Zazu: Supporting Curriculum Planning in a Distance-Education Environment

Judy van Biljon[1] Karen Renaud[2]

[1]Department of Computer Science, University of South Africa (vbilja@unisa.ac.za)

[2]Department of Computing Science, University of Glasgow (karen@dcs.gla.ac.uk)

Abstract

The task of selecting a student's curriculum generally implies a decision-making process in a system with many diverse options and constraints. The task can be complex and the available tools are often designed to represent the data of the organisation using a typically data-oriented approach rather than providing a tool for students and advisors which has been designed with the user in mind. The problems are exacerbated in a distance-education environment, because students take courses at different levels concurrently and often study for many years before completing a qualification.

A prototype electronic performance support tool was developed to address the problems related to paper-based curriculum planning. This paper introduces the prototype, called Zazu, and discusses the design rationale and usability evaluation thereof. The Zazu prototype was evaluated from the general usability perspective and also considered the use of recognised data visualisation techniques.

1 Introduction

Curriculum planning is undertaken by students at all Universities at least once a year. At a distance-education University this process is far more complicated since students can take modules at different levels concurrently. Exacerbating factors are phased-out modules, complicated module requirements and untrained advising staff. When these factors are taken into account the planning process becomes far more error-prone.

The University of XX is a distance-education University with 6 faculties and 65 different departments. Courses are presented in the form of modules that students register for in order to obtain credits for degree or diploma purposes. XX presents information about the modules each department offers in a set of books which constitute the yearly calendar. Students must obtain credits for two major subjects[1] in order to obtain a degree. First-entry students usually come to an advisor with an idea of the majors they would like to complete in order to get a degree. In order to do the chosen majors students need to complete modules equivalent to four subjects at first-year, three at second year — only then taking two majors at third year level, as shown in Figure 1.

XX students may take up to 10 years to complete their degrees and may, during these years, take different combinations of modules as and when they wish – subject to certain constraints. Modules may have *pre-requisite* modules or *concurrent-registration* modules. If Information Systems module a is a pre-requisite for module d in Figure 1, a student must pass module a before the student can enrol for module d. However, if Chemistry module b is a concurrent-registration requirement for module c and a student wishes to enrol for module c but has not yet passed module b, the student must register for modules b and c simultaneously.

The combination of the freedom accorded the student in enabling a relatively free choice of modules, as well as the related pre-requisite and concurrent-registration requirements, makes it difficult for students to make choices about modules to enrol for once they are in the system. It is also easy to make ill-advised choices when first

[1]A major subject usually comprises 3 or 4 modules.

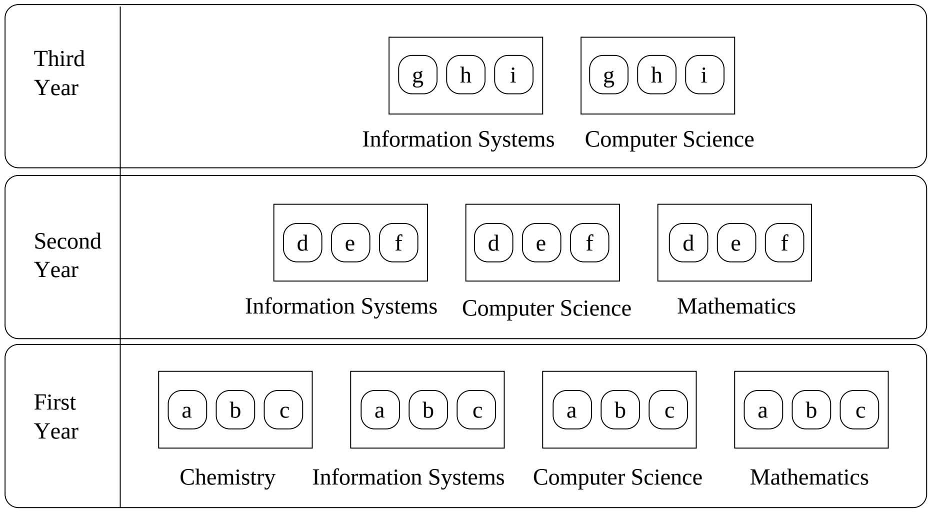

Figure 1: Degree Structure

enrolling which can prevent a student from enrolling for subjects in later years. The current format of the calendar, instead of supporting this complex decision-making process, presents both students and advisors with many problems:

1. users tend to suffer from information-overload because modules from *all* departments are described together.
2. it is difficult to find information since there is no index. Provision of a helpful index would be very difficult for such a document since the information required by each particular user is unique.
3. all information is related to, and ordered by, module code. The XX module codes are cryptic, in the form of, for example, COS311 and COS419, and tend to be meaningful only to XX staff. Using module codes is logical and sensible because it fits into their frame of reference — modules being referred as such within the University. It presents many problems to students and new lecturers, however, because the codes tend to confuse and detract from the information being presented in the calendar.
4. a limited amount of information is given about each module - which tends to be inadequate for some users.

The following section will give a more detailed description of the calendar-based advising process, and motivates the use of a software-based system to support this process.

1.1 Problem Description

Consider the following example of what the decision-making process entails. A student decides to select COS311 as a third year module and has to determine the pre-requisite modules to be completed on the first and second year levels. The XX calendar [24] provides the following information concerning COS311:

COS311 Advanced programming (3 hours)

Pre-Requisite: COS211, 212

The pre-requisites of these modules should also be considered in setting up a curriculum. The following information is provided in the XX calendar for COS211 [24]:

COS211 Programming: Data structures (3 hours)

Pre-Requisite: (COS111 or 114), COS112 and (COS113 or INF101)

Registration requirement: COS212

Note that COS212 is *both* a registration requirement for COS211 *and* a pre-requisite for COS311. Now consider the information given on the module COS212 in the XX calendar [24]:

> COS212 Programming: Practical (3 hours)
>
> *Pre-Requisite*: COS111 (or 114), COS112 and (COS113 or INF101)

The pre-requisites given for COS211 and COS212 are identical even though stated slightly differently. This slight inconsistency could easily cause confusion to students and novice advisors. The expression for pre-requisites, i.e. COS111 (or 114), COS112 and (COS113 or INF101) can be expanded into four independent options.

