Zazu — Investigating the Differences Between Experts and Novices in Using an Advisory Support Tool

JUDY VAN BILJON
University of South Africa
and
KAREN RENAUD
University of Glasgow
and
ESTELLE DE KOCK
University of South Africa

Curriculum planning is undertaken by University students at least once a year. Some Universities allow students to study part-time and to make up their degree in successive years by accumulating a few modules each year. The curriculum-planning process becomes far more complicated in this case since students can enrol for modules at different levels concurrently. Exacerbating factors are phased-out modules, complicated module requirements and inexperienced advising staff. When these factors are taken into account the planning process becomes far more error-prone.

A prototype advisor support tool, named Zazu, was developed to address the problems related to paper-based advising. In evaluating the tool a marked difference between the user-satisfaction of expert and novice advisors emerged. This paper discusses the differences between experts and novices in using the advisory support tool, and draws conclusions about how best to structure user interfaces for each advisor category based on the results obtained from the Zazu study.

Categories and Subject Descriptors: H.5.2 [**User Interfaces**]: Graphical User Interfaces, Interaction Styles, Screen Design, User-centred design; H.4.2 [**Information Systems Applications**]: Decision Support

General Terms: Advisory Tool

Additional Key Words and Phrases: User Interface, Student Advice, Experts vs Novices

1. INTRODUCTION

The University of South Africa (Unisa) offers part-time as well as full-time education. Courses are presented in the form of modules that students register for in order to obtain credits for degree or diploma purposes. Unisa presents information about the modules each department offers in a set of books which constitute the yearly calendar. Students must obtain credits for one or 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.

Unisa 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 possible to make ill-advised choices when first enrolling which can prevent a student from enrolling for desired subjects in later years. The current format of

[1]A major subject usually comprises 4 or 5 modules at third year level plus prerequisites at previous levels.

Author Addresses:
J van Biljon and E de Kock, Department of Computer Science and Information Systems, University of South Africa, P O Box 392, UNISA, 0003, South Africa; vbiljja@unisa.ac.za
K V Renaud, Department of Computing Science, University of Glasgow, 17 Lilybank Gardens, Glasgow, G12 8RZ, United Kingdom.

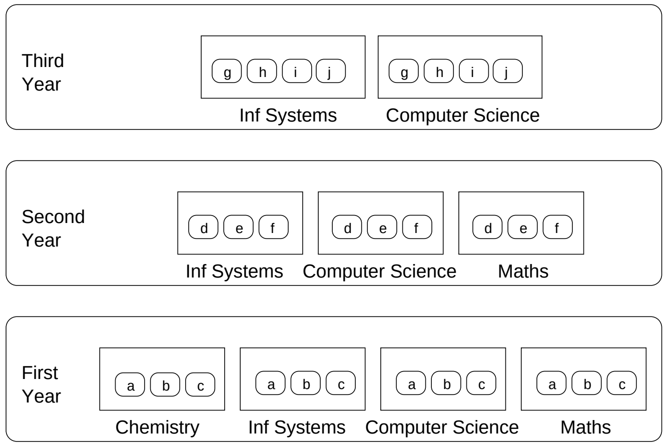

Figure 1. Degree Structure

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 calendar user is unique.
(3) all information is related to, and ordered by, module code. The Unisa module codes are cryptic, in the form of, for example, COS311 and COS419, and tend to be meaningful only to Unisa staff. Using module codes is logical and sensible because it fits into their frame of reference — modules being referred to as such within the University. It presents many problems to new students and lecturers, however, because the codes tend to confuse and detract from the information being presented in the calendar.

The following section gives 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 Unisa calendar [Registrar 2001] 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 Unisa calendar for COS211 [Registrar 2001]:

> 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 for the module COS212 in the Unisa calendar [Registrar 2001]:

> 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 type of 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: COS114, COS112, INF101; or COS114, COS112, COS113; or COS111, COS112, COS113; or 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 South Africa, including two 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 [Cho et al. 1999; Johnson et al. 1995; Lesgold 1988; Mawhinney and Morrell 1998], often based on computerised student performance measurement. Four main factors contribute to the uniqueness of the situation at Unisa:

(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 Unisa 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 old modules for pre-requisite purposes.

(3) Students can take a mixture of modules at any of three levels providing the prerequisites have been met, making the assessment of their current status and recommendation of their yearly curriculum tricky and error-prone.

(4) The University's policy is to allow students the flexibility to arrange their curriculum to suit their home and work circumstances and to set their own pace accordingly.

Hence it was decided that the situation warranted the development of a computer-based support system. In developing such software one could target different types of users: advisors, students or even the producers of the current yearly calendar book. It was decided that we would produce software aimed at supporting advising lecturers only. The lessons learnt from experiences of advisors using this software will then be used in the second phase of the project to produce curriculum-planning software directly for students. The prototype of this tool will henceforth be referred to as the Zazu[2] system.

Section 2 addresses task analysis for the performance support computer tool. Section 3 provides an overview of the Zazu prototype. Section 4 describes the prototype evaluation procedure and give the results of the evaluation. Section 5 concludes.

2. TASK-ANALYSIS FOR THE ZAZU SYSTEM

It is advisable for task[3] analysis to be done for performance support tools since it elicits knowledge for design purposes, provides a reference for evaluation and ensures the efficiency and accuracy of the resulting system [Hugo 1996]. Task analysis will help to ensure that the prototype will support the advisor's tasks.

