Teaching Case
Analysis of an Electronic Voting System

Nik Thompson
Murdoch University,
Perth, Western Australia
n.thompson@murdoch.edu.au

Danny Toohey
Murdoch University,
Perth, Western Australia
d.toohey@murdoch.edu.au

ABSTRACT

This teaching case discusses the analysis of an electronic voting system. The development of the case was motivated by research into information security and management, but as it includes procedural aspects, organizational structure and personnel, it is a suitable basis for all aspects of systems analysis, planning and design tasks. The material is based on real life analysis of currently used electronic voting systems, which have been generalized so as to highlight the wider issues and to not identify with any particular implementation of electronic voting. Suggested project deliverables are described in the teaching case, and these are complemented by the associated teaching notes which detail sample solutions and discussion points for class.

Keywords: Systems analysis and design, Face-to-face teaching, Teaching Case

1. INTRODUCTION

A variety of teaching tools are suitable for Information Systems education including problem-solving exercises, collaborative projects, role playing and case studies (e.g., Arling, Deeter and Eggers, 2010, Bee and Hayes, 2011). A key element of systems analysis and design is developing the ability to distil relevant facts surrounding an issue, formalize this understanding into a coherent and useful structure, and finally to communicate this understanding to others in the form of models, diagrams or reports. Case studies, in particular are widely used in systems analysis as they facilitate this understanding and communication by promoting active learning (Meyers and Jones, 1993). Indeed, the case study approach has proven to be a suitable framework within which to develop valuable skills, while also maintaining a level of consistency across students or groups that would not be possible were they to choose individual topics. Cases can enable students to develop their higher order skills in a way in which they can transfer their theoretical knowledge to practical real-world situations (Hackney, McMaster, and Harris, 2003).

2. BACKGROUND

The case is based on a study of electronic voting systems conducted by the authors. This is an area that, over the years, has received substantial coverage in both the research literature and popular media. The reliance on such technologies brings a large amount of scope for discussion and analysis. This case was chosen as the topic of this paper as it has been successfully used in several classes to date, and meets all of the teaching case development criteria laid out by Cappel and Schwager (2002). It is also a topic that students have found interesting, as it highlights the real world applicability of their systems analysis skills in a variety of problem domains.

The case described in this paper has been previously used as an individual major assignment in a semester long undergraduate Systems Analysis and Design course. This course is required in both Information Systems and Computer Science degrees. Students may follow this course with an optional Advanced Business Analysis course that focuses more heavily on business processes and modeling techniques such as BPMN. The assignment contains a number of deliverables designed to address the main learning objectives of the course, which include project management and scheduling, data modeling, process modeling and object oriented techniques. Students work independently and individually on this assignment, and while they are permitted to work on the tasks in class and consult with their tutors, the formal submission of all deliverables is done at the same time.

The assignment places an emphasis on the analysis tasks, as this is consistent with the weighting of topics taught in class. However, the case and the deliverables are also well suited for design assignments as the logical progression from the analysis deliverables is to translate these into some form of design. Instructors may even wish to use this as an entirely systems design-based assignment by providing students with the completed data and process models and tasking them with the construction of a working system to support some aspect of the functionality described.

This teaching case is based on the analysis of an electronic voting system. The concepts, concerns, and indeed some aspects of the functionality described are based on reality. Due to the scrutiny that this technology has received in the media and research, a number of high quality research projects have investigated the electronic voting systems provided by specific manufacturers. This research, in particular the work of Kohno, Stubblefield, Rubin and Wallach (2004), has been an indispensable resource in guiding the development of this case into something that closely resembles a real world situation.

The text of the case is presented in the following section; this includes some background information relating to the topic of electronic voting followed by the description of the election process. The remainder of the paper lists deliverables and tasks that may be based on the case for analysis or design, while the associated teaching notes elaborate further on potential solutions for the deliverables.

3. CASE OVERVIEW

AccuVote Inc. is a medium size enterprise specializing in the development and implementation of electronic voting systems. The company was founded in 2000 by Tobias Jones, then a final year university student studying computer engineering. Tobias had noted that electronic voting systems had been in use since the 1960's when they implemented rudimentary punch-card systems, and in spite of the vast leaps that technology had made since then, the current systems were still outdated. When he heard about the changes that would be required due to the proposed Help America Vote act, he saw this as an opportunity to turn his ideas into a commercial product that would also provide a benefit to society.

In addition to the replacement of the outdated punch-card systems, the Help America Vote act required that certain minimum levels of accessibility would need to be met to enable handicapped persons to be able to cast their votes easily. This gave further support to the use of electronic voting systems, as the use of touchscreens, audio prompts and other assistive technologies were already making their way into the market at that time.

