following process.

- Cell A1 finds the current date using the date function in excel
- B1 extracts the year from A1
- C1 makes the year a 2-digit number, which is the first component of the job number.
- E1 performs a 'count if' function on the register. The criterion for the function is to count the number of job identification numbers in the register that begin with the year shown in cell C1 of the data sheet. This is the third component of the job number.

- Applescript is used to prompt the user to choose a client from a list box. Applescript then pastes the choice into cell D4 of the datasheet.
- Cell D1 then performs a vertical lookup of the client list and code to determine the client code. This is the second component of the job number.
- Cell F1 concatenates the components of the job number.
- Applescript reads the created job number in cell F1 and copies it to the next blank line in the job register.

The functions of Applescript in the job registration process will be explained in the task modules solution section 5.3.1.

5.2 Vendor Analysis System Solution

The vendor analysis (VA) system is a spreadsheet that contains all of the information required for the procurement process. Each job has its own VA sheet, which is housed within the filing system. The VA system is capable of being used in isolation from the rest of the system allowing it to be viewed and used on any computer that runs Microsoft Excel.

The solution is an excel spreadsheet consisting of five main components. These are:

- Supplier and subcontractor registers.
- 50 Quote analysis sheets.
- A front sheet for summarising the job.
- A dashboard for graphical representations.
- A vendor analysis data sheet.

5.2.1 Supplier and Subcontractor Registers

This consists of two sheets, one for suppliers, coloured yellow, and one for subcontractors, coloured green. The decision was made to keep these separate for ease of differentiation i.e. a supply item such as fencing may be confused with the supply and install item fencing from a subcontractor. This may lead to quotes and other information being incorrectly filed and could cause confusion at a later date if information needs to be retrieved for any reason. The supplier register sheet is shown in figure 6 and is identical to the subcontract register in its format.

Sheets | Charts | SmartArt Graphics | WordArt

A | B | C | D | E | F | G | H

Chilli™ PROJECT SERVICES

Supplier Register

Package	Company	Contact Name	Phone	Email	Status	Price	Proposed Quote
Concrete	Company A	A Contact	0412 123 456	acontact@companya.com.au	Waiting	-	☐
	Company B	B Contact	0402 983 571	bcontact@companyb.com.au	Declined	-	☐
	Company C	C Contact	0414 831 797	ccontact@companyc.com.au	No Response	-	☐
	Company D	D Contact	3343 6079	dcontact@companyd.com.au	Complete	-	☐
Comparison sheet						-	☐
						-	☐
						-	☐
						-	☐
Comparison sheet						-	☐
						-	☐
						-	☐
						-	☐
						-	☐
Comparison sheet						-	☐

Figure 6 - Supplier Register

The supplier register sheet controls the first 25 quote analysis sheets (one for each package) and the subcontractor register sheet controls the second 25 quote analysis sheets.

The ‘package name’ is the identifier for the package that is to be quoted on e.g. concrete supply. It is entered manually on this sheet and is automatically transferred to the heading of the appropriate quote analysis sheet. If the first package name is “Concrete” then the heading on quote analysis sheet 1 will also be “Concrete”. Furthermore, under each package name is a cell containing the text ‘Comparison sheet’ that hyperlinks to the quote analysis sheet for that package. This assists with navigation and allows the user to retrieve information without needing to know which sheet number holds which package.

Column B holds the name of each company that has been approached to provide a quote. These are also entered manually on the register sheets and are transferred to the quote analysis sheets in the same manner as the package name.

The name, phone number and email address of a contact person at each company are all entered manually into the register sheet and are used in some of the task modules that will be discussed in section 5.3. They provide a quick and easy reference point for the user to find the contact details for a specific company if clarification of a quote is required. It also allows others, such as our clients, to see who has been approached for a quote when reviewing the estimate. This is also a good resource when starting a new job. Past jobs can be opened and companies easily found to populate the contact details for a new job.

The purpose of the status is to keep track of who has quoted, who has yet to submit their quote and who is not quoting. The status of the quote may be set to any of the five values shown in the boxes in figure 6 i.e. Waiting, Declined, No response, Complete or just left blank. These are combo boxes and draw their information from a table on the vendor analysis data sheet. Combo boxes work by using a lookup table containing the text that is displayed in the drop down menu along side numbers. When the user makes

a selection from the combo bow, Excel returns the corresponding number from the lookup table to a nominated cell. This number is used in several calculations in order to filter the information used.

The user does **not** enter the analysed quote price. Rather, it is taken directly from the appropriate quote analysis sheet and is linked accordingly. A 'proposed quote' check box allows the user to select the quote that they wish to use regardless of whether it is the cheapest. Once a box is checked, calculations become live and information appears on the front sheet and dashboard. Radio buttons could not be used to control the quote selection process since all radio buttons on a single sheet are linked together. This means that only one selection in the entire sheet could be made and a selection for each package was not possible. Check boxes overcame this problem, however it created another. The accidental selection of more than one quote per package was now possible and the data transferred to the front sheet may be incorrect. To safeguard against this situation, a mechanism to alert the user was designed into the front sheet using a simple 'if' statement and conditional formatting.

In addition to this information, there are a number of hidden cells used in performing various calculations or information storage. Figure 7 shows the hidden cells.

status value	winner STATE	Adjustment	sheet no	Column offset	Username	password
1	FALSE		1	g		
2	FALSE		1	i		
3	FALSE		1	k		
4	FALSE		1	m		
5	FALSE		1	o		
5	FALSE		2	g		
5	FALSE		2	i		
5	FALSE		2	k		
5	FALSE		2	m		

Figure 7 - Supplier Register Hidden Cells

The status value column is a cell required by Microsoft Excel in order to allow the combo boxes to function properly. When referring to the combo box values in formulas, it is not the text in the dropdown menu that is used but rather the number value in the status value cell that is linked to it.

The ‘winner state’ cells work for the proposed quote check boxes in the same fashion as the status value cells work for the combo boxes. The only difference is that Excel can only set the value to either ‘true’ or ‘false’ and a lookup table is not necessary. When a checkbox is selected, indicating the selection of a quote to use, the ‘winner state’ value is set to ‘TRUE’. At all other times these cells will be set to ‘FALSE’.

The ‘adjustment’ cell copies a value from the relevant quote analysis sheet. It is the difference between the quote value and the value currently used in the first principles estimate. It is used in a calculation on the front sheet but is housed in the register to enhance the ability to find information while constructing the system.

The ‘Sheet no’ and ‘column offset’ cells are values that are used in the ‘Return RFQ’ task module. They tell the task module which sheet and column to enter the returned quote into i.e. it identifies the correct package and company. Other methods using lookup commands etc. were tried unsuccessfully due to either limitations with the Applescript commands or with the formatting of the cells in Excel. This method proved to be the most reliable and it does not interfere with the aesthetics of the system since the cells are hidden.

The ‘RFQ’ task module uses the username and password columns. They are storage repositories for the case that a username or password has been forgotten. This allows them to be easily reissued without the need to create a new account.

All cells except those that require information to be entered manually are locked. This is to ensure that none of the formulas are accidentally deleted and that unauthorised people cannot access hidden information.

5.2.2 Quote Analysis Sheets

The quote analysis sheets are where comparisons between companies quoting on the same package occur. There are 50 of these sheets to satisfy the design requirements and each of them houses up to 100 items although the number of items housed may be easily expanded by copying and pasting existing rows. Each sheet holds information such as the bill of quantities and company names for one package only. Figure 8 shows a typical sheet. The user fills in the first four columns manually where the item number, item description, unit of measure and quantity are placed.

The package name and all company names, as entered on the supplier register, appear on the sheet. Each company has a pair of columns under it. These are the unit rate and $EXT (Extended price i.e. the unit rate x quantity) columns. The user (or Return RFQ task module) enters the unit rates into the appropriate column and the locked $EXT column is automatically populated. There is an additional pair of columns with the heading 'Estimate Value'. The estimate values are the rates that are currently being used in the first principles estimate and are shown in blue text to differentiate them from the quotes. The quote totals are shown at the bottom of each '$EXT' column with the difference between them and the first principles estimate value directly underneath. It is important to capture the first principles values since many of the calculations and statistics require the information.

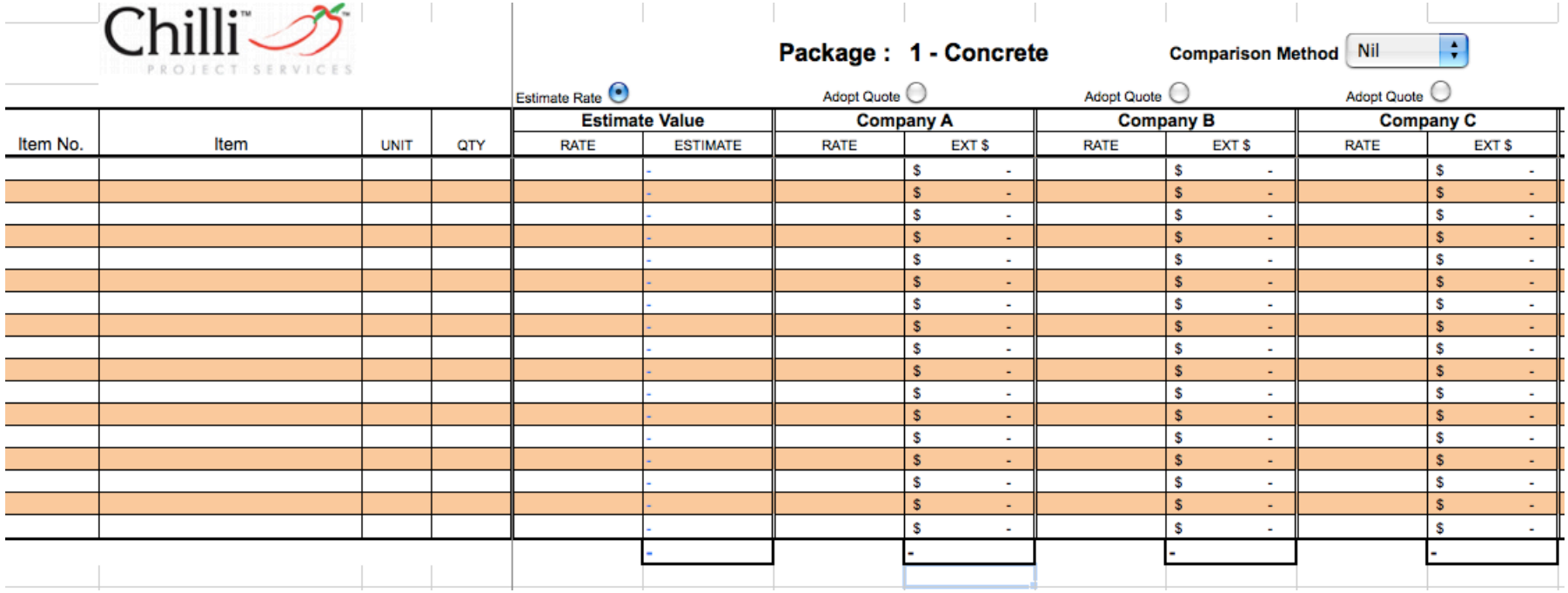


Figure 8 - Typical Quote Analysis Sheet

Not all quotes are in the same format. Some companies will have certain inclusions, exclusions or conditions in their quotes that others do not. The purpose of doing the analysis is to compare the quotes on an even baseline; therefore items that have not been priced by companies need to be added to their quotes. This gives a better indication of the true job cost. To the right of the heading is a combo box with the title 'Comparison Method'. This allows the user to select one of five options to compare the quotes.

