Rainy Day:
Actuarial Software and Disaster Recovery

Aleksey S. Popelyukhin, Ph.D.

Rainy Day:
Actuarial Software and Disaster Recovery

Aleksey S. Popelyukhin, Ph.D.

Abstract

Tragic events with disastrous consequences that are happening all around the Globe made Disaster Recovery and Continuity Planning a **much higher** priority for every company. Scenarios, in which data centers, paper documents and even recovery specialists themselves may perish, became more probable.

Both, actuarial workflow and actuarial software design should be affected by disaster recovery strategy. Actuaries may simplify recovery task and insure higher rate of success if they properly modify their applications' architecture and their approaches to documenting algorithms and storing structured data.

The article attempts to direct actuaries to strategies that may increase chances of complete recovery: from separation of data and algorithms to effective storage of actuarial objects to automated version management and self-documenting techniques.

The matter of continuity of actuarial operations is in the hand of actuaries themselves.

Rainy Day:

Actuarial Software and Disaster Recovery

Aleksey S. Popelyukhin, Ph.D.

Failed Assumptions

Presumably, every insurance company has a backup system. Files, databases and documents are copied to tapes or CDs and stored offsite. It gives protection against hard disk failure, rogue viruses and provides an audit trail.

Many of the existing backup solutions, however, are built on the ***assumptions*** that after disaster strikes restoration will be performed by the same personnel to the same (compatible) hardware/software system. As the events of September 11th painfully demonstrated, these assumptions may not exactly hold true.

The following unfortunate scenarios became much more plausible:

- *One may have tapes (or other media), but not know what to do with them*
- *One may know what to do with the tape, but not have a compatible system to perform a restoration*
- *One may restore the files, but not have the software to read them*
- *One may get files restored and software working, but not have anyone around to explain how to use it.*

Consequently, disaster recovery and business continuity plans have to address them.

Personnel can perish

A company's tapes stored offsite may survive a disastrous event, but it does not mean they can be used effectively for the restoration. It may not be immediately clear how to perform a restoration: on what hardware with what backup software and in which order. It may also happen that the backup/restore software is so old it requires an older Operating System (OS) not available anymore. It may not be evident how to reinstall software without the manual and a license key. Moreover, there might not be anyone who remembers where to restore, what to restore and in which order.

Thus, it is *imperative* to escrow not only tapes, but also:

- *installation software (OS, backup/restore and other environment programs),*
- *manuals and documentation,*
- *licenses and support info,*
- *and restoration* ***instructions****.*

Sure, it is not up to actuaries to perform the restoration tasks, but it is in their best interests to make sure their software is part of the restoration effort (including installation disks, manuals and licenses) and that they do *everything* possible to simplify that effort.

Restoration Priorities

Any BIA (Business Impact Analysis) study will assign very low priority to the restoration of an Actuarial subsystem. Indeed, experience shows (see [1]) that the most important service for the business continuity is communications.

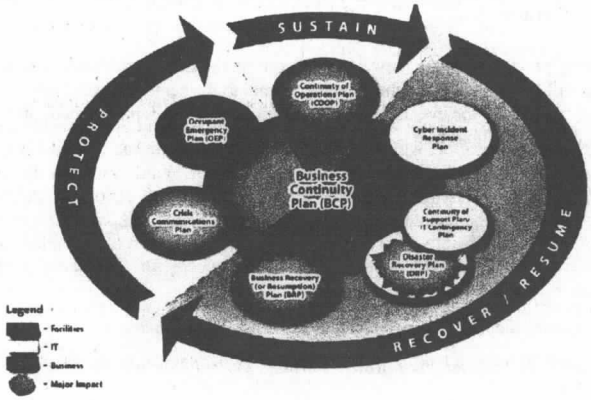


Fig 1 (see [2])

Experience shows that restoration priorities start with e-mail and end with the actuarial subsystem:

- *e-mail/communications*
- *accounting*
- *payroll*
- *trade/marketing*
- *underwriting*
- *claims*
- *actuarial applications*

There is nothing wrong with that picture: it just means that actuaries have to be ready to perform some or all restoration tasks by themselves and not wait for IT department help. It also implies that actuaries would be much better off if their applications were easy to restore or *reproduce* even if some knowledgeable personnel were not available.