1. COS114, COS112, INF101; or
2. COS114, COS112, COS113; or
3. COS111, COS112, COS113; or
4. COS111, COS112, INF101.

The modules COS111, COS112 and COS113 have a further requirement of certain grades for school-level Mathematics. This example illustrates the complexity of finding the pre-requisites for only one module. The process is complicated by the fact that any one subject major comprises at least four modules, which means that the above-mentioned process must be carried out recursively and iteratively to arrive at a complete curriculum. The following section will explore the possibility of providing a software system to support this process.

1.2 Exploring the Case for Software Support

An investigation into the existing systems for computer-aided support in curriculum planning at residential universities in YY, including two other distance-education universities, was carried out. We could not find a system which would satisfy our needs although we did discover some student projects which attempted to satisfy this need at other Universities. It was found that many systems exist for dynamically adjusting individual module curricula [4, 14, 16, 18], often based on computerised student performance measurement. This did not suit our purpose, however, and we concluded that, the situation at XX being quite unlike any other, it warranted a unique approach. Two main factors contribute to the uniqueness of the situation:

1. At most other tertiary institutions a set curriculum is prescribed, greatly simplifying the process of curriculum planning and diminishing the need for computerised support in curriculum planning.
2. Students at XX take much longer, on average, to complete their qualifications than residential students. Apart from the fact that this necessitates a re-assessment of their curriculum on a regular basis, it also means that modules are phased out and, in addition to understanding the current module structure, advisors also need to know how to substitute current modules for pre-requisite purposes.
3. Students can take a mixture of modules at any of three levels, making the assessment of their current status and recommendation of their yearly curriculum tricky and error-prone.

No existing system could be found that addressed the need for support in curriculum planning and therefore it was decided that the situation warranted the development of a computer-based system. The first step in investigating this possibility was to identify all the users of the calendar who would benefit from such software. We identified three distinct groups of users:

1. *lecturers* — at first it seemed that these users did not have undue problems with calendar-based advising. However, we found that even lecturers who understand the content of the calendar and the structure of the degree courses experience difficulty in finding the required information in the calendar. We discovered

that lecturers almost universally book-mark their calendars in a variety of eye-catching ways. In addition, lecturers generally give advice verbally, which students sometimes remember incorrectly by the time they arrive at the registration hall.

2. *students* — this group can be split into two groups:
   (a) *novice students* — these students are fresh out of school and have only a vague idea of the module structure of the degree and diploma courses. They have no understanding of the pre-requisite and concurrent-registration mechanisms.
   (b) *current students* — while these students have a good understanding of the structure and system they present advisors with an even bigger challenge. They have typically completed a number of modules at various levels and need to know which modules they can register for i.e. which modules they have satisfied the pre-requisites of, and, having identified such modules, which concurrent registrations are required.

3. *producers* — it is difficult to ensure correctness of the calendar. The calendar is produced centrally and once a year each department sends a handwritten list of required changes to the production department. The fact that errors creep in is predictable and inevitable. These errors are currently corrected by sending memoranda or emails to all advising staff during the registration period. This mechanism is flawed for two reasons:
   (a) lecturers have different information from that given to students; and
   (b) lecturers have to consult multiple sources of information during the advising process, which leads to errors.

Our proposed solution to these problems is that an *Electronic Performance Support System* (EPSS) be developed. The prototype of this tool will henceforth be referred to as the Zazu[2] system. According to De Kock [6] the goal of a EPSS tool is to enable people to perform new or complex tasks more effectively. Unlike paper-based mechanisms an EPSS tool provides an element of immediacy, a dynamic reference tool and a learning aid [3]. Hugo [12] defines an EPSS tool as:

> *integrated, readily available, up-to-date, just-in-time support that makes employees proficient on the job.*

The Zazu system meets these criteria by comprising intuitive, task-oriented screen design, task-oriented menu/system structures and processes, information database, application software and on-line help. In addition to the obvious benefits of keeping the calendar data in a database and providing users with a dynamic interface to that data, there are other benefits to be derived from this scheme:

1. *Effortless data collection* — By collecting some very basic demographic data we can determine what the real needs are of the students visiting the department asking for advice. This is currently almost impossible to do because advising is done by many different people over a period of 3 months and it is difficult to pool individual experiences to identify particularly common problems after the registration period is over.

2. *Ease of producing the calendar for different media* — It is relatively simple to use software to produce the calendar in various formats since the stored data is being used for each [5]. It will also be relatively trivial to generate separate departmental calendars from the same data and to ensure consistency of the different printed documents.

3. *Performance-related benefits* — improved productivity and efficiency, and having students' needs met promptly and correctly.

[2]The name Zazu alludes to the bird Zazu in the film *Lion King* from whom the lion Simba received performance support.

4. *On-the-job training* — Novice lecturers will be empowered to provide advice and the on-the-job learning curve will also be reduced.

Due to the high risk involved in giving incorrect advice in curriculum planning the prototype was initially developed and tested for use by lecturers only. This decision was made so that a second version of the prototype could be made available to students after problems identified during initial prototype evaluation had been corrected. Section 2 addresses task analysis for the performance support computer tool. Section 3 will provide an overview of the Zazu prototype. Section 4 will describe the prototype evaluation procedure and give the results of the evaluation. Section 5 gives details of future work on the project and Section 6 concludes.

2 Task-Analysis for the Zazu System

It is advisable for task[3] analysis to be done for EPSS tools since it elicits knowledge for design purposes, provides a reference for evaluation and ensures the efficiency and accuracy of the resulting system [12]. Task analysis will help to ensure that the prototype will support the user tasks[4].

