A Just-In-Time Approach for Tackling Year 2000 Problems

Dr. Eric Deakins and Stuart M. Dillon
Department of Management Systems
University of Waikato
New Zealand
edeakins@waikato.ac.nz, sdillon@mngt.waikato.ac.nz

Abstract

As the new millennium dawns, attention will focus on the fallout from Year 2000 (Y2K) computer problems. This paper describes a risk-based Y2K Decision Support System (DSS) that assesses the mission critical processes in an organisation's supply chain. A graphical user interface, data-mining and scenario-based contingency planning are key features. Problem formulation, model development, and limitations are described. A real organisation is used to illustrate the use of this decision support tool for Y2K risk assessment and reduction. A demonstration copy of the software is available on request.

Keywords:

Year 2000, Decision Support System, Supply Chain Management, Logistics, Contingency Planning, Risk Analysis, Inventory Management, Data Mining.

1. Background to the *Y2Kat* Project

1.1 Defining the Y2K Problem

Year 2000 (Y2K) computer problems stem from the common practice in computer systems, databases, and human communication of using the last two digits to represent a year in the 1900's. The approach of the new century and the growing need to process dates across a century boundary have resulted in an enormous, world-wide effort to adjust countless volumes of data processed by billions of lines of computer code in millions of interdependent systems.

This interdependency means that changes in a system managed by one organization may have unintended consequences, not only for other internal systems, but also for the critical systems of *other* organisations, possibly leading to a 'domino effect'. The complexity of the problem and impossibility of complete end-to-end testing ensures that Y2K events will occur and that remediation is as much about risk management, to minimise significant disruptions, as it is about prevention [10, 22].

1.2 Company Logistics and Y2K

Even if a company has considered every possible problem related to Y2K, the sheer volume of connections in the supply network virtually guarantee some glitches will occur. The potential consequences of such failures are immense in an era of Just In Time

(JIT) practices and mass customisation [14]. For these reasons supply chain exposure is reported as being the single biggest Y2K issue that businesses face [25].

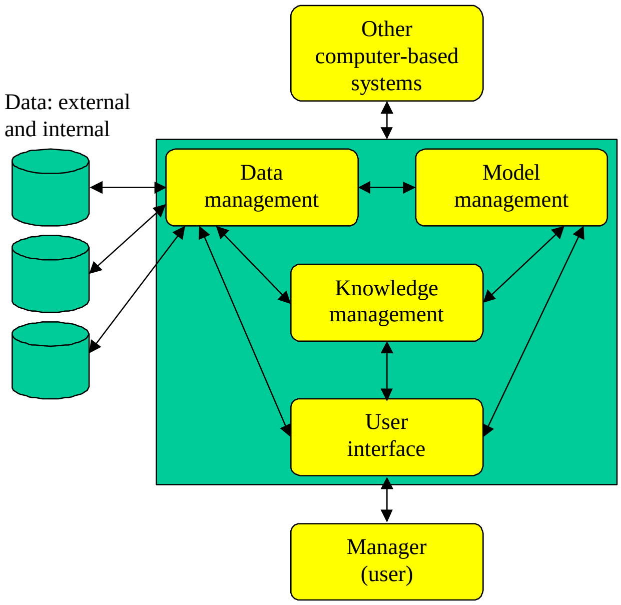
Yet despite the threat to their well being, few large companies have assisted their suppliers [19]. While they may have poured billions of dollars into Year 2000 remediation to make *themselves* internally compliant, the smaller companies that supply them are in a totally different position. By mid-July 1999 around 53% of small companies in the U.S. had yet to undertake a formal assessment of their Y2K compliance, with approximately 3.3 million firms waiting 'until the last minute' to prepare for the millennium [17]. Indeed, an April 1999 National Federation of Independent Business survey indicated that 28 percent of small businesses with exposure to possible Y2K problems did not intend to do anything to prepare for the date change [24]. It is noteworthy that even these figures are optimistic when compared with those of other countries [24].