Tobias founded the company with a fellow engineering student, George Chen, using a very modest industry grant of $5,000 that they obtained through their university research office. At this point they concentrated mostly on the software side of their system, working on developing the software required to count, analyze and report votes. Their aim was to use this to showcase their abilities and generate more revenue to start working on construction of the actual voting terminals.

Over a decade later, AccuVote currently employs 50 staff members who work mainly in development and design roles. There is very little in the way of marketing required, as the company operates on existing long term contracts with government or industry bodies for the supply and maintenance of entire voting systems. The construction of the voting terminals is outsourced to one of three manufacturing facilities, two of these are in China and one is in the USA. Thus AccuVote does not have any of their own manufacturing facilities or factories which saves on capital investment. To ensure that their strict quality assurance (QA) conditions are met, AccuVote QA experts personally supervise the processes at the manufacturing facilities during construction of the terminals after which they are shipped back to the AccuVote main office for final testing. This testing is carried out on two main levels. Firstly, system tests are conducted at the AccuVote main office during which the entire network is configured and run in a simulated environment. Secondly, user acceptance tests and stress tests are carried out after the system has been deployed into its final real world implementation.

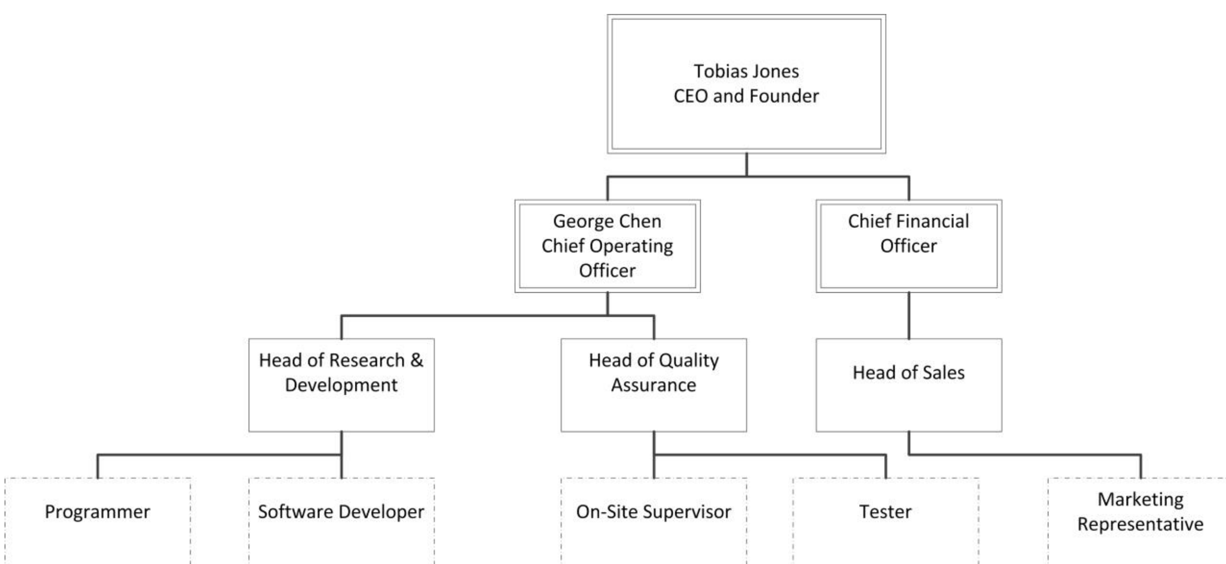

Figure 1 AccuVote Organizational structure

The organization has a relatively informal structure, with no rigid hierarchies. The development and design is considered to be more of a team effort and staff do not adhere to a fixed set of assigned activities. The organizational structure is detailed below – where there are multiple positions, only one has been shown for conciseness.

AccuVote has enjoyed steady growth in staff numbers as well as revenue since its inception in 2000, a fact of which the founders are very proud. They attribute this primarily to their cutting edge technology and superior products, but also from simply being in the market at the right time to gain a foothold. Electronic voting systems are now in widespread use, and have received a lot of public attention in recent years. This attention includes both good and bad publicity and there is concern that if some of the negative press associated with electronic voting systems is not addressed that it could harm the future of the company. Studies of similar electronic voting systems have revealed the potential for several security vulnerabilities. These vulnerabilities generally impact either the use of the voter smartcards or the stored election data on the terminal.