Task analysis, in the context of interface design, studies user needs and the definitions of functional requirements [30]. A list of frequently-asked questions and problems experienced with curriculum planning on the existing system was compiled based on lecturers' previous experiences. This list, together with the task analysis done on the existing system, form the basis of the user requirements for the Zazu system. Task-oriented interviews with lecturers experienced in curriculum planning revealed two primary tasks:

1. First enrolment advice for new students — referred to as *novice students* in Section 1.

2. Curriculum re-assessment for re-registering students — referred to as *current students* in Section 1.

In giving such advice there are two distinct types of activities: firstly, fact finding — gleaning information about modules offered and their requirements; and secondly, curriculum planning — constructing a curriculum for a specific student, based on his or her specific background. Advising is typically a recursive decision-making process. In general terms one can consider two distinct types of decisions to be supported [10]:

1. *Programmed* decision-making, which is a routine procedure worked out for dealing with a recurring situation. The programmed curriculum-planning tasks involve repetition of procedures such as, for example, finding the pre-requisites for a module.

2. *Non-programmed* decision-making, which is novel and unstructured and requires more thought and background knowledge. The fact-finding curriculum-planning tasks are user driven and non-programmed, involving iterative browsing and selection-type tasks. The goals are not always clearly defined and many actions are possible.

The tasks identified in the support of curriculum planning include both programmed and non-programmed tasks. The programmed tasks are not necessarily trivial given the structure of the calendar and the dispersed nature of the required information. The fact-finding part of the task becomes easier with experience. The non-programmed tasks are challenging for most advisers.

The prototyping effort focused on implementing representative tasks from each kind of the above-mentioned tasks. The programmed and non-programmed tasks are iteratively interspersed. A user needs the system to provide timely, current, context-sensitive information for decision making (non-programable). However, the selection based on the decision could be followed by a cumbersome manual task which could ideally be programmed. At this point the user may need to obtain more information and so the process recurses until the curriculum-planning

[3]By task we mean a *unit of human activity which is carried out in order to achieve a specific goal* [32].

[4]This study is limited to advisors' tasks in curriculum advising as described in Section 1. An investigation into incorporating the producers' role into the Zazu prototype will be carried out during a later stage of this project.

process is complete. The system will attempt to perform the programmed tasks on behalf of the user and support the non-programmed tasks as much as possible by providing well-structured interlinked information.

It is clear that there are definite benefits in automating this task to the point where the user indicates the module and all the pre-requisites are automatically displayed. We will use an adaptation of a simple iterative computer operating model (Figure 2) proposed by Author [2] which extends an example first published in Woodson, Tillman and Tillman [33], which is in turn based on the set of USA Military User Interface Design Guidelines. This type of model is eminently suitable to be used as the basis for formulating a strategy for task analysis. Norman's classical execution-evaluation model (as discussed by Dix *et al.* [7]), has similar stages. The following should be

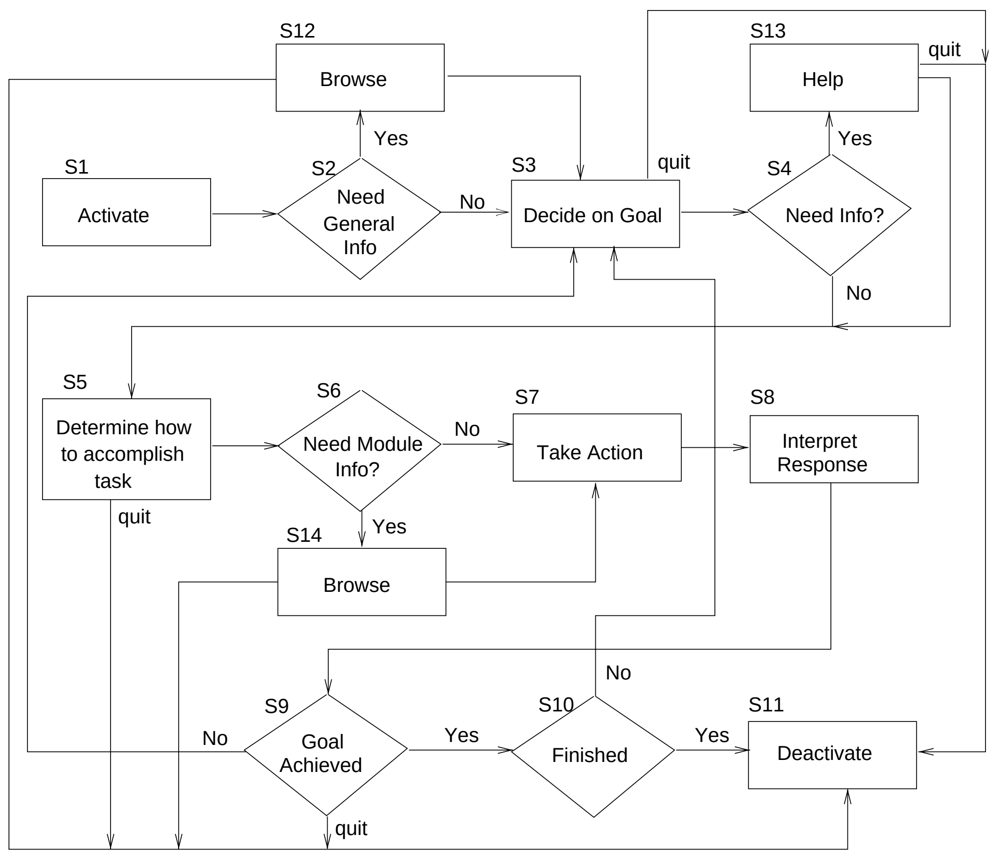

Figure 2: Task Analysis

noted about Figure 2 specifically when considering the tasks for the prototype:

1. The system is activated at S1 and deactivated at S11.

2. S12 and S14 represent the non-programmable tasks of browsing for information.

3. All the tasks focus on finding information and although most advice sessions will proceed from S1 to S11, provision is made for the case where a user wishes to deactivate during the browsing actions (S12, S14)

4. S7's implies that the user either chooses a module or indicates that a module has been completed.

5. The error feedback loop S9-S3 is included to provide for the possibility of a user redefining a goal. In that case S3 could possibly be re-labelled as goal adjustment.

6. In career planning many intermediary goals are defined and followed. Therefore the accomplishment of a goal is not necessarily the completion of the advice session, hence the loop S10-S3.

7. S13 represents information on the prototype and is applicable when the user has difficulty in understanding the interface of the prototype.

8. S10 is reached at the achievement of every goal and refers to the situation where the user is *finished* with the advice session.