1. *Nil*. This is the default value and allows free manipulation by the user where any value may be typed into a blank 'Rate' cell.
2. *Average*. This takes the average value of the other quotes for each unpriced item and applies it to the '$EXT' values. Blank items from all quotes are filled in concurrently.
3. *Highest*. This takes the maximum values from the other quotes and applies them to the unpriced items.
4. *Lowest*. Takes the minimum values from the other quotes and applies them to the unpriced items.
5. *Estimate*. This uses the first principles estimate values and applies them to the unpriced items.

An example of the formula in each 'EXT $' cell to handle all of the possible values is shown below.

```
=IF(AND($B9<>"",$U$6<5,$G$4<>0),IF(G9="",IF($U$6=1,$Q9,IF($U$6=2,$R9,IF($U$6=3,$S9,IF($U$6=4,$F9,IF($U$6=5,0,G9*$D9))))),G9*$D9),G9*$D9)
```

In order to differentiate between actual quoted values and those that were filled in through one of the above processes, the text is turned red and the rate left blank for automatically filled in values. This is done through the conditional formatting function in Excel.

Above each company name is a radio button with the text ‘Adopt Quote’ along side. This gives the estimator the ability to use a quote in the estimate and automatically set the values in the first principles estimate column. The text in the ‘ESTIMATE’ column is turned green and to indicate it the values are taken from a quote. The estimate ‘RATE’ column retains the original first principles such that the estimator can easily revert back if necessary. The selected radio button indicates which of the quotes has been adopted and the unit rates can be obtained from there.

As with the register sheets, there are a number of hidden cells. Figure 9 shows the hidden cells.

Avge(1)	High(2)	Low(3)	adopt control	comparrison
$12.00	$12.00	$12.00	6.00	5
$17.50	$21.00	$14.00		
$26.67	$32.00	$16.00		
#DIV/0!	$0.00	$0.00		
#DIV/0!	$0.00	$0.00		
#DIV/0!	$0.00	$0.00		
#DIV/0!	$0.00	$0.00		
#DIV/0!	$0.00	$0.00		
#DIV/0!	$0.00	$0.00		
#DIV/0!	$0.00	$0.00		
#DIV/0!	$0.00	$0.00		

Figure 9 - Quote Analysis Sheet Hidden Cells

The ‘Comparison’ column consists of one cell that is used as the control for the comparison method combo box. Similarly, the adopt control cell is the value linked to the set of radio buttons.

The ‘Avge(1)’ column performs the calculation to find the average of the supplied quotes. This value is then used in the ‘Average’ comparison method when the combo box value is set to 1.

The ‘High(2)’ and ‘Low(3)’ columns hold the highest and lowest supplied quotes for each item. They are used in the calculations in the ‘High’ or ‘Low’ comparison methods when the combo box value is set to 2 or 3 respectively.

5.2.3 Front Sheet

The front sheet summarises all of the selections and statistical information in one sheet. Figure 10 shows the layout of this sheet. Only the top half of the sheet (supplier packages) has been shown in figure 10 for clarity. The lower half of the sheet summarises the subcontractor packages in the same format.

The only information input on the front sheet are the values in the yellow boxes. All other information is taken from elsewhere and summarised on this sheet. The user inputted values are totals taken from the estimate, which is prepared using other software, and are the total dollar values of the plant, materials and subcontract items. Labour is not included since this system is used for supply items and subcontractors only. The plant value is only used when the plant will be hired and is subsequently vetted. In all other cases plant should be set to zero. Indirect or overhead costs are not generally vetted and therefore only the direct job costs should be used. These values are used in the calculation of coverage on the dashboard and will be discussed in the next section. If the coverage exceeds the estimate value then a warning appears next to the offending category as shown in figure 11.

The fourth yellow box allows the user to stipulate a percentage for overhead and margin. This is used to provide a more accurate calculation of the adjustment figure on the entire job.

.

Chilli™ PROJECT SERVICES

Job Summary

Show Estimate Adjustment -$2.04 Apply Margin and Overhead % 2

JOB NO: 11-USQ-007
Client : University of Southern

Plant Total From Estimate	**$100.00**
Materials Total From Estimate	**$100.00**
Subcontract Total From Estimate	**$100.00**

SUPPLI

	Number of Quotes					Details		
Package	Requested	Pending	Completed	Proposed Company	Price	Spread %	Exposure	VALUE
1 - Concrete	3	0	**3**	Company C	$ 42.00	26.19%	$ 11.00	100.00%
2 -	0	0	**0**					
3 -	0	0	**0**					
4 -	0	0	**0**					
5 -	0	0	**0**					
6 -	0	0	**0**					
7 -	0	0	**0**					
8 -	0	0	**0**					
9 -	0	0	**0**					
10 -	0	0	**0**					
11 -	0	0	**0**					
12 -	0	0	**0**					
13 -	0	0	**0**					
14 -	0	0	**0**					
15 -	0	0	**0**					

Figure 10 - Front Page Summary Sheet

Once a package name is entered on one of the register sheets, it is transferred to the front sheet and hyperlinked to its corresponding quote analysis sheet. This happens automatically and requires no effort from the user.

The ‘Number of Quotes’ summary takes its information from the relevant register sheet. It counts the number of cells in the ‘status value’ column of the register, for each package, that are not equal to 5 i.e. the status combo box is not blank, and reports the total in the ‘Requested’ column. Similarly the ‘Pending’ cells count the number of combo boxes set to ‘Waiting’ and the ‘Completed’ cells count those set to ‘Complete’. If the number of completed quotes is not equal to the number requested, the text is turned red via a conditional formatting function. This can be seen in figure 11.

The ‘Proposed Company’ column remains blank until a check box on the register is selected, at which point the name of the selected company is transferred to the front sheet. This is done by the use of a formula that looks at the ‘Winner State’ column on the register sheet and when it finds a ‘TRUE’ value it sets the cell value to the company name. As mentioned earlier, the potential for multiple check boxes to be selected exists therefore a checking mechanism is designed in. The formula has an ‘if’ statement that checks to see if more than one value in the ‘Winner Status’ column is set to true for each package. In the case that there is a multiple selection, the text ‘More than one quote selected’ appears and is coloured red as shown in figure 11.

The ‘Price’ column works the same way as the ‘Proposed Company’ column except it sets the cell to the price value from the register sheet rather than company name.

The ‘Spread %’ column activates once there is a name in the ‘Proposed Company’ cell. It uses the prices from the register sheets to calculate the spread of prices for each package. It is simply the ratio of the highest price to the lowest price expressed as a percentage. A lower spread indicates that quotes are similar in magnitude and hence a high degree of confidence in the prices. Packages with a high spread value need to have the quotes investigated to find the cause.

Chilli™ PROJECT SERVICES

Job Summary

JOB NO: 11-USQ-007
Client : University of Southern

Show Estimate Adjustment $777.60 | Apply Margin and Overhead % 20

Plant Total From Estimate	$100.00
Materials Total From Estimate	$100.00
Subcontract Total From Estimate	$100.00

COVERAGE EXCEEDS ESTIMATE VALUE

	Number of Quotes					Details		
Package	Requested	Pending	Completed	Proposed Company	Price	Spread %	Exposure	VALUE
1 - Concrete	3	0	3	Company C	$ 42.00	26.19%	$ 11.00	0.70%
2 - ReinforcingSteel	3	1	2	Reo world	$ 5,500.00	9.09%	$ 500.00	91.79%
3 - Quarry Products	3	0	2	MORE THAN ONE QUOTE SELECTED	$ 450.00	11.11%	$ 50.00	7.51%
4 -	0	0	0					
5 -	0	0	0					
6 -	0	0	0					
7 -	0	0	0					
8 -	0	0	0					

Figure 11 - User Alerts

The next column is driven by the combo box. There are three options in the drop down menu, these are:

- Exposure
- Opportunity
- Adjustment

Exposure calculates the difference between the selected quote price and the highest quote price. This parameter indicates the risk if the selected company is unable to undertake the work, e.g. due to clashing with another job at the same time, and another company needs to be engaged. The exposure parameter measures the worst case where all companies except for the highest quote are unavailable and is reported as a positive value since it adds cost to the estimate.

Opportunity is the opposite measurement to exposure. This is the difference between the selected quote value and the lowest quote for each package and represents a potential profit. This only occurs when the lowest quote has not been selected and there is usually a reason for this occurring e.g. poor reputation of the cheapest company. Another reason may be to reduce exposure on packages with a high spread and therefore the opportunity may well exist. Opportunity reduces the estimate magnitude and is therefore reported as a negative number.

Adjustment is the difference between the selected quote and the value used in the estimate for that package. Usually the estimate will adopt the proposed quote values but in the case that a late quote comes in or a change of selected company is required after the estimate has been “locked”, this function provides a quick and easy method to adjust the estimate total. The adjustment value is calculated for each package and may be reported as either a positive or negative number.

The aggregate exposure and opportunity are shown in red and blue text respectively at the bottom of the column. The total adjustment figure is shown via a ‘Show Adjustment’ check box. When checked, the total appears to the right of the box. The

conditional formatting function is used to control the colour of text. Blue is used when the estimate will be reduced, red when the estimate will be increased and white when the box is not checked. Figures 10 and 11 show the cases where the adjustment is positive and negative respectively.

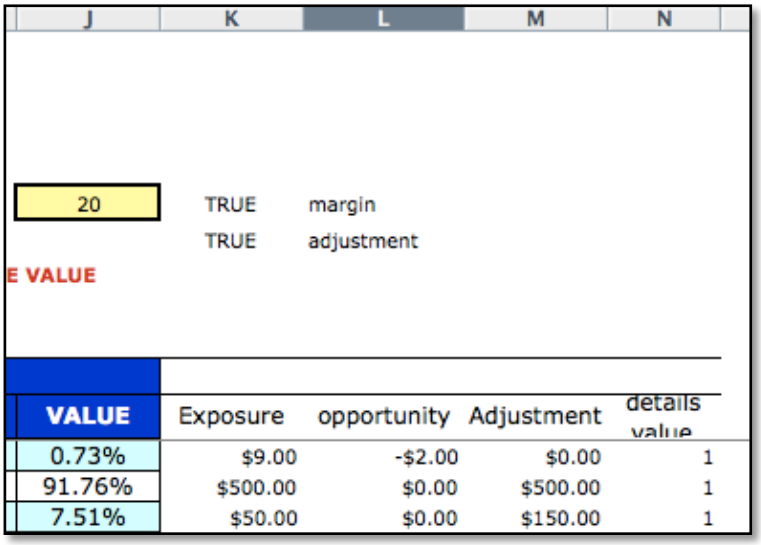
The 'Value' column shows the proportion of an individual package value as a percentage of the total package values. For example, figure 11 shows that the reinforcing steel package is worth 91.79% of the vetted total. This is used to identify which packages hold the greatest influence over the project.

A second check box operates the application of a margin to the price. Once the direct job costs are calculated in an estimate, the sell rate must then be calculated. The 'sell price' is the sum of the direct costs, overhead costs, risk adjustment, contingencies, profit margin and price escalation. Since the quotes in the vetting sheet are all direct costs, a change in the selected quote does not reflect the change to the sell price. A percentage value can be calculated from the estimate to reflect the increase from direct job cost to sell rate. This check box gives the user the option to use this conversion to assess the impact of an adjustment on the sell price.