Research has indicated that in many cases, smartcards do not perform any encryption in their default configuration. This is a significant weakness, and undermines one of the major advantages of smartcard technology as opposed to simpler magnetic stripe technology. The lack of encryption casts doubt over the authentication process, and implies that potentially untrusted (or counterfeit) smartcards may be used, assuming that it is possible to find out the details of the communications/smartcard protocol used. This may be done via privilege misuse on the part of election staff, or by attempting to replicate an actual voter card on the day of voting. The extent of the damage caused by an attack will be governed by the level of access that the counterfeit smartcard provides. In the most basic attack, a regular voter card may be copied allowing an attacker to cast multiple votes.

With this background in mind, the CEO, Tobias Jones, has commissioned an external consultant to carry out a systems analysis of the voting process. He has made himself available for face to face meetings and consultation during this process as it is his priority to ensure that any potential weaknesses or security vulnerabilities in the systems that his company develops should be addressed and should hold up to external scrutiny. Your job as a Systems Analyst is to conduct this investigation as requested. The first step is the kick-off meeting between your colleague, Mark Roberts, and Tobias Jones, the Client, to learn more about his requirements. The transcript is given in the following section.

3.1 Project kick-off meeting

Scene: *Mark Roberts, systems analyst, is meeting with Tobias Jones, CEO of AccuVote, at his office, Room 456, in Building 314 at the AccuVote Head Office in Harrisonburg, Virginia. Mark scheduled the interview with Mr. Jones in response to his request for a systems analysis of his electronic voting systems.*

Tobias: Good Morning, you must be Mark.

Mark: Yes I am sir, it's a pleasure to meet with you Mr Jones.

Tobias: Please call me Tobias. I'm glad we could get together at such short notice as this is quite a sensitive task I have requested you to undertake.

Mark: Why is that?

Tobias: Well there is a growing amount of public mistrust and concern about voting systems, partly due to a few incidents reported in the mainstream media. The closest to home is of course the investigation into the electronic voting machines implemented in Fairfax, Virginia. While this is old news as it occurred around a decade ago, this was the first impression that many people got about this technology, and still casts a shadow over subsequent implementations. If we look further afield, there have been documented issues with electronic voting in other states such as Florida and California, as well as overseas in Europe. All of these bad press incidents have the potential to be harmful to our organization.

Mark: I see, why don't we start at the beginning then so that I can get a feel for the situation. Could you tell me more about electronic voting systems in general?

Tobias: Yes of course. When people think of more old-fashioned or "traditional" approaches to voting they are usually thinking of something like punch-card technology which requires that voters punch a hole in a pre-printed card to make their selection. Alternatives may include optical mark recognition (OMR) in which the voter makes marks in ink on a ballot form, or other forms of paper ballots. In actual fact all of these "traditional" voting methods are still supported by electronic mechanisms. We take this to the next logical level and simply do away with the paper or card and allow voters to directly input their choice into the voting terminal.

Mark: That sounds quite straightforward; I wonder why people have concerns about this?

Tobias: Well, I think it stems from the fact that while older approaches also rely heavily on electronic mechanisms for processing, the existence of a physical record of a vote means that there is a strong audit trail and an increased level, or at least perception, of integrity.

Mark: Ok, so when you say electronic voting system you really mean "paperless"?

Tobias: Yes, our definition of the term "electronic voting" refers to an entirely electronic, paperless ballot system. Votes may be cast at special terminals, using a touch screen or other interface, and the votes are recorded digitally and later transmitted and collated by the voting administration body. Commonly cited advantages of this type of approach include the fact that there is less chance of errors due to increased automation, the possibility to instantly know results without a lengthy counting process, and that any voter could vote at any convenient location.

Mark: Plenty of advantages then, could you tell me more about the perceived disadvantages?

Tobias: This has been the subject of substantial discussion and mistrust, as it is often perceived by the public that they are losing control over the most important step of the process – that of counting and handling the votes.

I believe that the concern largely originates from the belief that while traditional voting procedures have been heavily scrutinized over the years, the use of a new system may introduce vulnerabilities and weaknesses into what is an otherwise robust and dependable process. The lack of a physical audit trail (of paper ballots) also implies that if the electronic record of votes is somehow damaged or compromised, that this may be irreparable, or worse still, go undetected.

To cap it off, there has been a lot of media attention regarding the potential weaknesses in electronic voting systems resulting in a number of detailed studies being conducted to investigate and disassemble the voting systems offered by various manufacturers. In many cases, these studies have revealed very substantial vulnerabilities in the electronic voting systems produced by our competitors. So this has put the spotlight on all manufacturers now, and not just those with known issues.