The results of the task analysis of the existing system is used as basis for the requirements specification of the new system. It is necessary to make cognitive skills visible in order to develop an effective EPSS [22] and this was done for the Zazu prototype by means of the preceding task analysis. The following section will describe the Zazu prototype system.

3 The Zazu System

The design method followed, and components identified, comply with the requirements for an extrinsic EPSS [22]. A stand-alone EPSS application for a specific task can be called a *Performance Support Tool* (PST) [12]. Zazu is a partially implemented EPSS which addresses specific performance requirements of the curriculum planning task and therefore it will henceforth be referred to as a PST. Performance support is generally based on a knowledge base of some kind, so PST design is inherently data-oriented. This traditional data-oriented perspective is contrary to the user-oriented perspective that we wanted to apply to this project in order to provide a truly useful tool. The challenge addressed in the next section is the design of the interface to provide a user-oriented visualisation of the calendar data.

The data and business rules were captured from the afore-mentioned calendar. The information database was designed to allow the rules to be data driven. This means that new modules and their relationships are added simply by additions to the database. The discussion of the implementation is beyond the scope of this document and it suffices to say that the implementation was done in a relational database with Delphi as the programming language. The challenge in designing the interface was to integrate the programmable and non-programmable tasks in such a way that all recurring, routine, programmable tasks are automated while allowing the user to use the system to provide performance support for non-programmable tasks.

3.1 Designing the interface for the Zazu System

It is not sufficient merely to place the information currently recorded in the calendar on the screen since this fails to anticipate the user's real world needs [1]. One needs rather to find a visualisation mechanism which *conveys* knowledge — making the difference between a simple data generator and an information searcher [1].

The decision-making process can be supported by a system which supports the user by organising his or her thinking rather than suggesting a course of action [1]. The interface components of an EPSS should therefore be designed to "lead, follow, or get out of the way" [22]. Users of the Zazu interface fit into three broad categories:

- novice users have the need to be *led* through the appropriate activities by providing the necessary scaffolding;
- intermediate users should be provided with a reactive interface and interrupted only when an error is made (*follow*);
- expert users should find the interface unobtrusive and transparent, matching their mental models and usual procedures (*get out of the way*).

The literature cites many valuable design principles which can be used to ensure the success of an interface. In the following discussion design principles are given, followed by a description of how the Zazu prototype was developed with the specific principle in mind.

1. *Make it easy to determine what actions are possible at any moment* [20]. The Zazu interface shows options as they are applicable from the start, allowing an application usage paradigm based on recognition of available options rather than recall of control sequences [8]. Instructions appear at the top of the screen to guide the users through the stages to achieve their goal.

2. *Make things visible, including the conceptual model of the system* [20]. The user model will be based on the system image (that which the user perceives) rather than on the design model (how the system really works). The design must support the user in building an internal model of how the application works. The internal model is created within the mind of the user by the act of visualisation [27]. The following data visualisation issues were implemented in the design:

   (a) *Presentation:* Modules are represented as square building blocks, arranged in three levels with the first year level at the bottom of the screen (Figure 3). This is intended to help the user to correlate the user image with the system image to form the correct mental model i.e. using the metaphor of constructing a building as being similar to the process of accumulating modules for a degree course. In both paradigms the builder starts on the ground (1st year) and builds on top of the previous level until one gets to the roof (3rd year). Furthermore, 3rd year modules are dependent on the supporting 1st and 2nd year modules. This metaphor supports the following design principles:

Figure 3: Representing Modules

   i. *Spatiality* — According to Spence [27] the use of spatiality in representation leads to a robust internal model. Spatial and visual consistency is supported in the prototype by assigning a location to each module according to the year level structure and consistently displaying the module in that location throughout.
   Plaisant [23] and Spence [27] warn about the danger of using scroll bars when presenting overviews as users forget to browse the complete image. The use of scroll bars is inevitable for this application but grouping the modules according to year level helps to support the notion that there is movement and direction associated with the structure.

ii. *Consistency & Stability* — The selected modules are marked by means of a tick boxes but modules are always displayed in the same location to preserve the stability of the view. In the first version of the prototype the layout algorithm generated a fresh layout of module boxes for different tasks but this change in location of basic elements can get users somewhat disoriented as was also experienced by Kumar [15].

iii. *Clarity* — Lokuge [17] mentions visual clarification as an important characteristic for reducing user's cognitive load in an interactive, visualisation system. Module colours were chosen in line with the previously mentioned metaphor. Supplementary modules are displayed in brown, first year modules in dark green, second year modules in lighter green and third year modules in blue. This denotes the progression from the earth (the foundation formed by supplementary modules), to vegetation (intermediary modules) to sky (final modules).

(b) *Responsiveness:* Feedback is provided for every choice or action that is taken i.e. left-click, right-click or fly-over. Left-click is used to signal an action while right-click provides a detail description of the module. The mouse roll-over reveals a short description of the module. The lengthy description is provided on demand to decrease the cognitive overload experienced by users.

The system is mainly user-driven and interaction is provided in each phase. The non-programmable tasks are completely user-driven, the functionality in the other phases is system driven with the user initiating the events. Programmable tasks are triggered by the user making some selections and observing the application's response — shown in Figure 4.

