SERT: Software Tool for Generating Student Engagement Reports

Judith Ann Ouimet
Office of University Assessment
University of Nevada, Reno

Sermsak Buntha
CSE Department
University of Nevada, Reno

Sergiu Dascalu
CSE Department
University of Nevada, Reno

Abstract

Drawing on our previous work on survey and reporting tools that compare faculty expectations with what students report experiencing in class, this paper presents details of a new assessment procedure applied at the University of Nevada, Reno and describes the software created to support this procedure. In order to identify discrepancies between what instructors and what students perceive is happening in the class, help evaluate the course curriculum, and close the assessment loop, detailed questionnaires were developed for faculty and students and two types of reports were generated: a survey-look-alike report and a quadrant analysis report. To automate data collection and reporting, the SERT software was created, as no other existing assessment software allows direct comparison of faculty and student responses. Details of this software, which significantly reduces the processing time of survey data, are provided in the paper in terms of database design, software specification, and user interface details.

1. Introduction

As a major responsibility of educational institutions is to improve their instructional performance, most American accredited colleges and universities collect survey data and make educational assessments regularly [1, 2]. To obtain precise evaluations, surveying and reporting tools have been proposed, for example Mark, Frezza, and Yoo have designed a web-based survey and reporting tool for the preparation of ABET EC2000 [3, 4]. Although this web-based tool provides convenient distribution and collection of survey data, its reporting capabilities do not include means of visualization and do not provide support for direct comparison of faculty and student responses. As another example, two survey instruments for students and faculty, developed by the Center for the Advancement of Scholarship on Engineering Education (CASEE) at the National Academy of Engineering (NAE) [5], do not allow a direct, automated comparison of faculty and student responses.

As many other universities in the USA, the University of Nevada, Reno collects feedback survey data from both students and instructors during the semester. The University's Office of Assessment is responsible for developing and distributing surveys, creating report documents, and performing all other assessment procedures. Because student engagement is a key factor to the success of the educational process, visualization reports have been created to identify the degree of congruency between what instructors and what students perceive is happening in the class. The creation of the reports draws on the main author's prior work on CLASSE (Class-Level Survey of Student Engagement), an assessment tool that compares faculty expectations with what students report experiencing in class [6, 7]. The new assessment procedure that includes these reports has been applied since Fall 2005 at the University of Nevada, Reno in Math, Psychology and Physics classes and is being currently extended to other disciplines. For example, in Fall 2005 the survey questionnaires contained 48 questions grouped in 3 categories and data collected consisted of 60 faculty and over 2000 student surveys (in total, 60 courses).

To automate the assessment procedure the software tool SERT (Student Engagement Reporting Tool) described in this paper was created from scratch because no other currently available software tool provides support for direct comparison of faculty and student responses. Using SERT the processing time per section surveyed was reduced significantly, from about four hours (as done manually) to several seconds.

The remaining of this paper is organized as follows: Section 2 provides details of the student engagement reports created, Section 3 details the database structure used in SERT, Section 4 describes

International Conference on Information Technology (ITNG'07)
0-7695-2776-0/07 $20.00 © 2007 IEEE

IEEE COMPUTER SOCIETY

the main elements of SERT's specification, Section 5 presents elements of the tool's user interface, and Section 6 discusses directions of future work and presents our conclusions.