This is why I am requesting an external analyst to study and report on our systems. What I am mostly interested in is exploring what potential there is for insecurities in electronic voting to creep up on us. These could possibly be procedural things about how the process is conducted, or they could be technical issues regarding the way we are implementing our technologies. I have worked with our lead engineer to develop the following process documentation (Accuvote Process Documentation) which describes the procedural aspects of the voting system and the concerns that we have. Please refer to it to guide your analysis.

Mark: Thank you so much for your time. I will contact you if I require further information.

3.2 AccuVote Process Documentation

This documentation describes the process of setting up and running an election using an AccuVote electronic voting system. As clients generally receive a customized solution for their needs these documents are not intended to directly replicate any particular system in use. Therefore it is essential to base any analysis decisions solely on the information provided in this case and not attempt to describe an actual implementation. An effort has been made to provide sufficient detail to support the subsequent analysis tasks, however if in any situations there is insufficient information then assumptions may have to be made. These assumptions should be documented and clearly stated in the deliverables.

The system under discussion uses individually coded smartcards to identify valid users. Voters or election officials who wish to interact with the voting machine are given these cards which uniquely identify them. Another element of the setup is the process of defining the issues to be voted upon and the options that are presented to the users. The election officials must specify the list of political offices and issues to be voted upon, as well as creating the list of candidates and their party affiliations. The Voter may also have a nominated party affiliation, and based on this he or she will be presented with a variation of the ballot. This background setup information is known as a *ballot definition*.

Having created the ballot definition, the voting equipment must be setup and configured in each of the polling offices where voting will take place. The ballot definitions must also be distributed to these locations in a secure manner. The integrity of the entire election could be compromised if this ballot definition was in some way corrupted.

On the day of the election, the voting terminals must be started up by an election official. This involves checking that the ballot definition was installed correctly before starting the actual election and allowing voters to cast their votes. As the administration is centralized, voters may attend any convenient polling office. Upon reaching the polling office, the voter must present a valid form of photographic identification to the election official. Valid forms of identification include driver's license, passport, proof of age card or any other state-issued photographic identification. After this has been verified, the voter is handed a smartcard which he or she may use to interact with the machine. This credit card sized plastic laminated card contains an embedded computer chip to store data, and is known as a *voter card* in this context. These voter cards are reprogrammed for each use, and returned to the election officials after the vote is cast so that it may be erased and used again.

In addition to the regular voter cards, there are also administrator cards and finalize cards, which have additional capabilities within the voting system. The former allows access to administrative functionality (such as copying and archiving votes) and both of these classes of cards permit the user to end the election process if used in conjunction with a simple 3 digit PIN.

The voting terminal has a smartcard reader in its front panel. This operates in the same way as the commonly used automated teller machines. The voter must follow the on screen dialogue and insert the voter card into the card reader on the terminal. The terminal performs some checking to ensure that the card is a valid voter card and that it has not been previously used before proceeding.

The voting screen is then presented; this is simply an on screen ballot form, upon which the user may select his or her options using the touch screen. As noted above, the voter may have nominated a specific political party preference, and if this is detected on the voter card the terminal will display a customized ballot form. The party preference nomination is not a mandatory component, and if this is not detected on the card then the machine will simply present a generic ballot form to the voter.

The voter may interact with the terminal using the touch screen to tick the boxes corresponding to his or her chosen candidates/options. Other interaction media are available for visually impaired voters, including headphones and keypads which are installed on all voting terminals. When the voter indicates that he or she has completed making the selections, a summary screen is presented in which the voter is asked to review his or her selections before committing the votes to storage. The final commit step is, of course, non-reversible. Once the votes are committed, the voter card is automatically cancelled by the machine, the user is then presented with a printed receipt and the card is returned. The step of cancelling the voter card ensures that the voting cards are not inadvertently (or intentionally) reused, while also finalizing the voter's interaction with the system. After this step the terminal is ready for another voter to use. The voter returns the cancelled card to the election official so that it may be reprogrammed for another user.

During its normal operation, the voting terminal stores log files which provide an auditable trail of activity. Summary reports are created automatically on an hourly basis and these are stored on the voting terminal for later transmission or viewing by the election officials.