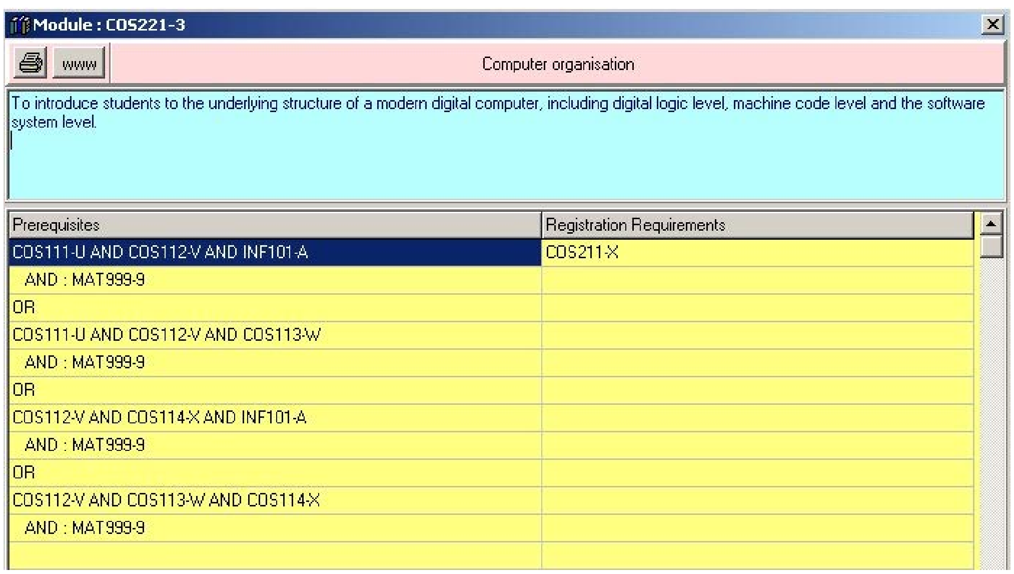

Figure 4: Module Description

(c) *Interaction:* The top of the screen operates as a navigation area to make the program state(mode)clear. The interface was designed to support the user in recognising the two main modes:

i. *browsing* for general information, or

ii. making *selections* for viewing module requirements-related information as described below.

Nielsen [19] encourages simplicity of design in ensuring interface consistency. There is always a trade-off between providing information and ensuring that the user does not suffer from information-overload. Sutcliffe [28] advises that only relevant information be provided and that the user's working memory should not be overloaded. Thus we purposely excluded certain features which could conceivably have added to the completeness of the system. However, ease of learning, ease of action and error avoidance was a higher priority than supporting full functionality of the prototype.

- Option 1 : *Registration requirements* — When a user left-clicks on a module box the pre-requisites appear in a window. In the first version the window appeared adjacent to that particular module box but the heuristic evaluation revealed that it could be confusing to have the information popping up at a different location for every module. The window currently consistently displays at the centre of the screen. If there are no pre-requisites for the module, a message informs the user.
- Option 2: *Recommended modules* (for future enrolment) — Only modules that have no pre-requisites are displayed at the start. Each module has two check boxes, one is labelled *completed* and the other *chosen* (see Figure 5). If the *completed* check box is selected, two actions take

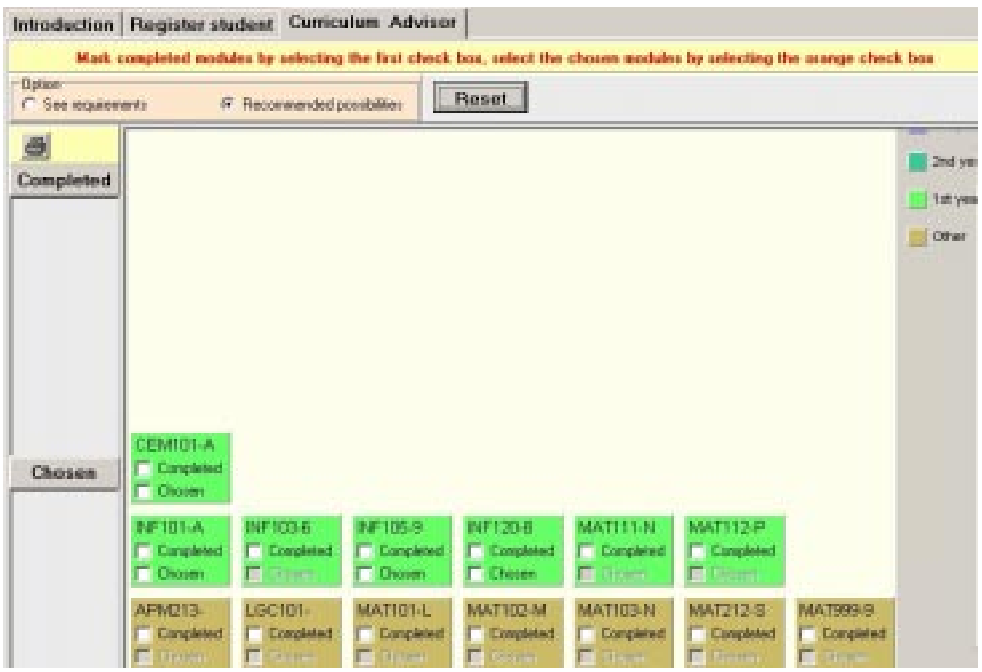

Figure 5: First Screen — Recommendations

place (shown in Figure 6):

  - *Clicking "completed":* Modules which become available as a result of having completed modules appear, i.e. higher level modules for which the particular module was a pre-requisite become available to the student — focusing the user's attention on modules of importance to their decision-making process [28]. At the same time a list is maintained on the left-hand side of the screen denoting already-completed modules.
  - *Clicking "chosen":* The module's code is added to the list of modules to be registered for on the left-hand side of the screen.

  Both task feedback (a tick appears in the box) and external feedback (list is updated on the left) are given as the user acts upon the interface. Goodman [11] has shown that providing both types of feedback has an additive effect on task performance.

3. *Make it easy to evaluate the current state of the system.* [20]

   (a) *Semantic Conflicts* — The same visual features, such as colour, transparency and motion should not be used by more than one visual element or process [17] This is done for the two options as follows:

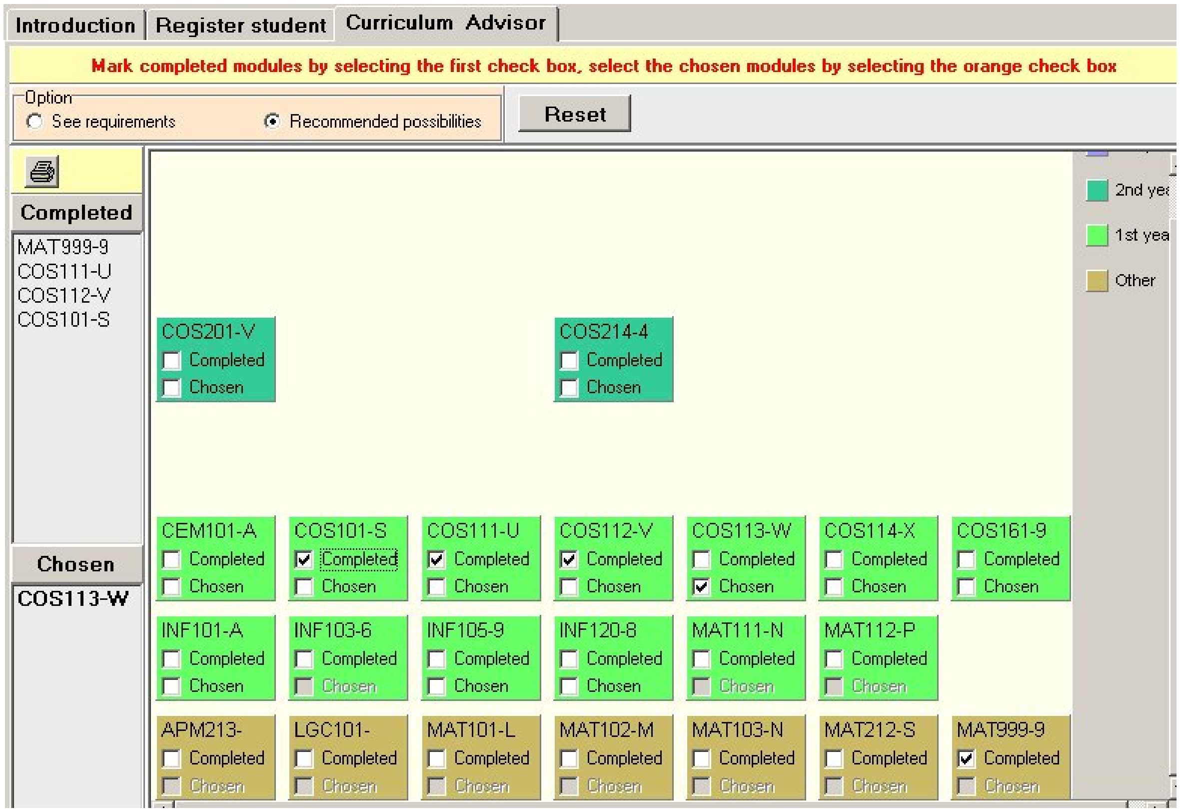

Figure 6: Possibilities Once Modules have been Selected

- Option 1: Registration requirements — displays requested information in a transient window.
- Option 2: Recommended modules (for future enrolment) — maintains a dynamic display which reacts to the user's specification of completed and chosen modules.

(b) *Constraints* — can be used to simplify the choice of actions [20].

i. *Satisfying Pre-Requisites:* Considering option 2, on first appearance only the 1st year and supplementary modules are displayed (only the available modules). As modules are selected, the modules that become available are added. It is therefore not possible for the student to choose modules for which the requirements have not been satisfied. This constraint reduces the number of alternative actions and filters out incorrect combinations.

ii. *Substituting current for phased-out modules:* Students cannot enrol for expired modules but the students may well have completed such modules. This rule is supported by not allowing the user to choose these modules for future enrolment, while still acknowledging completion of such modules. The system will automatically display the modules for which the substitute module is a pre-requisite so that the student can still progress towards their qualification.

(c) *Filtering* — both dynamic filters, which act on information as it is collected, and static filters that act on information that has already been captured and internalised [22], can be used. The act of displaying only the accessible modules at the outset is static filtering based on values in the database. The functionality of adding modules as they become available, based on the user's act of selecting modules as *completed*, provides dynamic filtering.

4. *Follow natural mappings between intentions and the required actions; between actions and the resulting effect; and between the information that is visible and the interpretation of the system state*[20]. A user selects

a module by means of a left click on the *chosen* or the *completed check box* inside the module box. This makes it easy to observe the reaction to the action. The convention of left-click for action and right-click for information is followed throughout the system.

5. *Undoability* — has been ensured by:
   (a) giving the user the ability to select and de-select the check boxes.
   (b) the dynamic nature of the display, which allows the user to exploit and explore all available combinations of modules until satisfied.
   (c) providing a reset button which allows the user to restart. The user can preview the possibilities before printing and if still not satisfied he/she can start all over again.
   (d) displaying the completed and chosen modules separately so as to minimise confusion. The
   (e) offering a print facility to allow users to print their final choices.

Zazu provides a visualisation environment offering overview, filtering and details-on-demand. Visual design techniques such as typography, colour coding and filtering contribute to structuring the information and supporting the users' mental model of the system. The reactive nature may well help users to retain the information presented in the calendar better than they currently do. This is based on the well-known psychology principle cited by Terrier and Callier [29] which states that one memorises best when one processes the *meaning* of information than when information is presented superficially. Zazu does this by dynamically presenting information based on user choices. The following section will report on the evaluation of Zazu.

4 Evaluation

According to the ISO 9241 [13] standard:

- "Usability is the effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments;
- Effectiveness is the accuracy and completeness with which specified users achieve specified goals in particular environments;
- Efficiency is the resources expended in relation to the accuracy and completeness of goals achieved;
- Satisfaction is the comfort and acceptability of the work systems to its users and other people affected by its use".

The ISO standard provides guidance on how to specify and measure the usability of products, using the stated factors. Frøkjaer *et al.* [9] presents evidence to suggest that effectiveness, efficiency and satisfaction should be considered independent aspects of usability and should be measured independently.

4.1 Usability evaluation methods

Wesson [32] identifies two ways to conduct usability analysis: analytically, by performing a simulation of how the users will perform activities; or empirically by the building of a prototype and the testing and observation of actual user performance with the prototype. Van Greunen [31] mentions *heuristic* evaluation, *observation* and *empirical* evaluation as usability evaluation methods. Heuristic evaluation refers to a method where usability specialists apply their knowledge of established usability principles to find usability problems without the need to involve users. Dix and Finlay [7] describe three main categories of usability principles namely learnability, flexibility and robustness which are applicable to heuristic evaluation. According to Rosenbaum *et al.* [25], usability testing based on observation — whether inside or outside of a lab facility — was rated highest as an effective usability methodology aimed at strategic impact.