2. Student Engagement Reports

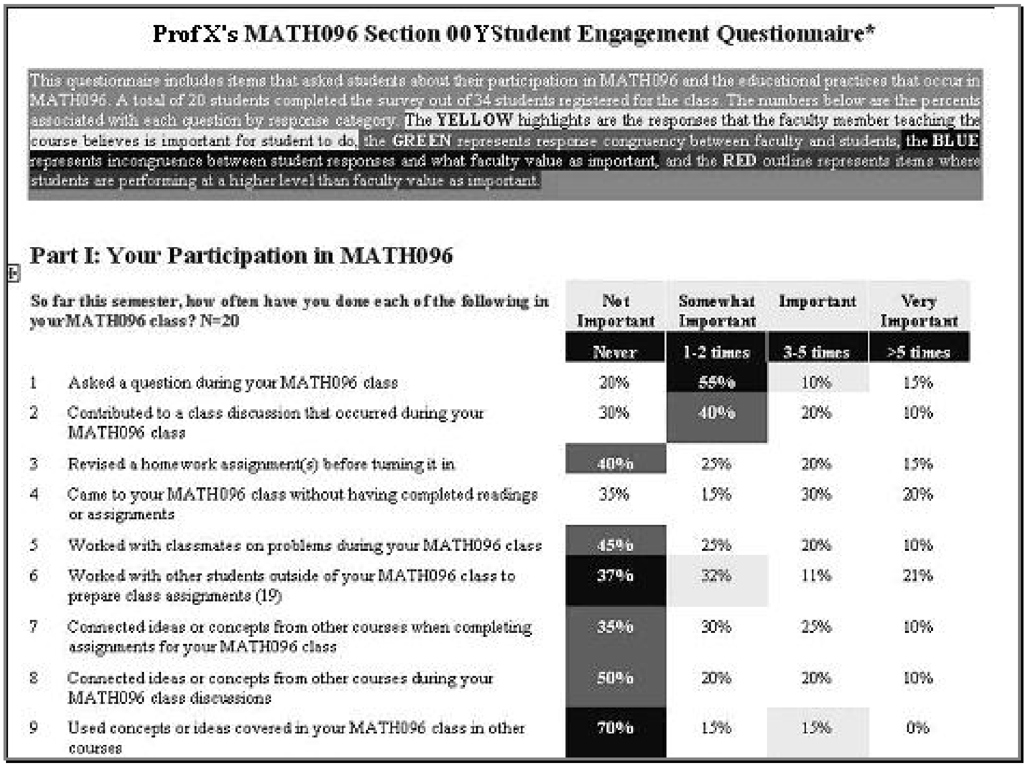
In order to identify differences between what instructors and what students consider is happening in the educational process, detailed questionnaires were developed for faculty and students and two types of reports were generated: *a survey-look-alike report* and a *quadrant analysis report* (the latter with two forms). The *survey-look-alike report* presents data in the same format questions were asked and highlights using color codes and box outlines to visually present the degree of congruency between faculty and student responses. The *quadrant analysis reports* are used as research devices to compare behavior and importance under three criteria: engagement, student techniques, and Bloom's taxonomy [6, 7]. Data for all three criteria are plotted in four quadrants highlighting "hits" and "misses". As detailed later in the paper a complete *report document* may contain multiple reports of both *survey-look-alike* and *quadrant analysis* types.

Due to limited space available, the complete reports are not shown in this paper. However, relevant excerpts of these assessment documents are presented in Figures 1-3.

The survey-look-alike table report is organized as shown in Figure 1. The report is divided into several parts and each part is further divided into several sets of related questions [5]. The overall student response for each question is indicated in percentages and highlighted in green (e.g., box "Somewhat important" of question 2 in Fig. 1) while faculty preferences are highlighted in yellow (e.g., box "Important" of question 1 in Figure 1). If there is match between them, the responses are highlighted in blue (e.g., box "Somewhat important" of question 1 in Figure 1). Other colors are also used, for example a red outline is associated with an item where students perform at a higher level than faculty deem it is important.

The two types of quadrant analysis reports used are shown in Figures 2 and 3, respectively. Each quadrant has a specific meaning, as follows:

- Quadrant I (upper-right): important for faculty and students report high-levels of engagement;
- Quadrant II (upper-left): not important for faculty but students report high levels of engagement;
- Quadrant III (lower-right): important for faculty but students report low levels of engagement;
- Quadrant IV: not important for faculty and students report low levels of engagement.



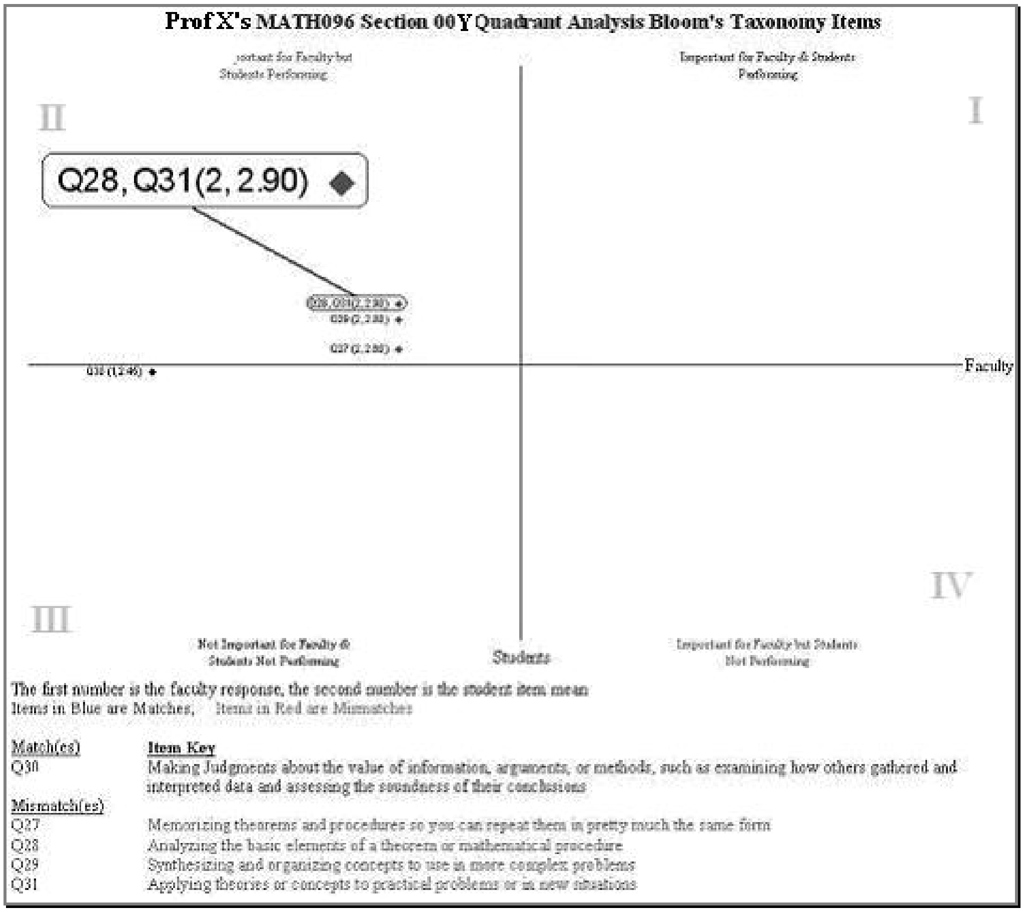
ProfX's MATH096 Section 00YStudent Engagement Questionnaire*

