ACIS 2002 Proceedings   Australasian (ACIS)

December 2002

Assessing a Web-Based Information System for Casual and Novice Users of External Statistical Data

Peter Hyland
University of Wollongong

Follow this and additional works at: http://aisel.aisnet.org/acis2002

Assessing a Web-Based Information System for Casual and Novice Users of External Statistical Data

P.N. Hyland

School of IT and Computer Science
University of Wollongong
peter_hyland@uow.edu.au

Abstract

This paper describes the development and testing of a Web-based Information System to provide casual and novice users with access to external statistical data. External statistical data is provided by sources including national statistical agencies, government departments and by many businesses. The system described in the paper integrates three recently developed technologies, OLAP, MDDB and the Web to provide a simple system that allows casual and novice users to manipulate statistical data by adding or removing variables, drilling down or rolling up the data and customising their view of the data by filtering out unwanted variables.

Keywords

External Statistical Data, Online Analytical Processing, OLAP, MDDB, Benchmarking

INTRODUCTION

Millions of dollars are spent annually by public sector organisations and private enterprises, gathering and publishing statistical data used by people **outside** of those organisations. These end-users regard such data is external statistical data (ESD) (e.g. Shawkey, 1990). Typical sources of ESD include international organisations, national statistical agencies, state and national governments and many businesses.

External statistical data (ESD) is made available because of its importance to users such as researchers, business analysts or members of the wider community. Users, such as business managers and researchers, are often **casual** users of ESD while many members of the wider community use ESD so infrequently that they remain **novice** users of ESD.

Little research has been done about these users of ESD, but some studies (ABS, 1993) have shown that there are a significant number of casual and novice users of ESD. Studies by Hyland and Gould (1998) suggest that many casual and novice users of ESD have problems with the systems used to disseminate ESD.

These problems stem from three sources:

- ESD is often provided as non-interactive, summary data. Users cannot change the level of aggregation of such data, add or delete variables or view information for subsets of the population – tasks which they often need to carry out.

- Some ESD is provided in relatively unaggregated form with the expectation that it will be manipulated using spreadsheets and statistical packages. Unfortunately, these packages often do not support the type of analysis carried out by casual and novice users.

- Relatively unaggregated ESD is often provided with custom-built packages to manipulate the data. Unfortunately, the packages are often difficult to learn and the existence of many such custom-built packages means that ad hoc users of ESD often have to learn several packages.

Casual and novice users of ESD need systems suited to the tasks that they carry out and providing a level of support appropriate to their skill and experience. These systems should support many types of ESD, so that users need only learn a single application.

A number of technologies have been used to provide access to ESD, including spreadsheets, databases and statistical packages. These have typically been used by more experienced users, although some recent developments have made them more appropriate for casual and novice users of ESD (e.g. Jones, 1996; Wilkinson and Brown, 1997). More recently, GIS have been used with great success to provide easy access to ESD with a strong geographical or spatial dimension (e.g. Exeter, 1996). The number of technologies that have been used to provide ESD reflects the ongoing problem of finding a suitable level of functionality for different types of users.

Recently developed technologies, such as Online Analytical Processing (OLAP), Multi-Dimensional Databases (MDDB) and the World Wide Web (web), could be integrated to produce an easy-to-use system that provides the functions needed by casual and novice users of ESD. This study describes the development of such a system and its evaluation by casual and novice users of ESD. The paper presents a brief overview of MDDB and OLAP technologies, a description of the development of the system and the results of user testing of that ESD system.

MULTI-DIMENSIONAL DATABASES (MDDBS)

MDDBs differ significantly from relational databases both in structure and in their intended use. While relational databases store records in a series of related tables, MDDBs store summary values in an n-dimensional matrix or **hypercube**. For example, in a business application, a hypercube might store values or **measures** such as the *Number of Sales* and the *Value of Sales*. The MDDB stores these measures in relation to other **dimensions** of interest to the users. These **identifier** dimensions might include *Location*, *Date of Sale* and *Product Type*. Each dimension has a finite set of possible values, called **members**. The Location dimension might have as its members, the location of ten stores - each store would be a **member** of the Location dimension. Date of Sale and Product Type dimensions would have their own members.