4.2 Evaluation of the Zazu prototype

Frøkjaer *et al.* [9], used task completion time as an indicator of efficiency and the quality of the solution as an indicator of effectiveness. In the evaluation of the Zazu prototype, time was also used as the metric to test efficiency. The ISO 9241 [13] standard further refines effectiveness into completeness and correctness. However, an answer had to be complete to be correct so correctness and completeness could not be measured in isolation. Heuristic evaluation followed by usability testing and a questionnaire was chosen for the Zazu prototype. Clearly the number of users was too small for an in-depth empirical evaluation, especially if the users had to be subdivided further into expert and novice users.

The first version of the prototype was scrutinized for usability problems by two usability experts. The results of this heuristic evaluation included specific changes to improve ease of use and effectiveness of the prototype but also identified further user needs. Their recommendations which necessitated fundamental changes to the design were implemented in the second prototype. Rosenbaum [25] acclaims this focus on specific changes as a reason for the success of usability testing as opposed to field studies. The second prototype was evaluated by means of observing lecturers using the prototype to and involved two questionnaires. Section 4.4 discussed the process in more detail.

4.3 Selection of evaluators for the questionnaire

Van Greunen [31] notes that a focus group of 8-12 participants are found to be an appropriate size for research while Rubin [26] recommends 4-8 participants for a formal usability test. In this evaluation ten evaluators, which included 5 novice and 5 expert advisors, participated in the evaluation. Olivier [21] warns against the possibility of learning interfering with the results when the same users evaluate different systems. As a precaution five did the evaluation on the current system(using calendar books) first followed by the Zazu evaluation while the other five did the Zazu system first and then used the book. Two sets of questions namely, set A and set B, were used randomly to minimize the effect of the question set on the result.

4.4 Evaluation process using the questionnaire

The lecturers were presented with a set of questions(Set A or Set B) and the calendar. A computer program was used to monitor and support the evaluation process, the lecturers were required to log in and provide some demographic information for the questionnaire (see Appendix A). The user was then presented with a number of questions. The questions encompassed fact finding, selection and decision-making tasks. Decision-making tasks involve selection and fact-finding and for the purpose of the questionnaire, decision-making tasks are defined as a combination of fact finding and selection tasks.

The questions were selected to be representative of the curriculum planning task. The lecturers were required to type the answers in. In order to automate the whole evaluation all possible answers to the questions would have to be provided and this could mask the cases where the evaluator could not find any answer and could provide clues to finding the correct answer. At the end of both testing sessions (Zazu and the calendar books) an interview was held to get feedback on the system and a post test questionnaire (see Appendix B) was completed to get some indication about the users satisfaction with the system.

4.5 Results obtained from the questionnaire evaluation

- Effectiveness: 2 of 5 novice users and 4 of the 5 expert users got all the answers correct using the paper-based calendar system. In contrast 4 of 5 novice users and 5 of the 5 expert users got all the answers correct using the existing calendar system. The leads to the conclusion that the Zazu system did improve efficiency and the improvement was greater for novice users.

- Efficiency: The evaluators time in using the Zazu system was on average X minutes better than without the system for novice users and X minutes better for experienced users. This means an improvement of ... in effectiveness as measured as a function of time.

- Satisfaction: The biggest gain was in user satisfaction where users reported increased task satisfaction attributed to the use of the Zazu system. The post test questionnaire(See Appendix B)was used to gauge the user's perception of the task satisfaction experienced. The following results were obtained.

Property	Percentage for calendar	Percentage for Zazu
Mental demand	70 %	50%
Physical demand	20%	30%
Performance level achieved	20%	20%
Effort expended	20%	20%
Frustration level	20%	20%
Annoyance experienced		

It can be concluded that the Zazu system greatly improved user's satisfaction in performing the task of curriculum planning.

The main limitations identified were the increased design and programming effort needed to create a user centred system image that supports the user image. The scalability of the design provided challenges. When data is represented by items in scrollable list boxes the addition of new elements cause few problems. However, when a new object is dynamically created for each new data item and the presentation properties such as colour, location etc of the data item depends on the specific data item, careful planning is needed.

The complex, unstructured business rules which underlie the curriculum planning made it very difficult to automate and support the whole process of curriculum planning. While Zazu does support the complex, recurring tasks it does not provide complete performance support for all tasks.

5 Future Work

There are many databases at XX which could be incorporated into such a system in order to provide an even better tool for users. For example, students could benefit from knowing how many other students are currently registered for a particular module. They would also probably like to know the pass-rate for a particular module so that they have an idea of the difficulty of the module. The initial prototype was developed for use by lecturers. Once the initial evaluation has been completed the limitations identified the Zazu prototype could be extended and tested for use by students via the Web. Further research should include work on adapting the interface to reflect the multi-cultural composition of the student population.

6 Conclusion

The Zazu PST is proposed as a starting point toward an interactive visualisation environment for providing people with the knowledge, skills, tools and reference material necessary for them to perform in their job of curriculum planning. Considering both the heuristic evaluation and the results of the questionnaire, two main benefits of the Zazu system had been identified. Firstly, the cognitive load on the user was greatly reduced by the identification and automation of complex programmable tasks. Secondly, narrowing the gap between the user model and the design model by designing the presentation, selection and interaction according to principles that support the forming of a mental model. The main contribution of this project is the display of the data elements as independent objects that can be directly observed and manipulated. This is in stark contrast to the traditional approach where data items are items on lists and do not feature as independent, atomic objects with properties and relationships. The usability advantage of the data visualisation approach is clearly expressed in the Zazu system where the use of existing, relational database technology and application software together with user-centred design principles and data visualisation provided an eminently usable PST.

Acknowledgements

This project was partly supported by a grant from the Faculty of Science at the University of XX. We also acknowledge the contributions of Aurona Gerber.

References

[1] M. J. Albers. Goal-Driven Task Analysis: Improving Situation Awareness for Complex Problem-Solving. In *ACM 16th International Conference on Systems Documentation*, Document Design and Production, pages 234–242, 1998.

[2] Author. Task based evaluation Metrics for E-commerce sites. Technical Report XX-TR-2001-26, University of XX, 2001.