This questionnaire includes items that asked students about their participation in MATH096 and the educational practices that occur in MATH096. A total of 20 students completed the survey out of 34 students registered for the class. The numbers below are the percents associated with each question by response category. The **YELLOW** highlights are the responses that the faculty member teaching the course believes is important for student to do, the **GREEN** represents response congruency between faculty and students, the **BLUE** represents incongruence between student responses and what faculty value as important, and the **RED** outline represents items where students are performing at a higher level than faculty value as important.

Part I: Your Participation in MATH096

	So far this semester, how often have you done each of the following in yourMATH096 class? N=20	**Not Important**	**Somewhat Important**	**Important**	**Very Important**
		Never	**1-2 times**	**3-5 times**	**>5 times**
1	Asked a question during your MATH096 class	20%	55%	10%	15%
2	Contributed to a class discussion that occurred during your MATH096 class	30%	40%	20%	10%
3	Revised a homework assignment(s) before turning it in	40%	25%	20%	15%
4	Came to your MATH096 class without having completed readings or assignments	35%	15%	30%	20%
5	Worked with classmates on problems during your MATH096 class	45%	25%	20%	10%
6	Worked with other students outside of your MATH096 class to prepare class assignments (19)	37%	32%	11%	21%
7	Connected ideas or concepts from other courses when completing assignments for your MATH096 class	35%	30%	25%	10%
8	Connected ideas or concepts from other courses during your MATH096 class discussions	50%	20%	20%	10%
9	Used concepts or ideas covered in your MATH096 class in other courses	70%	15%	15%	0%

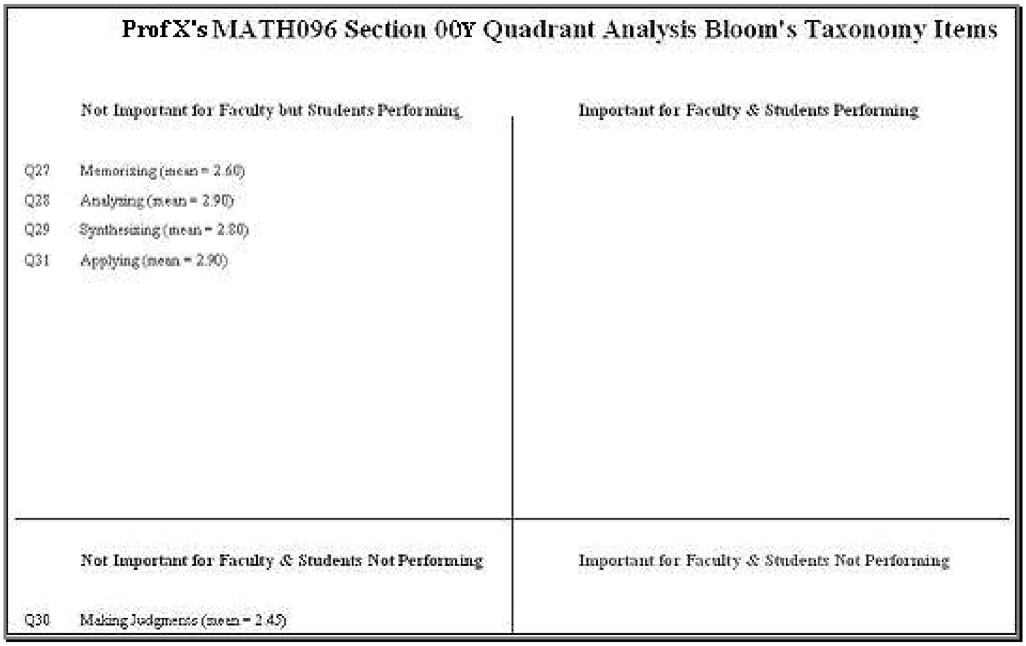
Figure 1: Survey-like report showing degree of congruency between faculty and student responses