Small businesses are reluctant to address Y2K issues for various reasons; either they do not understand the potential consequences of date-related failures, or they do not feel they have the time to fix them, or they think they cannot afford the Year 2000 consulting fees, e.g. [18]. The perceived need for a self-help Y2K assessment *system* for small and mid-sized businesses prompted the study described in this paper. The authors were approached by a consulting firm in late 1998 to assist in the design and testing of a Y2K Decision Support System (DSS) that would be especially suitable for smaller firms and business units.

The remainder of this paper outlines the chosen modelling approach and its embodiment in a commercial DSS application that is currently being marketed globally, via the internet, as ***Y2Kat***. Problem formulation, model development, and limitations are described. A case study illustrates its application in a New Zealand company.

2. Decision Support System (DSS) Description

A classical model of a DSS, comprising several subsystems is presented in Figure 1.



1. Schematic View of DSS [23].

Its basic features include data storage and retrieval, plus data drilling and manipulation of that data. In common with many DSS, *Y2Kat* provides support for lower level, structured, cognitive tasks [21]. However, it also supports a level of unstructured decision making, via scenario planning and futurecasting.

2.1 DSS Development Process

The Y2K problem-solving strategy employed in the development of the DSS was based on the methodology first proposed by Bahl and Hunt [1]. This is an approach that recognises the organisational setting and interpersonal characteristics of its likely domain of use. It involves:

- *Taking a descriptive approach that incorporates contextual properties*

The literature is replete with normatively based DSS's which, although used for academic purposes, have for whatever reason not found their way into practical decision making situations. It is being increasingly acknowledged that the low level of uptake is a product of the gap between what decision-makers do and what the normative literature prescribes they should do [3].

- *Emphasising the decision maker rather than the decision*

It is often (erroneously) believed that the purpose of a DSS is to support the decision. Although that is what the name suggests, it is the decision-maker who should be the beneficiary of this support [1]. In addition, there is a far greater level of similarity between decision-makers than there is between the decisions they make.

- *The concept of DSS design as a joint undertaking for organisational problem-solving*

The literature abounds with work that describes the value of user involvement in systems development, e.g. [16, 12]. It was no accident that members of the *Y2Kat* development team had extensive experience within the domain of both managerial decision making and Y2K issues. Their input was particularly significant in the development of the user interface.

2.2 Problem Formulation

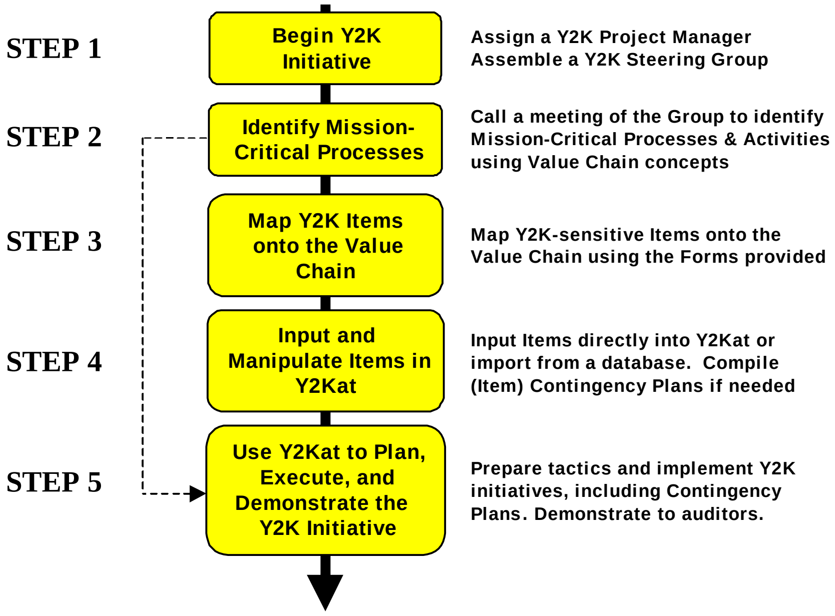
At the time the project was initiated, contemporary approaches to Y2K remediation had involved up to 5 basic steps: *Awareness, Assessment, Renovation, Testing,* and *Implementation*. However, given the *forecasted* project completion date of late 1999, allied to a small company's inability to fix problems without external assistance, the authors' approach focused on the *Awareness* and *Assessment* steps.

