

Experiences in Building and Using Decision-Support Systems in Postgraduate University Courses

Arthur Tatnall and Stephen Burgess
Centre for International Corporate Governance Research
Victoria University, Melbourne, Australia

Arthur.Tatnall@vu.edu.au Stephen.Burgess@vu.edu.au

Abstract

In this paper we will relate some of our experiences in building and using simple Decision Support Systems (DSS) for use in two postgraduate subjects at Victoria University in Melbourne. These subjects are not about Decision Support Systems, but are Information Systems (IS) subjects that introduce, amongst other things, concepts of decision support and DSS. It has been our experience that the complexity, an inability to examine the logic behind decisions that have been recommended, the proprietary nature and the cost of commercial DSS make their use in teaching less than ideal. Our solution was to produce our own simple systems using Excel, Visual Basic and Visual Basic for Applications, and this has proved to be quite successful. The paper reports on building and using Decision Support Systems in this way.

Keywords: Decision support, university courses, postgraduate studies, business students, programming.

Introduction

The concept of Decision Support Systems (DSS) has been around since the late 1960s (Power, 2003) and in recent years these systems have become increasingly important in business. As would be expected, a great deal has been written about Decision Support Systems – a Google search in February 2007 revealed over 242,000,000 pages of information, and a great deal of support material is also available. Unfortunately, for our purposes not much of this material is particularly useful. Our interest is in introducing the basic concepts of Decision Support Systems in two postgraduate subjects that are part of business Masters' programs at Victoria University in Melbourne. These are not subjects *about* Decision Support Systems, but Information Systems subjects that introduce, amongst other things, concepts of decision support. Many of our students are general business students who have little background in computing or information systems.

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@InformingScience.org to request redistribution permission.

What we need is thus a 'light' treatment of this topic, preferably with some concrete examples that the students can view, manipulate and customise. In this paper we will relate our experiences in building and using such simple Decision Support Systems. In the subject *Executive and Mobile Computing* we also introduce the concept of an Executive Information System (EIS), which Power (2003) notes evolved from relational

database products and model-driven Decision Support Systems. In the subject *Small Business Internet and Information Systems* we introduce the DSS within the framework of investigating only such systems that are suitable for and affordable to small business. We make use of a simple programming language, available within Microsoft Office to do this.

Although there are many teaching resources available for Decision Support Systems (such as, Holsapple and Whinston (2005), Power and Kaparathi (2002) and Monash University (2005)), for our subjects we have encountered three significant problems in sourcing suitable DSS examples:

- 1) Difficulty in learning to use the DSS – many of the available systems are too complex for our students to learn and use successfully within the short time that we have available. This is, of course, counter-intuitive to the notion that DSS are meant to be easy for managers to use.
- 2) Inability to examine the logic of the system:
 - a) A business that uses a DSS will want this to be transparent in its operation, but in teaching about DSS we want the students to be able to see and understand how the system operates. It is difficult to understand what is going on unless you can understand the logic behind decisions recommended by the system.
 - b) Need to customise – we would like students to be able not just to see how the system works, but to be able to alter assumptions, to customise it, to make changes and to observe their affect.
- 3) Cost – with a small budget available to support the purchase of software for teaching purposes we need a system that is free, or at least inexpensive.

The way we have attempted to solve these problems is to build our own simple systems for use by the students. To do this we have made use of Microsoft Office (especially Excel), Visual Basic (VB) and Visual Basic for Applications (VBA). In the future we will probably make use also of the facilities available in VB.Net. Although others have reported on use of VBA and Excel for building decision support systems by students (Palocsay & Markham, 2002; Ragsdale, 2001), most of these are typically working with students majoring in some form of computing rather than with general business students. The remainder of this paper will describe how we have attempted to solve this problem with our students, and discuss some of the consequences.