Task analysis, in the context of interface design, studies user needs and the definitions of functional requirements [van Dyk 1999]. A list of frequently-asked questions and problems experienced with advising using the existing system was compiled based on lecturers' previous experiences. This list, together with the task analysis done on the existing system, provide 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.

(2) Curriculum re-assessment for re-registering students.

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 three distinct types of decisions to be supported [Gibson et al. 1991]:

(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.

[2]The name Zazu alludes to the bird Zazu in the film *Lion King* from whom the lion Simba received advice and support.
[3]By task we mean a *unit of human activity which is carried out in order to achieve a specific goal* [Wesson 1997].

(3) *Wicked* decision-making, which no amount of information can ease [Simon 1969]. For example, the advisor cannot know how well a student will cope in a particular module based on previous performance and personal circumstances. Some work is currently taking place using Bayesian nets to support decision making in the face of this type of problem [Dekhtyar et al. 2001] but this work is currently in the preliminary stage and details of their interface have not yet been published. At this stage we do not intend to provide support for this type of decision.

The tasks identified to support advising 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 prototyping effort focused on implementing representative tasks from each kind of the above-mentioned tasks. The programmed and non-programmed tasks are iteratively interspersed. An advisor needs the system to provide timely, current, context-sensitive information for decision making (non-programmable). However, the selection based on the decision could be followed by a cumbersome manual task which could ideally be programmed. At this point the advisor may need to obtain more information and so the process recurses until the curriculum-planning 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 advisor selects the module and all the pre-requisites are automatically displayed. We use an adaptation of a simple iterative computer operating model (Figure 2) proposed by van Dyk and Renaud [van Dyk and Renaud 2001] which extends an example first published in Woodson, Tillman and Tillman [Woodson et al. 1992], which is in turn based on the set of USA Military User Interface Design Guidelines. This type of model is suitable to be used as the basis for formulating a strategy for task analysis. Norman's classical execution-evaluation model [Norman 98] has similar stages. This task model is, of course, the optimal way of carrying out the task and is not intended to be prescriptive or to suggest that this is the way that things would have to occur.

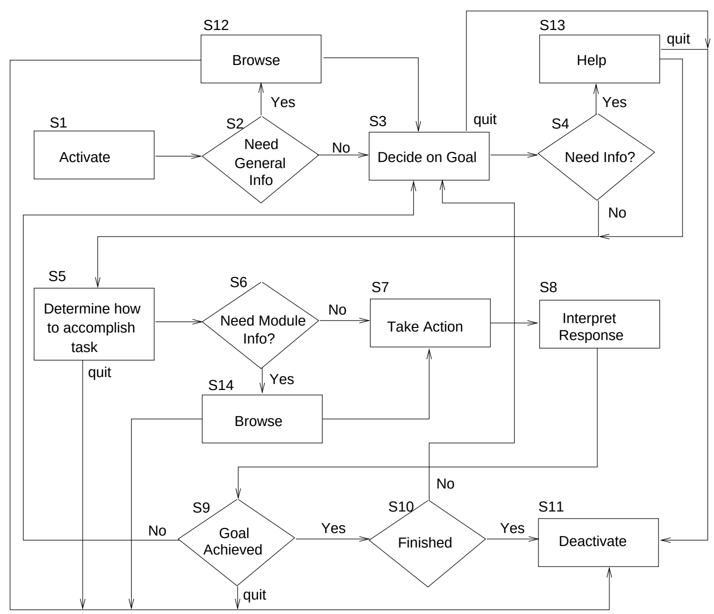

Figure 2. Task Analysis

The following should be noted about Figure 2 specifically when considering the tasks for the prototype:

(1) S12 and S14 represent the non-programmable tasks of browsing for information.
(2) 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)
(3) S7 implies that the user either chooses a module or indicates that a module has been completed.
(4) 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-labeled as goal adjustment.
(5) 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.

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

(7) S10 is reached at the achievement of every goal and S11 refers to the situation where the user is *finished* with the advice session.

In analysing the advising task it became clear that the expert and novice advisors operated in distinctly different ways:

—Experts were familiar with the contents of the calendar, understood the module requirements and had internalised the structure of the various degree and diploma courses. Much of their knowledge was not explicitly recorded in the calendar, and had been accumulated over the years. Experts were not always aware of changes in the calendar and those who did not keep track of the changes were in danger of inadvertently giving incorrect advice.

—Novices attempted to answer queries without this extensive background knowledge. They often resorted to consulting experts when they could not link together the disparate sources of information to answer a difficult query.

It seemed that the process of maturing into an expert advisor was long and arduous and since the experts were so familiar with all the intricacies of the calendar they did not always understand the difficulties novices had with the material. Unfortunately the calendar was always produced by experts.

The following section will describe the Zazu prototype system.

3. THE ZAZU SYSTEM

This section will explain the design decisions made in the development of the Zazu system.

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 advisor to use the system to support 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 [Albers 1998]. One needs rather to find a visualisation mechanism which *conveys* knowledge — making the difference between a simple data generator and an information searcher [Albers 1998].

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 [Albers 1998]. The interface should therefore be designed to "lead, follow, or get out of the way" [Piskurich et al. 2002]. 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.