At the end of the voting period, the election must be formally closed. A poll worker may do this by inserting a specially coded *finalize card* into the machine. This card is only used to end the election. When the machine detects the insertion of this card it enters into the finalize stage in which additional identification and authorization PIN codes are requested before proceeding. If this information is entered correctly then the terminal prompts for confirmation before proceeding to commit any pending data transactions to permanent storage, closes any open files and packages the vote data into a single archive file. This archive file of votes may be written to removable storage, or directly transmitted to a networked central server depending on how the terminal is configured. The central server will collect these archives from the various voting terminals to collate the separate data files, error check and create a data file containing a detailed summary and log of all events and results of vote counts. This file will remain stored locally on the central server from which the (authorized) Election Officials may view it or take a copy for public reporting.

4. SYSTEMS ANALYSIS ASSIGNMENT

This section details the suggested deliverables associated with the case. These include tasks for scheduling, data modeling, process modeling and object oriented techniques. The case is given to students close to the start of semester, with the intention that the students read and familiarize themselves with the content and structure of the case study before the specific skills required to complete the deliverables are taught.

The course is taught in a lecture plus tutorial format with 12 teaching weeks spread across a 14 week semester. In each teaching week there is a two hour lecture and a two hour tutorial session. Lectures are delivered to the entire group of enrolled students in a traditional large-group format, with the presentation covering the background and theoretical aspects of a new topic each week. Small in-class exercises and practical tasks are also conducted in lectures but due to the large group format, this is all done in a group work setting with a lot of discussion and interaction between students. The weekly tutorial sessions are where the majority of the practical work takes place as these sessions are limited to 15 participants each to facilitate more individual attention. The tutorial format includes a small amount of revision of lecture material with the remainder of the session devoted to practical tasks and putting the skills into practice. It is during these tutorial sessions that the teaching case or assignment would be discussed.

Deliverables in the case follow a similar order and pattern to the topics taught in the course. Therefore it is possible for students to apply their newly developed skills each week on the relevant section of the case. For example, the Project Management tutorial would include discussion of scheduling, feasibility and problem analysis and would have a series of in-class practical tasks. The instructor may then discuss how these tasks relate (or overlap) with the Project Management deliverables for the teaching case and allow students to independently work on their own assignments.

Students are also given the opportunity to present their completed deliverables to the instructor before the formal submission. This allows them to gauge their progress and gain valuable feedback on areas that may need further development before the actual submission. This has proved to be quite beneficial to students as it enables a more "formative" approach to assessment. By gauging the students' progress early it is possible to more appropriately identify and respond to their individual needs. From the students' perspective, this also means that there are no nasty surprises as there is some assurance that they are on the right track and have not misunderstood any of the questions. Employing this approach to assessment has in our opinion improved the overall quality of teaching in the course. This, and other techniques for formative assessment are well documented in Education literature and instructors may be interested to read Nicol and Macfarlane-Dick's (2006) discussion of good feedback practices.

There are a number of suggested deliverables associated with this case, for reference a brief summary of all deliverables is presented below.

Project Management

a) Develop a project outline or statement of work the problem description. This must contain the anticipated benefits or outcomes of the project and the scope or capabilities of the proposed solution.

b) Create an initial plan and project schedule. This should include task breakdown, estimated start and end dates and be submitted as Gantt or PERT chart.

Problem Analysis

a) Perform root cause analysis. This should include an Ishikawa diagram analyzing groups of problems leading to the issues and be accompanied by explanation or class discussion.

Use Case modeling

a) Develop a list of use cases. This should be submitted in a tabular format with use case name, actor and 1-2 line descriptions.

b) Translate this into a use case model diagram. UML notation must be used, and consideration given to any subsystems or opportunities for reuse where appropriate.

c) Document a detailed use case description of the "Place Vote" use case. The brief use from Part a should be expanded now, by including flow of activities, other actors , exception conditions and presented in a tabular form using the class template.

Process Modeling

a) Create a Context Data Flow diagram. This should illustrate the scope and boundaries of the system and include all external agents.

b) Elaborate further by developing a Diagram 1. This must also include data stores.

c) Optional decision trees or activity diagrams may supplement this section if additional focus on process modelling is desired.

Data Modeling

a) Construct a list of the main data entities. This should be presented as a tabular listing showing entity name and 1-2 line description. Most of the data requirements and sources of data will have been identified in previous steps, so this deliverable is a step toward providing a logical model of the data.

b) Create an ERD relationship diagram (ERD). This should provide a logical model of the data for the voting data. It must show all entities, attributes, relationships and cardinality. Primary and foreign keys must be clearly labeled in the diagram.

The following sub-sections individually discuss these main deliverables in the teaching case.