Decision Support Systems Based on a Spreadsheet

The goal of *Executive and Mobile Computing* is, as its name suggests, simply to examine how business executives can benefit from the use of computing (and especially mobile computing) technologies. In this subject we introduce the students to a simple DSS based on an Excel spreadsheet. The scenario involves (the fictitious) Gutenberg Publications; a small publisher operating in a specialist market with only one serious

	A	B	C	D	E	F
1	Gutenberg Publications				19-Feb-05	
2						
3	PRINTING				MARKET	
4	Copies	3,500			Expected total market	5,000
5	Pages	320			Competitor's Price	\$20.00
6	Pages/sheet (1 or 2)	2				
7					Sales base (libraries etc)	50
8	INCLUDE?		Y/N			
9	Paper	\$10,080.00	Y		Marketing budget	\$10,000.00
10	Covers	\$2,975.00	Y		Selling Price	\$21.00
11	Wire Binding	\$0.00	N			
12	Plates	\$630.40	Y		ESTIMATES	
13	Printing ink	\$4,460.00	Y		Estimated sales	3,284
14	Printing labour	\$6,030.77	Y		Possible sales revenue	\$68,964.00
15	Collating/binding	\$1,470.00	Y		Estimated market share	65.68%
16						
17	COSTS				WARNINGS	
18	Unit Production Cost:	\$7.37				
19	Warehouse costs	\$54.00			RECOMMENDATION	Publish
20						
21						
22	PRODUCTION COST:	\$25,805.17			GROSS RECEIPTS:	\$68,964.00
23						
24					NET PROFIT	\$33,104.83
25						

Figure 1: Gutenberg Publications DSS

competitor (Tatnall, Davey, Burgess, Davison & Wenn, 2002). The company needs a DSS (Figure 1) to assist them in deciding whether to publish a given book, and if so, at what price to sell it, and how many copies to print. It begins with intelligence on the likely total market for a new book, along with the price at which its main competitor sells comparable books. It then determines a marketing budget and sets a price for the book. These figures are entered into the DSS which then suggests how many books will be sold.

The DSS has two parts: one to calculate the cost of printing a book (certainly not a DSS) and another to suggest the number of sales, based on the size of the expected total market, the competitors' price, the selling price and the marketing budget. Students are shown the model that calculates the expected total sales, which involves quite a complex formula that initially raises 'a few eyebrows':

$$=INT(F7+(F4*LOG(1+F9/Costs!C14))*(F5/F10)^2)$$

The first exercise the students perform on the DSS is to check the range of each of the input variables for which the DSS gives a 'sensible' result. The main advantage of using a simple DSS like this is that with no copyright restrictions on it, students can investigate, modify and customise the DSS in a multitude of ways, giving them a better understanding of the way that a DSS works than they could otherwise obtain. The results produced by the model seem plausible, and the students then look at the model and try to see if they can improve it. The model makes more sense if seen in pseudocode as:

$$\text{Integer (Sales base + (Expected Total Market * Log (1 + Marketing Budget / Market Factor)) * (Competitors Price / Selling Price)^2)}$$

Students discuss the place of each factor in the model, and the affect that it has on the overall result. They can try leaving a factor out, removing the log value or changing the power from 2 to 1 or 3. The question is then 'do any of these changes make the model produce a result that seems to fit better with any known case studies, or with our experiences?' (Unfortunately it is not, of course, possible to test it empirically.) It is only after there has been some considerable discussion of the model that we ask the students how they think it might have been derived, and finally admit that it was just constructed by us as something that would produce a plausible looking result – in other words that there is no sound theoretical justification for its construction! This opens up another useful discussion, this time about models.

In another example the students look at an Excel-based DSS to recommend whether a small business should lease or purchase a car. The scenario is to compare lease costs over a three year period (or any other period), with the cost of borrowing the money to purchase a car and to resell it at the end of this period. In this case the model is quite simple:

$$\text{If (Total Lease Cost) < (Total Purchase (and Sale) Cost) Then Recommend Lease} \\ \text{Else Recommend Purchase}$$

and students can investigate how the result varies with changes in interest rates, leasing fees and the cost of the car. Again, this is a very simple DSS but this is just the point: it is simple enough for students to fully understand, and to easily investigate and modify. This is the advantage of teaching with a spreadsheet-based DSS, and all students know enough about Excel to be able to easily investigate the model and make changes, without first having to learn about the application or about programming.