Forced upgrading

The implicit assumption behind the majority of existing recovery plans is that restoration would be performed on the same version of hardware/OS combo that was used for backup. Or (given the long upgrade cycles of the recent past) quite similar and compatible versions. Not anymore. Every major OS upgrade may render backup/restore software useless; every advance in drive technology may make backup tapes unreadable. Skip a couple upgrade cycles and you may fail to find the appropriate drive to read your tapes and have no software to recognize recording format. And if the company's computers are destroyed, the company may be *forced* to upgrade.

Thus, downloading patches for backup software should be done as vigorously as downloading for anti-virus and security purposes. It is also crucial to monitor availability of the tape drives backward-compatible with existing tapes.

The same forced upgrading trap may occur with actuarial software. During restoration, one may discover that new computers come only with the newer OS, Utilities and Spreadsheet versions, which are not necessarily compatible with existing files. Imagine if one had to read VisiCalc or WordStar files today. Nobody guarantees that Oracle 7 will run on Windows XP or that Excel will properly interpret that old trustworthy *.wk3 file. It is even more of a problem for third-party proprietary software. It has to be maintained compatible with the latest OS, compiler, hardware key protection software and, possibly, a spreadsheet or a database: quite a formidable task.

Third-party actuarial applications

Sales data from the suppliers of actuarial software imply that actuaries heavily rely on "shrink-wrapped" applications from the third parties. Development, distribution and compatibility of theses applications are controlled by their vendors. Yet, disaster recovery cannot be completed without restoring full functionality of these programs. Actuaries cannot do much about these applications except to make sure that they can be restored.

Adequate code protection

Actuaries may require that their license agreement include a contractual obligations from the supplier for:

- *Adequate code base protection and*
- *Technology Assurance*

Adequate code base protection should include measures taken by the vendor to protect the application code with backups and offsite escrow. In addition, the vendor has to guarantee access to the code in case it goes out business or cannot longer support an application.

Technology Assurance is a fancy name for the continuous compatibility upgrades and patches that would guarantee application compatibility with the ever-changing software environment. Vendors should make sufficient effort to maintain their applications capable of running under the latest OS and interoperating with the latest spreadsheet or the underlying database.

Hardware keys

Actuaries also have to clarify a procedure for restoring third-party applications that utilize hardware-key protection schemes. In a plausible disaster scenario, hardware keys may cease to exist rendering an application useless. In that case

- *Does the License Agreement provide for* replacement *keys?*
- *Can the vendor deliver replacement keys from Australia (England, Connecticut)* fast*?*
- *Does the vendor provide a* downloadable *temporarily unprotected version?*

All these questions have to be answered before the disaster strikes: this way, actuaries can avoid a few unpleasant surprises during restoration.

In-house development

Programming cycle

Aside from analysis, actuaries perform some activities that closely resemble software development. Indeed, no matter what computer language they are using (Lotus, PL/1, APL, Mathematica, VBA), they are programming. Thus, as programmers, they have to conform to development cycle routines established in a programming world. Documentation, versioning, testing, debugging – these activities are well studied, and even automated.

Both actuarial workflow and software design should be affected by disaster the recovery strategy. Actuaries have to design their applications in such a way that somebody else other than the designers can *understand* the spreadsheet, the code and the logic.

Separating Data and Algorithms

A spreadsheet is a very popular actuarial tool. It is so versatile: it can be used as a database and as a calculation engine, as an exchange format and as report generator, as a programming environment and as a rich front-end to Internet. Actuaries use spreadsheets in all these aspects; the problem, though, arises when they use multiple features in one file. More precisely, when they use single spreadsheet file as the *engine* for calculations *and* as the *storage* for results of these calculations, creating multiple copies of the engine.

Actuaries do realize that input data like loss triangles, premiums vectors and industry factors come from outside and do not belong to their calculation template. What they rarely realize (or don't realize at all) is that output results such as predicted ultimates or fitted distribution parameters *do not belong* to the template either, and that they (results) have to be *stored outside* just like input data.