Prof X's MATH096 Section 00Y Quadrant Analysis Bloom's Taxonomy Items
portant for Faculty but Students Performing
Important for Faculty & Students Performing
II
I
Q28, Q31(2, 2.90)
Q28, Q31(2, 2.90)
Q29 (2, 2.80)
Q27 (2, 2.60)
Q30 (1, 2.45)
Faculty
III
IV
Not Important for Faculty & Students Not Performing
Students
Important for Faculty but Students Not Performing

The first number is the faculty response, the second number is the student item mean
Items in Blue are Matches, Items in Red are Mismatches

Match(es)	Item Key
Q30	Making Judgments about the value of information, arguments, or methods, such as examining how others gathered and interpreted data and assessing the soundness of their conclusions
Mismatch(es)	
Q27	Memorizing theorems and procedures so you can repeat them in pretty much the same form
Q28	Analyzing the basic elements of a theorem or mathematical procedure
Q29	Synthesizing and organizing concepts to use in more complex problems
Q31	Applying theories or concepts to practical problems or in new situations

Figure 2: Quadrant analysis report showing congruency coordination for each question in a study item group



Prof X's MATH096 Section 00Y Quadrant Analysis Bloom's Taxonomy Items

Not Important for Faculty but Students Performing	Important for Faculty & Students Performing
Q27 Memorizing (mean = 2.60) Q28 Analyzing (mean = 2.90) Q29 Synthesizing (mean = 2.80) Q31 Applying (mean = 2.90)	
Not Important for Faculty & Students Not Performing	**Important for Faculty & Students Not Performing**
Q30 Making Judgments (mean = 2.45)	

Figure 3: Quadrant analysis report showing list of questions in a study group separated by quadrant location

The horizontal axis of a quadrant analysis report is associated with faculty responses, while the vertical axis is related to student responses. The headers of response columns in Figure 1 are mapped into numerical values on these axes as follows: 1–Not important, 2–Somewhat important, 3–Important, and 4 – Very important. Thus, on both axes of a quadrant analysis report the origin is placed at 2.5, meaning that on the negative side of an axis values between 1 and 2.5 are shown, while on the positive side of the axis values between 2.5 and 4 are displayed. To keep the visual representation simple, the values 1, 2, 2.5, 3, and 4 are not shown on the horizontal and vertical axes of the quadrant analysis reports.

In contrast to the report shown in Figure 2 (the quadrant analysis coordinate location report), the report depicted in Figure 3 (the quadrant analysis question list report) does not give a precise location of the faculty and student responses. However, it provides other details; specifically, based on responses, it classifies each question by quadrant location and lists the question's ID, keywords, and average student response value (on the scale 1 to 4, from "Not important" to "Very important").

Plots in quadrants I and IV indicate matches between instructor and student responses and are represented by small blue diamonds (e.g., Q30 in Figure 2) while plots in quadrants II and III indicate mismatches and are indicated by small red diamonds (Q27, Q28, Q29 and Q31 in Figure 2, with overlapping positions for Q28 and Q31 as both have the same faculty and student average responses).

3. Database Design

Microsoft Access [8] is a database system that provides a good embedded environment for general database management. Moreover, by using MS Access further development is possible with other MS tools, particularly for user interface creation with Visual Basic for Applications [9]. Based on the above factors, we decided to use Microsoft Access as the database system for our SERT tool.

For storing and manipulating student survey data, we designed four tables in a Microsoft Access Database (*.mdb file): *Questions, Faculty_Data, Student_Data,* and *Report*. Each table has been created for a specific purpose and the tables have to be filled with data in the order indicated next.