Like others before them, the authors also favoured a *triage* approach to Y2K asset management. The principle of triage involves examining Y2K-sensitive items (of hardware, software, etc.) on a risk/impact/cost basis, without intending to make them all compliant, e.g. [8, 9]. Many authors agree that as 1 January 2000 draws near, triage become the first choice, e.g. [15, 13, 11]. Zerega asserts that the same process occurs when companies examine their trading partners; most manufacturers focus on mission-critical suppliers. They begin by identifying potential hotspots, assessing their risk, and then fixing problems [25].

There have been many so-called Y2K remediation methodologies developed to assist companies with the process of Y2K *repair* [14]. However, a general lack of confidence in them, plus company exposure to *external* Y2K failures has forced supply chain organisations to consider the need for backup *contingency plans* [25, 26]. This trend was also incorporated into the design of the DSS from an early stage.

2.3 Model Flowchart

The *major steps* in using the *Y2Kat* DSS are shown in Figure 2; they are cognisant both of the organisational setting and the interpersonal characteristics of its likely domain of use:-

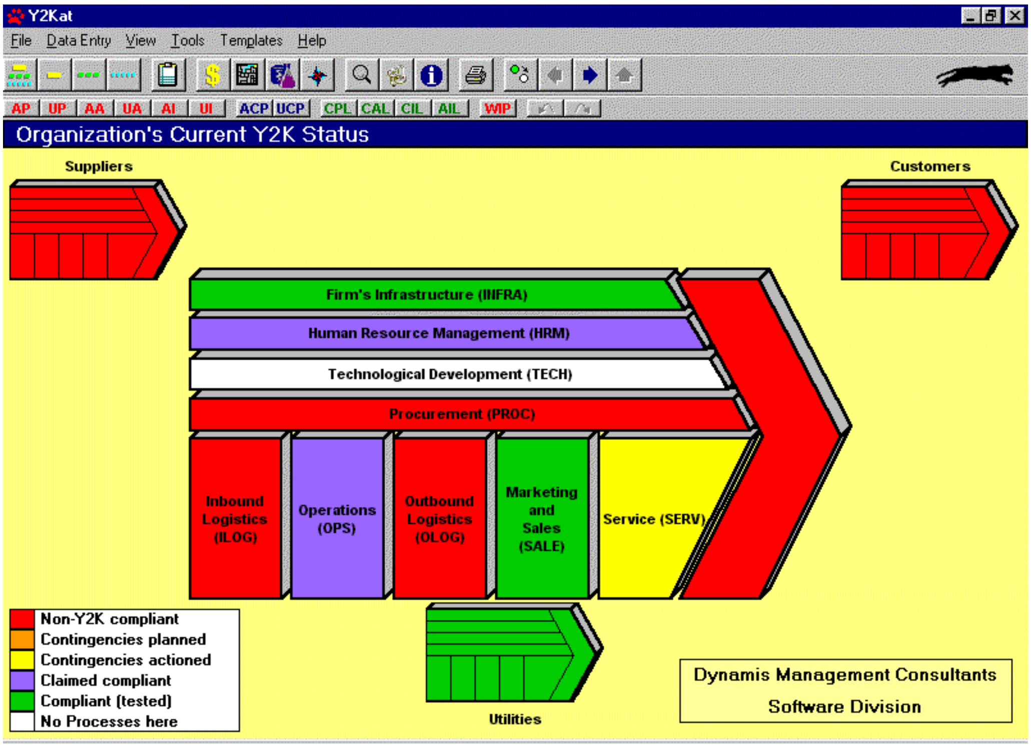

STEP 1
Begin Y2K Initiative
Assign a Y2K Project Manager
Assemble a Y2K Steering Group
STEP 2
Identify Mission-Critical Processes
Call a meeting of the Group to identify Mission-Critical Processes & Activities using Value Chain concepts
STEP 3
Map Y2K Items onto the Value Chain
Map Y2K-sensitive Items onto the Value Chain using the Forms provided
STEP 4
Input and Manipulate Items in Y2Kat
Input Items directly into Y2Kat or import from a database. Compile (Item) Contingency Plans if needed
STEP 5
Use Y2Kat to Plan, Execute, and Demonstrate the Y2K Initiative
Prepare tactics and implement Y2K initiatives, including Contingency Plans. Demonstrate to auditors.