4.1 Project management

The first phase of any project often considers a broad view of the requirements, root causes of any issues and, of course, project management related tasks including scheduling and feasibility. This initial set of deliverables is derived from the basic project management concepts taught in class. In this course, the emphasis is more strongly on systems analysis with only one session devoted to project management. Therefore the tasks are targeted at main principles, and are appropriate for the level of proficiency that the students may possess at this time. These tasks include the development of a project outline/statement of work and the development of scheduling models for time and resource tracking/allocation. The first task is for students to write a brief project outline (sometimes known as statement of work) to accompany their final report. There are three components to this document: the problem description, the anticipated benefits or outcomes of the project and the scope or capabilities of the proposed solution. The format and content of this requested documentation requested is based on the type of project being undertaken. For instance, this project is about conducting an in depth analysis of a system, so there will be less emphasis on a "proposed solution" as compared to a project that was to develop an item of software. While this is not meant to turn into a technical writing course, students should be made aware of the value of high quality documentation and the fact that this is also highly regarded by potential future employers. Students also generally progress onto more advanced software development and design projects, such as the final-year capstone project. Documentation and communication skills are weighted quite strongly in such projects, and students who develop these skills earlier on often prove to be the highest performers in these advanced projects.

Students are next asked to plan and schedule their project. As the deadline for project completion is something fixed (i.e. before the election), the students' scheduling model must be based on this. Task breakdown should be sensible and feasible and should be presented as a Gantt chart and/or PERT chart. Students should be deterred from simply rehashing generic SDLC phases as one-size almost certainly does not fit-all! To ensure that students produce a sensible work breakdown additional guidance can be given in class. This particular topic has also proved to be very well suited for interactive class discussions as students enjoy sharing their own experiences and approaches to time management and organization. Students may also consider dependencies and how to allocate resources to these tasks based on any information given to them in class.

As a class discussion point, the instructor may ask the class to consider a scenario and assess what type of scheduling may be appropriate to use. The follow up question is to then ask the students to consider their own techniques, in particular their study techniques and how they schedule their study plan and work for assignments. The concept of reverse-scheduling (i.e. working back from the due date) is almost always the response received from the class; however further discussion yields interesting insights into how the students weigh up priorities and estimate the time and effort required for individual job elements.

4.2 Problem analysis

The next task encourages students to consider the root causes of any issues rather than devoting their attention to (potentially superficial) symptoms of the issues. The Ishikawa or fishbone diagram is ideal for this kind of brainstorming and also forms a very popular class exercise. The project will ultimately address and understand these problems, but an analysis of the causes will help to shed light on this during the initial stages. An Ishikawa diagram is suitable for this high level problem analysis (Ishikawa & Loftus, 1990). Students will then be encouraged to identify the *groups* of problems that are contributing as well, rather than simply generating a "to-do" list of things to remedy. Instructors may wish to give students some guidelines on the classes of problems for which they should be looking, or simply look at a few of the commonly used Ishikawa templates in order to derive a useful set of groups for the given problem domain. Those commonly used in industries such as manufacturing may be a useful starting point for students (e.g. Manpower, Machine, Method, or Who, What, Why, When, Where). To aim for some consistency, a template or partially filled out example may be provided to students. Please refer to the Teaching Notes for a blank template which is suitable for printing on A4 to discuss and annotate in class if required.

4.3 Use Case modeling

Use case modeling is a valuable tool as it focuses on the users of the system rather than the system itself. Thus, the real needs and necessary functionality of the system is identified at an earlier stage. Deliverables for object oriented techniques may include use case diagrams and detailed use case descriptions. The details of the case specify a number of tasks that must be supported by the system, these relate to the overall setup of the election, election-day operations, and finally collation and reporting of results. A number of distinct stakeholders (actors) have been identified as well.

The first task should therefore be to provide a list of use cases. While this may appear to be a trivial task to the experienced analyst (especially here as this information is quite well presented in the case itself), it is a very good starting point for students to grasp the scope and extent of their future work. For each use case, students must provide a brief description and identify participating actors. Assumptions may be made where necessary, although these must be clearly stated in the document. Most of the use cases and their descriptions and actors should be able to be identified from the case. Some students may identify other use cases based on their prior knowledge. These are acceptable too, but not necessary for full marks.

To develop the documentation and presentation skills further, this information is submitted in several forms. Having created the tabular listing of use cases in the previous section this information should then be translated into a use case diagram. It should show use cases and the actors that initiate the use cases, with use cases grouped into several likely subsystems if appropriate. The instructor may wish to nominate one or more of these use cases to be studied in further detail.