There are four hidden columns on the front sheet. These are located to the right of the 'Value' column and are used in the calculations. Figure 12 shows these cells when in their visible state.

Column K houses the two control cells for the checkboxes. These are both shown as 'TRUE' in figure 12 and have labels to identify which boxes they control next to them. Below this are the exposure, opportunity and adjustment columns where the calculations for these parameters, as discussed above, are carried out. Putting these calculations in the hidden cells allows us to choose which one we see for individual packages via the combo box in the 'Details' column, while simultaneously showing the totals for all of the categories.

The fourth hidden column is the ‘details value’. This is simply the column that is linked to the ‘Details’ combo box that determines which of the exposure, opportunity or adjustment values will be displayed.



J	K	L	M	N
20	TRUE	margin		
	TRUE	adjustment		
E VALUE				
VALUE	Exposure	opportunity	Adjustment	details value
0.73%	$9.00	-$2.00	$0.00	1
91.76%	$500.00	$0.00	$500.00	1
7.51%	$50.00	$0.00	$150.00	1

Figure 12 - Front Sheet Hidden Cells

5.2.4 Dashboard

The dashboard was a late addition to the vendor analysis system. Some of the information being produced by the system can be appreciated better and more quickly via graphical representation. This information is

- The relative value of each of each package
- Supplier coverage
- Subcontractor coverage

The first item is a graphical representation of the ‘Value’ column on the front sheet as described in the previous section. The next two items refer to coverage. Coverage is the ratio between the value of the supplier and subcontractor items in the estimate to the value of the proposed quotes. This gives an idea of the risk involved in the final price. For example, if we received quotes for all works and supplies and used these values in

the estimate, then we have 100% coverage. This indicates that we know the cost of the project and hence the actual cost should not differ from the estimate. If, however, we only had 15% coverage then we would be less confident in the actual cost of the project since a large proportion of the cost is estimated. Figure 14 shows the layout of the dashboard page. The layout has been designed such that the dashboard display will fill the entire screen of the widescreen monitors that Chilli Project Services utilise when viewed at 100% magnification. This is for maximum clarity.

Figure 13 shows a magnified view of the relative value graph. Here we can see that the x-axis is automatically populated with the package names and the y-axis has its units in percentage. The graph is setup such that the y-axis automatically adjusts its own scale depending on the greatest value that it needs to display.

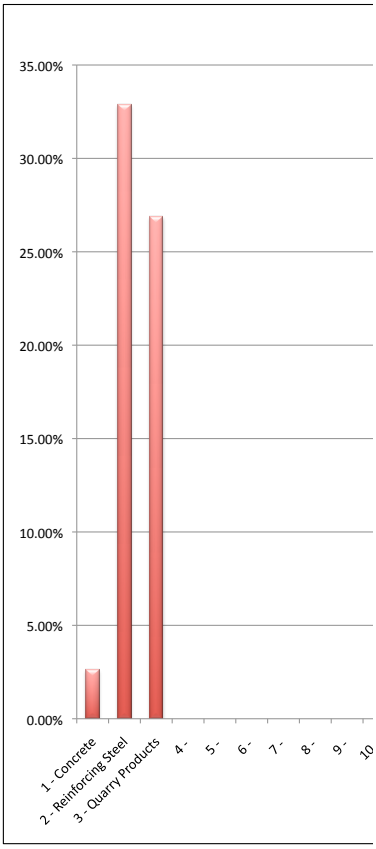


Figure 13 - Relative Value Chart

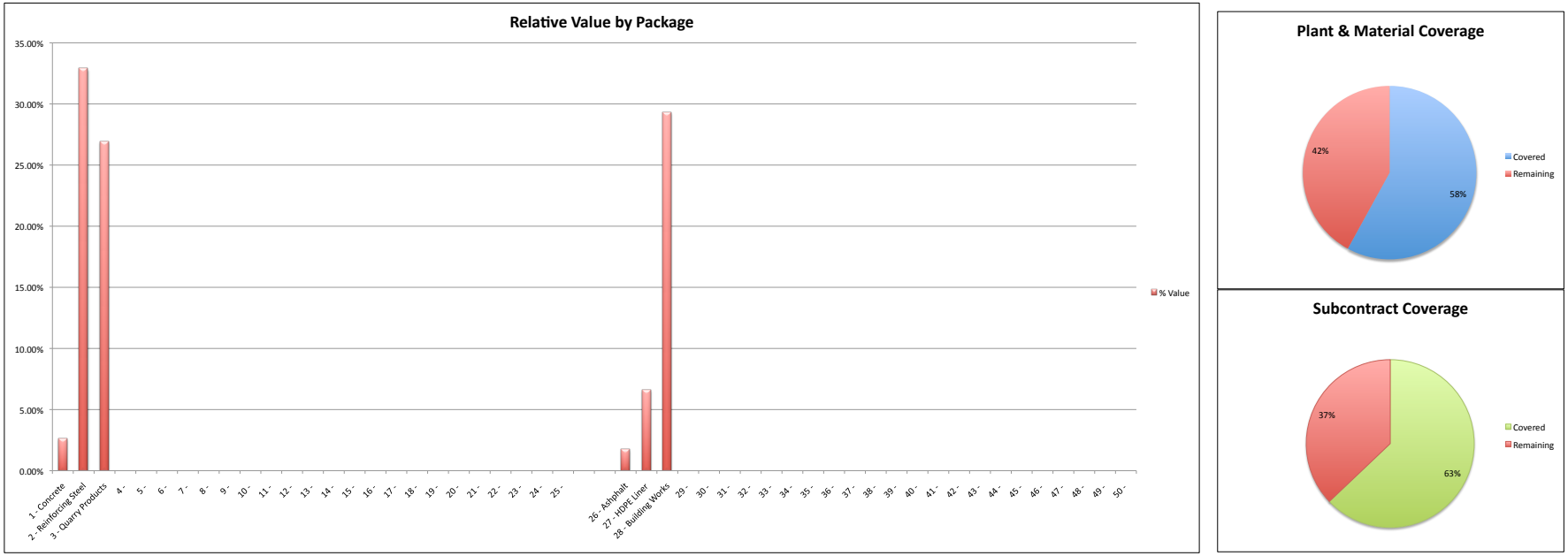


Figure 14 - Dashboard Layout

The supplier coverage chart is shown in figure 15. The blue section shows that we have attained 65% coverage on the materials and plant for the project and the red shows that we have 35% of the price uncovered. This is a quick and easy way to visualise the magnitude of the risk attributed to procurement.

A problem with the dashboard was that hiding the cells that perform the coverage calculation caused the charts to stop functioning. In order to work around this, the cells that are responsible for calculating the data are placed directly behind the charts that they refer to. This way, they do not need to be in hidden cells but are still masked from view and it is also easy to find the cells generating the chart data if troubleshooting becomes necessary.

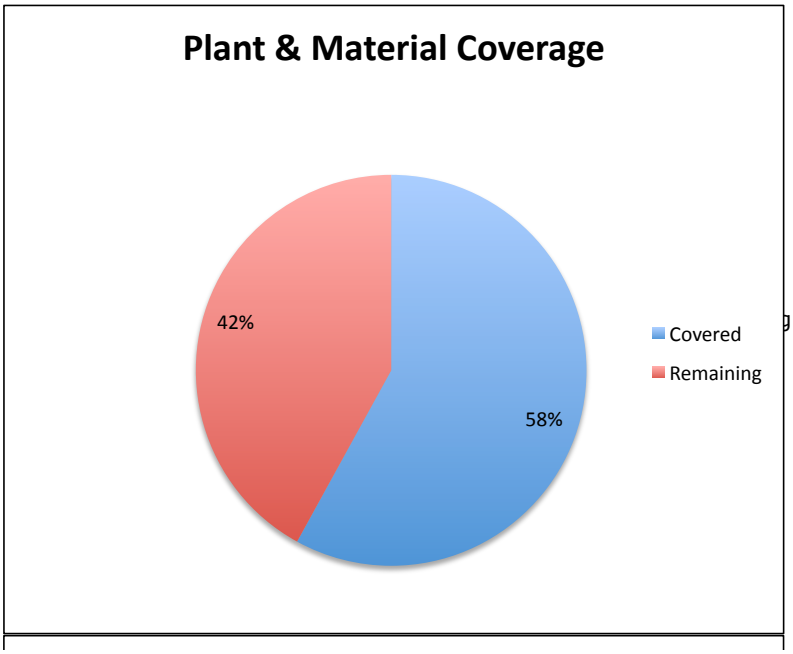


Figure 15 - Supplier Coverage Chart

5.3 Task Modules Solution

The task modules are a set of Applescript based functions designed to automate repetitive actions. Initially, I used Automator as the platform for the Applescript compiler since I was unfamiliar with the Applescript scripting language. Automator is an application that is part of the standard OSX operating system and offers the user the ability to create actions by dragging preprogrammed pieces of Applescript code into a sequential order. In addition to this, Automator allows the user to create an Applet, which is an icon that sits on the desktop and executes the script when double clicked. Figure 16 shows the two created Applets as they appear on the desktop of the computer.

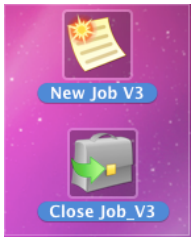


Figure 16 - Applets

The applets proved to be an easy way to install the scripts onto the other computers in the office and a convenient way for staff members to either initiate the registration of a new job or close out an old one.

The generic Automator scripts were far too limited to use for some of the other task modules, therefore they were written in Applescript and placed into the 'scripts' folder of the appropriate application e.g. Microsoft Excel. The scripts menu is located to the right of the 'Help' menu on the task bar of the applications and operates the same way as the *file, edit* or *insert* menus. This allows the user direct access to the functions from within the application they are using.

There are five task modules as part of this system. These are:

- New Job
- Close Job
- Update Folders
- RFQ
- Return RFQ

The first two task modules were created using Automator and are applets. These are essentially icons that are kept on the desktop and are launched by double clicking on them. The next two modules are Applescript scripts that are placed in the scripts menu of Microsoft Excel while the last task module is a script that is placed in the scripts menu Microsoft Word.

5.3.1 New Job Task Module

This was the most obvious of all the task modules that needed to be created. Job registration occurs for every job and is done in the same way every time. The tasks involved in registering a new job are:

- Generate and assign a unique job number
- Capture all of the job details such as location, client, description and job type
- Allocate the job to a staff member
- Set up all folders and files for the new job on the business server

These steps are able to be executed via Applescript and lend themselves to the development of a task module.

The requirements for consistent information formatting and user friendliness are best handled using Applescript prompts. The job registration process is initiated by starting the application ‘New Job’. This is a set of sequential steps controlled by Applescript

code as depicted in figure 14. All “Run AppleScript” steps contain code that was written to perform the required functions while the remaining steps are standard functions that are part of Automator in the Macintosh OSX operating system.

This task module exists as an icon on the desktop of the users computer and is activated by double clicking on it. There is no need for the user to open any applications or search for any files since the module performs these tasks automatically.

Step 1 (Client List) in figure 17 connects the laptop being used to the Chilli Project Services server. The server and all company laptops have been previously setup on a Virtual Private Network (VPN). This allows full remote access to the server at any time and complies with the requirement that the system functions remotely. ‘Client List’ then opens the Job Register spreadsheet in Microsoft Excel and finds the next blank line at the bottom of the list. This is where it will enter the new job information.