A hypercube containing this data would have the members; *Location*, *Date* and *Product Type*, along the three axes of the cube. The values of the measures; *Number of Sales* and *Value of Sales* would be stored within the cube at the points of intersection formed by the members along each axis. For example, values for the *Number of Sales* and *Value of Sales* of Groceries sold in January in Perth would be stored at the point of intersection in the hypercube of the members; *Groceries*, *January* and *Perth*. A cube can have many identifier dimensions. For example, adding dimensions like *Salesperson* and *Payment Type* would produce a 5-dimensional hypercube.

Hypercubes also support **hierarchical** identifier dimensions i.e. the members in an identifier dimension can be aggregated to higher levels. For example, the *Location* dimension has individual stores as its lowest member. The *Value of Sales* for these stores could be aggregated to give values for each Region, for each State etc. When storing hierarchical dimensions, MDDBs make no distinction between the hierarchical members, like States, and the original members, like stores.

The structure of MDDBs supports rapid manipulation of the summary values stored in the hypercube. This manipulation is usually carried out using OLAP tools, which are also used to define and populate the hypercubes with data.

ONLINE ANALYTICAL PROCESSING (OLAP)

The functions typically provided by OLAP systems typically include:

- Providing a simple means of producing and accessing MDDBs, which typically contain historical data drawn from a number of sources;
- Supporting complex analyses, such as calculations and modelling, across dimensions and through hierarchies;
- Carrying out these analyses interactively with a response time of 5 to 20 seconds and

- Providing tools that allow users to:
    - Add or remove entire dimensions from an analysis (rotation)
    - Aggregate or disaggregate (roll up/ drill down) data for any hierarchical dimension
    - Filter out particular members of a dimension and
    - View a sub-population of the data (slicing)

(Thomsen, 1997)

The functions provided by MDDBs and OLAP appear to match the requirements of casual and novice users of ESD. In combination with the Web, it appears that they could provide simple, online access to a wide variety of ESD. *Abacus*, the system developed in this research system, will demonstrate the suitability of such a combination of technologies to meet the needs of casual and novice users of ESD.

DEVELOPMENT OF THE *ABACUS* SYSTEM

Specifications for *Abacus* were gathered from a number of sources. A literature review concerning casual and novice users provided several useful insights into the type of functions that would be required (e.g. Preece, 1994; Shneiderman, 1992; Cuff, 1980). Other functional requirements were gathered by examining two existing systems used to disseminate ESD, and from interviews with potential ESD users (Hyland and Gould, 1998; Hyland and Hasan, 1997). Analysis of the data collected gave rise to the following specifications.

Abacus should be able to:

1. Provide access to multiple sets of ESD having 5 or more dimensions
2. Produce up to 5-way tables
3. Aggregate any of the dimensions into pre-defined or user defined groupings
4. Move the dimensions on multi-level tables inwards or outwards on the table
5. Move any dimension from a row to a column or vice versa
6. Hide one or more members of a dimension
7. Show only the values for a subset of the population
8. Show users the hierarchical organisation of dimensions and members of dimensions
9. Access data from several sources
10. Quickly integrate new sources of data or update existing data
11. Allow the user to move effortlessly between data sets
12. Transfer tables and graphs to other applications e.g. word processors
13. Be accessible through the Internet

The development environment and problem domain

Abacus was developed using Gentia ©, a client/server, OLAP/MDDB development environment. Although Gentia is not widely known, at the time it was selected it was one of the more advanced OLAP development environments. *Abacus* was developed using a Pentium 1 personal computer (running Windows NT) as the client and a Solaris Enterprise 1 server. User testing was done on a Pentium computer using the Netscape browser.

The data used in *Abacus* was a 1% sample of the 1991 Australian population census and contains unit data for 63000 dwellings, 67000 families and 168000 people. There are over 60 dimensions and the number of members in each dimension varies from 2 (sexes) to 58 (qualification). The data appeared to be large enough and complex enough for our purposes.

The development process

Based on the user requirements given above, an initial prototype was developed providing four modes of interaction. Pilot tests carried out prior to the full-scale usability testing suggested that

- a) The four modes of interaction took too long to test and
- b) Users had a great deal of difficulty selecting an appropriate mode of interaction

Consequently, the prototype was simplified to only two modes of interaction: Value Mode and Table Mode.