Programmable versus Non-Programmable Tasks. The interface needs to support both fact-finding and decision-making. These activities are interspersed and it would be ideal to offer them from the same interface. However, supporting decisions about modules to be taken based on modules already completed necessitates the display of only a particular subset of all the available modules. The dynamic nature of the display shows the advisor which modules are available to the student based on modules already completed. Unavailable modules do not clutter the display.

However, if this mechanism is used and the advisor wants to get information about some module which is not immediately available he or she needs to be able to activate an information display for that module.

The decision was therefore made to provide two distinct windows — one to support fact-finding and another to support decision-making. The user is able to switch between the two windows by means of tabs. The windows will be referred to as follows:

—*Fact-finding window* — this has a top-down structure and supports programmable tasks. This is labeled "See Requirements".

—*Decision-making window* — this has a bottom-up structure and supports both programmable and non-programmablenon-programmable tasks. This is labeled "Recommended Possibilities".

Module Naming — In designing the interface a decision had to be made about how to name the modules in the interface. Experts insisted on the use of module codes while the novice advisors had difficulties remembering the code-module content associations. One option would have been to offer a customisable interface which would display either module codes or module names. While this would satisfy experts it would not assist novices to assimilate the module codes, which must be one of the goals of this software since a working knowledge of these codes is essential to function adequately within the University. Since novices significantly outnumber experts it was decided that *both* the module code and the module name would be displayed, the former to satisfy the experts, and the presence of both to assist and train novices.

Module Status — The following should be displayed:
Module level or year. Modules are represented as square building blocks, arranged in three levels with the first year level at the bottom of the screen (Figure 4). This uses the metaphor of constructing a building as being similar to the process of accumulating modules for a degree course. The student 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.

Module colours were chosen in line with the 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).

Whether the module is available to the student. Both dynamic filters, that act on information as it is collected, and static filters, that act on information that has already been captured and internalised [Piskurich et al. 2002], can be used. The act of displaying only the accessible modules at the outset in the decision-making window performs 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.

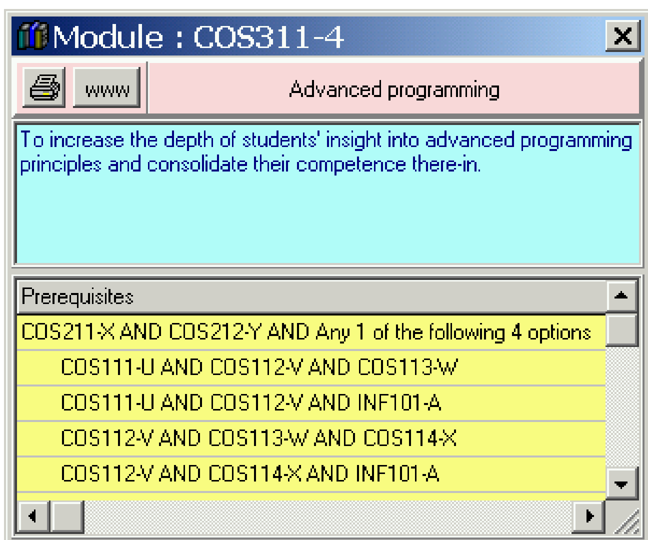

Figure 3. Module Description

Pre-requisites & concurrent modules. Displaying this information is not a matter of simply highlighting those modules that are pre-requisites for a specific module. As can be seen from Figure 3, the pre-requisites can be one of a list of different sets of modules. One could highlight one set at a time and allow the advisor to step through them, but this could be very confusing since the advisor would struggle to remember the first set by the time the last set is reached. It was decided to list all pre-requisites in the traditional format in a pop-up window and this is done via the fact-finding window. This is already an improvement on the information recorded in the calendar since it is easier to isolate a particular set from the pre-requisite requirements. This information is displayed using module codes since the use of module names made the display overly verbose and difficult to parse correctly.

Since the above-mentioned mechanism for displaying pre-requisites was less than optimal, we decided to augment this display by indicating these requirements dynamically in the decision-making window, linking it to available module actions.