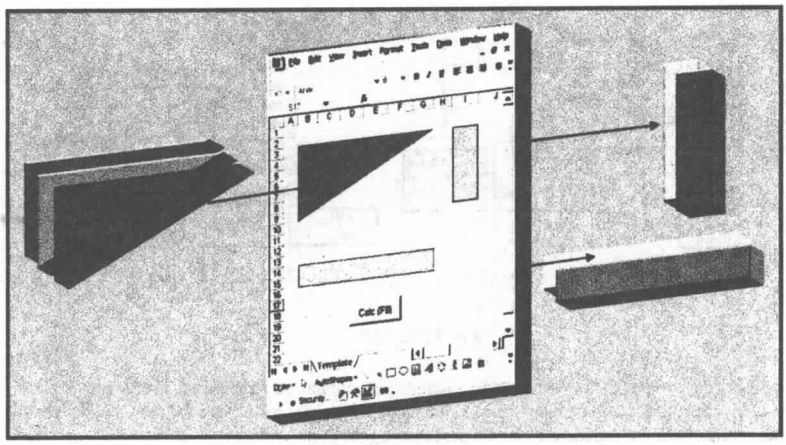


Fig 2

Usually, it is not the case: actuaries routinely create 72 files with the same algorithm for 72 lines of business rather than keep one file and storing 72 answers separately. This strategy creates an obstacle for the effective:

- *debugging/versioning,*
- *modifications/improvements,*
- *integrity/security,*
- *reporting and*
- *multi-user access.*

Indeed, correcting an error in one file takes 72 times less time than correcting the same error in 72 copies of that file. Extracting answers for reports from 72 files requires much more effort then summarizing 72 records in the database. And it is much harder to guarantee that nobody modified the 57th file incorrectly.

From a recovery standpoint, restoring a single file with formulas and separate data records is definitely easier then restoring 72 files with commingled data and formulae, especially given that the probability of a corrupted file is 72 times greater for 72 files.

It would be wise for actuaries to modify their workflow and spreadsheet design in order to separate data and algorithms. Rethinking their methodology in this light, actuaries *inevitably* will arrive at the idea to store data along with some kind of description, that is, to treat data as Actuarial Objects[♣] (see [3]).

[♣] See [5] to learn how to program custom objects in Excel VBA.

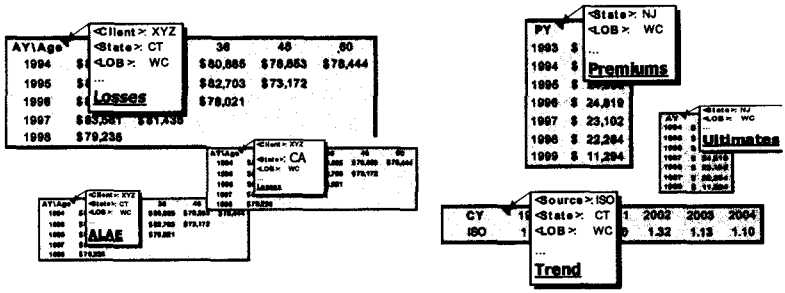


Fig 3 (see [4])

The logical extension of this idea would be to modify a calculation template in such a way that it understands these descriptions and acts upon them, serving like a traffic cop for the data objects. In other words, build an engine for objects processing (see [4]).

This architecture would streamline actuarial workflow, encourage debugging and modifications, simplify reporting and improve enormously recovery success chances.

Input and output to simpler formats

If (in addition to all its functionality) Microsoft Excel incorporated simulations, there would be only one actuarial software program: Excel. And, thus, no worries about file formats, conversions and availability of the file reader programs. Fortunately or unfortunately, this is not the case, and actuaries have to rely on applications with different file formats. The problem is that, as time passes, it will be harder and harder to find a reader program for some obscure and proprietary files. So, for the sake of disaster recovery actuaries should rely on the most ubiquitous file formats.

In the foreseeable future one can count on the ability to read ASCII (including XML and HTML), *.xls and *.doc files. Perhaps, *.dbf and *.pdf readers will be easily available too. Consequently, it is always a good idea to store an extra copy of the most crucial data in one of the aforementioned formats.

For example, SQL Server and Oracle tables can be dumped into ASCII*. Microsoft Access can import/export tables in *.xls or *.dbf formats. Excel files can be seamlessly converted to XML (structured ASCII). And, unless there are trade-secret considerations, it is always a good idea to export VBA modules to *.bas text files.

From a disaster recovery standpoint, it is important that third-party software has the capability to read and write to one of the ubiquitous formats (a side benefit of that capability would be possibility of data exchange and integration with other software programs).

---

* In fact, the whole database can be restored from ASCII files by running SQL scripts (ASCII) that recreate database structure and loading tables' content from the dump file (ASCII).

Version management

If a calculation algorithm is used (or is going to be used) *more than once*, it needs versioning. Indeed, if a "separation of data and algorithms" paradigm is embraced and implemented, it becomes quite practical and useful to maintain a version of the algorithm (in case of Excel: a version of template used for calculations).