Of course, the idea of spreadsheets being used for decision support has been around for a while, and our experiences with spreadsheet-based DSS are by no means unique. Other references to the use of spreadsheets in DSS applications include: Hanna, Ahuja and Winston (2003) and Bhargava, Sridhar and Herrick (1999). Spreadsheets provide users with the capability to alter figures

and to see the effects the alterations have on recommendations. Although spreadsheets have been associated with the concept of decision support for a number of decades (Stair & Reynolds, 1999), there are few documented examples of their successful use in small businesses. Much of the limited research into small businesses has investigated the success factors for information technology, based upon the current use of IT and DSS, or the design and development of specific DSS for SMEs. Little work has been done specifically to identify those areas that have not been adapted to DSS, but show potential for its introduction for the small business (Duan, Kinman, & Xu, 2002). Spreadsheets have been used as decision support tools in many different ways. For instance, one example describes the use of spreadsheets to allocate production resources and combine raw materials in an optimal mix in wood panel manufacturing (Buehlmann, Ragsdale, & Gfeller 2000).

Building a Decision Support System with Visual Basic

Like the spreadsheet examples, the main advantage of building your own small system using a programming language like Visual Basic is that with no copyright issues the students can look at the code (- many of our students have done some VB programming at some stage) and so better understand how the system works. Even those who have not studied any programming are able to gain some understanding of what the simple program code does. The students can also make modifications and improvements in the system's operation.

One example we use in *Executive and Mobile Computing* is an Executive Information System for Felix Auto Imports (Tatnall et al., 2002). The EIS (Figure 2) extracts data from an Access database (also available to the students) and displays critical success factors (CSF) for Felix operations across the country extracted from this. If the user clicks on an Australian State or capital city, the map changes to show just this State and the CSF relating to its operations. The EIS allows drill-down for several levels until eventually reaching down into the Access database (Davey & Tatnall, 1995, 1996).

Another example is a system for Wotan's Wonders – a specialty cake shop (Figure 3). One information system (also written in VB) shows students a