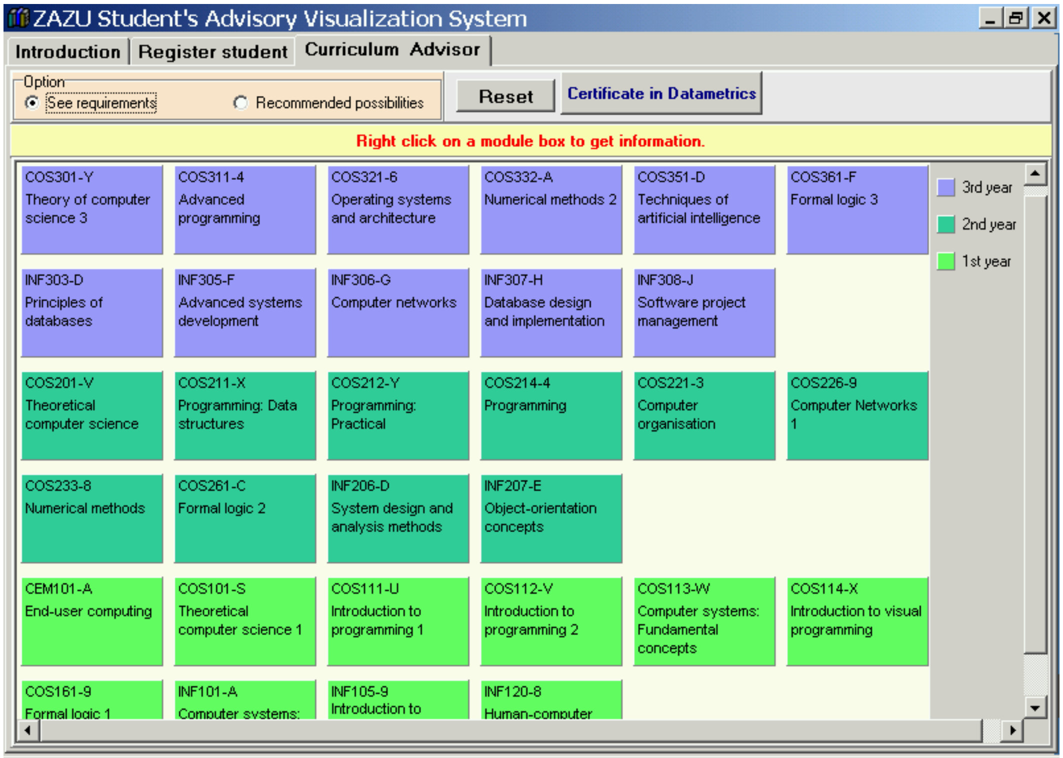

Figure 4. Representing Modules

Only modules that have no pre-requisites are displayed at the start. Each module has two check boxes, one is labeled *completed* and the other *chosen* (see Figure 5). Advisors using this facility will select all modules already completed by a particular student thereby indicating that the modules have been completed. If the *completed* check box is selected, two actions take place (shown in Figure 5):

—*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 advisor's attention on modules of importance to their decision-making process [Sutcliffe 1997]. 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. This list of completed modules and chosen modules supports accountability as it provides the evaluator with some evidence of the recommendation based on module codes given.

Screen Layout. 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 make users feel disoriented [Kumar et al. 1995]. We also experienced this problem in initial usability testing and so a module is thus allocated a position on the screen and will always appear in that position.

The top of the screen operates as a navigation control area to make the program state (mode) evident. The interface was designed to support the advisor in recognising the two main modes: *browsing* for general information in the fact-finding window; or making *selections* for viewing module requirements-related information in the decision-making window.

Plaisant [Plaisant et al. 1995] and Spence [Spence 2001] 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.

Linking to additional module information. The information traditionally included in the calendar is aimed at informing about module content and requirements. A student could conceivably require more information, such as the lecturer name or the name of the prescribed book or a more detailed description of the module content. These are already provided on the department's website and Zazu thus links directly to these pages by means of a button labeled *www*.

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 advisor to choose these modules for future enrolment, while still acknowledging completion of such modules. The system will automatically display the modules for which the expired module is a pre-requisite so that the student can still progress toward their qualification.

Prototype Functionality. We wanted to obtain information about how lecturers would make use of the prototype and identify the mode of usage they found most helpful before developing a tool with full functionality. Thus certain features were purposely excluded which could conceivably have added to the completeness of the system. For example, the advisor has to manually select the modules a student has completed while this is trivial to automate. However, most advice is given to first year students and it was deemed important to test the building of "what if" scenarios from scratch. Focusing separately on each mode of the system to determine ease of learning, ease of action and error avoidance was a higher priority than supporting full functionality of the prototype.

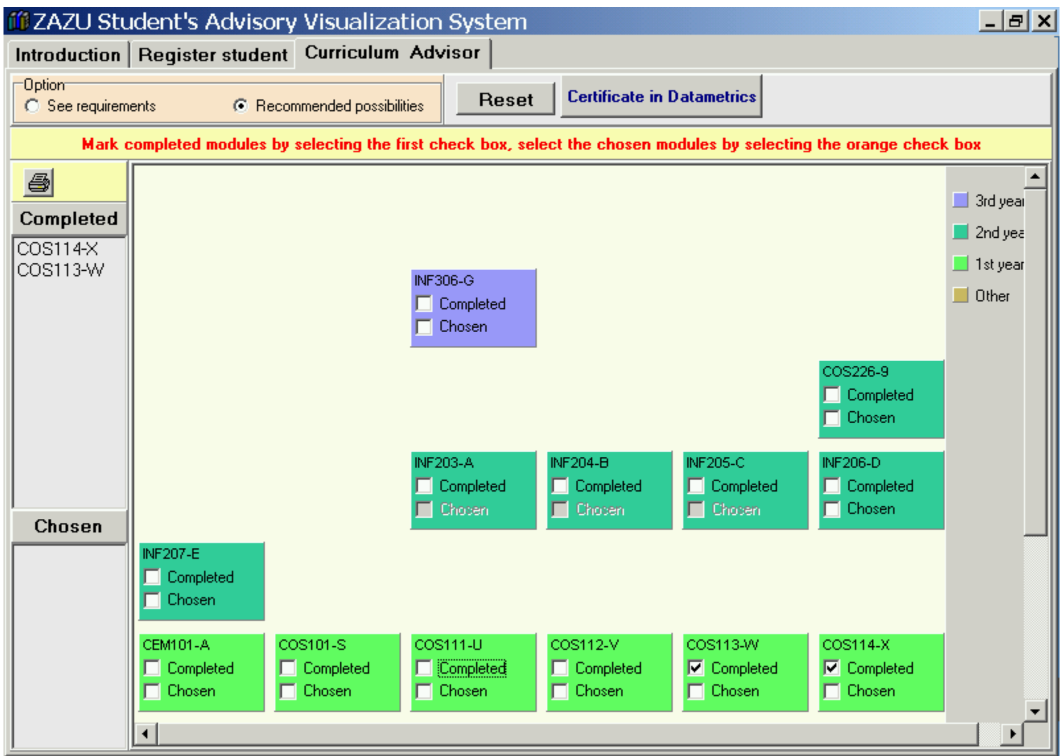

Figure 5. Decision-making Screen

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, dynamic nature may well help users to retain the information presented in the calendar better than they currently do [Terrier and Callier 1999]. The following section will report on the evaluation of Zazu.