2. Major Steps Involved in Using *Y2Kat*.

In essence, the first 4 steps involve identifying items of Y2K-sensitive hardware, software (etc.), and assigning them into a suitable classification scheme.

If time to act is very short a more appropriate approach may be to action contingency plans for (mission-critical) activities, ignoring their items for the time being (dotted arrow). This enables the user to safeguard vital operations and to then examine Y2K-sensitive items knowing that some degree of protection is in place. The user may even decide to remedy only the items that actually fail since this can be a major cost consideration in those organisations that lack the resources to locate and remedy all of their suspect items. *Y2Kat* caters for either course of action. It assigns Y2K-sensitive items of hardware, software (etc.), into a classification scheme based on the earlier work of Michael Porter, concerned with the value chain [20].

In essence, *Y2Kat* aims to ensure that vital parts of the value chain (chain of value-adding processes) are protected against Year 2000 glitches and failures, Figure 3. It also considers how such failures being experienced by key Utilities, Suppliers, and Customers impact the external Supply Chain. This is cognisant of recommendations by various authors for a 2-stage action plan: *compliance internally and with supply chain partners*, e.g. [5, 22]. *Y2Kat* divides a company (or a business unit) into the distinct processes it performs to do business. Similar processes are grouped into categories. *Y2Kat* identifies 5 primary value-adding categories that are represented by a chain moving from left to right in the lower part of figure 3. Each named category contains a collection of value-adding processes that physically create a product or service and move it onto the customer.


Y2Kat
File Data Entry View Tools Templates Help
AP UP AA UA AI UI ACP UCP CPL CAL CIL AIL WIP
Organization's Current Y2K Status
Suppliers
Customers
Firm's Infrastructure (INFRA)
Human Resource Management (HRM)
Technological Development (TECH)
Procurement (PROC)
Inbound Logistics (ILOG)
Operations (OPS)
Outbound Logistics (OLOG)
Marketing and Sales (SALE)
Service (SERV)
Utilities
Dynamis Management Consultants
Software Division

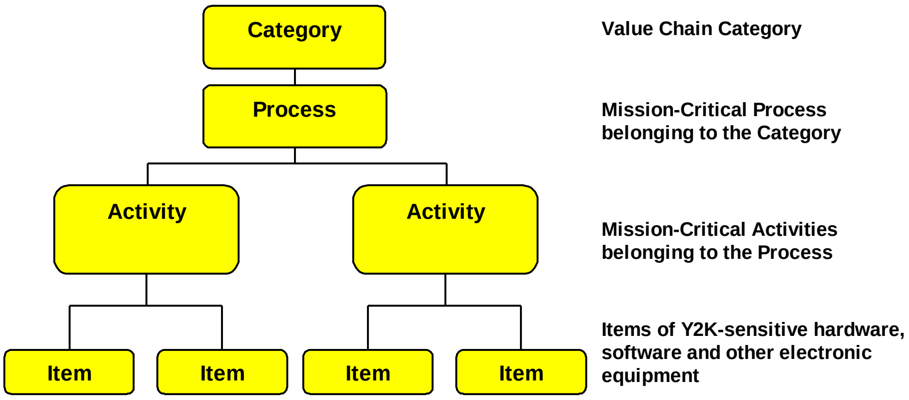
🟥 Non-Y2K compliant
🟧 Contingencies planned
🟨 Contingencies actioned
🟪 Claimed compliant
🟩 Compliant (tested)
⬜ No Processes here

3. Value Chain as Utilised in *Y2Kat*, as adapted from [20].

Following classification of the primary (value-adding) processes and the secondary (support) processes (upper part of figure 3) by the senior management team, the m*ission-critical processes* are identified and their risk (and impact) of Y2K failure used to prioritise them for action.