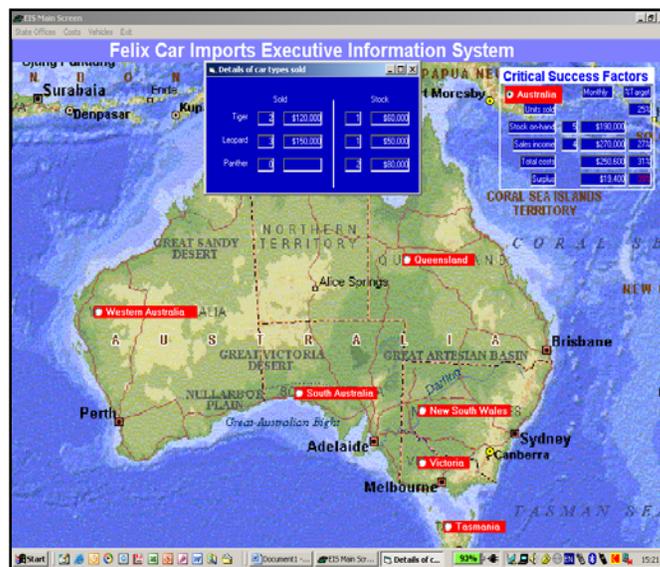


Figure 2: Felix EIS

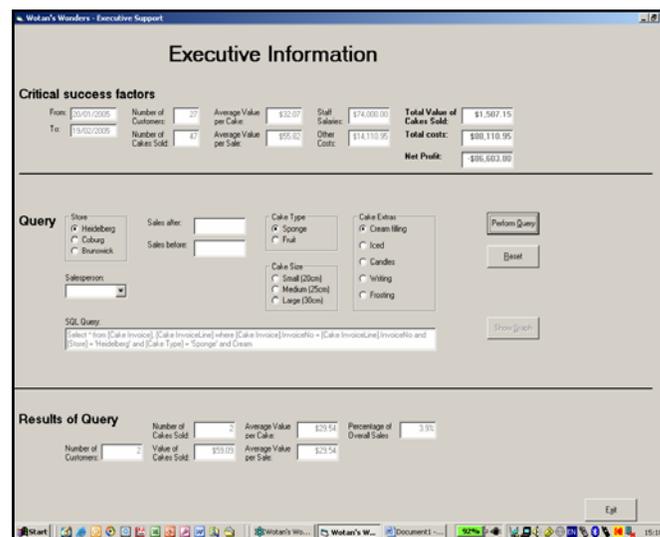


Figure 3: Wotan's Wonders Queries

transaction processing system (TPS). Another related system is a simple EIS that extracts its data from the Access database created by the TPS. One part of this system offers executive support through menu-driven SQL enquiries. Using this system, the user builds up a query by clicking on appropriate option buttons. Although, of course, not necessary for the operation of the system the SQL query is also displayed on screen.

Students are encouraged to examine the code that produces this simple system, snippets of which are shown below:

```
Private Sub optCoburg_Click()
    ExtraQuery(1) = " and [Store] = 'Coburg'"
End Sub
```

```
Private Sub optCream_Click()
    ExtraQuery(7) = " and Cream"
End Sub
```

```
Private Sub optFrosting_Click()
    ExtraQuery(7) = " and Frosting"
End Sub
```

```
Private Sub optFruit_Click()
    ExtraQuery(5) = " and [Cake Type] = 'Fruit'"
End Sub
```

```
Private Sub cmdQuery_Click()
    QueryTotalValue = 0
    QueryTotalCustomers = 0
    QueryTotalCakes = 0
    NewInvoice = 0
    PreviousInvoice = 0

    SQLQuery = "Select * from [Cake Invoice], [Cake InvoiceLine] where [Cake Invoice].InvoiceNo = _
    [Cake InvoiceLine].InvoiceNo"
    For Z = 1 To 7
        SQLQuery = SQLQuery & ExtraQuery(Z)
    Next Z
    txtSQLQuery.Text = SQLQuery
    datQuery.RecordSource = SQLQuery
    datQuery.Refresh

    Do While Not datQuery.Recordset.EOF
        QueryTotalValue = QueryTotalValue + datQuery.Recordset("Price")
        NewInvoice = datQuery.Recordset("[Cake InvoiceLine.InvoiceNo]")
        If NewInvoice <> PreviousInvoice Then QueryTotalCustomers = QueryTotalCustomers + 1
        PreviousInvoice = NewInvoice
        QueryTotalCakes = QueryTotalCakes + 1
        datQuery.Recordset.MoveNext
    Loop
    txtQueryCakeValue.Text = Format(QueryTotalValue, "Currency")
    txtQueryNumbCustomers.Text = QueryTotalCustomers
    txtQueryNumbCakes.Text = QueryTotalCakes
    If QueryTotalCakes <> 0 Then
        txtQueryAverageCake.Text = Format(QueryTotalValue / QueryTotalCakes, "Currency")
    Else
        txtQueryAverageCake.Text = "0"
    End If
    If QueryTotalCustomers Then
        txtQueryAverageSale.Text = Format(QueryTotalValue / QueryTotalCustomers, "Currency")
    Else
        txtQueryAverageSale.Text = "0"
    End If
    txtQueryPercentage.Text = Format(QueryTotalValue / TotalValue, "#0.0%")
End Sub
```

Building a DSS using VBA

Most small businesses now have computers (Burgess, Tatnall, & Darbyshire, 1999), and a high proportion of them use integrated applications packages such as Microsoft Office to carry out their day to day record keeping and other functions. Many small business owners and employees are, however, not aware that this package comes with a powerful programming language built in – Visual Basic for Applications (VBA). The programming interface for VBA is similar to that of Visual Basic, with VBA offering greater capabilities in relation to being able to interact with functions specific to a particular application (such as a cell in a spreadsheet), but having less functionality in other areas (such as in forms used to input data). VBA code created in different versions of Office (such as 2000, XP and 2003) is fairly (but not completely) portable across versions.