4. EVALUATION

An initial evaluation of the Zazu system was carried out to test the efficacy thereof in supporting the advising process [van Biljon et al. 2002]. The outstanding features of the evaluation were the perceived difference between the user satisfaction experienced by novice and expert advisors respectively and the fact that experts performed much worse than expected with the current paper-based calendar system.

During this evaluation we also were struck by the obvious difference in the response to the system by expert and novice advisors. Our preliminary observations seemed to indicate that experts were so entrenched in the calendar-based mode of advising and that they felt they had no need of Zazu. Novices, on the other hand, felt that they benefited considerably from Zazu and were very enthusiastic about the project. It was therefore decided that a further investigation would be carried out to determine the nature of the differences between expert and novice advisors' experiences with Zazu. In carrying out this investigation we wanted to answer the following questions:

(1) which techniques experts used that enabled them to give advice more easily.

(2) whether there was any perceivable difference between performance improvement of experts and novices using Zazu.

(3) whether the different groups favoured one or the other of the provided Zazu interfaces. (The fact-finding screen offers an interface which is similar to the calendar-based information, whereas the decision-support interface offers a dynamic interface which seeks to support decisions directly.)

(4) whether different groups expressed a preference either for the calendar-based or for the advisory system. If they preferred the calendar-based system, to determine the reasons for this.

The major challenge of the Zazu system evaluation would be to answer the above questions thus enabling us to strive to meet the requirements of both the expert and novice advisors more effectively in the next phase of Zazu.

4.1 Usability evaluation methods

The ISO 9241 [ISO924-1 1997] standard provides guidance on how to specify and measure the usability of products, using stated factors. These factors include efficiency, satisfaction and effectiveness of software in assisting users achieving their goals. Frøkjaer *et al.* [Frøkjaer et al. 2000] presents evidence to suggest that effectiveness, efficiency and satisfaction should be considered independent aspects of usability and should be measured independently.

Usability analysis can be conducted in two ways: 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 [Wesson 1997]. In conducting the latter three methods can be applied: *heuristic* evaluation, *observation* and *empirical* evaluation [van Greunen and Wesson 2001]. Usability testing based on observation — whether inside or outside of a lab facility — has been rated highest as an effective usability methodology aimed at strategic impact [Rosenbaum et al. 2000]. The Zazu evaluation thus combines observations with empirical evaluation by means of a questionnaire as discussed in the following section.

4.2 Evaluation Procedure

Researchers either recommend a focus group of 8-12 participants [van Greunen and Wesson 2001] or 4-8 participants [Rubin 1994] for formal usability testing. We chose to carry out the experiment with ten lecturers. Two sets of questions (Appendix A) were used randomly to minimize the effect of specific question set on the result. The first 5 questions relate to calendar content and the latter 5 relate to the evaluators satisfaction with the system. Five lecturers answered one questionnaire using calendar books first followed by the other questionnaire using Zazu, while the other five used Zazu first and then the calendar. The lecturers were observed while completing the questionnaires.

In the previous evaluation a significant difference was discovered between the preferences of expert and novice advisors and this questionnaire includes some questions aimed at investigating these preferences.

4.3 Classification of Advisors

In the first evaluation ten lecturers were requested to rate themselves as expert or novice and to state their experience in curriculum planning in years. Only three lecturers with respectively 11, 15 and 20 years' experience rated themselves as experts. The rest of the group, with between 1 and 4 years of experience, rated themselves as novices. Subsequently, in this second evaluation ten lecturers were requested to rate themselves as *novice, intermediary* or *expert* and to state their experience in giving advice in full years. The following values for years of experience were obtained:

Self-Classification	Years
expert	2,6,30
intermediary	2,3,10,13,18
novice	0,2

Since a duty roster is set for giving advice on curriculum planning and most lecturers therefore get approximately the same exposure to this duty, it was decided, for the purposes of this investigation, to classify *experts* as advisors with more than five years of experience in curriculum planning. The expert category includes lecturers who rated themselves as experts and those with more than 5 years of experience who rated themselves as intermediate. Lecturers with less than 5 years of experience were rated as novices. Our classification thus produced the following sets were: expert [6,10,13,18,30] years and novice [0,2,2,2,3] years.

4.4 Evaluation Metrics

4.4.1 *Effectiveness*

The ISO 9241 [ISO924-1 1997] standard further refines effectiveness into completeness and correctness, so these aspects can be reported separately:

—*completeness*: The Zazu system did improve completeness. For example, when a student has completed some modules and wants to know what options are available (see question 3 in the questionnaire), all users improved the completeness of their solutions by using Zazu. Furthermore, the Zazu system provided implicit information which novices did not detect when using the calendar. For example, the module INF307-H has no prerequisites but it does have another module, INF303-D, as registration requirement. This means that it effectively has the same prerequisites as INF303-D, a subtlety not detected by all novices.