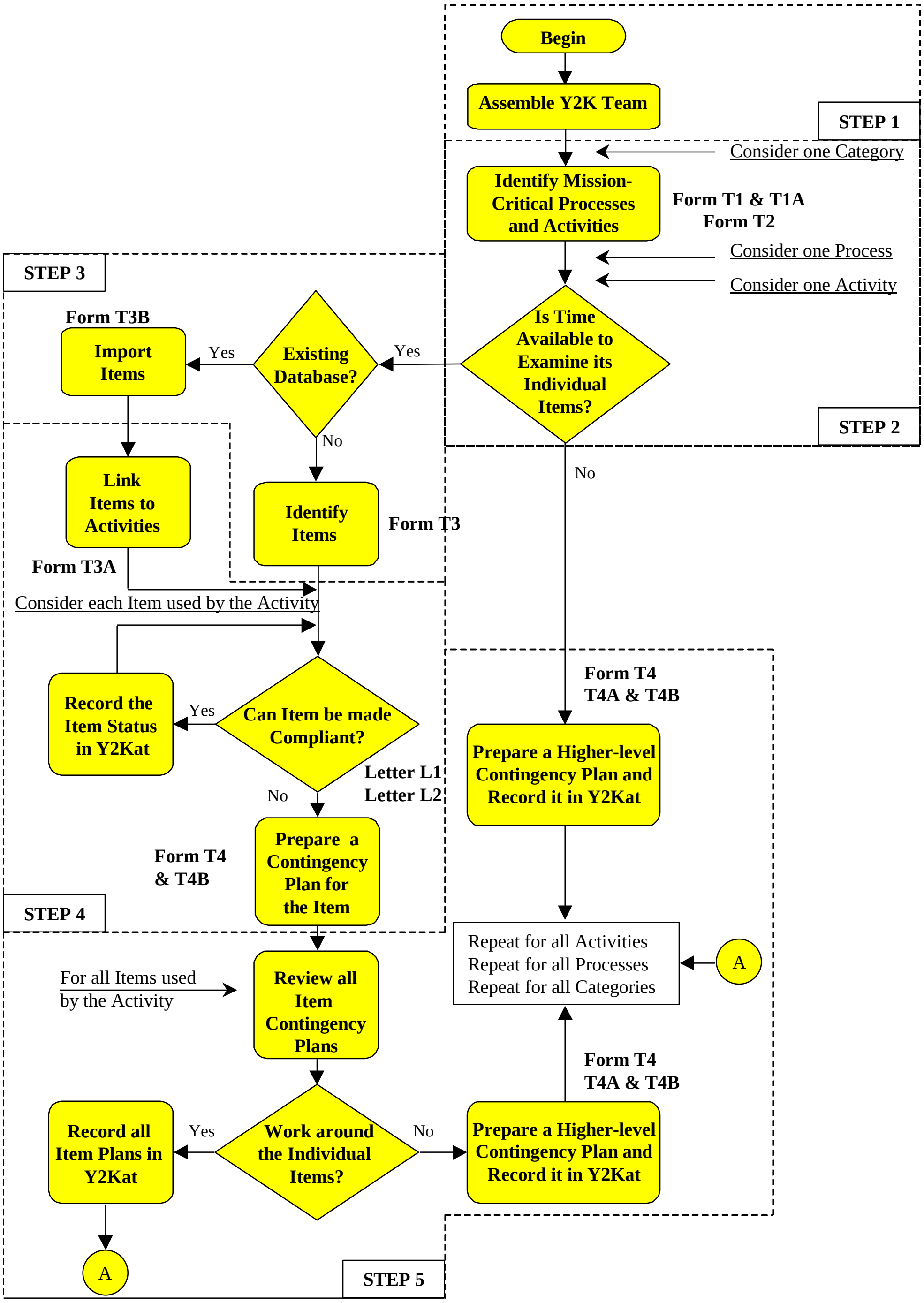
A process comprises a series of end-to-end (non-overlapping) activities. Activities, in turn, comprise a sequence of actual tasks in the organisation that have been grouped together to form a suitable candidate for a contingency plan workaround. Each activity will have both a physical component (the physical task/s) and an information-processing component (tasks that handle information necessary to perform the activity).