A 2001 study of 133 small manufacturing businesses in the UK (Duan et al., 2002) revealed limited use of decision support systems. A lack of staff time to analyse needs and identify solutions was the primary reason given for the lack of use. Where used effectively firms with a more ‘strategic’ outlook implement them. They mainly take the form of previously developed packages and most of them are targeted towards supporting routine decisions. Because of these factors, there is an opportunity for effective decision support tools to make a real impact on small businesses.

The aim of the subject *Small Business Information and Internet Systems* is to introduce students to a broad range of topics relating to the field of information technology and small business, including: the role of information systems and the Internet in small business, selecting applications for small business, integrating information and Internet systems with small business processes, selecting hardware and operating systems for small business, networking for small business, building small business applications, office suite programming, and integrating office and Internet applications. It can be seen that many of these relate to integrating information and applications and building applications. A major part of the assessment is around building a decision support application using VBA.

The students are provided with a case study of a (mythical) small business. Usually this business is involved with manufacturing, and has a head office, manufacturing plant and a small number of retail outlets. Being a small business, it usually has around 20 employees. Typically, the business has out of date computer equipment and employees that range in computer skill levels from complete novices to those with adequate skills. The owner has a desire to integrate the information systems of the business to provide greater decision making capability. The first part of the student assignment is devoted to students acting as consultants and preparing a specifications document for the business to upgrade their systems. The second part shows students how to automatically access data from their transactions database ‘live’ into a spreadsheet. Changes in transactions data in the database can subsequently be automatically updated from the spreadsheet if required. Figure 4 shows the screen where data is automatically imported from

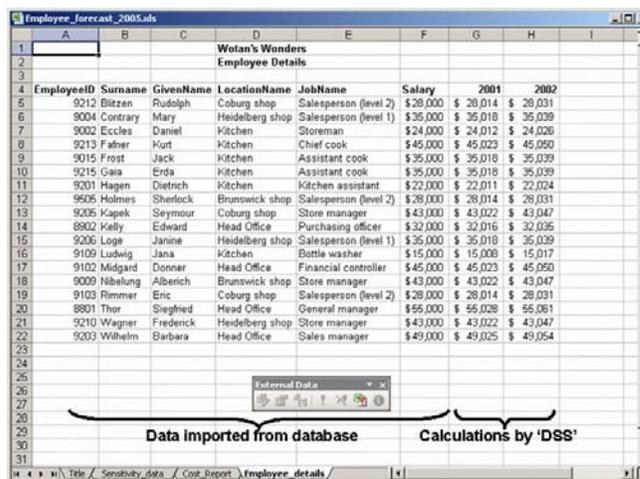


Figure 4: Data imported from MS Access to MS Excel and resultant DSS calculations

the Microsoft Access database, and subsequent calculations (2001, 2002 – which are based on projected salary increases entered by students). The ‘External Data’ box towards the bottom of the figure controls how the data is updated in the spreadsheet. This screen would not normally be seen by a ‘user’ of the spreadsheet.

Before transferring the data across from the Access database, VBA is used to build an input form for forecasting (refer to Figure 5). This does some error checking of the input of the user, and when working effectively the program stops users from entering letters instead of numbers, percentages for increases or decreases that are obviously out of range and so forth. It provides the flexibility for targeted error messages to be provided to users depending upon the entry error made. Once acceptable input data has been recognised, the VBA program inserts it into the spreadsheet and the user can simulate ‘what if’ situations with different percentage changes.