3.1 Questions Table

In regular SERT session operation, the *Questions* table is the first table that needs to be completed. Data in this table are used for defining the structure of the *Faculty_Data* and *Student_Data* tables, which are further detailed in the following sub-sections. The *Questions* table is also used for creating survey forms. Table 1 provides detailed field descriptions of the *Questions* table. It can be noticed that there are both Student Question ID and Faculty Question ID fields in this table; these are not redundant fields, because while most of the questions retain the same ID between student and faculty questionnaires, in some cases their IDs could be different, particularly because student questions and corresponding faculty questions may not be listed in the same order in the respective questionnaires. For example, in the survey we used in 2005/2006, student question #27 corresponds to faculty question #30. Furthermore, there are some student and faculty questions that do not have any correspondent in the counterpart questionnaire. In this case, the ID of the question in the other table is zero and the content of the question is left blank.

Table 1: Details of the Questions Table

Field Name	*Type*	*Description*
ID	Num	ID of student question
StudentQuestion	Text	Content of student question
Keyword	Text	Keyword of student quest.
FacultyQID	Num	ID of faculty question
FacultyQuestion	Text	Content of faculty question
QuestionStyle	Num	Style of survey question
NumberOfChoices	Num	# of available choices
StudentChoicesList	Text	List of student choices separated by "$*$"
FacultyChoicesList	Text	List of faculty choices separated by "$*$"

3.2 Faculty Data Table

The main purpose of this table, detailed in Table 2, is to be used for collecting faculty survey responses. The number of fields in this table is defined from the *Questions* table. The first nine fields are permanent fields but the rest depend on the number of questions and the style defined by the *Questions* table. Some question *styles* (e.g., single choice, multiple choices, text, etc.) require only one field while some require more, e.g. questions that allow participants to select more than one choice from a list of possible answers. The *Faculty_Data* table must be filled after the *Questions* table has been completed.

Another function of the *Faculty_Data* table is to provide options for the designer to specify the contents of the report in user interface drop-down menus (particularly useful for data entry). By looking in this table, the users are able to get the list of years, semesters, departments, courses, and sections used in interface drop-down lists. These lists are displayed when the user specifies the content of the report by selecting various options. Because *Faculty_Data* table is relatively small in terms of fields (not actual contents, which can be very large), the list of available features is displayed fast enough on the user interface.

Table 2: Details of the Faculty Data Table

Field Name	*Type*	*Description*
Department	Text	Department that provides the course
CourseNo	Text	Course code; e.g. MATH096
SectionNo	Text	Section code; e.g. 001.
Year	Num	Year the class took place
Semester	Text	Semester the class took place
FacultyFirst Name	Text	First and middle name of the instructor
FacultyLast Name	Text	Last name of the instructor
NumOf Students	Text	Number of students in this section
Received Date	Date	Date when the questionnaire was received
Xxx	Num/txt	The answer to question Xxx
…		
Zzz	Num/txt	The answer to question Zzz

3.3 Student Data Table

The Student_Data table is included in SERT for the purpose of collecting and storing the responses of student surveys. Similar to *Faculty_Data* table, the first nine fields are permanent fields but the remaining fields may vary according to the number of questions including in the questionnaire and their styles. This table is expected to contain the maximum number of records among all the tables (after all, there many more students than professors involved in a usual teaching proces). Due to space limitations, the field descriptions of *Student_Data* table are not shown here.

3.4 Report Table

The *Report* table is designed for customizing report documents (Table 3). This table allows the user to define groups of questions to be included in a survey questionnaire and to provide a title for each group of questions. The report document is organized according to the order and the contents of the records that populate this table (each record specifying a way of grouping and presenting questions). As shown in Table 3, there are five fields in the *Report* table. The first field, Type, gives information on the meaning of a record, specifically Part and Division are grouping indications (Division being a subset of Part), Foot stands for footnote to be added at the end of the previous Part or Division record, and Graph indicates displaying the information pertaining to the record using the quadrant analysis forms shown in Figures 2 and 3. Both survey-look-alike and quadrant graph analysis reports can be included in the same report document described by a Report table. Further in this table, QBegin and QEnd fields specify, respectively, the first and the last question pertaining to the record while BlnkLines stores the number of separation (white) lines between the presentation (displaying) of this record and that of the following one.

Table 3: Details of the Report Table

Field Name	*Type*	*Description*
Type	Num	Record type: 1=Part, 2=Division, 3=Foot, 4=Graph
Title	Text	Title of this record
QBegin	Num	First question pertaining to this record
QEnd	Num	Last question pertaining to this record
BlankLines	Num	# of blank lines at the end of this record

4. Software Specification and Design

In this section details of SERT's software specification and design are presented. In developing SERT a systematic software process was followed, primarily an incremental process, but also with some elements of evolutionary development, including a couple of initial prototypes that have been later discarded [10]. The software model of SERT includes functional and non-functional requirements, a use case diagram, use cases and scenarios, a class diagram, sequence diagrams, interaction diagrams, activity charts, statecharts, and pseudocode descriptions [11, 12]. Details of some of these are presented next.