The usefulness becomes obvious once one considers saving the version of the calculation engine along with results of calculation. Doing so helps immensely in audit trailing, debugging and, of course, recovery.

The practicality derives from the fact that (presumably) the number of calculation engines/templates is limited (usually the same algorithm can be reused for analysis of multiple contracts, LOBs and products). So maintaining version information for a few files is not an overburdening chore.

Microsoft Office applications provide adequate facilities for versioning: Word automatically updates "Revision number" (File/Properties/Statistics) and Excel allows custom properties to be linked to cells inside the spreadsheet.

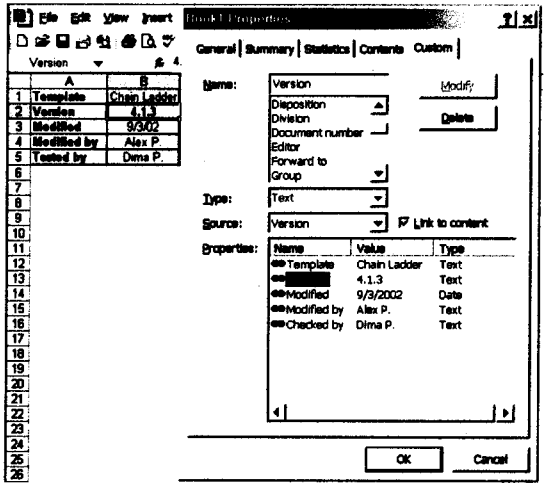


Fig 4

If the user would dedicate a cell in a spreadsheet to store version info and add one line of VBA code to the Workbook_BeforeSave event, he would get a "poor man" versioning mechanism for free.

```
Private Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean)
    Range("Version").Value = InputBox("Version number: ", "Properties", Range("Version"))
End Sub
```

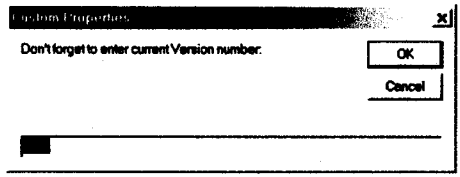


Fig 5

If there is a necessity to synchronize the version number through several files, the cell linked to the custom property can contain a formula referring to the information in the other (main) template.

Using files properties for versioning (and, possibly, other information*) has some nice side benefits: one can use them for targeted file searches (File/Open/Tools/Search/Advanced).

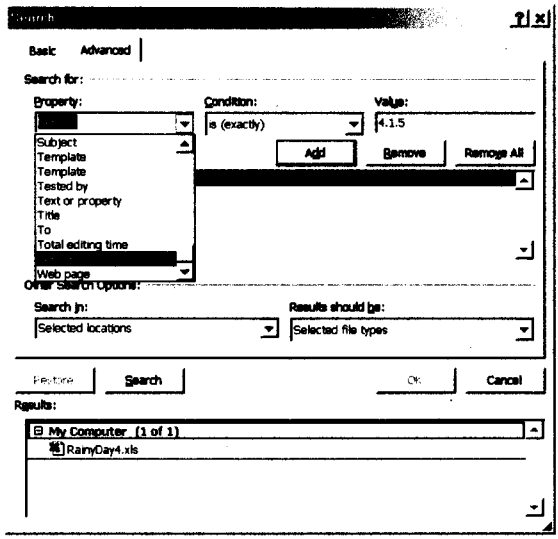


Fig 6

* It is always a good idea to dedicate an area (with named cells) in a spreadsheet to meta-information about it and link these cells to the Custom Properties like "Created by", "Last Modified by", "Verified by", etc...

More important, however, is the fact that Version info can be stored alongside with Results generated by the template, providing perfect means for the audit tracking.

Documentation

Knowledge about available documentation features and familiarity with restoration techniques may help actuaries to *design* their software in order to greatly simplify potential recovery efforts.