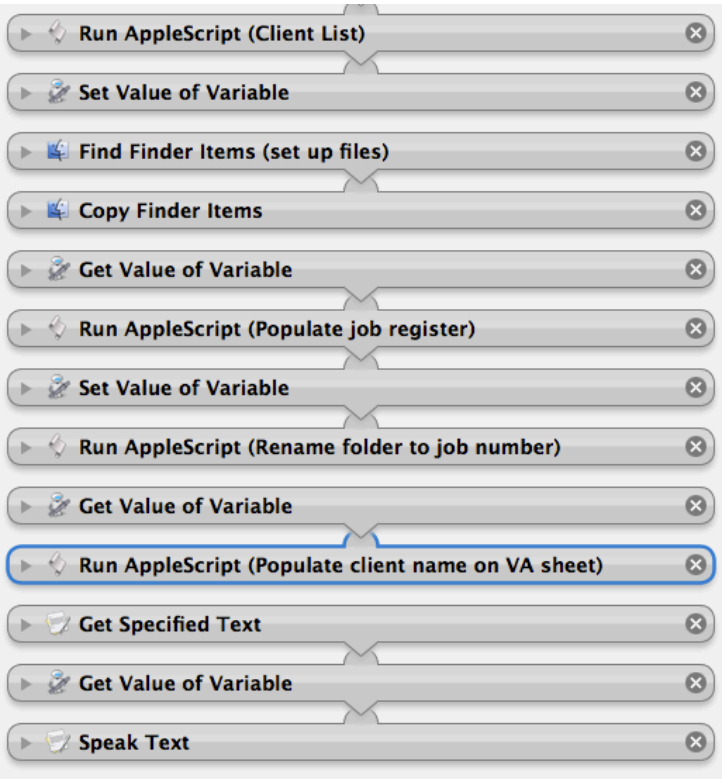


Figure 17 - New Job Application

A popup list is generated from the client list on the Job register data sheet. This list can be seen in figure 18. The user is asked to select a client from the list. If the client does not exist then the user chooses **NEW ITEM** and is asked to enter in the client name and nominate a client code. This information is automatically appended to the end of the client and code lists on the data sheet, which in turn makes it available for selection next time the application is run.

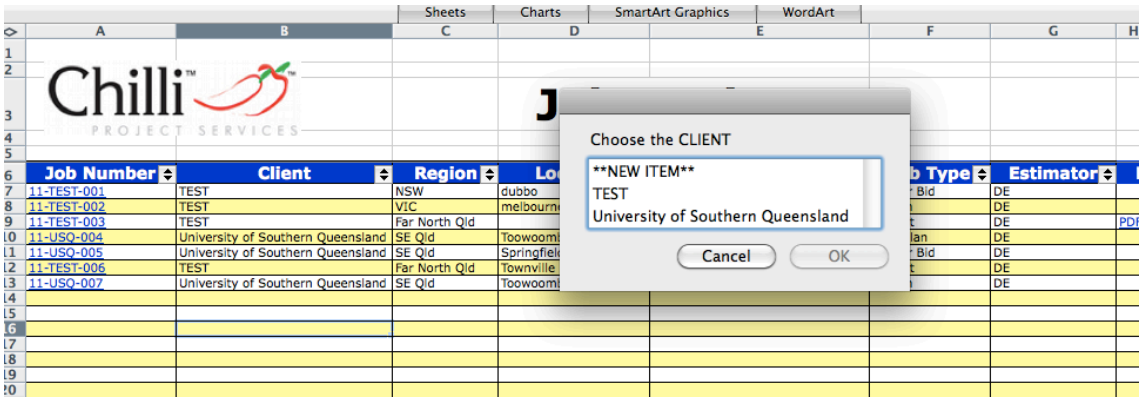


Figure 18 - Popup Menu

Step 2 sets the value of a variable ‘clientname’ to the selection made by the user.

Steps 3 and 4 set up the file framework for the job. They use the file browser to find the file framework template called ‘New Job Number’ and copy it to the ‘Current Projects’ folder on the server. This includes the creation of a blank Vendor Analysis sheet with a generic name.

The ‘Populate Job register’ step then looks at the data sheet of the Job Register spreadsheet. It pastes the client name to cell D4 and a job number is generated as described in section 5.1.2. The generated job number is copied to the register sheet and hyperlinked to the Vendor Analysis sheet generated in the previous step. The remaining fields are populated by copying known data from the data sheet or prompting the user to select from popup dialog boxes. The region, job type and estimator fields are populated via the use of popup lists whereas the Location and Project fields require the user to type a description into a dialog box.

The next two steps extract the generated job number and use it to rename the copied framework and blank Vendor Analysis sheet hence creating a unique set of folders and unique Vendor Analysis sheet for the new job.

Setting up the filing framework is conducted in the background meaning that there is no indication to the user that it has been achieved. The last three steps in this task module were added to provide an audible indication that the process has been successfully completed.

5.3.2 Close Job Task Module

The close job task module was written exclusively using Applescript code and is designed to move all files associated with a job to a closed jobs folder on the company server. The functions of this task module are shown in the flow chart below and the code may be found in Appendix B.

Microsoft Excel uses the entre path name for hyperlinks even if the hyperlink is within the same document. Therefore it is necessary to update all of the hyperlinks in the vendor analysis spreadsheet so that it will still function after the files have been moved. It is an extremely time consuming process to update 200 hyperlinks manually but the use of Applescript allows it to be done in a matter of seconds.

The job register needs the existing hyperlinks to be updated so that it continues to function after closing a job. New hyperlinks are put in place to link the job register with both .pdf and editable job reports. The job status is also updated in the job register to indicate that the job is now closed.

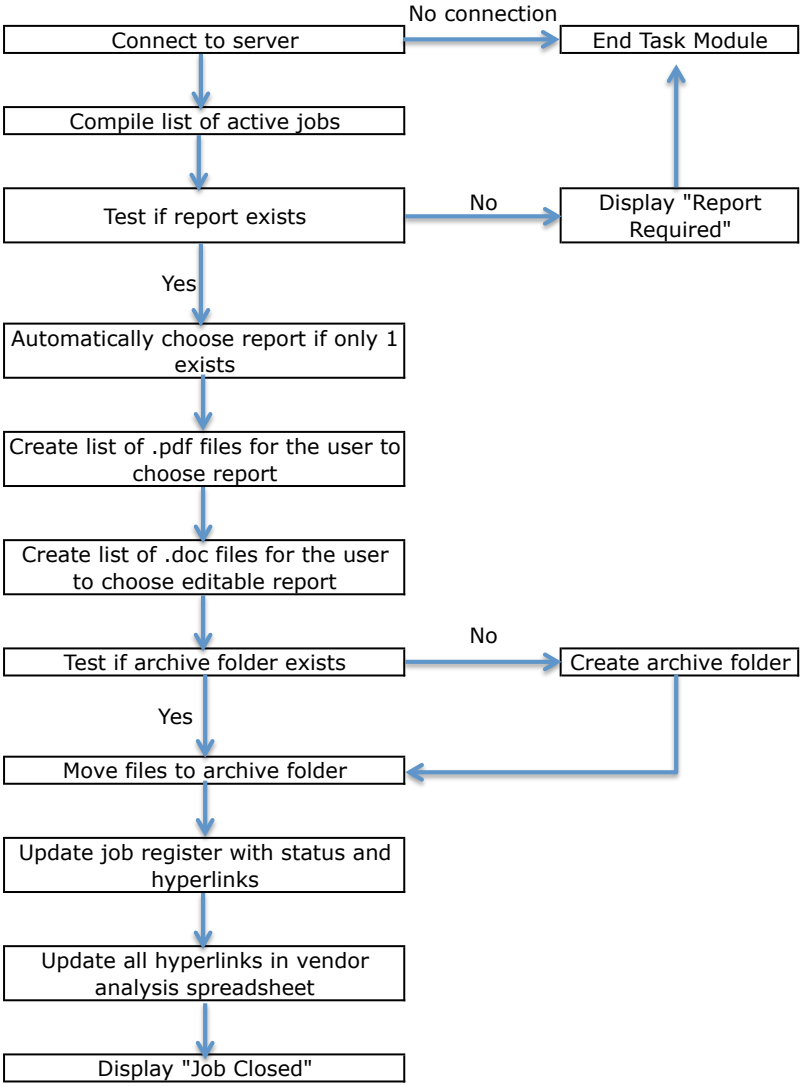


Figure 19 - Close Job Flow Chart

The close job task module has a step where it tests for the existence of an archive folder. The archive folders are housed along side the current projects folder in the file management system. The task module looks at the selected job number and can determine the year in which it was activated from the first two digits. It then searches the server for a folder whose name is made up of that year and the words ‘Closed Jobs’. If this exists then the entire job is moved to the folder, otherwise the folder is created before the move occurs.

5.3.3 Update Folders Task Module

This task module is written using Applescript and exists as a script file in the Microsoft Excel scripts menu. The purpose of this task module is to create folders on the server that have the same name as the supplier an subcontractor packages that are entered into the vendor analysis system by the user. These folders are used to store documents that are returned by vendors regarding each package e.g. terms and conditions, quotes or product information. It is important to maintain a consistent naming convention in order for the file management system to work effectively and creating the folders using Applescript ensures that this is achieved.

Selecting 'Update Folders' from the scripts menu while using Microsoft Excel activates the Update Folders task module. This script is available in the scripts menu for all Excel files, not just the vendor analysis files, although it will not function properly in any file other than a vendor analysis one.

The code first searches for the job number on the front sheet of the vendor analysis file. Next, it enters a loop where it gathers the first package name from the supplier register and tests if a folder with that name exists on the server (see figure 3 for the location). If the folder exists then the next package name is selected and tested, otherwise a new folder is created before progressing to the next package name. The entire process is then repeated for the subcontractor register. The Applescript code can be found in full in Appendix C.

5.3.4 RFQ Task Module

The RFQ task module also exists in the scripts menu of Microsoft Excel. It is used to automatically generate a request for quotation. The user must click on a company name in either the supplier or subcontractor register and then select 'RFQ_V3' from the scripts menu. This task module completes the following tasks:

- Checks that the selected cell contains a company name. Simply checking that the address of the selected cell has 'B' as its column identifier achieves this. If this is not the case then the message 'You must click on a company name first' appears and the program ends.
- Gathers information from the register sheet. The following information is gathered and stored as variables within the task module:
    - Vendor type – supplier or subcontractor
    - Company name
    - Contact name – the name of the person to whom the letter will be addressed.
    - Email address – the email address of the contact person.
    - Sheet name – this is a value in a hidden cell that is the name of the corresponding quote analysis sheet for the relevant package. This was the easiest solution to identify the required quote analysis sheet.
    - Job Number – the unique identifier for the job
- Sets the status of the quote to 'Waiting' in the combo box. This is achieved by setting the value of the linked cell to 1 for the company concerned. This then causes the combo box to display 'Waiting'.
- Gathers information from the quote analysis sheet. The following information is gathered and stored as variables in the task module:
    - Package name – this is the user defined package name e.g. concrete
    - Item list – This is a list containing all of the items on the quote analysis sheet bill of quantities.
    - Quantity List – this is the list of quantities corresponding to the above items.
    - Unit list – The units of measure corresponding to the above items.
    - Item Number List – the item numbers corresponding to the above items.
- Checks that the selection comes from one of the register sheets and sets the path within the file management system to store the generated RFQ. If the selection does not come from one of the register sheets then the message 'Invalid selection for RFQ' appears and the task module is terminated.