Value Mode

Value Mode provides a highly structured dialogue that leads the user through the process of defining a single value. On entering Value Mode, the screen displays a single button, labelled 'Define a Rule', and a default table showing the total number in the population for the data set. After selecting 'Define a rule', the user selects the dimension and the member of that dimension that he or she wants to constrain. Once a member is selected, a rule of the following form is displayed:

The value '40 to 64' is **included** for the variable 'Age'

Once a rule is defined, the user is provided with buttons that allow him or her to change the rule from Include to exclude, cancel the rule without applying it or apply the rule as it is shown (see Figure 1).
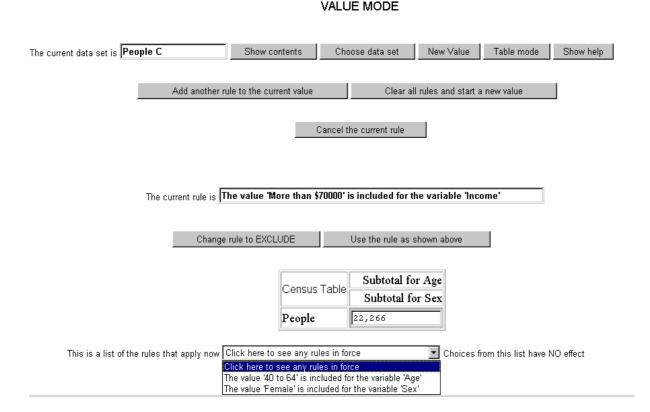

Figure 1: A screen from Value Mode showing the process of defining a rule

Once a rule is applied, the selected member is either the **only member included** in the table or is **excluded** from value in the table, depending on the sense of the rule. The user can apply additional rules until the desired population has been fully described e.g. all Greek-speaking, females, aged between 40 and 64. A drop down list, shown at the bottom of Figure 1, shows the rules that are currently in force.

Table Mode

The second mode of interaction is called Table mode and a typical screen is shown in Figure 2.

Figure 2: A screen showing Table Mode

Table Mode provides an extensive range of OLAP functions in an undirected manner, so users can choose to use any function whenever it is appropriate. The user is initially presented with a single-celled table showing the total for the population in the data set.

The buttons in the upper row are navigation functions while the central block of buttons provides the OLAP functions, allowing the user to:

- Add or remove a dimension from the table
- Move dimensions from one axis to the other
- Move a dimension inwards or outwards on multi- level tables
- Start the same 'define a rule' process used in Value Mode
- Activate the slicer, shown as a set of drop down lists shown above the table in Figure 2.

Figure 2 shows a 2-way table, by Age and Sex. Both the Age and Sex dimensions have been disaggregated one level. Bold-faced, underlined headings, such as '**Up to 5**', indicate the member can be disaggregated while headings in plain text, such as 'Male', indicate that they cannot be disaggregated. The slicer allows the user to select specific subsets of the population in relation to some identifier dimensions without needing to add those dimensions to the table being displayed. For example, the third drop down list box in the slicer shows the entry 'Married'. This means that the user has chosen to restrict the data on the table to Married people, without actually adding the dimension 'Marital status' to the rows and columns of the table itself.

ASSESSING THE EFFECTIVENESS OF THE TECHNOLOGIES USED

One of the main goals of this research is to demonstrate that OLAP, MDDB and the Web can be integrated to provide a system suitable for casual and novice users of ESD. To demonstrate this, the effectiveness of *Abacus* was assessed by user testing and by critically comparing the functions of *Abacus* with the system specifications.

The system was tested by 36 volunteers: approximately equal numbers of academics, university students and local government employees who were known to be users of statistical data from external sources. Volunteers completed a pre-test questionnaire about their use of ESD and, on the basis of their responses, were classified as either novice (N=13), casual (N=18) or expert (N=5) users.

After 20 minutes training with *Abacus,* participants were videotaped as they completed a set of 18 tasks. A typical task would be:

> 'For females having an income in the range $3001 - $5000, were there more who spoke English "Not well" than spoke English "Not at all". (Answer YES or NO)'

Following this, participants completed an Evaluation Sheet containing validated instrument for measuring users' perceptions of usefulness and ease of use (Davis, 1989). Each factor is measured using six 7-point Likert scales. The evaluation sheet also asked several open-ended questions about user satisfaction.