The activity may use Y2K-sensitive items of hardware, software, equipment, or embedded microchips, and these may now be located and inventoried, Figure 4.


Category
Value Chain Category
Process
Mission-Critical Process belonging to the Category
Activity
Activity
Mission-Critical Activities belonging to the Process
Item
Item
Item
Item
Items of Y2K-sensitive hardware, software and other electronic equipment

4. Hierarchy of Entities used in the Database.

The DSS model flowchart in Figure 5 below, shows the major decision points.


Begin
Assemble Y2K Team
STEP 1
Consider one Category
Identify Mission-Critical Processes and Activities
Form T1 & T1A
Form T2
Consider one Process
Consider one Activity
Is Time Available to Examine its Individual Items?
STEP 2
Yes
No
Existing Database?
Yes
No
STEP 3
Form T3B
Import Items
Link Items to Activities
Form T3A
Identify Items
Form T3
Consider each Item used by the Activity
Record the Item Status in Y2Kat
Yes
Can Item be made Compliant?
No
Letter L1
Letter L2
Form T4
& T4B
Prepare a Contingency Plan for the Item
STEP 4
Form T4
T4A & T4B
Prepare a Higher-level Contingency Plan and Record it in Y2Kat
Repeat for all Activities
Repeat for all Processes
Repeat for all Categories
A
For all Items used by the Activity
Review all Item Contingency Plans
Form T4
T4A & T4B
Record all Item Plans in Y2Kat
Yes
Work around the Individual Items?
No
Prepare a Higher-level Contingency Plan and Record it in Y2Kat
A
STEP 5

5. DSS Model Flowchart.

The focus on triage (Step 2) and on contingency planning (Step 5) can be readily appreciated. An example of one major decision occurs in Step 2, when the user initially elects to examine the individual items used by an activity rather than immediately planning a higher-level contingency plan for the activity.

The final decision to work around each individual item used by the activity, or to work around higher groupings of items, will depend on factors such as the number of non-compliant items used by the activity, plus the time (and other) resources needed to remedy these items. This is compared with the resources needed to work around higher-level item groups (items grouped by activity, process, or whole category).

2.4 Database

The DSS does not incorporate an internal database. Instead, it communicates with an independent (Microsoft Access) database through data access drivers. Such data independence allows multiple applications access to the same data. The database is also able to undergo enhancements or alterations without the need to immediately upgrade its associated applications [2].

2.5 User Interface

"The success/failure of a DSS is increasingly judged less by its processing speed and problem size, than by its communication capabilities and the interface it provides for the human-computer interaction process.” [7]

Considerable attention was placed on the development of the user interface. In fact the user interface, to a large degree, influenced the design of the other DSS components. Historically, this has not generally been the case; the user interface has often been described as one of the weakest aspects of a DSS [4, 6].

2.6 Other DSS Features

As was mentioned earlier, the model described is part of a self-contained Y2K DSS package, known as *Y2Kat*, that comprises a *system* of user procedures, guides, forms, letters, and DSS software. *Y2Kat* was designed in close collaboration with consulting firm staff and typical end users. The *system* assists the user to:

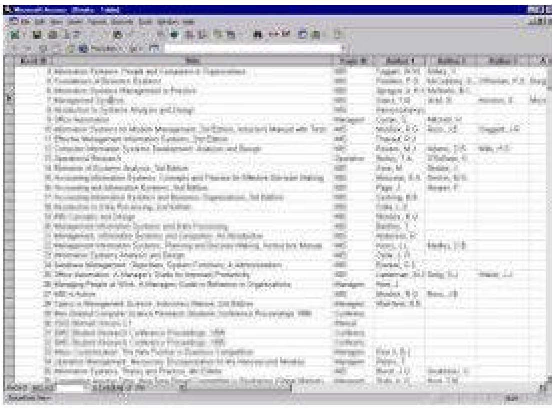
- build an inventory database, or import details from an existing database
- identify and prioritise the organisation’s mission-critical activities
- capture and track the progress of (multi-level, cascading) contingency plan workarounds, including dates and costs
- assess the organisation’s Y2K status at a glance, then drill down to the individual activities and the Y2K-sensitive items they contain
- identify items being used by several activities, or by mission-critical activities with a large failure risk, impact or hazard
- identify processes that may offer a strategic opportunity
- evaluate the future impact of alternate Y2K tactics with the aid of a scenario planner
- search the Y2K inventory database and print letters and forms from inside *Y2Kat*
- hotlink to the *Y2Kat* website at www.y2kat.com, that provides user assistance and Y2K news, plus links to external databases containing Y2K compliance information
- record a secure audit trail
- allow others on a network to view and print from the *Y2Kat* database, aiding intra-company communication