—*correctness*: One of the 5 novices and one of the 5 experts got all the answers correct using the paper-based calendar system. In contrast four novices and three experts got all the answers correct using the Zazu system. The efficiency improvement was better for novices than for experts. All the advisors who made mistakes with the Zazu system were using the fact-finding window at the time. This mode requires less interaction as advisors do not have to select each individual modules but it requires more complicated decision-making and reasoning skills and advisors risk missing information — as evidenced by the results.

4.4.2 *Efficiency*

The evaluators' average time improvement by using the Zazu system is indicated in Tables I and II. Four of the 5 novices' times improved. Only two of the 5 experts improved their performance - one significantly and the other marginally. Two took slightly longer with Zazu and one stayed the same. The time difference is depicted in the graphs in Figure 6. One of the obvious effects of Zazu is that it reduces the variation in time that is evident in calendar-based advising.

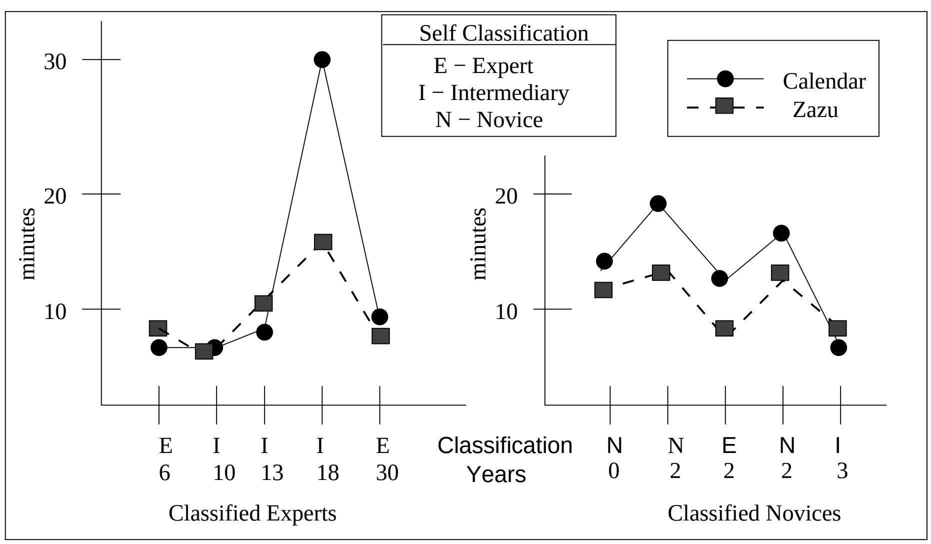

Figure 6. Time Difference

4.4.3 *Satisfaction*

The post-test questionnaire (Appendix B) was used to compare the advisors' experience with the different systems. All evaluators were positive about the system. The Zazu system improved confidence only for two experts, and three reported no difference. All novices were more confident about their performance with the Zazu system. Tables I and II indicate that both novices and experts find data visualization useful. It appears that experts are satisfied with transient, information-intensive displays (for example, Figure 3) which they integrate themselves. They appear to use their pre-existing knowledge to support decisions, using Zazu merely as an information-gathering device. Novices, who do not have an internalised semantic structure, prefer all the information linked together by Zazu so as to avoid the risk of missing important information.

Property	Calendar	Zazu
Completeness	60%	100%
Correctness Achieved	20%	60%
Average Time improvement		12%
Best Data Visualisation	0%	100%
Format preference	40%	40%

Table I. Experts' Satisfaction

Property	Calendar	Zazu
Completeness	20%	100%
Correctness	20%	80%
Average Time improvement		35%
Best Data Visualisation	0%	100%
Format preference	0%	100%

Table II. Novices' Satisfaction

4.5 Observations

The following was observed:

(1) The decision-making window displays only the set of modules with no requirements initially, and modules are then added as their prerequisites are selected. Novices were happy with the incremental increase in information but some experts would have preferred to use the fact-finding window with pre-requisites being shown by highlighting or colour coding. The decision-making interface appears to experts to be more tedious but both experts and novices who used this mode made no mistakes. This confirmed our previous findings about the performance and efficiency improvement provided by the Zazu system but showed that the improvement was significantly greater for novices. This difference appears to be at least partly due to a certain resistance to the use of advising software support evidenced by expert advisors. Another factor that influenced performance was caused by the mechanism for correcting errors in the printed manual. This was done by sending a general email out to all advisors. It was found that some advisors had neglected to update their calendars and this caused errors.

(2) The inclusion of the web browsing facility caused problems and reflected badly on the system. Novices who selected this option got lost in the web pages and poorly designed web pages were erroneously associated with the system.

(3) The scalability of the design provided challenges. When data is represented by items in scrollable list boxes the addition of new elements causes few problems. However, when a new object is dynamically created for each new data item in the central area and the presentation properties such as colour and location of the data item depends on the specific data item, careful planning is concerned.