The videotapes and evaluation sheets were analysed and the following well-established performance measures (e.g. Bloom and Hautaluoma, 1990) were recorded for each participant:

- The number of correct answers
- The total time used in all tasks
- The number of errors made
- The time spent dealing with errors
- The Perceived Usefulness – the mean value of the six Likert scales
- The Perceived Ease of Use – the mean value of the six Likert scales

There are no benchmarks available for the first 4 of these performance measures for OLAP-based systems, so it is difficult to say whether the values given in the following sections are acceptable. One of the contributions of this research is to establish initial benchmarks for testing such systems. A benchmark for Perceived Usefulness and Ease of Use is provided by Nielsen and Levy (1994) who carried out a meta-analysis of the use of subjective performance measures in 40 studies comparable to the current study. They suggest that a 'good' system would have mean and median values for subjective preference ratings of 5.6 (on a seven-point scale).

Number Correct

The mean Number Correct for all participants was 14 (SD = 4.1) and the median was 16. Neither of these values are particularly high but they are not unreasonable for a first attempt, particularly by casual and novice users. The median suggests that most participants got the vast majority of tasks correct. In fact, twenty-two participants (61.1%) completed 15 or more tasks out of 18 correctly. This is equivalent to a score of 83%.

While this seems a reasonable result, bear in mind that only about 8% of the group got all of the tasks correct which would be an alarming result of the data being retrieved were used for decision making. Even more alarming is the performance of some of the lower scoring participants. At the bottom end of the scale, 16.7% (N=6) of participants failed to find the correct answer in more than 60% of the tasks. However, these results were quite atypical of the participants generally and do not seem to indicate a significant problem with *Abacus*.

Secondly, because the prototype provides feasible answers even when used wrongly, those users who have made errors often indicated that they were confident that they had the correct answer. This was even the case when answers were clearly wrong, e.g. when the population recorded by the participant was a negative number. This is likely to be an ongoing problem, particularly for casual and novice users, who are less likely to be familiar with the domain of their statistical inquiry and so be less able to determine whether an answer presented by the prototype system is, in fact, likely to be correct.

Thirdly, the participants in the study are using the prototype for the first time and one might expect their performance would be better over subsequent interactions. Without the results

of longitudinal testing, this cannot be shown conclusively but, even within the current study, there is evidence that user performance improves rapidly with experience. It was noted that participants performed better in the Mixed tasks than in the tasks in Table Mode or Value Mode, presumably because they had more familiarity with the *Abacus* prototype by the time they attempted the Mixed tasks. This improvement in performance occurred despite the fact that the Mixed tasks were more complex than most of the tasks in Table Mode or Value Mode.

Total Time

The mean Total Time spent on all 18 tasks was 45 minutes (SD = 13) and the median was 45.5 minutes. 86.1% (N= 31) of participants completed all tasks in less than 1 hour. It is noteworthy that the Total Times for 5 participants (13.8%) were very high (in excess of 65 minutes). These have had a significant effect on the mean Total Time.

Number of Errors

The mean Number of Errors made by each participant was 17.7 (SD = 13.1) and the median was 15.0. This seems an acceptable level of error, with 63.9% (N= 23) of participants making less than one error per task. Given the novelty of the application and the conceptual difficulty of the tasks, this would appear to be a reasonable performance. In fact, 63.9% (N= 23) of participants made less than one error per task, on average.

Error Time

The Error Time is the total time spent by each participant dealing with errors. The mean Error Time is 7.9 minutes (SD = 5.3) and the median is 6.3 minutes. Since Error Time is across all eighteen tasks, the mean Error Time represents an average of less than 30 seconds per task. Given that the average time to complete a task was 2 minutes, Error Time accounts for only 17.3% of the time spent on a task. Sixteen participants (44.4%) spent less than 0.22 minutes in error per task. These values are significantly affected by the presence of 4 outliers, whose Error Times exceed 17.9 minutes.

Benchmarking the results

The results above are difficult to assess because of the lack of an appropriate benchmark. One of the contributions of this research is to establish an initial benchmark for testing this type of system.

Fortunately, suitable benchmarks do exist for the next two performance measures. Nielsen and Levy (1994) conducted a meta-analysis of 57 studies in which comparisons were made between two or more systems. Of these 57 studies, 40 involved the measurement of subjective preferences using rating scales such as the ones used in this study. In total, 78 systems were subjected to pairwise comparisons on the basis of these scales (each of the studies involved two or more systems).