Usually, big nice printed manuals and an interactive online help system[♣] are reserved for very large projects only. It is unreasonable to expect an actuary to write a manuscript for every Excel spreadsheet he creates in a hurry. Nevertheless, several simple approaches can be employed to greatly simplify restoration tasks:

- *Self-documenting,*
- *Excel Comments,*
- *Code Remarks.*

Self-documenting features

Microsoft Office programs provide adequate assistance for self-documentation attempts. If an author follows a few unobtrusive styling conventions, then Microsoft Word can easily generate an outline or table of content. Microsoft Access has an indispensable utility called Documenter (Tools/Analyze/Documenter):

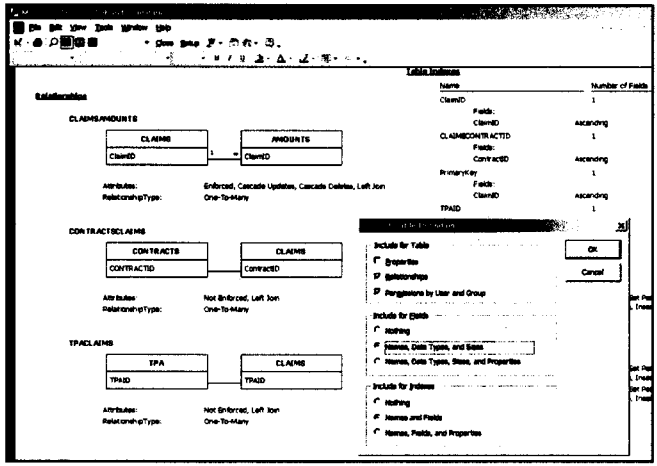


Fig 7

♣ Several packages on the market, most notably RoboHelp from eHelp, can convert Microsoft Word file(s) into a full-featured interactive help system, either Windows or HTML based.

Documenter generates Notepad or Excel files with the information about any object in a database with as many details as necessary.

Microsoft Access also provides facility for fields' descriptions (along with tables, queries, forms and reports descriptions): it would be unwise not to use it.

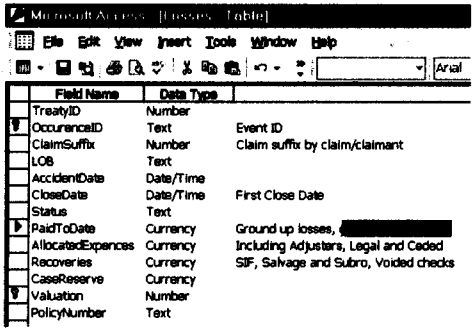


Fig 8

Excel is, by its very nature, self-documenting: by clicking on a cell the user can see the value in a cell and a formula behind it in a Formula Bar. To view formulae in multiple cells one can open a second window of the same spreadsheet (Window/new Window) and switch its mode to Formula View (Tools/Options/View/Formulas or just *press* CTRL - ~).

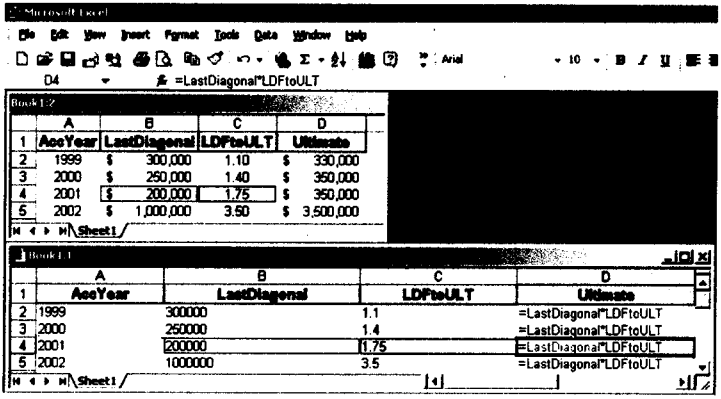


Fig 9

In recognition that building models in Excel is, essentially, some kind of programming, Microsoft added a quintessential debugging tool to Excel: Watch Window (Tools/Formula Auditing/Show Watch Window or *right click on cell*/Add Watch). Watch Window allows the user to track values

and simultaneously see formulas of multiple cells located anywhere in a spreadsheet. The tool's value is not only in debugging, but also in ad-hoc goal seeking and audit trailing. Accompanying it is the step-by-step Formula Evaluation tool (Tools/Formula Auditing/Evaluate Formula), which used to be Excel4 Macro debugging instrument.

Sure, cells are not the only place for formulae and settings: PivotTables, Solver add-in, Links, External Data ranges, Web Queries and even Conditional Formatting contain important information which could be crucial for understanding the functionality of an algorithm.

To display Calculated Fields and Items in a PivotTable or PivotChart, click anywhere on a PivotTable and then in a PivotTable toolbar select Formulas/List Formulas.