(4) Many factors influence a person's ability to provide advice in curriculum planning and the correctness and time-efficiency achieved did not always correspond with the advisor's self-classification which seemed to be primarily based on years of experience. For example, some self-classified experts made more mistakes than some self-classified intermediates. Furthermore, people with almost the same experience classified themselves differently. Apart from the practical problems involved, providing different performance support systems to experts and novices is not a viable option since there is no clear division between experts and novices.

(5) In order to build confidence in a performance support system such as Zazu it is important to have some history of the user's actions in order to check performance. This is necessary since users cannot always remember what they did and what state the system is in. The list of modules chosen displayed in the decision-making view is useful in this regard.

4.6 Answers to Evaluation Questions

We can now consider the answers to the posed questions:

(1) *the kind of techniques experts used* — Experts have many years experience advising from the calendars, with all their imperfections. They have obviously overcome the difficulties and now developed the skill of integrating the masses of information from all over the calendar in order to answer queries and provide advice. Zazu provides a new paradigm of obtaining this information, one where the information is integrated on behalf of the advisor and presented as a *fait accompli*. The advisors resisted the paradigm change and used the fact-finding window, which offers information as an animated calendar, and used a backward-chaining strategy to answer the questions. They also obviously relied on their previous knowledge. Some experts found Zazu offering information that conflicted with what they believed to be correct and challenged the system. Upon consulting the calendar they found that something had changed and that their fore-knowledge had been faulty. This over-reliance on fore-knowledge, together with the resistance to Zazu's automation, caused the errors they made.

(2) *the performance difference between experts and novices* — Experts made more errors than one would have expected given their years of experience. This is probably due to the fact that they rely on their heuristic knowledge and do not check their facts. This causes problems since the information that decisions need to be based on changes every year. Novices were less sure of themselves and checked up on all options. The mistakes they made were due to inexperience and due to the fact that facts not explicitly linked in the calendar had to be connected in order to answer a question.

(3) *which interface was favoured by the different groups* — experts find the Zazu system useful but prefer more information-intensive displays than novices. They tended to make more use of the fact-finding window. They used this window and made the decisions based on information they integrated themselves. This pre-established internal mental model precipitated most of the errors experts made. Novices tended to make more use of the decision-making window — allowing Zazu to support their decision-making process.

(4) *preferences of different groups* — the novices all preferred Zazu to the calendar. Experts were more divided with two preferring Zazu, two preferring the calendar and one being ambivalent. The reasons for this are clear from the observations of experts using Zazu as an animated calendar, and of novices' tendency to utilise its full capabilities.

5. CONCLUSION

The findings of this study indicate that to accommodate both experts and novices, what is really needed is a system that reduces risk, thus protecting advisors and students from the consequences of incorrect advice. This functionality incidentally protects *both* novices and experts in the light of the findings of this study. Novices seemed willing to accept constraints imposed by the decision-making window in order to minimize risk but experts were more divided on the issue. The importance of ease-of-use was emphasised throughout the evaluation of Zazu, and the fact that Zazu was so much easier for novices to use appears to be what makes it so attractive to them.

The questionnaire evaluation verified the observation that subject-experts and -novices have very different needs where performance support is concerned. Novices find the decision-support helpful while experts prefer a data-intensive display.

6. FUTURE WORK

The initial prototype was developed for use by lecturers only and this limited the number of evaluators. Our findings with experts and novices can be extrapolated to extending and testing Zazu for use by students via the Web. Students are the ultimate novices and it is essential for the expert/novice interface requirements to be resolved before delivering Zazu to students. Further research should include work on adapting the interface to reflect the multi-cultural composition of the student population. Once the system is operational and advice sessions are recorded this could provide valuable information about the general student queries and the performance paths of students who had curriculum advice.

Acknowledgments

This project was partly supported by a grant from the Faculty of Science at the University of South Africa. We also acknowledge the contribution of Aurona Gerber. Our thanks to Phil Gray for his helpful comments on an earlier draft of this paper.

REFERENCES

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

CHO, B.-I., MICHAEL, J. A., ROVICK, A. A., AND EVENS, M. W. 1999. A curriculum planning model for an intelligent tutoring system. In *Proceedings of the Twelfth Florida Artifical Intelligence Symposium (FLAIRS '99)*. Hotel Royal Plaza, Walt Disney World Village, Orlando, Florida, 197–201.

DEKHTYAR, A., GOLDSMITH, J., LI, H., AND YOUNG, B. 2001. The bayesian advisor project: Modeling academic advising. Tech. Rep. TR-323-01, University of Kentucky.

FRØKJAER, E., HERTZUM, M., AND HORNBÆK, K. 2000. Measuring Usability: Are Effectiveness, Efficiency and Satisfaction Really Correlated? *CHI 2000 Conference Proceedings 1,* 1 (April), 345–352.

GIBSON, J. L., IVANCEVICH, J. M., AND DONNELLY, J. H. 1991. *Organizations Behaviour Structure Processes*, 7 ed. Irwin, Homewood IL 60430 Boston.

HUGO, J. 1996. Interface Design for Performance Support Systems. *Proceeding of Fourth CBE/CBT Conference and Workshop 1*, 111–128.

ISO924-1. 1997. Iso924-1: Ergonomic requirements for office work with visual display terminals. Tech. rep., ISO.

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

KUMAR, H., PLAISANT, C., AND SHNEIDERMAN, B. 1995. Browsing Hierarchical Data with Multi-Level Dynamic Queries and Pruning. Tech. Rep. CS-TR-3474, HCIL Department of Computer Science University of Maryland. Mar.