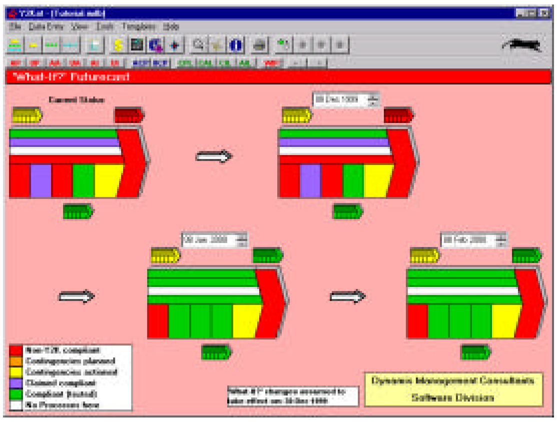
3. Case Study and Selected Results

A subsidiary of a large manufacturing organisation was used as a pilot study for this project.

The company had already initiated a Y2K project that involved construction of a ‘flat’ database file containing several hundred Y2K-sensitive items, Figure 6. However, company officers were experiencing great difficulty interpreting the large amount of information it contained. Using *Y2Kat* procedures, mission-critical processes and activities in the organisation were identified. Relevant items were then imported from the existing database and linked to their activities.

Candidate Y2K strategies were developed and the ‘What-If?’ facility used to ascertain which strategy would offer the greatest protection to the organisation, given its limited resources and project timeframe. Figure 7 shows how the ‘What-If?’ planner can be used to provide a futurecast view of the likely outcome of a particular strategy. The effect of contingency plans coming to fruition can be clearly seen.





6. Client Database.

7. ‘What-If?’ Planner in action.

Y2Kat was able to offer a choice of scenario alternatives ranging from remedying the individual items, all the way up to working around entire categories in the value chain (earlier figure 3).

Overall, the company was able to quickly get to grips with its supply chain exposure to Y2K and to rapidly deploy scarce resources to mitigate the likely effects of date rollover. Early work was commenced on items that were used by several activities, especially where a high degree of failure risk was involved. Such items had not been discernible prior to using *Y2Kat*.

4. Discussion and Limitations

The susceptibility of large companies to Y2K problems suffered by their (upstream and downstream) supply chain partners is a particular concern given the possibility of a domino effect that would impact all companies in the network, -regardless of their own state of Y2K readiness. Yet there are few reports of large firms assisting their smaller suppliers with their Y2K efforts [19].

Regardless of company size, directors are liable for losses suffered by their company as the result of Y2K, -which as a known problem is generally not insurable. The possibility of litigation by injured parties is also high and such problems are predicted to last right into 2001 [24].

Small businesses are generally reluctant to address Y2K issues, either because they do not understand the potential consequences of date-related failures, or they do not feel they have the time to fix them, or they think they cannot afford the Year 2000 consulting fees [18].

A DSS was developed to address these concerns. It has been shown to make Y2K assessment more accessible to smaller businesses that, until now, have not had the knowledge, funds, or experienced personnel to conduct a traditional Y2K initiative.

Although the DSS has been designed for a wide range of businesses and business situations, and for decision makers that may be termed ‘novice users’, it has limitations. For example, the DSS does not support the implementation phase of the decision-making process: intelligence, design, choice, and implementation. Furthermore, what may be considered its greatest strength in one situation may be its greatest weakness in another. In this regard the following limitations may apply:-

- the DSS fails to support all decision-making styles
- the standard operating procedures mean that users do not have full control over all steps of the decision-making process
- the user cannot completely adapt the DSS but must ultimately modify their Y2K initiative to fit

Acknowledgements

The authors gratefully acknowledge the co-operation of the case study company during the preparation of this paper.