To view the query behind the External Data range, right-click on it and select Edit Query item from the menu (or choose Data/Get External Data/Edit Query). The same procedure works for Web Queries. What's important is that in both cases Excel provides an option to save a query as a text file (*.dqy in case of Data Queries and *.iqy for Internet Queries). It is highly recommended to do so. The benefit is threefold: a) queries get documented, b) it is easier to modify them in this text form and c) it is so easy to execute them - just double-clicking on a *.dqy or *.iqy file.

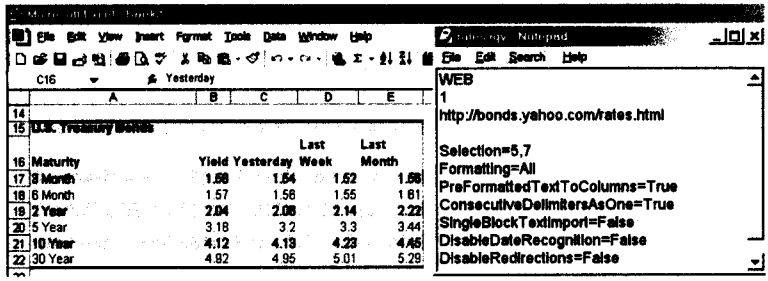


Fig 10

Solver* is a very popular goal-seeking tool and, thankfully, Excel preserves Solver settings, but only one per worksheet. This means that if Solver is used multiple times on the same sheet, it is a good practice to save its settings (Tools/Solver/Options/Save Model) in a descriptively labeled area.

Important settings are stored in Conditionally Formatted as well as Data Validated cells. To see validation settings and format conditions, navigate to these cells using Edit/Go To/Special dialog box.

* To access Solver select Tools/Solver from the Excel menu. If Solver is not listed in the menu, check whether it's installed (run Office install) and/or enabled (checkmark in Tools/Add-ins).

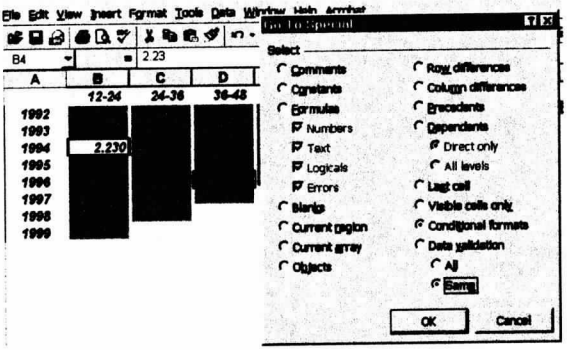
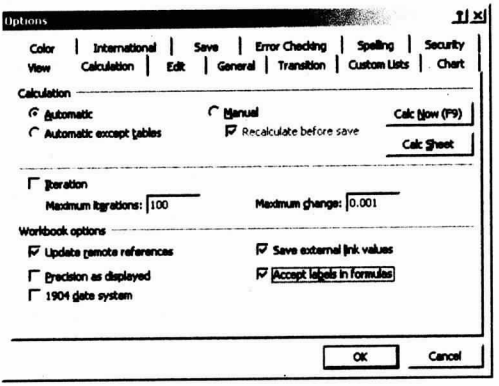


Fig 11

Most users use Names (Insert/Name/Define/Add) only for naming ranges. However, Excel allows giving a Name to any formula, even a User-Defined one. To display the list of Names with their definitions, press F3♠/Paste List. Names in Excel are too important -- and too convenient tool for documenting a spreadsheet -- to be ignored. In the ideal world, there should be no unnamed references in Excel formulas: every variable, input region and output location has to be named. Good naming conventions along with the habit of naming ranges and cells may prove invaluable not only for disaster recovery, but for debugging, modifications and education of the new employees.

Excel creators believe in named references so much they actually supply Names even if the user himself didn't define any. Since version 97, users can use column and row labels as if you created range names for rows and columns (since Office XP, the same syntax works for PivotTables♠♠). To enable this functionality, check the Tools/Options/Calculation/Accept Labels in Formulas option.



♠ Use F3 to paste a Name while typing formula text in a Formula Bar to avoid misprints.
♠♠ To get data from the PivotTable use GetPivotData function.

SUM =LastDiagonal*LDFtoULT

	A	B	C	D
1	AccYear	LastDiagonal	LDFtoULT	Ultimate
2	1999	$ 300,000	1.10	$ 330,000
3	2000	$ 250,000	1.40	$ 350,000
4	2001	$ 200,000	1.75	=LastDiagonal*LDFtoULT
5	2002	$ 100,000	3.50	$ 350,000
6				

Fig 12

Structured Comments

Excel comments, if used creatively, represent an amazingly powerful tool. Available through the Reviewing toolbar, comments can be toggled (by moving mouse over commented cell) or displayed permanently (Show Comment). They can be printed "in place" or as footnotes (File/Page Setup/Sheet/Comments *dropdown*). Moreover, comments (as in any programming environment) are invaluable for documenting designer's intentions and understanding algorithm's logic.