Figure 5: Simple Data Entry Form

The usefulness of the VBA code, however, does not end there. Recent advances have allowed the addition of a ‘foreign exchange’ aspect to the assignment, where students are able to see the effects of their ‘what if’ scenarios in various foreign currencies, according to the latest rates retrieved ‘live’ via VBA code from the Internet (this can be done even more effectively using VB.Net.). The following VBA sample code illustrates how an external web page can be accessed and certain areas (such as a particular exchange rate) ‘found’ and copied into a spreadsheet cell (in the case, a range).

```
Sub Button7_Click()
    Dim oBk As Workbook
    Dim org As Range
    Dim exch As Double
    Dim xstr As String

    Set oBk = Workbooks.Open("http://www.x-rates.com/d/AUD/table.html")
    Set org = oBk.Worksheets(1).Cells.Find("American Dollar")
    xstr = org.Offset(0, 1).Value
    Workbooks("Wotan_Emp_forecast_2005_Word_report.xls").Worksheets("Cost_Report").Range("usexch").Value = xstr
    Workbooks("Wotan_Emp_forecast_2005_Word_report.xls").Worksheets("Cost_Report").Range("curr_text").Value = "USDollars"
    oBk.Close
End Sub
```

VBA code in Excel is also used to open up a Microsoft Word document automatically and paste the forecast values from the spreadsheet into a Word report. To round off the functionality, VBA code is used to open up Microsoft Access Forms so that details in the database can also be altered (which are then subsequently updated in the spreadsheet). In this manner, the spreadsheet becomes a central area for accessing and changing data in the database, producing professional reports and accessing up-to-date information from the Internet – all tied together with the functionality of VBA.

It is not expected that small business owners or employees would be capable of all of the application development described here. However, they should at least be aware of these capabilities –

and training our students to be the consultants who can develop these applications for decision support for small business, using tools that the businesses already possess (Microsoft Office and VBA), can provide an important means for small businesses to develop these applications.

Web-Based Decision Support Systems

We have not, as yet, looked at web-based DSS (Power & Kaparathi, 2002) but in *Small Business Information and Internet Systems* this would be most appropriate, and VB.Net might be just the tool to develop these applications with the students. Decision support portals (Moon & Burstein, 2005; Tatnall, 2005) is another area of interest here and offer plenty of possibilities that are within the capabilities of postgraduate students and, perhaps more importantly, are quite doable within the time frame and resource restrictions of a university business subject.

Student Responses to this Approach

Over the five or six years we have been doing this work we have surveyed students at the end of each semester to find out about their experiences. After a minor glitch when one of the subjects was first offered, students (almost universally) indicated that they found being able to build and 'play with' the simple decision support systems to be a very useful experience. They indicated that they thought they learned much more in this way that they would have by just being told about these systems or seeing (closed) commercial systems in operation. Several students even indicated that they would now undertake more training in Visual Basic and VBA so as to be better able to build these systems.

Conclusion

Even postgraduate business students have trouble really understanding the concept of how decision support systems work unless they are able to get some experience using and playing with them. While ideally we would make more use of real decision support systems, self-built systems using spreadsheets, VB and VBA (that can easily access business databases and other relevant online data sources) have a number of significant advantages: they can be made easy for students to learn and use, it is possible for students to easily 'see inside', both to see how the system works and to be able to customise it, and they are inexpensive. Furthermore, our research with students shows that they enjoy using and building these simple systems and get something useful from this experience at the same time.