References

[1] H.C. Bahl, R.G. Hunt, *Problem-solving Strategies for DSS Design*, Information & Management, 8 (1985), pp 81--88.

[2] C.J. Date, *An Introduction to Database Systems*, (1995), Addison-Wesley, 6th Edition.

[3] S.M. Dillon, *Descriptive Decision Making: Comparing Theory with Practice*, Operational Research Society of New Zealand, Proceedings of the 33rd Annual Conference, Auckland, New Zealand, August 31 - September 1, 1998, pp 99--108.

[4] B.L. Dos Santos, C.W. Holsapple, *A Framework for Designing Adaptive DSS Interfaces*, Decision Support Systems, 5 (1989), pp 1--11.

[5] T. Farris, *Ounce of Prevention for Y2K: Protecting Against Barbarians at the Gate*, Journal of Business Logistics, 4 (1998).

[6] B. Fazlollahi, M.A. Parikh, S. Verma, *Adaptive Decision Support Systems*, Decision Support Systems, 20 (1997), pp 297--315.

[7] G. Fischer, *Human-Computer Interaction Software: Lessons Learned, Challenges Ahead*, IEEE Software, 6(1) (1989), pp 44--52.

[8] J. Good, *A Survivor’s Guide to the Year 2000 Problem – A Non-Technical Handbook for All New Zealanders*, Shoal Bay Press (1997).

[9] W.M. Fletcher, *Computer Crisis 2000*, Self Counsel Press (1998).

[10] M. Haselkorn, reported in Internet WWW page at URL: http://www.mindspring.com/~pci-inc/Year2000/y2ktech.htm (version current at 5 November 1999).

[11] I.S. Hayes, W.M. Ulrich, *The year 2000 Software Crisis: Challenge of the Century*, Prentice Hall (1998).

[12] B. Ives, M.H. Olson, *User Involvement and MIS Success*, Management Science, 30(5) (1984), pp 586--603.

[13] T.C. Jones, *Getting Down to the Wire*, Application Development Trends, 5(4), 1998, pp 37--46.

[14] R.A. Leach, L. Whitman, K.J. Rogers, and D.R. Underdown, *An Enterprise Transformation Methodology to Mitigate Y2K Risk*, Logistics Information Management, 12(3), (1999), pp 269--179.

[15] L.J. Marcoccia, *Building Infrastructure for Fixing the Year 2000 Bug: a Case Study*, Journal of Software Maintenance: Research and Practice, 10, (1998), pp 333--352.

[16] J.D. McKeen, T. Guimaraes, J.C. Wetherbe, *The Relationship Between User Participation and User Satisfaction: An Investigation of Four Contingency Factors*, MIS Quarterly, 4 (1994), pp 427--451.

[17] J. McKendrick, *Small Businesses Ignore Y2K at their Peril*, Midrange Systems, 12(3), March 1 1999, pp 44--46.

[18] L.L. McMurchie, *Small Businesses Slow to take First Steps on Y2K*, Computing Canada, 25(10), March 12 1999.

[19] K. Melymuka, *Big Risks from Small Suppliers*, Computerworld, 33(30): 44, July 26 1999.

[20] M.E. Porter, *Competitive Advantage*, New York Free Press (1985).

[21] F.J. Radermacher, *Decision Support Systems: Scope and Potential*, Decision Support Systems, (12(4/5) (1994), pp 257--265.

[22] S.G. Rostron, *Millennium Bug – Are you solving the right problem?*, Logistics Information Management, 3 (1999).

[23] E. Turban, *Decision Support Systems and Intelligent Systems*, (1998), Prentice Hall, 5th Edition.

[24] US Senate Committee. Internet WWW page at URL: http://www.y2k.gov/new/3rdquarterly.html#summary (version current at 5 November 1999).

[25] B. Zerega, *Y2K Bites the Supply Chain*, Infoworld, 24 August (1998).

[26] M. Zetlin, *The Computer Time Bomb: How to Keep the Century Date Change from Killing Your Organization*, American Management Association, New York, NY (1999).