In addition to their special role in documentation of a spreadsheet, and consequently in any restoration effort, comments may play an even bigger role if used as Object's descriptors. Indeed, given that comments are

- *an ASCII text*
- *associated with a range and*
- *can be manipulated (created, used, modified and deleted) through VBA*

comments can be used for storing *structured attributes* of an object a la XML (see [4]):

AY\Age	<State>: CT <LOB>: WC	36	48	60
1994	$ ...	107,847	$ 115,288	$ 124,592
1995	$ Shape --> Triangle	110,271	$ 112,562	
1996	$ Amount--> Losses	104,029		
1997	$ Cumulative- True			
1998	$ 105,647			

Fig 13 (from [4])

One can think of a comment as a "price" tag attached to an Actuarial Object. A user-defined VBA function that accepts such ranges as input may read that tag and decide what to do with associated ranges (if it is a triangle of losses the function may divide it by the corresponding claim counts; if it is a vector of loss development factors a function may multiply it by the last diagonal and if it is column of loss ratios the function will prohibit any attempt to add inflation factors to it).

Auto backup copy

To increase the chances of recovery of the most important Excel files, it is wise to enable a built-in facility for the automatic creation of backup copies. By launching "Save Options" dialog (File/Save As/Tools/General Options) and choosing "Always create backup" option, user can be assured that every time he saves the file an extra copy with the extension XLK is generated.

Still, for occurrences when files are corrupted or incompletely restored from the tapes, Excel 2002 has beefed up its file repair utility. Available through the (File/Open/Open *dropdown*/Open and Repair) menu item, the utility does a formidable job in recovering corrupted files.

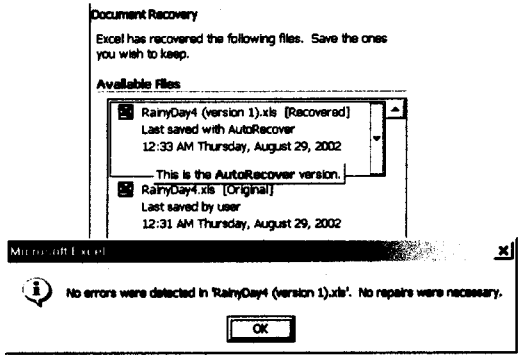


Fig 14

In a rare case, when an attempt to repair fails, as a last resort one can try to paste the content of the corrupted file into a new spreadsheet (see [6]). To do that, open two new files, select cell A1 and copy it to clipboard, switch to the second file and after pasting the link (Edit/Paste Special/Paste Link), change the link to the corrupted file. In most cases, Excel allows the user to access this way as much content as it could recover. The rest of the file (VBA modules, External Data queries and Pivot Table cubes) can be imported from the ASCII files.

Documenting Workflow

As crucial as preservation of files and documentation of algorithms is the process of diagramming actuarial workflow. The order of actions grouped by stages with the references to file locations and processes is an invaluable restoration asset.

There are many ways to document workflow. The most natural and powerful, though, is to use "smart diagramming" software like Microsoft Visio or Micrografx iGrafx by Corel. In addition to their ability to document, analyze and simulate workflow, these packages (empowered by VBA) may execute some actions automatically.