4.1 Specifications

SERT's software specification was defined starting from functional and non-functional requirements written in the practical and efficient style proposed by Arlow and Neustadt [13]. The main advantages of these "one-liners" consist of clarifying details of system operation, providing a good basis for contract negotiation, and supplying a clear set of conditions for system test and validation. For illustration purposes, some of the more important requirements of SERT are presented below (note that "shall" repeated in the text is part of the traditional specifications' jargon in software engineering [10]).

Functional requirements

R1 SERT shall allow the user to define database files for collecting student engagement survey data.

R2 SERT shall allow the user to specify the class section that needs to be reported.

R3 SERT shall allow the user to select the class sections of a specific course, department, and semester or year that need to be reported.

R4 SERT shall allow the user to define the computer location of the report document files.

R5 SERT shall allow the user to select either the generation of an individual class section report or the generation of a multiple class section report.

R6 SERT shall support the generation of a student engagement aggregate report.

R7 SERT shall allow the user to work on several different student engagement survey databases.

Non-functional requirements

N1 The system shall use Microsoft Access databases files.

N2 The system shall not slow down significantly other processes during report document generation.

N3 The system shall operate on machines on which Microsoft Word and Excel applications are installed.

4.2 Use Case Modeling

The use case diagram is one of the main UML diagrams used in software specification and modeling [11, 12, 13]. In summary, a use case diagram provides an overview of a system's desired functionality. In our case, SERT's functional capabilities were grouped in 6 use cases, as depicted in Figure 4. Details of use cases are as follows:

UC1: Connect db (shortcut for "Connect to database") provides functions to link SERT to a specific database file (or, in short, a "database" that contains a specific student engagement survey's data) and to browse and query its data. Thus, in SERT's context "connect to database" means "select the survey database" for which report documents will be generated.

UC2: Define content provides drop-down options in SERT's interface to allow the user to specify the content (year, semester, department, course, and course section) of the report document to be generated for the database with which the system is connected.

UC3: Define location provides two options for saving the report document generated. One option allows the user to save the report document in the same location where the database is, the other allows the user to specify this location (path in the file system) manually.

UC4: Section report allows the user to specify the desired (course) sections of a report document and generate this document. In this use case, the user can define the structure of one or more report documents at a time. If the user does not select a specific course section for the report document, the program will create a report for all sections under the lowest level content that has been defined. For example, if the user only indicates the department as MATH, the program will generate a report for all sections in every course that has been taught in the MATH department (and for which data are available in the current database). The section report document is generated as a *.doc file, which can be further edited by MS Word. The report filename is defined by the program using the *[course]_[section]_[semester][year].doc* template, for example *MATH128_003_FALL2005.doc*. For generating the report, SERT constructs graphs in MS Excel and then imports them into the Word document file.

UC5: Aggregate report allows the user to create an aggregate report. An aggregate report is a report that contains several section reports.

UC6: Disconnect db (shortcut from "disconnect from database") provides a function to release the current database ("unselect" the currently selected database) and thus bring the user in a position in which he or she can connect to another survey database available.