The final step in the use case modeling deliverables is to submit a detailed use case description for the selected use case. This provides a more challenging task in which the students must consider the perspective of the user and analyze and document the flow of events that take place during a particular use case. In this instance, the *Place Vote* use case was selected as it was considered to be central to the whole scenario. As a guideline, students follow the sections detailed in Satzinger, Jackson and Burd (2008). This is a useful starting point as it reminds students of the kind of detail that is required for this task. A copy of the use case template given to students is included in the associated Teaching Notes.

At all stages in this and other deliverables, students are reminded that outcomes of analysis are very often different depending on the perspective taken. While it is important to be unbiased and base documentation only on the facts given, at times it is necessary to make assumptions about the scenario. This is more common when basing the analysis on a teaching case, rather than a real world situation in which it might be trivial to find out more detail where required. Students are told that it is entirely acceptable to make assumptions, as long as they are indeed necessary, and that they are clearly and fully documented. The reader should never have to second-guess the intentions of the analyst.

4.4 Process modeling

The data flow diagramming technique is a fundamental technique used in the traditional approach to systems analysis. In the past, this technique has been quite heavily weighted in teaching in this course. It is believed that this does not reflect the necessity of industry, and consequently the data flow diagramming tasks have been somewhat reduced in recent years. However, this is not to be excluded altogether. At a minimum, a context level data flow diagram gives students an insight into the scope and boundaries of the system, while developing their ability to represent their understanding of the system in a variety of different forms.

The process modeling aspect of the case involves the creation of data flow diagrams to model the entire electronic voting system as described. The body of the case details the organization of the voting system and the external entities that provide data; this is sufficient data with which to create a context DFD and Diagram 1 for the electronic voting system. Students must therefore document their understanding of this system using the DFD techniques taught in the course. This should provide a clear view of the scope and boundaries of the system, as well as providing a basis for visualization of the data processing requirements.

It is worth noting that most process modelers will exclude data stores from a context level diagram, as these are expected to be included within the system itself. In this course students were encouraged to consider the logical separation of data and processing, and therefore to still consider what data stores may exist, even if they are only being asked to construct a context level DFD. This is useful information later on if a Diagram 1 is also being constructed, as this would require the students to identify data stores. Instructors may wish to follow their own preferred approaches to these techniques based on their own requirements and course.

This case is pitched at a higher level for the problem analysis tasks, however there are also a number of additional techniques such as decision trees, or activity diagrams/flowcharts which may be useful for capturing more detail of the processes being undertaken and instructors may wish to incorporate these techniques if they find it appropriate to support their own desired learning outcomes.

4.5 Data modeling

Having identified most of the data requirements and sources of data from the previous analysis tasks, the next step is to construct a list of the main data entities and entity relationship diagram (ERD) that provides a logical model of the data for the voting data. It must show all entities, attributes, relationships and cardinality. Primary and foreign keys must be clearly labeled in the diagram.

The first deliverable is a tabular listing of the main entities discussed in the case. Students may find that the entities are quite similar for either an electronic or a traditional voting system, so they may use their own understanding of how votes are cast to help in the data modeling process, or supplement their understanding with additional readings elsewhere. However, as with previous sections, it is mandatory that if any assumptions are made they must be documented. Differences in student answers can be based on differing assumptions; an interesting in-class exercise is to take some of these assumptions and discuss how they will change the model. The submitted tabular listing may follow any format that the instructor finds appropriate, and at a minimum should include a brief description and suitable name for each entity.

From our experience in teaching both data and process modeling courses, students may sometimes become confused between data and process modeling; this confusion often results in a poor analysis of the system. As such, it is worth the instructor emphasizing that the ERD addresses the *data requirements* of the system, *and not the processes that act on the data.*

5. CONCLUSION

This paper has introduced a teaching case based on a timely and relevant real world scenario. The case has been used in an undergraduate level systems analysis and design course on several occasions, with approximately 200 students at 3 campuses. Students and instructors have found this to be a very useful case, and of adequate length and complexity to match the skill levels of students, while having enough flexibility to allow more advanced students to further refine and develop their solutions.

Feedback from the staff and students has been positive and the high quality of work has reflected the interest and engagement shown by the students. The most valuable feedback however, comes from the teaching staff who helped to refine and develop the course materials in the early stages by keeping track of common misunderstandings and frequently asked questions. The case and deliverables were adjusted on the basis of this feedback in order to clarify and improve the quality of the case so that it may better support the course learning outcomes. It was generally observed that the tasks were of an appropriate skill level and difficulty to be engaging and interesting to the students, and to allow them to develop their own systems analysis skills. This was further evidenced by the generally high performance in both this assignment, as well as the final examination which revisited many of the same concepts.