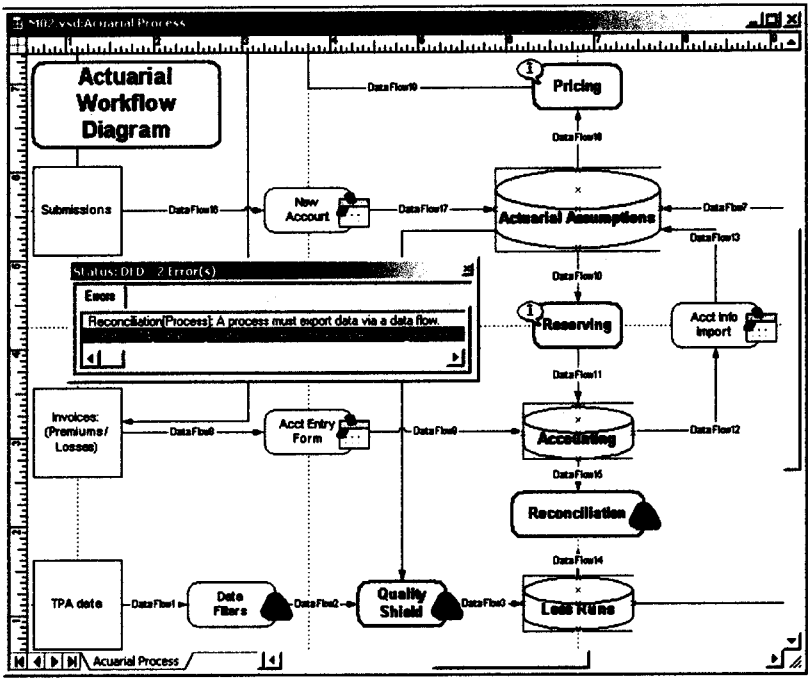


Fig 15

Workflow diagrams - as important as they are for disaster recovery – provide additional benefit as a way to look at the actuarial process as a whole, and possibly to streamline and simplify it.

Telecommute

Telecommuters present yet another challenge for flawless disaster recovery. A whole additional layer of network subsystems (terminal services, VPN access, firewall) has to be restored in order to enable their access to the company's applications. Home and mobile computers and devices represent an additional hazard for security and maintenance.

However, provided that security, connectivity, maintenance and support issues are solved, home computers will become a decentralized independent distributed file storage system: an additional chance to restore a copy of this most important lost file. Also, if configured accordingly, remote computers may serve as a temporary replacement system until restoration of the main system is complete. Indeed, many applications can be scaled to work on a standalone machine: major databases have compatible "personal" versions, while Office and many third-party actuarial applications are "personal" by nature. The synchronization with the "main" system can be possibly achieved via import/export to/from ubiquitous file formats.

Paperless Office

If the backup is set up and working smoothly, files are copied to tapes and stored offsite, then there is *no excuse **not to*** scan every paper document greatly reducing risk of losing it. Indeed, with advances in scanning quality and OCR (optical character recognition) accuracy, it makes perfect sense to convert all paper documents into computer readable files. The ubiquitous PDF (portable document format) file preserves the look of the original, while at the same time enabling index, catalog and search services to scan through its content as if it were simple text file. Even Internet Search Engines are now PDF-enabled, so Internet search queries are capable of looking for information inside PDF files. Thus, scanned paper documents can be organized into a useful searchable hierarchical "knowledge base" instead of lying in some storage boxes, being hard to find and, probably, unused.

Once again, an action geared toward better disaster protection may turn out to have a great side benefit, perhaps even greater than the initial purpose of the action.

Conclusion

Any type of action – from a big radical change of architecture in order to "separate data from algorithms" to a small conventional "enabling of auto-backups" in Excel - is better than no action.

Besides, all aforementioned recommendations help not only in the case of devastating disaster, but also in the event of a virus attack, malicious user actions, and staff rotation. In fact, benefits from such *preparation* measures as

- *clear documentation of actuarial procedures,*
- *streamlined algorithms and*
- *more effective workflow*

may far outweigh the potential payback from the original objective of disaster preparedness. These measures are more than worthy by themselves. Surely, the **cost of precautions should not exceed estimated damages**. However, side benefits such as audit trail capabilities, design discipline and improved understanding of calculations can easily justify disaster preparedness efforts.

Dedication

To Giya Aivazov and all friends and colleagues affected by The September 11.

Stamford, 2001

Bibliography

[1] Annlee Hines. *Planning for Survivable Networks: Ensuing Business Continuity.* 2002, Wiley Publishing

[2] http://csrc.nist.gov/publications/nistpubs/800-34/sp800-34.pdf

[3] Aleksey S. Popelyukhin. *The Big Picture: Actuarial Process from the Data Processing Point of View.* 1996, Library of Congress

[4] Aleksey S. Popelyukhin. *On Hierarchy of Actuarial Objects: Data Processing from the Actuarial Point of View.* Spring 1999, CAS Forum

[5] Michael Koflet. *Definite Guide to Excel VBA.* 2000, apress

[6] Mark Dodge. *Microsoft Excel Version 2002 Inside Out.* 2001, Penguin Books

Disaster Recovery websites:

[7] http://www.fema.gov

[8] http://www.disasterplan.com

[9] http://www.disasterrecoveryworld.com