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

MAWHINNEY, C. H. AND MORRELL, J. S. 1998. The IS Undergraduate Curriculum: Predicting Student Outcomes in the IS' 97.5 Course (Programming, Data, File, and Object Structures). In *Proceedings of the ACM SIGCPR Conference (SIGCPR-98)*, R. Agarwal, Ed. ACM Press, New York, 133–135.

NORMAN, D. A. 98. *The design of everyday things.* MIT Press, London, England.

PISKURICH, G. M., BECKSCHI, P., AND HALL, B. 2002. *The AST Handbook of Training Design and Delivery*, 1 ed. McGraw-Hill.

PLAISANT, C., MILASH, B., ROSE, A., WIDOFF, S., AND SCHNEIDERMAN, B. 1995. LifeLines:Visualizing personal histories. Tech. Rep. CS-TR-3523, HCIL Department of Computer Science University of Maryland.

REGISTRAR. 2001. *Year book 2002: Part 3.* Vol. 1. XX Press.

ROSENBAUM, S., ROHN, J. A., AND HUMBURG, J. 2000. A Toolkit for Strategic Usability: Results from Workshops, Panels and Surveys. *CHI 2000 Conference Proceedings 1,* 1 (April), 337–345.

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

SIMON, H. A. 1969. *The Sciences of the Artificial.* The M.I.T Press, Cambridge, Massachusetts.

SPENCE, R. 2001. *Information Visualization.* Addison-Wesley.

SUTCLIFFE, A. 1997. Task-related information analysis. *International Journal of Human-Computer Studies 47*, 223–257.

TERRIER, P. AND CALLIER, J.-M. 1999. Depth of processing and design-assessment of ecological interfaces: Task analysis. *International Journal of Human-Computer Studies 50*, 287–307.

VAN BILJON, J. A., RENAUD, K. V., AND DE KOCK, E. 2002. Zazu: Supporting curriculum planning in a distance-education environment. Tech. Rep. UNISA-TR-2002-1, University of South Africa. January.

VAN DYK, T. J. 1999. Usability and Internet Banking. M.S. thesis, University of South Africa.

VAN DYK, T. J. AND RENAUD, K. V. 2001. Task based evaluation Metrics for E-commerce sites. Tech. Rep. UNISA-TR-2001-26, University of South Africa.

VAN GREUNEN, D. AND WESSON, J. 2001. 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.

WESSON, J. 1997. An investigation into Design Methodologies for Usability:a case study approach. Ph.D. thesis, University of Port Elizabeth, University of Port Elizabeth.

WOODSON, W. E., TILLMAN, B., AND TILLMAN, P. 1992. *Human Factors Design Handbook*, 2nd ed. McGraw Hill.

Appendix A - Example of a Question Set

(1) What are the prerequisites for INF307-H?

(2) A student plans to take COS332-A on the 3rd level. Does the following combination of modules provide the minimum required prerequisites? COS233-8; COS111-U, MAT111-N; MAT112-P; MAT101-L; MAT103

(3) A student plans to take INF305-F on the 3rd level. Does the following combination of modules provide the required prerequisites? INF105-9; INF206-D; INF207-E; COS111-U, CEM101-A:

(4) A student has successfully completed the modules given below. What third year modules can the student select now? COS111, COS113, CEM101, INF120, INF203, INF204, INF105

(5) A student has successfully completed the following modules: COS111, COS113, CEM101, INF120. The student plans to register for INF207 and INF306. Is this registration in order?

(6) Is the following curriculum acceptable for the certificate in Datametrics? (Note: The certificate in Datametrics requires 10 modules of which at least 3 must be second or third year) COS111, COS112, COS113, CEM101, INF105, COS211, COS212, INF205, INF207, INF303

Appendix B - Questionnaire

(1) The Zazu system displays prerequisite modules as different options while the calendar books display it in a more compact format. Which display do you find more useful? (A) Calendar books (B) Zazu

(2) The Zazu system displays prerequisite modules for all the previous levels(second and first) while the calendar book displays only the previous level. Which display do you find more useful? (A) Zazu (B) Calendar books

(3) The Zazu system displays modules as screen objects with properties, do you find this useful or would it be better to save space by displaying modules as data items on lists. (A) Display objects (B) Display on lists

(4) In the "See recommendations" mode the Zazu system dynamically adds the modules that become available, as objects on the screen. Would it be sufficient to show these module codes in the list as done on the left of the display or should the modules also be displayed as objects. (A) Display only on lists (B) Display as objects too

(5) Which of the Zazu windows did you find more useful. (A) See Requirements (B) Recommended possibilities

(6) Which of the Zazu windows did you use the most? (A) See Requirements (B) Recommended possibilities

(7) The Zazu system does not provide the functionality to replace the calendar system but for the functions, which it does provide, what will you prefer to use: (A) Calendar books (B) Zazu

(8) Were questions easier or harder to answer using Zazu? (A) Easier (B) Harder

(9) How confident were you with the answers you gave, based on the calendar books? Express your confidence as a mark out of 5, with 5 as the highest.

(10) How confident were you with the answers you gave, based on the Zazu system? Express your confidence as a mark out of 5, with 5 as the highest.

(11) Suggest ways in which Zazu can be made more useful or more friendly.