- Creates a request for quote letter. The task module copies the template letter document and renames it with the correct job number and company name.
- Check that the folder to store quotes for the package exists and runs the *Update Folders* script if it does not.
- Checks that the RFQ document does not already exist. If it does, then a revision number is appended to the end of the new file name. The revision number is determined by counting the number of RFQ files that already exist. This provides a degree of traceability allowing us to see which version of the bill of quantities each vendor has.
- Populates the RFQ letter. The following information is transferred to the letter automatically.
    - Reference number – this consists of the job name and quote analysis sheet number and is required by the 'Return RFQ' task module.
    - Company Name
    - Date and time of issue.
    - Contact name – addresses the letter to this person
    - Bill of quantities – this is constructed from the lists taken from the quote analysis sheet.
- Generates a login name and password. The task module uses the package name and company name to generate a user name and password. These are not presently used in this system but are intended for use to download documents from a website upon further development of the system. The username and password are then stored in hidden cells against the company name on the appropriate supplier or subcontractor register sheet.
- Generates an email. The task module then uses the gathered email address to generate an email with the RFQ letter attached and indicating what the username and passwords are. A link to the website will also need to be included when this function becomes operational.

Figure 20 shows an example of a typically generated RFQ letter.

Reference Number : 11-USQ-007 2
Company : Reo world
Date : Tuesday, 20 September 2011 3:36:44 PM

Request For Quote

Dear Darryl Evans

Chilli Project Services is seeking a quote from your company for the following items. Please fill in the unit prices and return this document. Include a separate file disclosing any conditions as necessary.

Item No.	Item Description	Quantity	Unit	Unit Price ($)
1.1	Deformed Bar	200	Tonne	
1.2	SL81 mesh	300	Sheet	

Darryl Evans
Civil Estimator

Email: devans@chilliprojectservices.com.au

Phone: +61 7 3861 0233
Fax: +61 7 3861 0277

Website: www.chilliprojectservices.com.au

Figure 20 - Automatically Generated Request

5.3.5 Return RFQ Task Module

The Return RFQ module can only be used if the bill of quantities on the generated RFQ has been filled out and returned. This task module exists in the script menu of Microsoft Word 2008 and performs the following actions.

- Gathers information including:
    - Job number – this is extracted from the reference number
    - Quote analysis sheet number – this is the last number in the reference number.
    - The company name.
    - Item list – this is the list of items in the bill of quantities table. Although this list already exists in the quote analysis sheet, it is used in a checking procedure later on.
    - Price list – this is the list of quoted prices corresponding to the items on the returned RFQ sheet.
- Opens the unique vendor analysis sheet for the job and then finds the correct quote analysis sheet within it.
- Enters the quoted prices under the correct company name.
- Performs a check. The task module compares the quoted item list to the bill of quantities in the quote analysis sheet. If there is a difference then the user is alerted that there has been an alteration made by the vendor and the quote needs to be checked manually.
- Saves the returned quote. The quote is saved to the correct package folder within the file management system. It is renamed so that it is identified as a returned quote. Version control is applied in the same manner as that for a generated RFQ.

The user is not required to click on any particular place in the document before launching the Return RFQ task module. The document must be a .docx or .doc file open in Microsoft Word, however, in order to access the script. This increases the chances that the user will look at the quote and not just blindly enter it into the system without first perusing it.

6.0 Testing and Implementation

The system development occurred locally on my laptop to minimise the chances of corrupting or losing documents from the company server. It also allowed work to progress without hindering staff in the office, since there were no disruptions to the server. Upon completing the entire system, the necessary adjustments to the Applescript code (i.e. path names etc.) were made and the system was then operating live on the company server. The only software that needed to be copied to the server was the filing system template; all other software exists on individual laptops.

Each employee had a package of 5 task modules to be installed on each of their laptop computers. This was a simple case of emailing the modules to staff with instructions describing which location to copy and save each file to on their laptops. The RFQ task modules were modified slightly for each staff member so that they use an RFQ template with their own signature.

Staff training consisted of one on one demonstrations and tutorial sessions. This was deemed to be an appropriate training delivery method due to the low number of staff requiring training (only 2 other staff members). A copy of the file framework diagram (figure 2) was supplied to each staff member to aid with initial navigation through the system.

6.1 Alpha Testing

Testing was an iterative procedure and occurred concurrently with the development of the system. As each part of the system was developed, it was tested to see if it behaved in the expected manner. Any detected problems were resolved before proceeding with the next phase. All parts of the system were tested in this manner, however not every user action can be predicted and some unexpected behaviors may become apparent when it is used in the field.

The checking procedures and warning messages that have been incorporated into the system are a direct result of this testing process. Some examples are:

- The “More than one quote selected” warning message on the front sheet of the vendor analysis system.
- The “Coverage exceeds estimate” warning message on the front sheet of the vendor analysis system.
- Version control in the RFQ and Return RFQ task modules
- Checking for the existence of folders before creating new ones in all task modules.

6.2 Beta Testing

There have been varying degrees of opportunity to use different parts of the system in the field. This is due to factors such as the timing of different parts of the system being ready for use and the type of work Chill Project Services was engaged to undertake i.e. not all jobs require us to perform the procurement role. The following table shows the number of times each part of the system has been used in the field.

System Component	Number of Times Field Tested
New job task module	13
Close job task module	5
File management system	13
Vendor analysis system	2
RFQ task module	0
Return RFQ task module	0

Table 1 – Field Testing Opportunities

The RFQ and Return RFQ task modules have not been used on live jobs at the time this paper was written. Although we have had two jobs that required us to undertake the procurement role, our client had very specific RFQ documentation that they wanted sent out to the market. This resulted in the RFQ modules being unsuitable for these particular jobs and as such beta testing was not possible.

The new job and close job task modules performed as expected on live jobs. No errors have been found at this stage after using both of these modules a number of times. The file management system also performed as expected without any problems.

The vendor analysis system was found to contain some errors where references were made to incorrect cells or hyperlinks were made to incorrect sheets. This was mainly due to the expansion of the system from its initial capacity of 20 vendor sheets to its current capacity of 50 vendor sheets without making all of the necessary changes to the copied formulas. The advantage of using a template system is that once the master copy has been corrected, all subsequent jobs will function correctly. The disadvantage is that if an error is picked up after a number of jobs have been completed then retrospective corrections need to be made on all previous jobs including an analysis of the impact - (if any) it may have had on the information produced.

7.0 Project Outcomes

This section will report how well the project objectives were met. The major objectives of the project as described in section 1.2 are:

1. Develop a vendor analysis system
2. Increase efficiency through automation of business practices
3. Develop a file management system

Furthermore the overall aim of the project was to develop a functioning prototype to act as a specification for a dedicated software package to be written.

7.1 Vendor Analysis System Outcomes

The first minor objective of this category was to streamline the procurement process in order to increase efficiency. Due to the low number of actual jobs that the system was used on, there is insufficient data to conclusively show that this has been achieved. Typically, jobs of the size and nature that we used this system on require a full time procurement officer to manage the procurement process. While using this system, one person was able to perform the procurement on two jobs simultaneously and work on a third job. At this early stage, it seems that the vendor analysis system has had a substantial impact on the efficiency of managing the procurement process.

The second minor objective was to store vendor contact information. The system performed very well in this respect, capturing all of the required parameters and providing an easy method to see which companies have been approached for each package. There were no great differences between using this system and creating a contact list using an excel spreadsheet as far as storing contact information is concerned. The difference came with the integration of the other functions with the contact information. The management of quote statuses, the ease of selecting a vendor

with their post analysed price displayed along side the contact details and the summary of proposed companies for each package on the front sheet greatly added to the value of the contact list.

The third minor objective was to allow the comparison of quotes. The vendor analysis system met this criterion with the inclusion of the quote analysis sheets. The user can manually make adjustments as required in order to compare quotes on a level playing field. The system exceeded this basic functionality and included a choice of four automatic quote comparison methods. These automatic methods were used to analyse a number of the packages in the two field-tested jobs. This performed the analysis instantaneously and produced the same results as an equivalent manual process.

The ability to adopt a quote via selection of the radio buttons was used a number of times for both pre and post-analysed quotes. This provided a quick and easy way to keep the first principles estimate values up to date, without requiring the estimator to invest their time in manually entering the values. User feedback indicated that when a quote is adopted, that vendor should automatically become the proposed vendor, which does not occur presently. Currently, the user is required to select a check box on the register sheet in addition to selecting a radio button for adopting a quote. Making this process a single action would reduce the confusion that arises in this regard.

The ability to assess potential exposures, opportunities and spread of quotes was the fourth minor objective. The system performed well in this regard. This information was available as a total for the job or broken down for each package. Non-compliant quotes e.g. installing kerb and channel without having the correct profile, lead to these calculations being skewed. Deleting the non-compliant items and using the automatic quote comparison tool to analyse the quote handled this situation. This information was always available instantly and required no effort on behalf of the user to generate it. This was a vast improvement over compiling this information manually for each package.

The system was also required to calculate below the line adjustments. This relies on the user maintaining the first principles estimate values and quantities to match those that have been used in the estimate. This function was not used in the field-tested jobs; however, it provided us with another unexpected quality control tool. By looking at the adjustment figure, we had an independent method of checking that the value of the quoted items in the estimate was correct and that errors in translating the data have not occurred.

The power of the instant reporting tool became apparent at meetings with our clients. Of particular interest was the ability to report the market coverage and give a report on the quote status. The reporting was designed to provide information at the conclusion of the procurement process as well as a means to manage the process for the procurement officer, however it was used extensively to provide progress updates throughout both jobs. There was no need to invest any time into producing progress reports since the information was already summarised on the front sheet and dashboard of the vendor analysis system.

Some of the limitations of the vendor analysis system are its restricted ability to house data. Presently the system can only cater for 50 packages with 4 vendors in each package. The need for the system to be organic and have the flexibility to add as many packages and vendors as required must be addressed when the dedicated software is developed. The bar graph on the dashboard always shows 50 packages regardless of how many are being used. This gives the graph the appearance of being empty on smaller projects. This may give a client the impression that the job is not progressing well when in fact it is. Once again, this issue needs to be addressed in the development of dedicated software.

7.2 Automation of Business Practices Outcomes

Three staff members have used the new job task module on 13 jobs over the space of 7 months. The table below shows a comparison of the time taken to set up a new job with respect to the method used.

Method	Time Taken	Comment
Manual without templates	10 mins	Various forms of file structure set up, if any. Errors with job numbers and captured information.
Manual with templates	5 – 7 mins	Better consistency of file structure. Still errors with numbers and captured information.
Task Module	45 - 60 secs	Consistent file structure and captured information.

Table 2 – Job Registration Comparison

While there is a saving in time for the job registration process, it is not significant over a period of time due to the infrequency with which a new job is registered. Over a seven month period the time saved in the registration process was 2 hrs, at the most. The real advantage of the task module lies in the consistency of information collected and the file management structure that it provides. No errors in information translation have been detected since the task module has been operational and all electronic documents have a known place to be stored and retrieved. This has improved the quality of stored information, enabled us to gather and analyse data about the jobs we have undertaken and increased the ability for all staff to find the documents that they require.