[3] W. R. Bezanson. Performance support: Online, integrated documentation and training. In *ACM 13th International Conference on Systems Documentation*, pages 1–10, 1995.

[4] B.-I. Cho, J. A. Michael, A. A. Rovick, and M. W. Evens. A curriculum planning model for an intelligent tutoring system. In *Proceedings of the Twelfth Florida Artifical Intelligence Symposium (FLAIRS '99)*, pages 197–201, Hotel Royal Plaza, Walt Disney World Village, Orlando, Florida, 1999.

[5] E. Cloete and L. Miller. Structured Mapping of Digital Learning Systems. In *Proceedings of SAICSIT 2001. Annual Conference of the South Africa Institute for Computer Scientists and Information Technologists*, Pretoria, South Africa, Sept. 2001.

[6] H. C. de Kock and S. A. du Plessis. An Electronic Performance Support System to Provide Computer Based Coaching of Parallel computing, Expert Systems, Induction and Solution Methods for Linear Programming. *Proceeding of Fourth CBE/CBT Conference and Workshop*, 1:137–145, October 1996.

[7] A. Dix, J. Finlay, G. Abowd, and R. Beale. *Human-Computer Interaction.* Hemel Hampstead: Prentice Hall International, United Kingdom, 2nd edition, 1998.

[8] A. J. Dix. Closing the Loop: modelling action, perception and information. In T. Catarci, M. F. Costabile, S. Levialdi, and G. Santucci., editors, *AVI'96 - Advanced Visual Interfaces*, pages 20–28. ACM Press, 1991.

[9] E. Frøkjaer, M. Hertzum, and K. Hornbæk. Measuring Usability: Are Effectiveness, Efficiency and Satisfaction Really Correlated? *CHI 2000 Conference Proceedings*, 1(1):345–352, April 2000.

[10] J. L. Gibson, J. M. Ivancevich, and J. H. Donnelly. *Organizations Behaviour Structure Processes.* Irwin, Homewood IL 60430 Boston, 7 edition, 1991.

[11] J. S. Goodman. The Interactive Effetcs of Task and External Feebdack on Practice Performance and Learning. *Organizational Behavior and Human Decision Processes*, 76(3):223–252, 1998.

[12] J. Hugo. Interface Design for Performance Support Systems. *Proceeding of Fourth CBE/CBT Conference and Workshop*, 1:111–128, October 1996.

[13] ISO924-1. Iso924-1: Ergonomic requirements for office work with visual display terminals. Technical report, ISO, 1997.

[14] D. R. Johnson, M. Ruzek, and M. Kalb. Earth System Science Education: an Internet-based electronic curriculum. In T. I. Stein, editor, *1995 International Geoscience and Remote Sensing Symposium, IGARSS '95. Quantitative Remote Sensing for Science and Applications*, pages 572–574 (vol. 1), 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1995. IEEE Computer Society Press.

[15] H. Kumar, C. Plaisant, and B. Shneiderman. Browsing Hierarchical Data with Multi-Level Dynamic Queries and Pruning. Technical Report CS-TR-3474, HCIL Department of Computer Science University of Maryland, Mar. 1995.

[16] A. Lesgold. Towards a theory of curriculum design for use in Intelligent Instructional Systems. In H. Mandl and A. Lesgold, editors, *Learning Issues for Intelligent Tutoring Systems*, pages 114–137. Springer Verlag, New York, 1988.

[17] I. Lokuge and S. Ishizaki. GeoSpace: An Interactive Visualization System for Exploring Complex Information Spaces. *ACM CHI'95 Proceedings*, pages 409–414, 1995.

[18] C. H. Mawhinney and J. S. Morrell. The IS Undergraduate Curriculum: Predicting Student Outcomes in the IS' 97.5 Course (Programming, Data, File, and Object Structures). In R. Agarwal, editor, *Proceedings of the ACM SIGCPR Conference (SIGCPR-98)*, pages 133–135, New York, Mar. 26–28 1998. ACM Press.

[19] J. Nielsen. *Coordinating User Interfaces for Consistency.* Academic Press, 1 edition, 1989.

[20] D. A. Norman. *The Design of Everyday Things.* MIT Press, 2000.

[21] M. S. Olivier. *Information Technology Research: A practical guide.* Rand Afrikaans University, Johannesburg, 1999.

[22] G. M. Piskurich, P. Beckschi, and B. Hall. *The AST Handbook of Training Design and Delivery.* McGraw-Hill, 1 edition, 2002.

[23] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Schneiderman. LifeLines:Visualizing personal histories. Technical Report CS-TR-3523, HCIL Department of Computer Science University of Maryland, 1995.

[24] Registrar. *Year book 2002: Part 3*, volume 1. XX Press, 2001.

[25] S. Rosenbaum, J. A. Rohn, and J. Humburg. A Toolkit for Strategic Usability: Results from Workshops, Panels and Surveys. *CHI 2000 Conference Proceedings*, 1(1):337–345, April 2000.

[26] J. Rubin. *Handbook for Usability Testing: How to Plan, Design and Conduct Effective Tests.* John Wiley & Sons, 1994.

[27] R. Spence. *Information Visualization.* Addison-Wesley, 2001.

[28] A. Sutcliffe. Task-related information analysis. *International Journal of Human-Computer Studies*, 47:223–257, 1997.

[29] P. Terrier and J.-M. Callier. Depth of processing and design-assessment of ecological interfaces: Task analysis. *International Journal of Human-Computer Studies*, 50:287–307, 1999.

[30] T. J. van Dyk. Usability and Internet Banking. Master's thesis, University of South Africa, 1999.

[31] D. van Greunen and J. Wesson. Formal Usability testing Informing Design. In *Proceedings of SAICSIT 2001. Annual Conference of the South Africa Institute of Computer Scientists and Information Technologists*, Pretoria, South Africa, Sept. 2001.

[32] J. Wesson. *An Investigation into Design Methodologies for Usability:A Case study approach.* PhD thesis, University of Port Elizabeth, University of Port Elizabeth, Feb. 1997.

[33] W. E. Woodson, B. Tillman, and P. Tillman. *Human Factors Design Handbook.* McGraw Hill, 2nd edition, 1992.