References

- Bhargava, H. K., Sridhar, S. & Herrick, C. (1999). Beyond spreadsheets: Tools for building decision support systems. *IEEE Computer*, 32(3), 31-39.
- Buehlmann, U., Ragsdale, C. T. & Gfeller, B. (2000). A spreadsheet-based decision support system for wood panel manufacturing. *Decision Support Systems*, 29(3), 207-227.
- Burgess, S., Tatnall, A. & Darbyshire, P. (1999). *Teaching small business entrepreneurs about computers*. EuroPME - Entrepreneurship: Building for the Future, Rennes, France, Groupe ESC, Rennes.
- Davey, B. & Tatnall, A. (1995). Introducing object environments: Cognitive difficulties. *Software Education Conference: SRIG-ET'94*, Dunedin, New Zealand, IEEE Computer Society Press.
- Davey, B. & Tatnall, A. (1996). Tools for client-server computing. In M. Purvis (Ed.), *Software Engineering: Education and Practice* (pp. 280-285). Los Alamitos, CA: IEEE Computer Society Press.
- Duan, Y., Kinman, R. & Xu, M. (2002). Use of decision support systems in small businesses. In S. Burgess (Ed.), *Managing information technology in small businesses: Challenges and solutions*. Hershey, USA: Idea Group Publishing.

- Hanna, M. M., Ahuja, R. K. & Winston, W. L. (2003). *Developing spreadsheet-based decision support systems using VBA for Excel*. Gainesville, USA: Innovation Center.
- Holsapple, C. W. & Whinston, A. B. (2005). Decision support systems - A knowledge-based approach. Retrieved Feb 2005, from <http://www.uky.edu/BusinessEconomics/dssakba/instmat.htm>
- Monash University. (2005). Monash DSS lab. Retrieved Feb 2005, from <http://dsslab.sims.monash.edu.au/>.
- Moon, J. & Burstein, F. (2005). Intelligent portals for supporting medical information needs. In A. Tatnall (Ed.), *Web portals: The new gateways to Internet information and services* (pp. 186-204). Hershey, PA: Idea Group.
- Palocsay, S. W. & Markham, I. S. (2002). Teaching spreadsheet-based decision support systems with Visual Basic for Applications. *Information Technology, Learning, and Performance Journal*, 20(1), 27-35.
- Power, D. J. (2003, 31 May). A brief history of decision support systems (Version 2.8.). Retrieved Feb 2005, from <http://DSSResources.COM/history/dsshhistory.html>
- Power, D. J. & Kaparathi, S. (2002). Building web-based decision support systems. *Studies in Informatics and Control*, 11(4), 291-303.
- Ragsdale, C. T. (2001). Teaching management science with spreadsheets: From decision models to decision support. *INFORMS Transactions on Education*, 1(2).
- Stair, R. M. & Reynolds, G. W. (1999). *Principles of information systems* (4th ed.). USA, Course Technology - ITP.
- Tatnall, A., Ed. (2005). *Web portals: The new gateways to Internet information and services*. Hershey, PA, Idea Group Publishing.
- Tatnall, A., Davey, B., Burgess, S., Davison, A. & Wenn, A. (2002). *Management information systems - concepts, issues, tools and applications*. Melbourne: Data Publishing.

Biographies



Dr Arthur Tatnall is an Associate Professor in the Graduate School of Business at Victoria University in Melbourne, Australia. He holds bachelors degrees in Science and Education, a Graduate Diploma in Computer Science, and a research Master of Arts in which he explored the origins of business computing education in Australian universities. His PhD involved a study in curriculum innovation in which he investigated the manner in which Visual Basic entered the curriculum of an Australian university. His research interests include technological innovation, information technology in educational management, information systems curriculum, project management and electronic commerce. He has written several books relating to information systems and has published numerous book chapters, journal articles and conference papers. He has just completed editing an Encyclopaedia of Portal Technology and Applications.



Dr Stephen Burgess is a Senior Lecturer in the School of Information Systems at Victoria University, Melbourne, Australia. He has a bachelors degree in Accounting and a Graduate Diploma in Commercial Data Processing, both from Victoria University, Australia; a Master of Business (Information Technology) from RMIT, Australia and a PhD from Monash University, Australia in the area of small business to consumer interactions on the internet. His research and teaching interests include the use of IT in small business, the strategic use of IT, B-C electronic commerce and IT management education. He is currently Director of the Electronic Commerce Research Unit at Victoria University and convener of the IRMA Special Research Cluster on Small Business and Information Technology (www.business.vu.edu.au/sbrit).