The second minor objective of this category was not fully met. The objective was to automatically file returned quotes. The return RFQ Task module provides a method of doing this, however, experience has shown that vendors are reluctant to provide a quote on a supplied word document. Vendors, especially large companies, tend to provide quotes as .pdf documents on company letterhead. The reasons for this may be that they have a quoting system that they use to make the process more efficient, or for document security. Companies do not want send out quotes in a format that can be altered by other parties and prefer the security of .pdf files. It is difficult to automatically detect where a

quote should be filed when the quote files arrive in so many different formats. This finding has generated further investigation into how this objective may be better achieved. The objective to automatically update the vendor analysis sheet from returned quotes shares the same problems.

The closing or archiving of completed jobs is required in order to keep the active jobs list at a manageable level. 5 jobs have been closed using the close jobs task module and all were done successfully. The time taken to simply move a job from the current jobs folder to a closed jobs folder is only a matter of seconds, however, updating all of the hyperlinks so that they still function may take a substantial amount of time. The following is a list of hyperlinks that need to be updated when closing a job.

- Job number in the job registration sheet
- PDF and Editable reports in the job register
- Up to 50 sheet hyperlinks within the vendor analysis system

If we assume that a hyperlink can be manually updated, error free, in 20 seconds, this equates to 17.6 minutes. I was able to close 5 jobs in 3 minutes using the task module leading to a potential time saving of 85 minutes. Closing jobs are more likely to occur in groups rather than singly, therefore the impact of the time saving becomes more obvious.

The requirement to automatically generate a request for quote was met with the RFQ task module. Unfortunately, there have been no opportunities to field test this module to date. Alpha testing of this task module gave promising results and positive feedback from staff who received training for the system. Creating individual letters for each vendor can be a time consuming process and carries with it the potential for incorrect information to be sent out. For example, a letter may be written to company A and then copied and altered before sending to company B. It is possible that some of company A's details may not be altered on the copied letter before it is sent to company B. It looks unprofessional to send a letter out with the wrong details on it and may be

offensive to the party receiving it. This task module ensures that errors of this nature do not happen.

Navigation through the system has been improved by the addition of hyperlinks such as:

- The job number in the job register is linked to its vendor analysis package
- The PDF and EDITABLE report listings in the job register are linked to their reports.
- Every package name on the front sheet and supplier and subcontractor registers within the vendor analysis system is linked to its quote analysis sheet.
- The Chilli Project Services logo on every sheet in vendor analysis system linked to the front sheet. It acts as a home button to bring the user back to the front sheet.

The system also gave the user the option to navigate through the file management system using the finder file browser to locate the vendor analysis or report that they required, if they were more comfortable doing it this way.

The transfer of information has improved through reduced data entry and automatic transferring of information via task modules. Errors in calculations for spread, exposure, opportunity and adjustment did not occur on the field-tested jobs. This improved the quality of information that was produced and reduced the time to produce it.

7.3 Document Management System Outcomes

The document management system was the first part of the system that was developed. Even though a training session and a copy of figure 2 (file management framework) was provided to each staff member, it still took some time for staff to adjust to the new system. Early problems were mainly due to a lack of familiarity with the file structure, which led to documents being misfiled. As staff gained more exposure to the file structure these problems disappeared. There have been occasions where documents

from old jobs needed to be retrieved and were easily located because they were stored under this system.

A shortcoming of the system became evident when using the vendor analysis system for the first time in the field. All quotes and documents received from vendors are filed under the package name on the server. This proved to be disorganized since vendors will not only send a quote but may also send terms and conditions, product information, confidentiality agreements and other documents leading to many documents from many companies being stored in one folder. The system allows the flexibility to manually create folders for each vendor, however in my opinion modifying the Update Folders task module to create them automatically would be more efficient in the long term. The RFQ and Return RFQ task modules will also require updating so that the documents that they generate are filed under the correct vendor folder.

Another disadvantage that was found with the file management system was the time taken to transfer client-supplied documents onto the server. Previously the entire contents of a CD or DVD was simply dragged onto the server and stored in the format in which it was supplied. With the introduction of this system, each file must be stored in a particular place, which meant that more time had to be invested to ensure that the files were stored in the correctly on our system. The disadvantage of this has been outweighed by the benefits we have seen with document retrieval, especially on closed jobs when the original client format had been forgotten and documents become difficult to find.

The requirements under ISO 9001 section 4.2.4 ‘Control of Records’ have been met. The identification, storage and retrieval procedures occur through the use of the system. Identification of documents is done by the job number process as well as using pre existing folders in the file management system e.g. the specification for a job will be stored in the specifications folder of a certain job number. This not only provides a means for storage and retrieval, but also provides identification for what the file is. Any new documents created by the system are named with the job number, description of the file and a revision number e.g. 11-USQ-007_RFQ_Concrete_rev1.

Protection, retention and disposal of records were outside the scope of this project but have been handled by our Information Technology department.

8.0 Conclusion

This project has shown that it is possible to develop a functioning prototype for a vendor analysis and file management system using the standard software available on a Mac Book running the OSX Snow Leopard operating system and Microsoft Office for Mac.

Significant improvements to efficiency of the business and quality were achieved by implementing this project. The integration of managing the procurement process with analysing and reporting in real time proved to be the noticable advantage. The ability to keep track of procurement progress with up to date statistics increased the quality of work produced and decreased the time taken to produce it. Clients were able to have questions about procurement answered immediately with the most current information at any time.

The task modules improved quality by reducing transcription errors and reducing the time taken to perform tasks. They also eliminated the need for the user to remember to perform every part of a task. This increased the quality of the data that was gathered, ensured the job register was always up to date, eliminated the chance of duplicating a job number, guaranteed that hyperlinks for closed jobs still functioned and maintained a consistent file management system. Further investigation into the automatic handling of returned quotes is required and Chilli Project Services are currently undertaking this investigation as a result of this project. The task modules are an essential part of the system and need to be incorporated into a dedicated software package when it is written.

The vendor analysis system has the ability to stand alone on any computer that has Microsoft Excel installed. It has successfully operated as a tool for quote management, quote analysis, progress reporting, quality control and information reporting on each of the jobs for which it was used. The system performed better than expected, with the ability to provide real time progress reports proving to be a valuable tool. Task modules are not essential to run this system, however they have the potential to enhance it. The

RFQ and Return RFQ task modules are specifically designed to make the transfer of information in and out of this system quicker, easier and more accurate. The limitation in the number of quotes that can be housed in the system needs to be addressed for a dedicated software package. The ability for the system to grow organically as required is something that was not easily achievable using Microsoft Excel but can be achieved with dedicated software. Instead of having 50 available package spaces and 4 vendors within each package, the system should allow the user to add as many packages and vendors as they like and only as required. This would improve the appearance and flexibility of the system.

The graphical reporting shown on the dashboard sheet of the vendor analysis system is limited by the graphs available in Microsoft Excel. Better graphical tools such as gauges and interactive charts need to be incorporated into a final software package. These graphical tools allow a complete review of the procurement process very quickly to identify areas that need to be focused upon. Developing or improving these tools further will enhance the system further.

The file management system works well for the identification, storage and retrieval of documents. The template system allowed staff to become familiar with the file framework and hence become proficient at navigating through it. The file management system does not eliminate the chances of misfiling or using superseded documents, however, it does reduce them. The control of documents was outside the scope of this project, but version control was used when generating any new documents. Permissions and version control for all documents form the next logical step with the file management system.

The implementation of the file management system was barrier free. By sticking to a simple hierarchical system of folders that could be navigated using the standard operating system file browser, users felt comfortable with the workings of the system and only had to familiarise themselves with the structure of the folders. Once this was achieved, all jobs could be navigated easily since the same type of document was kept in the same place for every job. The value of this system was not immediately realised

since an adjustment period was required and files provided by clients needed to be rearranged to fit our framework. It was not until three or four jobs were done using the system that staff became competent with browsing it. Searching for documents in past jobs that were registered using the system, became far more efficient since staff knew where files should be located.

The job register has allowed Chilli Project Services to manage their workload better by providing data on the type of jobs that are active and individual staff workloads. Other data for geographic location, clients and job description are also being collected for use by the business. The register provides a means for locating subcontractors and suppliers in different geographic locations. As the number of times the vendor analysis system is used increases, the ability to find past vendors from the system will also increase. This acts as a rough database for suppliers and subcontractors and will be extremely useful since finding these vendors is often a time consuming process.

Preliminary findings suggest that this project has been very successful. Chilli Project Services has successfully used the system in the field and is satisfied with the performance of the system. It has improved quality and business efficiency even in its current prototype form. There is an industry need for software of this nature and this prototype is able to serve as a specification for a dedicated software package to be written while being used as an interim solution to a business problem.

8.1 Further Work

Linking this system to a database is the next logical step for this project. This would enable the user to perform functions such as:

- Search for vendors by geographic location and discipline
- Track market trends by geographic location allowing an estimate of price escalation to be made.

Reducing the time taken to perform tasks is a key element of this project and there are further areas in which this can be achieved. This will not only improve efficiency, but will also reduce costs. When contacting vendors, up to 200 telephone calls may be required at a cost of say $0.20 per call. That is a cost of $40 in phone calls and 16hrs (assuming 5 min per call) in labour. Using a database for preferred vendors by geographic region and discipline could open up the possibility of automatically sending out email notifications when packages are ready to be quoted on. This will potentially eliminate the phone call costs and save two days labour.

The method for vendors to download large files needs to be developed in such a way that the RFQ task module can automatically provide a website link along with the username and password. The method of using a wiki hosted by the Chilli Project Services server has potential issues with the security of other sensitive information housed on the server. A method whereby the packages are uploaded to a website and then downloaded by the vendors needs to be investigated. If it is possible, a method for determining who has downloaded the documents and when they were downloaded is advisable. This allows the management of quotes in the vendor analysis system and provides a method for document control i.e. when a new document is released after the downloaded date a notification is sent to those vendors that require it.

Chilli Project Services are currently investigating these future options and hope to have a marketable software package in the near future.

List of References

Bjork, B-C 2003, 'Electronic document management in construction - research issues and results', *ITcon*, vol. 8, pp. 105-17, viewed 22/3/2011, <http://www.itcon.org/cgi-bin/works/Show?2003_9>

Chimes, R 2009, 'What Exactly IS Electronic Document Management?', *IDMi (Information & Document Management International)*, no. 58, pp. 26-

Davidson, P & Griffin, R 2006, *Management 3rd Australasian Edition*, John Wiley & Sons Australia, Ltd, Milton, Qld.

Gary P. Johnston, DVB 2005, 'The Benefits of Electronic Records management Systems : A general review of published and some unpublished cases', *Record Management Jornal*, vol. 15, no. 3, pp. 131 - 40 viewed 21/3/2011,

Maguire, R 2005, 'Lessons Learned From Implementing an Electronic Records Management System', *Record Management Jornal*, vol. 15, no. 3, pp. 150 - 7

Robertson, B 2004, 'Document control advances quality', *Quality*, vol. 43, no. 6, pp. 54-7

Ryan, D 2005, 'The future of Managing Electronic Records', *Record Management Jornal*, vol. 15, no. 3, pp. 128 - 30 viewed 21/3/2011,

Standards Australia 2002, AS ISO 15489.1-2002 : Records Management, General, Standards Australia, Sydney.

Standards Australia & Standards New Zealand 2008, ISO 9001, Quality Management Systems - equirements, Standards Australia, Sydney.