In the report on these studies, Nielsen and Levy found the mean for subjective preference was 4.82 (with a 95% confidence interval of +/- 0.19) and a median of 4.9. These measures of central tendency (the mean and the median) are not suitable as a benchmark because they have been affected by both systems that users liked and systems that users disliked. In effect, the mean and median represent the users' rating of satisfaction with an 'average' system. When only those systems that were **preferred** by the users are included in the calculation, the mean and the median value were both 5.6 (on a seven-point scale). The authors recommend the use of the 5.6 level as a benchmark for good quality systems.

Perceived Usefulness

The mean Perceived Usefulness of the *Abacus* prototype was 5.5 (with a 95% confidence interval of +/- 0.33) and a standard deviation of 1.1. The median was 5.7. Clearly, both of these values suggest that the Perceived Usefulness of the *Abacus* prototype was comparable to the subjective preference ratings of 5.6 for both measures of central tendency for better quality systems, as reported by Nielsen and Levy (1994). These results were also

confirmed by comments made in the open-ended responses by participants (see below). Overall, these results indicate that the system is perceived as being useful.

Perceived Ease of Use

As with the results given for Perceived Usefulness above, these results appear to indicate that participants found the prototype very easy to use. The mean Perceived Ease of Use was 5.4 (with a 95% confidence interval of +/- 0.39) and a standard deviation of 1.1. The median was 5.5. Once again these compare favourably with the preferred benchmark suggested by Nielsen and Levy of 5.6 for both measures of central tendency. These results were also confirmed by comments made in the open-ended responses by participants

Flexibility of the prototype

One of the system specifications described above was that the system should be flexible and interactive. These characteristics have not been measured using any specific instrument but one indication of these characteristics would be the number of distinct strategies used by participants to complete the tasks. In all 22 separate strategies were recorded, 6 in Value Mode, 15 in Table Mode and 1 in Mixed tasks. Once again the larger number in Table Mode suggests that, as expected, it is the more flexible of the modes of interaction. The mean number of strategies used, in all modes was 3.8 with a standard deviation of 1.22 and a median of 4.0. Bearing in mind that participants worked in both modes, it was necessary for them to use at least 2 strategies. The average value of 3.8 indicates that participants felt the need to use more than the most obvious strategies and were able to do so.

Indications that the prototype is easy to learn

All of the 36 valid participants completed the training in less than 24 minutes and more than 80% (N=29) of participants completed it in less than 17 minutes. The mean time spent on the training exercise was 15 minute) with a standard deviation of 4.1 minutes. Once again, no reliable benchmark is available for the amount of time such training might take or the level of competence that should be expected of users after such training.

Following this training, participants were able to complete the vast majority of the tasks required of them, including the production of complex tables with multiple dimensions and the location of specific data elements within these tables. Many users commented on how easy the prototype was to learn and to use.

Participant's responses to open-ended questions about *Abacus*

Additional information was gathered about each participant's experience with *Abacus* via the open-ended questions on the Evaluation Sheet. Participants were asked to describe their **overall reaction** to *Abacus*. Participants often made two or three separate comments about their experience and these have been categorised as follows. Thirty participants (83.4%) had an overall impression of *Abacus* that was predominantly positive.

Positive only:	all the comments made by the participant were clearly positive
Positive, learning:	all the comments were positive but some mention initial difficulty
Positive, condition:	all the comments were positive but some included a condition e.g. easy in a specific mode or good for some type of data
Negative only:	all the comments made by the participant were clearly negative
Other:	all of the comments made were neither positive nor negative

Table 1 shows a frequency distribution based on these categories. On the basis of these results, approximately 83.4% (N=30) of respondents had an overall impression of the prototype that was predominantly positive.

Category	% of respondents	Typical comments
Positive only	36.1% (N=13)	very good, makes it simple; easy to use and effective
Positive, learning	16.7% (N=6)	good once I got used to it; easy to use after initial difficulty
Positive, condition	30.6% (N=11)	good if there are many variables; excellent tool for complex data
Negative only	5.6% (N=2)	searches need to be more flexible; table was confusing
Other response	11.1% (N=4)	value easier than table; can it support all types of data ?