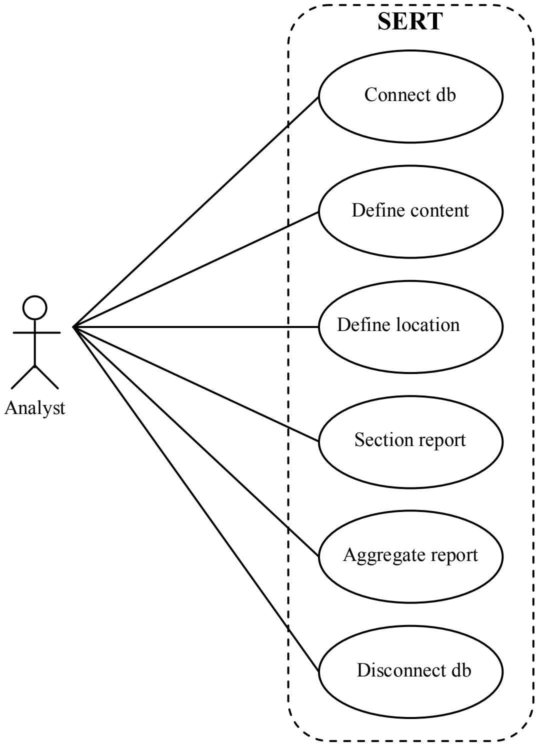


Figure 4: Use case diagram of SERT software

4.3 State Diagram

The state diagram (state-chart) shown in Figure 5 describes SERT's high level behaviour (report generation part only – because, due to space limitations, the tool's entry data facilities are not described in this paper). Transitions between SERT states are triggered by the user. When started, the program enters its *Idle* state, in which the program is not connected to any database. After the user selects a database for connection, the SERT program is linked to that database and enters its *Connect* state. In this state, the user defines the content of a report that he or she wants to be generated. After the content of the report is defined, the program is ready for the report document generation (at this point SERT enters its *Ready* state). From the *Ready* state, when triggered by the user, SERT moves into its *Process* state, in which report documents are generated (an activity that may take between few seconds and several minutes, depending on how many report documents are generated in this run, how many records are contained on the database, the structure of the report document, and so forth). While SERT is in its *Process* state, the user is able to perform other activities using the computer, including editing other MS Word or Excel files. After the generation process is completed, SERT goes back to its *Connect* state. From this state, if the user wants to work with another survey database, the current database needs to be disconnected and thus allow SERT to re-enter its initial state *Idle*. Exiting the program is possible in all SERT states.

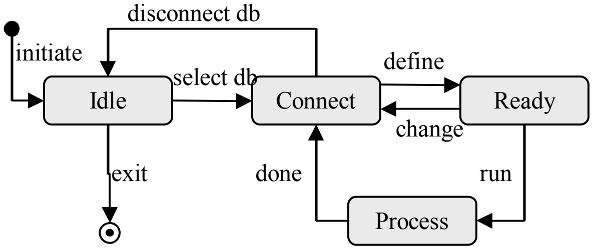
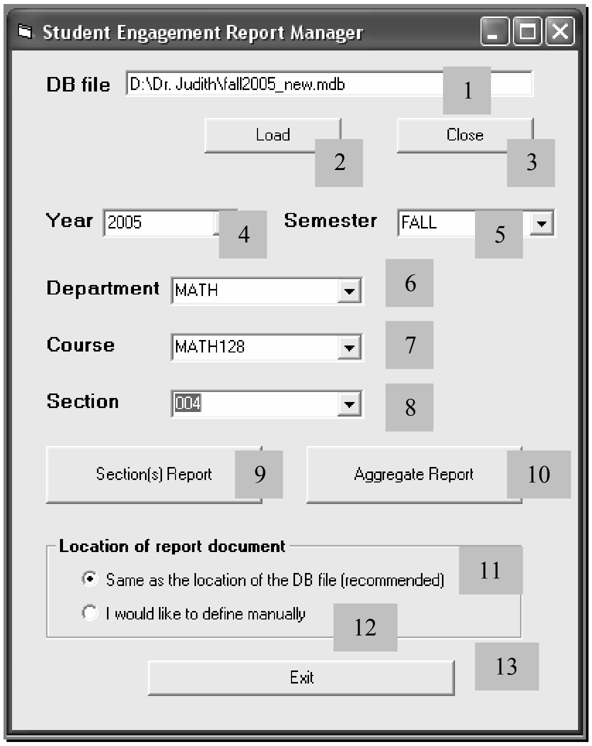


Figure 5: SERT state machine for user interaction

5. User Interface Details

Figure 6 shows the main interface of the SERT program. For explanation purposes, each item on the interface (interface widget/control) was marked with a number from 1 to 13. Details of functionality available through these interface controls are as follows:

Item 1 indicates the location (pathfile) and name of the database file to which the program is currently (or will soon be) connected. When this item is empty (blank text) SERT is in its *Idle* state.


Student Engagement Report Manager

DB file: D:\Dr. Judith\fall2005_new.mdb

Year: 2005
Semester: FALL
Department: MATH
Course: MATH128
Section: 004

Location of report document
● Same as the location of the DB file (recommended)
○ I would like to define manually

Figure 6: SERT main user interface

Item 2 is a button for connecting SERT to a survey database. A dialog for browsing and selecting a database appears when this button is pressed. The database location and filename are shown on Item 1.

Item 3 is a button for disconnecting SERT from a survey database. Pressing this button takes SERT back to its *Idle* state (and empties the content of Item 1).

Items 4-8 are drop-down menus for specifying report content. The options associated with these items will appear in strict order, i.e. an option will be displayed only if the previous option in this group has been specified. For example, if Item 4 (Year) has not been specified, items 5-8 will show only the default value ALL and specific option selection will not be available. Therefore, if an item in group 4 to 8 is not selected, the program will generate reports of all available sections or aggregate all data of these sections. The main benefit of performing option selection in preset sequence is to eliminate user error.

Item 9 is a button for generating a section report document. Pressing this button will lead to the generation of one or more report documents. If any item in the group 4 to 8 is ALL, more than one document will be created.

Item 10 is a button for generating a single aggregate report, regardless whether ALL is indicated in any item of group 4 to 8. If ALL is indicated, the program

International Conference on Information Technology (ITNG'07)
0-7695-2776-0/07 $20.00 © 2007 IEEE

will combine the data from all corresponding individual report documents.