Further support for the effectiveness of this course of study, comes from the students ability to apply their skills to novel and realistic situations. The capstone project course is a compulsory real world project that all students must undertake in their final year of study. This project requires students to apply their (pre-requisite) systems analysis course skills in many areas including directly liaising with an industry based client to conduct a formal requirements analysis. Academic results indicate that students enter into this course extremely well-prepared and have a firm grasp of the pre-requisite systems analysis skills. Students consistently perform well in this course and achieve excellent grades on average. Furthermore, this observation holds true across the different campuses in different geographic regions.

A common concern with teaching cases or assignments is whether the material is too abstract so as to prevent the students from fully engaging with the topic. As this case is based on a real world scenario of which most students will have some experience, it is anticipated that this issue would be diminished if not eliminated. Student feedback on this particular issue has been extremely positive and indicates that this goal has indeed been achieved.

The deliverables are largely aimed at analysis tasks; this reflects the nature of the course being taught. However, it is possible to easily incorporate more design related tasks and additional complexity if required to match the specific needs of a course. Additionally, as noted above, this case was used for an individual assignment. This does not preclude it from being used in a collaborative group or team oriented setting. Deliverables may be distributed amongst team members to work in isolation or in parallel on them. If this is to be done, a suggestion is to encourage the students to consider how they will split up the work, what dependencies may exist between their deliverables, and how to implement tasks such as versioning and change control if team members are working on separate work packages.

6. ACKNOWLEDGEMENTS

Thanks to the Systems Analysis and Design classes in 2011 and 2012 who used this case in class and provided valuable feedback which was used to refine the case through several iterations. In particular the teaching staff at the Singapore campus that used the very first iteration of the teaching case in early 2011, before it was updated and run locally in Australia. Their feedback was an indispensable asset.

Many thanks also to the editorial staff and anonymous reviewers for JISE who have provided extremely well considered and valuable feedback during the review process. This feedback has greatly strengthened our publication and we are grateful for their efforts.

All diagrams created in classes and provided in the teaching notes were created with Microsoft Visio 2010 Professional. This software is used in teaching the analysis and design courses, and is provided free of charge to our students through the Microsoft Developer Network Academic Alliance (MSDNAA) Program.

7. REFERENCES

Arling, P.A., Deeter, C., & Eggers, H. (2010). A Systems Analysis Experiential Case Study: Repeatable Real-World Problem Solving. *Decision Sciences Journal of Innovative Education, 8*(2), 417-422

Bee, S., & Hayes, D.C. (2011). Using The Jeopardy Game to Enhance Student Understanding of Accounting Information Systems (AIS) Exam Material. *Review of Business Information Systems (RBIS), 9*(1), 69-78

Cappel, J. J., & Schwager, P. H. (2002). Writing IS Teaching Cases: Guidelines for JISE Submission. *Journal of Information Systems Education, 13*(4), 287-293.

Hackney, R. A., McMaster, T., & Harris, A. (2003). Using cases as a teaching tool in IS education. *Journal of Information Systems Education, 14*(3), 229-234.

Ishikawa, K., & Loftus, J. H. (1990). *Introduction to quality control*. Tokyo: 3A Corporation.

Kohno, T., Stubblefield, A., Rubin, A. D., & Wallach, D. S. (2004). Analysis of an electronic voting system. *Proceedings of the IEEE Symposium on Security and Privacy*, 27-40. Oakland, USA.

Meyers, C., & Jones, T. B. (1993). *Promoting Active Learning. Strategies for the College Classroom.* San Francisco, CA: Jossey-Bass Inc

Nicol, D. J., & Macfarlane-Dick, D. (2006). Formative assessment and self‐regulated learning: a model and seven principles of good feedback practice. *Studies in Higher Education, 31*(2), 199-218.

Satzinger, J., Jackson, R., & Burd, S. (2008). *Systems Analysis and Design in a Changing World*. Boston: Course Technology.

AUTHOR BIOGRAPHIES

Nik Thompson is a Lecturer in Information Technology at Murdoch University in Western Australia. He holds MSc and PhD degrees from Murdoch University and teaches in the area of Systems Analysis and Data Resource Management. His research interests include affective computing, human-computer interaction and information security.

Danny Toohey is Associate Dean of Teaching and Learning in the School of Engineering and I.T at Murdoch University. A 13 year veteran of teaching, Danny mentors new academics and has received a number of awards for his outstanding contributions. He teaches and researches in the area of Databases and Business Intelligence and is currently conducting research in the area of transnational education.

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096