Table 1 Distribution of responses to open-ended questions about the prototype

The second open-ended question asked participants what were the **best features** of the prototype. Roughly 40% of the comments made about the best features of the system were very general, describing the prototype as a whole. Consequently, comments made about the best features were often similar to those made about the participants' overall reaction to the prototype. Thus, to get a complete picture of the participants' overall reaction, some of the responses about best features (question 2) must be included with the responses to overall reaction (question 1). For example, in describing their overall reaction, 14 participants commented that the prototype was easy to use but a further 4 participants made this same comment as one of the best features of the prototype. Thus, 50% (N= 18) of participants, described the prototype as being easy to use (in one or other response). Of course, other participants may also have thought the prototype was easy to use but not have thought to mention it. The same is true of the other comments, both positive and negative, described below.

Other comments made in response to one or other of these two questions included the prototype being user friendly (8.3% (N= 3) of participants), easy to learn (13.8% (N= 5) of participants) and good or great (16.6% (N= 6) of participants). Several participants made additional positive comments about either Value or Table Mode being easy to use or easy to learn while other participants made similar comments about the prototype as a whole but with some condition e.g. "good for finding single values". The overall impression given by the responses to the first two questions is a very positive one.

The third open-ended question was about the **worst features** of the prototype. All of the comments about worst features were about specific problems that participants had with the prototype. A brief description of the four most frequently cited problems and the percentage of participants who reported on these are given below.

- Difficulties locating values in large tables and loss of screen icons – 30.6 % (N= 11)
- The inability to remove a single rule from a set of rules   – 25% (N= 9)
- Confusion about the way the prototype operated   – 13.9% (N= 5)
- Lack of colour in the interface or poor layout of screen elements– 5.6% (N= 2)

The third and fourth comments may not be very significant for two reasons. Firstly, the number of participants making each comment is not very large and secondly, it is to be expected that some participants will be confused about the way the system operates or about aesthetic considerations. The first and second comments, on the other hand, are both more frequent and the problems described are more substantial. In response to the first comment, it might be said that the large tables are a natural result of large data sets with complex sets of dimensions and members. The census data used in *Abacus* was intentionally chosen to create an environment in which large tables would be produced. It is only to be expected that participants would have trouble with such large tables. What is more important is the fact that participants were unable to use several OLAP functions to control the size of tables, even after these functions had been demonstrated in the training session.

The second comment identifies one of the most serious shortcomings of the *Abacus* prototype. In both Table and Value Mode, users can define multiple rules to constrain the

population. Once a set of rules is defined and applied, *Abacus* does not let the user remove or replace one of the defined rules. This was an unintentional oversight in the development process that was not detected until the usability testing was well under way. It could, however, be easily corrected and does not reflect a serious problem with the underlying technologies.

By and large, the responses to open-ended questions are largely positive and support the assertion that the prototype is easy to learn, flexible and useful.

Variation in Performance due to Learning effects

During usability testing it was observed that many of the participants (55.6 %) got all four of the Mixed tasks correct. This was surprising, given that these tasks were conceptually more difficult than many of the ones encountered in Table or Value Mode. In addition to complexity of the tasks themselves, there was also the added complexity of selecting an appropriate mode in which to carry out the tasks. It was expected that participants would score no better on these than on the rest of the tasks. One explanation for the higher scores in Mixed tasks was that those tasks **had** to be done last and that participants had more experience with *Abacus* by the time they attempted the Mixed tasks. The higher scores may reflect a learned ability with the prototype.

To test this informal hypothesis, paired sample t-tests were conducted for the four performance measures: Number Correct, Task Time, Number of Errors and Error Time for the Mixed tasks and for four comparable tasks. The results are shown in Table2.

	Mixed tasks	**Four comparable tasks**	**Std error**	**t**	**p value**
Number Correct	3.29	2.83	0.16	2.944	.003
Task Time	509 seconds	522 seconds	24.72	-0.501	.320
Number of Errors	4.3	4.7	0.94	-0.431	.334
Error Time	79.8 seconds	168.8 seconds	37.31	-2.386	.012

Table 2 Performance measures in 4 Mixed tasks and 4 tasks of comparable difficulty in earlier modes

The values in Table 2 indicate that the differences in means for all four performance measures are in the expected direction. The mean for Number Correct is higher for Mixed tasks than for four comparable tasks and the result is statistically significant at the 5% level. For the other three measures the means are lower for Mixed tasks than for four comparable tasks because the learning effect would lead to a reduction in Task Time, Number of Errors and Error Time. Although the differences in means for these three measures are in the expected direction the difference is only significant at the 5% level for Error Time. Nonetheless, it appears that there is an improvement in performance as users become more familiar with the prototype and that this improvement is statistically significant for both Number Correct and Error Time.