Items 11 and 12 provide options for defining the location in which to save the generated report document(s). Selecting item 11 will save the report at the same location where the database file is stored while item 12 will open a dialog that allows the user to specify this location directly.

Item 13 is a button for exiting the program.

6. Conclusions and Future Work

The SERT software application presented in this paper may not look very sophisticated but it has, in our opinion, at least three desirable characteristics: its operation is reliable, its services are needed, and its utilization has saved significant amount of resources (both in terms of time and effort).

Moreover, while the tool may look simple and straightforward, to "engineer its simplicity" (as Grady Booch well said many years ago [14]) took the three authors of this paper about six months in which traditional software engineering phases where followed, from requirements specification, to design, implementation, testing, and deployment (in our University's Office of Assessment).

Also, the importance of the educational assessment may not be apparent to all the readers, but the people working in academia know very well the complex challenges and rigorous demands of the course curricula and degree program accreditation processes. In fact, there are very few software tools described in the literature that support assessment activities – we find this rather surprising, given that hundreds of colleges and universities in USA alone undergo periodically (every 1 to 10 years) detailed accreditation processes by third parties in which well documented and comprehensive assessment data covering all academic semesters are mandatory.

SERT, the software tool presented in this paper, offers a concrete, practical and effective model on which further applications that support assessment-related activities can be built. Furthermore, while currently used only in academia, SERT's utilization could possibly be extended to other areas, for example assessment of training in industry or military.

Regarding future work, we plan to enhance SERT in several ways. Currently, the SERT software is a stand-alone system. Although the program is very useful for automatically generating report documents (while reducing the report generation time from several hours to several seconds), in the process of acquiring survey responses the users still need first to distribute and collect paper surveys and then manually enter the survey responses into SERT databases. Therefore, the primary direction of future work consists of creating online surveys and thus allowing faculty members and students direct access via web browsers to components of the student engagement assessment process. Other planned directions of future work include enhancing the statistical analysis of data, extending ways of visualizing the results, adding facilities for advanced customization of the questionnaires, and applying the student engagement assessment procedure supported by SERT to more courses and training programs.

References

[1] Bjorklund, S.A. and N.L. Fortenberry, "Developing national survey instruments to assess engagement in best instructional practices and attainment of targeted student outcomes," *Procs. of the 35th ASEE/IEEE Frontiers in Education Conf.*, 2005, pp. F1E/20-21.

[2] Derrick, E.J. and K.T. Stevens, "Using emerging technologies to assess student software development projects," *Procs. of the 35th ASEE/IEEE Frontiers in Education Conference*, 2005, pp. F4G/19-21.

[3] Mark, F., Frezza, S. and W. Yoo, "Web-based course-exit survey for ABET EC2000," *Procs. of the 2003 ASEE Annual Conference and Exposition*, 2003.

[4] Mark, F., Frezza, S. and W. Yoo, "Enhancing ABET EC2000 preparation using a web-based survey/reporting tool," *Procs. of the 33thIEEE Frontiers in Education Conf.,* 2003, pp. T2B/17-22.

[5] NSSE: National Survey of Student Engagement, Center for Postsecondary Research, Indiana University Bloomington, IN, 2006. Technical reports available as of November 2006 at http://nsse.iub.edu

[6] "CLASSE: Collegiate learning assessment survey of student engagement," St. Olaf College, MN, 2006. Available as of Nov. 2006 at http://www.stolaf.edu

[7] Ouimet, J.A. and Smallwood, R.A., "Classe – The class-level survey of student engagement," *Journal of Assessment Update: Progress, Trends, and Practices in Higher Education*, Nov.-Dec. 2005, vol. 17, No. 6.

[8] Prague, C.N., Irwin, M.R., and Reardon J., *Access 2003 Bible*, Wiley & Sons, Hoboken, NJ, 2004.

[9] David, J., Loton, T., Gunvaldson, E., and Bowen, C., *Professional Visual Studio Team System*, Wrox, 2006.

[10] Sommerville, I., *Software Engineering*, 8th edition, Addison-Wesley, 2006.

[11] Booch, G., Rumbaugh, J., and Jacobson, I., *The Unified Modeling Language: User Guide*, 2nd edition, Addison-Wesley, 2005.

[12] Object Management Group, *UML 2.0,* Available as of November 2006 at http://www.omg.org

[13] Arlow, J., and Neustadt, I., *UML and the Unified Process*, Addison-Wesley, 2002.

[14] Booch, G., *Object-Oriented Analysis and Design with Applications*, 2nd ed., Benjamin Cummings, 1994.