Want, C 2009, 'Bring Your Organisation up to Speed: The Benefits of Electronic Document Management', *IDMi (Information & Document Management International)*, no. 59, pp. 14-

Bibliography

Apple Inc 2007, 'Applescript Overview', Apple Developer Library, viewed 15/4/2011, <http://developer.apple.com/library/mac/#documentation/AppleScript/Conceptual/AppleScriptX/AppleScriptX.html%23//apple_ref/doc/uid/10000156-BCICHGIE>

Kong, CW, Li, Hr & Love, PED 2001, 'An e-commerce system for construction material procurement', *Construction Innovation (Sage Publications, Ltd.),* vol. 1, no. 1, pp. 43-54

Leonhard, EB & John, FT 1991, 'Vendor Analysis for Best Buy in Construction', *Journal of Construction Engineering and Management*, vol. 117, no. 4, pp. 645-58

Neuburg, M 2006, *Applescript: The Definitive Guide, 2nd edn*, O'Reilly Media Company, Sepastopol, California. viewed 16-march-2011.

Noorul Haq, A & Kannan, G 2006, 'AN INTEGRATED APPROACH FOR SELECTING A VENDOR USING GREY RELATIONAL ANALYSIS', *International Journal of Information Technology & Decision Making,* vol. 5, no. 2, pp. 277-95

Tagaras, G & Lee, HL 1996, 'Economic Models for Vendor Evaluation with Quality Cost Analysis', *Management Science*, vol. 42, no. 11, pp. 1531-43

The Open University 2011, *Models and Modelling*, M883_1 course notes. viewed 24/3/2011, <http://openlearn.open.ac.uk/mod/oucontent/view.php?id=397581&section=4.1

Zhang, Y & Cai, S 2005, 'The Coordination Mechanism Selection from Vendor Perspective: A Grey Relational Analysis Approach', *Journal of Grey System*, vol. 17, no. 3, pp. 257-62

Appendix A – Project Specification

FACULTY OF ENGINEERING AND SURVEYING

ENG4111/4112 Project

PROJECT SPECIFICATION

FOR: **DARRYL EVANS**

TOPIC: Simplifying Document Management and Quote Analysis

SUPERVISORS: Dr. David Thorpe
Toby Casley, Director Chilli Project Services

ENROLMENT: ENG4111 - S1, 2011
ENG4112 - S2, 2011

SPONSORSHIP: Chilli Project Services Pty Ltd

PROJECT AIM: The objective of this project is to develop a system to control the following

- Job registration and initial file management setup
- File management within each job
- Register and manage quotes from suppliers and subcontractors
- Quote comparison and basic analysis
- Calculation of parameters such as exposure, opportunity, spread, estimate adjustment figures and response rate
- Streamline the work process and increase efficiency.
- Close and archive old jobs

PROGRAMME: **Issue A, 11th March 2011**

1. Undertake a literature review of electronic document management systems, with particular reference to construction procurement systems.
2. Analyse the system requirements.
3. Design an appropriate system.
4. Develop a functioning prototype using standard software packages readily available to engineers.
5. Using selected projects, test the prototype.
6. Prepare and submit the dissertation to university requirements

As time permits:

7. Make necessary modifications
8. Implement the System

AGREED:

_____ (student) _____, _____

____/___/_____ ____/____/_____ ____/____/_____

Appendix B – Close Job Applescript Code

```
on run
	display dialog "Connecting to Server" giving up after 2
	try
		set return_value to true
		mount volume "Chilli Project Services"
	on error
		set return_value to false
	end try

	-- Compile list of job names

	set listOfNames to {}
	tell application "Finder"
		set joblist to every folder of folder "Current Projects" of
				folder "REGISTER" of disk "Chilli Project
				Services"
		repeat with currentFolder in joblist
			set currentFolderName to (the name of currentFolder)
			copy currentFolderName to the end of listOfNames
		end repeat
	end tell
	set closejob to (choose from list listOfNames) as text
	set theyear to (characters 1 thru 2) of closejob

	--Test if any report exists and choose correct versions

	set listOfpdfs to {}
	tell application "Finder"
		set filelist to (get every item in folder "06 Report" of
				folder closejob of folder "Current Projects" of
				folder "REGISTER" of disk "Chilli Project
				Services")
		if length of filelist = 0 then
			display dialog "You need to have a report on the
					server before closing"
			error -128
		end if
		set pdflist to every item in folder "06 Report" of folder
				closejob of folder "Current Projects" of folder
				"REGISTER" of disk "Chilli Project Services"
				whose name ends with ".pdf"
		repeat with currentFile in pdflist
			set currentFileName to (the name of currentFile)
			copy currentFileName to the end of listOfpdfs
		end repeat
	end tell
```

```
if length of pdflist < 2 then
	set pdfhyper to listOfpdfs as text
else if length of pdflist > 1 then
	set pdfhyper to (choose from list listOfpdfs with prompt
		"Select the Current Report Version") as text
end if
set listOfdocs to {}
tell application "Finder"
	set doclist to every item in folder "06 Report" of folder
		closejob of folder "Current Projects" of folder
		"REGISTER" of disk "Chilli Project Services"
		whose name does not end with ".pdf"
	repeat with currentdoc in doclist
		set currentFileName to (the name of currentdoc)
		copy currentFileName to the end of listOfdocs
	end repeat
end tell
if length of doclist < 2 then
	set dochyper to listOfdocs as text
else if length of doclist > 1 then
	set dochyper to (choose from list listOfdocs with prompt
		"Select the Current EDITABLE Report Version")
		as text
end if

-- Test if archive folder exists and create if it doesn't

tell application "Finder"
	if (exists ("Chilli Project Services:REGISTER:20" & theyear
		& " Closed Jobs")) = false then
		make new folder at folder "REGISTER" of disk "Chilli
			Project Services" with properties
			{name:("20" & theyear & " Closed Jobs")}
	end if

	-- Move the job to the closed job folder

	move folder closejob of folder "Current Projects" of folder
		"REGISTER" of disk "Chilli Project Services" to
		folder ("20" & theyear & " Closed Jobs") of folder
		"REGISTER" of disk "Chilli Project Services"
end tell

-- Update the job register status and hyperlink

tell application "Microsoft Excel"
```

```
        open "Volumes:Chilli Project Services:REGISTER:Job
                Register.xls"
        tell sheet "Sheet1"
            set closejobcell to find range ("A7:A1000") what
                    closejob
            clear contents of closejobcell
            set formula of closejobcell to
                    ("=hyperlink(\"file://localhost/Volumes/C
                    hilli Project Services/REGISTER/20" &
                    theyear & " Closed Jobs/" & closejob &
                    "/JOB SUMMARY" & closejob & ".xlsx\",\""
                    & closejob & "\")")
            set statuscell to get offset of closejobcell column
                    offset 9
            set formula of statuscell to "Closed"

            --  pdf report set hyperlink

            set pdfcell to get offset of closejobcell column offset 7
            clear contents of pdfcell
            set formula of pdfcell to
                    ("=hyperlink(\"file://localhost/Volumes/C
                    hilli Project Services/REGISTER/20" &
                    theyear & " Closed Jobs/" & closejob &
                    "/06 Report/" & pdfhyper & "\",\"" & "PDF"
                    & "\")")
            set doccell to get offset of closejobcell column offset
                    8
            clear contents of doccell
            set formula of doccell to
                    ("=hyperlink(\"file://localhost/Volumes/C
                    hilli Project Services/REGISTER/20" &
                    theyear & " Closed Jobs/" & closejob &
                    "/06 Report/" & dochyper & "\",\"" &
                    "Editable" & "\")")
        end tell
        close active workbook with saving
    end tell

    -- Update hyperlinks to quotes in job summary

    tell application "Microsoft Excel"
        open file ("Chilli Project Services:Register:20" & theyear & "
                Closed Jobs:" & closejob & ":JOB SUMMARY" &
                closejob & ".xlsx")
        tell sheet "SUPPLIER REGISTER"
```

```
            repeat with lineno from 6 to 55
                if value of range ("b" & lineno) is not "" then
                    set suppliername to get value of range ("b" &
                            lineno) as text
                    set oldpath to address of every hyperlink of
                            range ("b" & lineno)
                    set formula of range ("b" & lineno) to
                            ("=hyperlink(\"file://localho
                            st/Volumes/Chilli Project
                            Services/REGISTER/2010
                            Closed Jobs/10-TEST-017/"
                            & oldpath & "\",\"" &
                            suppliername & "\")")
                end if
            end repeat
        end tell
        tell sheet "SUBCONTRACTOR REGISTER"
            repeat with lineno from 6 to 55
                if value of range ("b" & lineno) is not "" then
                    set subbyname to get value of range ("b" &
                            lineno) as text
                    set oldpath to address of every hyperlink of
                            range ("b" & lineno)
                    set formula of range ("b" & lineno) to
                            ("=hyperlink(\"file://localho
                            st/Volumes/Chilli Project
                            Services/REGISTER/2010
                            Closed Jobs/10-TEST-017/"
                            & oldpath & "\",\"" &
                            subbyname & "\")")
                end if
            end repeat
        end tell
        close active workbook with saving
    end tell
    display dialog ("Job " & closejob & " has been closed")
end run
```

Appendix C – Update Folders Applescript Code

```
display dialog "Processing Data" giving up after 2
tell application "Microsoft Excel"
	tell sheet "Front Page" of active workbook
		set jobno to get value of range ("D4")
	end tell
end tell
set rowcount to 1

repeat with supcount from 1 to 10 by 1
	set rowcount to (rowcount + 5)
	if rowcount = 11 then set rowcount to 12
	tell application "Microsoft Excel"
		tell sheet "Supplier Register" of active workbook
			set sup to get value of range ("A" & rowcount)
			#if value of range ("A" & rowcount) = (not ("")) then
			#set Status to get locked of range ("A" & rowcount)
			#set locked of ("A1") to true
			#end if
		end tell
	end tell
	tell application "Finder"
		if not (exists ("Macintosh HD:Users:danevans:PROJECT
				SERVER:Current Projects:" & jobno & ":03
				Supplier Quotes:" & sup)) then
			make new folder at ("Macintosh
					HD:Users:danevans:PROJECT
					SERVER:Current Projects:" & jobno & ":03
					Supplier Quotes") with properties
					{name:sup as string}
		end if
	end tell
end repeat

set rowcount to 1

repeat with supcount from 1 to 10 by 1
	set rowcount to (rowcount + 5)
	if rowcount = 11 then set rowcount to 12
	tell application "Microsoft Excel"
		tell sheet "Subcontractor Register" of active workbook
			set sup to get value of range ("A" & rowcount)
		end tell
	end tell
	tell application "Finder"
		if not (exists ("Macintosh HD:Users:danevans:PROJECT
				SERVER:Current Projects:" & jobno & ":04
```

```
                    Subcontractor Quotes:" & sup)) then
                make new folder at ("Macintosh
                        HD:Users:danevans:PROJECT
                        SERVER:Current Projects:" & jobno & ":04
                        Subcontractor Quotes") with properties
                        {name:sup as string}
            end if
        end tell
    end repeat
    display dialog "Folders have been updated" giving up after 2
```