Comparison with the system specifications

Providing that the system specifications adequately reflect the needs of the intended users, the combination of technologies in *Abacus* can be shown to meet the needs of those users if *Abacus* meets the system specifications. With three exceptions, *Abacus* does meet the system specifications.

Abacus does not support user-defined groupings (hierarchies). Within the Gentia environment, it **would** be possible to allow users to define their own groupings, but the interaction that would be necessary would be extremely complex and far too time consuming. Consequently, it appears that the use of user-defined groupings is beyond the capabilities of the OLAP/MDDB approach.

Although *Abacus* does provide a Browse function i.e. users can see a list of all the dimensions and members in a data set, *Abacus* does not display the hierarchical structure of

the data set. Since the hierarchies are part of the data model, it would be relatively simple to make this information available in future versions of *Abacus*.

Although *Abacus* provides 'a seamless interface between data from different sources', the integration of a new source of data requires the definition of a new MDDB that is a very time consuming process that could not be undertaken by a casual or novice user. Even the updating of existing MDDBs is time consuming. This requirement was unrealistic in the first place.

Apart from these 3 exceptions, *Abacus* successfully met all of the system specifications.

CONCLUSION

Abacus demonstrates that OLAP, MDDB and the Web can be combined to produce a tool, which is certainly useful to casual, and novice users of ESD and which provides a high degree of user satisfaction. *Abacus* appears to meet the majority of functional requirements that were identified in the preliminary studies and via the literature review. Users appear to be able to use these technologies to carry out quite complex tasks with only limited training. However, those users who did have problems with the prototype made quite significant errors and were often unaware that their responses were grossly incorrect. Perhaps additional feedback from the prototype would help make this apparent to users, although it is difficult to imagine how this feedback might be structured

The study has also established initial benchmarks for the testing of OLAP/MDDB systems, which would be an advantage in future comparative studies concerning these technologies.

REFERENCES

ABS (Australian Bureau of Statistics) (1993) A Proposal for Enhanced ABS Client Servicing from a Corporate Marketing and Sales Program, ABS Management Meeting, April 1993.

Bloom, A.J. and Hautaluoma, J.E. (1990) Anxiety Management Training as a Strategy for Enhancing Computer User Performance, Computers in Human Behaviour, **6**(4), 123-134.

Cuff R. N. 1980, On Casual Users, International Journal of Man-Machine Studies, **12**. 163-187.

Davis, F.D. (1989) Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology, MIS Quarterly, September 1989, 319-340.

Exeter, T.G. 1996, Taming the Tiger for 2000, American Demographics, June, 1996, 10-13.

Hyland P.N. & Gould E. (1998) External Statistical Data: Understanding Users and Improving Access, International Journal of Human Computer Interaction, **10**(1), 71 – 83.

Hyland, P.N. & Hasan, H. 1997, The Use of External Statistical Data in Local Government, Urban Policy and Research Journal, **15**(4), 279-289.

Jones, A. 1996, Statistical Data Plots Thicken with 3-D VRML, Infoworld, **18**(34), 27.

Nielsen, J. & Levy, J. (1994) Measuring Usability: Preference Vs. Performance, Communications of the ACM, **37**(4), 66-75.

Preece J. (with Rogers Y., et al) 1994, Human Computer Interaction, Addison Wesley, Reading, MA.

Shawkey, B. (1990) How Do You Compare: Peer-Group Statistics Put Your Financial Vital Signs in Perspective (Part 4), Credit Union Magazine, **56**(12), 18-22.

Shneiderman, B. 1992, Designing the User Interface: Strategies for Effective Human Computer Interaction, Addison Wesley, Reading MA.

Thomsen, E. (1997) OLAP Solutions: Building Multidimensional Information Systems, John Wiley and Sons, New York.

Wilkinson, L. & Brown, R. 1997, Statistical Software Takes Advantage of Powerful Desktop Systems, Research and Development, **39**(1), 56-57.

COPYRIGHT