Appendix D – RFQ Applescript Code

```
display dialog "Gathering Data" giving up after 2

-- Check active cell is a company name

tell application "Microsoft Excel"
	set columncheck to (get address local of (get offset of active
		cell)) as text
end tell
set columnchar to character ((offset of "$" in columncheck) + 1) in
	columncheck
if not columnchar = "B" then
	display dialog "You must click on the company name first"
	error -128
end if

## Gather information from Register Sheet

tell application "Microsoft Excel"
	set vendortype to get name of active sheet as text
	set company to value of active cell
	set contact to value of (get offset of active cell column offset
		1) as text
	set email to value of (get offset of active cell column offset 3)
		as text
	set sheetname to (value of (get offset of active cell column
		offset 10) as integer) as text
	set jobno to value of range "D4" of sheet "FRONT PAGE"
	set status to (get offset of active cell column offset 7)
	set value of status to "1"

	## Gather Information From Vendor Sheets

	tell sheet sheetname of active workbook
		set package to value of range "H2"
		set lastRowCell to (get end range ("B" & (count rows))
			direction toward the top)
		set testrow to get address local lastRowCell
		set itemlist to get value of range ("B6:" & testrow) as list

		set testrow to get offset lastRowCell column offset 2
		set qtyrow to get address local testrow
		set qtylist to get value of range ("D6:" & qtyrow) as list

		set testrow to get offset lastRowCell column offset 1
		set unitrow to get address local testrow
		set unitlist to get value of range ("C6:" & unitrow) as list
```

```
            set testrow to get offset lastRowCell column offset -1
            set itemnorow to get address local testrow
            set itemnolist to get value of range ("A6:" & itemnorow) as
                        list
      end tell
      set listsize to count of itemlist
end tell

if vendortype = "SUPPLIER REGISTER" then
      set vendorfolder to "03 SUPPLIER QUOTES"
else if vendortype = "SUBCONTRACTOR REGISTER" then
      set vendorfolder to "04 SUBCONTRACTOR QUOTES"
else
      display dialog "Invalid selection for RFQ"
      error -128
end if
##copy the template RFQ document and rename it with the job
        number, sheet number and RFQ.


set packagename to characters ((offset of "-" in package) + 2) thru
        (count characters in package) of package as text
set newname to jobno & "_" & company & "RFQ.docx" as text

-- CHECK THAT ALL FOLDERS EXIST AND CREATE IF THEY DON'T
tell application "Finder"
      if not (exists (folder packagename of folder vendorfolder of
                  folder jobno of folder "Current Projects" of folder
                  "PROJECT SERVER" of folder "danevans" of folder
                  "Users" of disk "Macintosh HD")) then
            set thescript to "Macintosh
                        HD:Users:danevans:Documents:Uni
                        Stuff:2011:Files:Project Update Folders
                        revB.scpt:" as alias
            run script (thescript)
      end if

      -- CREATE WORD RFQ FROM TEMPLATE Extracting package
            folder name from excel VA sheet

      set rfq_path to folder packagename of folder vendorfolder of
                  folder jobno of folder "Current Projects" of folder
                  "PROJECT SERVER" of folder "danevans" of folder
                  "Users" of folder "Macintosh HD"
```

```
        set rfqdoc to ("Macintosh HD:Users:danevans:PROJECT
                SERVER:Current Projects:" & jobno & ":" & vendorfolder
                & ":" & packagename & ":" & newname)

        duplicate (file "RFQ.docx" of folder jobno of folder "Current
                Projects" of folder "PROJECT SERVER" of folder
                "danevans" of folder "Users" of disk "Macintosh HD") to
                rfq_path

        -- Version Control
        if exists rfqdoc then

            set rev to count (every file of rfq_path whose name
                    contains company)


            set name of file "RFQ.docx" of rfq_path to (jobno & "-" &
                    company & "RFQ_rev" & rev & ".docx")
            set rfqdoc to ("Macintosh HD:Users:danevans:PROJECT
                    SERVER:Current Projects:" & jobno & ":" &
                    vendorfolder & ":" & packagename & ":" & jobno
                    & "-" & company & "RFQ_rev" & rev & ".docx")

        else
            set name of file "RFQ.docx" of rfq_path to newname

        end if

        open file rfqdoc
    end tell

    tell application "Microsoft Word"
        ## fill in extracted fields
        insert text (jobno & " " & sheetname) at word 4 of paragraph 2
                of active document
        insert text company at word 3 of paragraph 3 of active
                document
        set rfqdate to (current date) as text
        insert text rfqdate at word 3 of paragraph 4 of active document
        insert text contact at word 2 of paragraph 9 of active document

        ## Make and populate Table
        make table at paragraph 12 of active document with properties
                {number of rows:(listsize + 6), number of columns:5,
                alignment:align row center}
        set mytable to table 1 of active document
```

```
        set content of text object of cell 1 of row 1 of mytable to "Item
                No."
        set content of text object of cell 2 of row 1 of mytable to "Item
                Description"
        set content of text object of cell 3 of row 1 of mytable to
                "Quantity"
        set content of text object of cell 4 of row 1 of mytable to "Unit"
        set content of text object of cell 5 of row 1 of mytable to "Unit
                Price ($)"

        set rowcount to 2
        repeat with rowcount from 2 to listsize + 1
            set content of text object of cell 1 of row rowcount of
                    mytable to item (rowcount - 1) of itemnolist
            set content of text object of cell 2 of row rowcount of
                    mytable to item (rowcount - 1) of itemlist
            set content of text object of cell 3 of row rowcount of
                    mytable to item (rowcount - 1) of qtylist
            set content of text object of cell 4 of row rowcount of
                    mytable to item (rowcount - 1) of unitlist

        end repeat
        auto fit every column of mytable
        set bold of text object of row 1 of mytable to true

        close active document with saving
end tell

-- Generate username and password
        (http://hintsforums.macworld.com/showthread.php?t=97903)
set username to ((characters 1 thru 3 of company) & (characters 5
        thru 7 of package)) as text
set password_chars to
        "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWX
        YZ1234567890"

set password_chars_count to number of characters of
        password_chars
set password_string_count to number of characters of username

set seeded to false

set password_encrypted to ""
set next_num to 0

repeat with char_num from 1 to password_string_count
```

```
        set ascii_num to ASCII number (character char_num of
                username)

        if (seeded is false) then
            set random_num to random number from 1 to
                    password_chars_count with seed 12345
            set seeded to true

        else
            set random_num to random number from 1 to
                    password_chars_count
        end if

        set next_num to ((next_num + ascii_num + random_num)
                mod password_chars_count) + 1

        set password_encrypted to password_encrypted & (character
                next_num of password_chars)

end repeat

-- Store login and password info on VA sheet

tell application "Microsoft Excel"
        set logcell to (get offset of active cell column offset 12)
        set value of logcell to username
        set passcell to (get offset of active cell column offset 13)
        set value of passcell to password_encrypted
end tell

-- Create mail message and attach above RFQ
set theAttachment to alias rfqdoc
tell application "Mail"
        set theMessage to make new outgoing message with properties
                {visible:true, subject:"Request for Quote",
                content:("Dear " & contact & "," & return & return &
                "Please complete the attached Request For Quote sheet
                and return it via email. The relevent documentation
                including scope of works and drawings can be found at
                the link below. Please use the username and password
                provided to access these documents" & return & return
                & "Username: " & username & return & "Password: " &
                password_encrypted & return & return)}
        tell theMessage
            make new to recipient at end of to recipients with
```

```
                    properties {address:email}
    end tell
    tell content of theMessage
        make new attachment with properties {file
                    name:theAttachment} at after last paragraph
    end tell
    activate outgoing message
end tell
```

Appendix E – Return RFQ Applescript Code

```
-- Connect to server
display dialog "Connecting to Server" giving up after 2
try
	set return_value to true
	mount volume "Chilli Project Services"
on error
	set return_value to false
end try

-- READ JOB NUMBER AND SHEET NUMBER FROM RETURNED RFQ
tell application "Microsoft Word"
	set refno to get content of words of paragraph 2 of active
		document as text
	set companyid to content of words of paragraph 3 of active
		document as text
	--GET UNIT PRICES FROM RFQ
	set mytable to table 1 of active document
	set quotelist to {}
	set itemcheck to {}
	repeat with countrow from 2 to count rows of mytable
		set prices to (get content of text object of cell 5 of row
			countrow of mytable) as text
		set returnitem to (get content of text object of cell 2 of
			row countrow of mytable) as text
		-- REMOVE THE CARRIAGE RETURN FROM CELLS IN THE
			LIST
		if (count (word in prices)) > 0 then
			set listprice to words 1 thru ((count (word in prices)))
				of prices
			set RFQitem to (characters 1 thru ((count (characters
				in returnitem)) - 2) of returnitem as text)
		else
			set listprice to {""}
			set RFQitem to ""
		end if
		copy listprice to the end of quotelist
		copy {RFQitem} to the end of itemcheck

	end repeat
end tell

-- MAKE CORRECT FORMATTING OF INFORMATION
set company to characters ((offset of ":" in companyid) + 2) thru
	((count characters in companyid) - 1) of companyid as text
set jobno to characters ((offset of ":" in refno) + 3) thru ((offset of
	":" in refno) + 13) of refno as text
```

```
set sheetno to last word of refno as text
set sheetnointeger to sheetno as integer

-- PUT PRICES INTO QUOTE SHEET

tell application "Microsoft Excel"
	open file ("Macintosh HD:Users:danevans:PROJECT
		SERVER:Current Projects:" & jobno & ":JOB SUMMARY"
		& jobno & ".xlsx")
	if sheetnointeger < 26 then
		set sheetname to "SUPPLIER REGISTER"
		set savefolder to "03 Supplier Quotes"
	else
		set sheetname to "SUBCONTRACTOR REGISTER"
		set savefolder to "04 Subcontractor Quotes"

	end if
	set savepath to ("Macintosh HD:Users:danevans:PROJECT
		SERVER:Current Projects:" & jobno & ":" & savefolder &
		":")
	tell sheet sheetname
		set quotecolumn to find range "B6:B100" what company
		set quotecell to value of (get offset of quotecolumn
			column offset 11) as text
		set status to (get offset of quotecolumn column offset 7)
		set value of status to "4"
	end tell
	tell sheet sheetno
		repeat with countrow from 1 to count items of quotelist
			set formula of range (quotecell & countrow + 5) to
				item countrow of quotelist
		end repeat
		set packageid to get value of range "H2"

		set itemlist to get value of range ("B6:" & "B" & 5 +
			(count items of quotelist)) as list

	end tell
end tell
set packagename to characters ((get offset of "-" in packageid) +
	2) thru (count characters in packageid) of packageid as text
set theoutpath to (savepath & packagename & ":" & jobno &
	company & "RFQ_RET.docx")

--Version control and record keeping
tell application "Finder"
```

```
        if exists theoutpath then
            set rev to count ((every file of folder packagename of
                    folder savefolder of folder jobno of folder "Current
                    Projects" of folder "PROJECT SERVER" of folder
                    "danevans" of folder "Users" of disk "Macintosh
                    HD") whose name contains (company &
                    "RFQ_RET"))

            set theoutpath to (savepath & packagename & ":" & jobno
                    & company & "RFQ_RET_rev" & rev & ".docx")
        end if
end tell

tell application "Microsoft Word"
        save as active document file name theoutpath
end tell
if not (itemcheck is equal to itemlist) then
        display dialog "RFQ Table has been altered. Check quote
                manually"
else
        display dialog "Quote has been entered into the system"
end if
return {itemcheck, itemlist, quotelist}
```