DIGITAL PRESERVATION EUROPE

DIGITAL PRESERVATION CHALLENGE

1st edition

SOLUTIONS WALKTHROUGH REPORT

José Miguel Araújo Ferreira
Department of Information Systems
University of Minho
4800-058 Guimarães, Portugal

mferreira@dsi.uminho.pt

July 2007

Table of contents

1 Introduction

The Digital Preservation Challenge is an initiative of the Digital Preservation Europe (DPE) consortium that invites individual participants to overcome the barriers that hinder the access to the content of digital objects. The challenge is composed of six real-life preservation scenarios that the applicant is supposed to overcome in order to gain access to the information carried within the objects. These scenarios intend to make the challenge more accessible to participants from all backgrounds while not trivialising the serious nature of the digital preservation challenges that currently face society.

This report attempts to describe all the steps undertaken to solve the six preservation scenarios promoted by the Digital Preservation Europe (DPE) in its first Digital Preservation Challenge. The report includes a description of the relevant software applications used during the problem-solving steps of this challenge, a description of the most cited digital preservation strategies and solution walkthroughs for all the proposed preservation scenarios. In the end some conclusions are drawn by the author.

This report is organised as follows: section 2 provides a brief description of all the software tools that were used to resolve the proposed preservation problems. Section 3 provides a description of the most relevant preservation strategies in practice. Section 4 depicts all the steps taken to solve the proposed scenarios. And finally, section 4 draws some conclusions on this challenge.

2 Software tools

This section describes the assortment of software tools that were used to solve the proposed digital preservation scenarios. Tools are grouped by type, such as, operating systems, format identification tools, readers/viewers, enabling applications and emulators.

Note: Most of the descriptions included in this section were quoted directly from Wikipedia[1] with a few minor edits.

2.1 Operating systems

Three different operating systems were used thoroughly throughout the digital preservation challenge: Mac OS X[2], Ubuntu Linux[3] and Windows XP[4]. The reason

[1] http://www.wikipedia.org

[2] http://www.apple.com/macosx/

[3] http://www.ubuntu.com/

[4] http://www.microsoft.com/windows/products/windowsxp

for this was that although we had a base operating system (i.e. Mac OS X), some of the problems claimed the use of tools only available in other operating systems.

Mac OS X

Mac OS X is a proprietary, graphical operating system developed, marketed and sold by Apple Inc., the latest of which is pre-loaded on all currently shipping Macintosh computers. Mac OS X is the successor to the original Mac OS, which had been Apple's primary operating system since 1984. Unlike its predecessor, Mac OS X is a Unix-like operating system built on technology that had been developed at NeXT through the second half of the 1980s and up until Apple purchased the company in early 1997. Because Mac OS X is based on UNIX, most software packages written for BSD or Linux can be recompiled to run on it [1].

Windows XP

Windows XP is a proprietary operating system developed by Microsoft for use on general-purpose computer systems, including home and business desktops, notebook computers and media centres. Windows XP is the successor to both Windows 2000 and Windows Me, and is the first consumer-oriented operating system produced by Microsoft to be built on the Windows NT kernel and architecture. Windows XP was first released on October 25, 2001, and over 400 million copies are in use, according to a January 2006 estimate by an IDC analyst [2].

Ubuntu Linux

Ubuntu is a widely used Linux distribution predominantly targeted at personal computers. Based on Debian GNU/Linux, Ubuntu concentrates on usability, regular releases, ease of installation, and freedom from legal restrictions. Ubuntu is sponsored by Canonical Ltd., a private company founded by South African entrepreneur Mark Shuttleworth [3].

2.2 Format identification tools

Droid

DROID[5] (Digital Record Object Identification) is a software tool developed by The UK's National Archives to perform automated batch identification of file formats. It

[5] http://droid.sourceforge.net

is one of a planned series of tools utilising PRONOM to provide specific digital preservation services. DROID uses internal (byte sequence) and external (file extension) signatures to identify and report the specific file format versions of digital files. These signatures are stored in an XML signature file, generated from information recorded in the PRONOM technical registry. New and updated signatures are regularly added to PRONOM, and DROID can be configured to automatically download updated signature files from the PRONOM website via web services [4].

Unix file

file is a standard Unix program for determining the type of data contained in a file [5].

The Single Unix Specification (SUS) specifies that a series of tests are performed on the file specified on the command line [5]:

- if the file cannot be read, its status undetermined, or its type undetermined, file will indicate that the file was processed and its type was undetermined.
- *file* must be able to determine directory, FIFO, socket, block special and character special types
- zero files are identified as such
- an initial part of file is considered and file is to use position-sensitive tests
- the entire file is considered and file is to use context-sensitive tests
- the file is identified as a data file

file's position-sensitive tests are normally implemented by matching various locations within the file against a textual database of magic numbers. This differs from other simpler methods such as file extensions and schemes like MIME. [5]

2.3 Readers/viewers

ChessBase Light 2007

ChessBase[6] is a company that markets chess software, maintains a chess news Web site, and operates a server for online chess playing [6].

ChessBase 2007 (the current version) is a popular commercial database program for storing and searching records of chess games which runs under Microsoft Windows. ChessBase uses a proprietary format for storing games, but can also handle games in

[6] http://www.chessbase.com/

portable game notation (PGN). The proprietary format uses less hard drive space and manages information that is not possible in PGN. The software converts files from PGN to ChessBase format or from ChessBase to PGN [6].

ChessBase also has a free version of its ChessBase program called ChessBase Light. The free version is a pared down version of ChessBase 6 that has a limit of only 8000 games per database - too few for accessing the one million game database that comes with a comprehensive database, such as ChessBase Mega (currently containing approximately three million games), but adequate for storing one's own games and other specialized collections, such as the weekly instalments of The Week in Chess [6].

Framework III

Framework[7], launched in 1984, was the first office suite to run on the PC 8086 with DOS operating system. An even earlier integrated suite, actually comparable to the original Macintosh of 1984 and Lisa of 1982 was produced by Epson, a complete integrated work station based on the previous Z80 processor and CPM operating system with GUI interface and WYSIWYG ("what you see is what you get") typography on the monitor and printing. Framework offered all this however in the first all-in-one package to run on any PC platform. Framework was not separate products with similar look and feel that could share or import or "plug-in" other modules or files, but was a single workspace construct that could contain any combination of word processor, outliner, mini-database and spreadsheet "frames." The spreadsheet program was superior in its day, offering true 3D capability, where any cell could be "opened" to reveal a separate spreadsheet -- a feat of sheer convenient function never again seen [7].

In 1983 Robert Carr and Marty Mazner founded Forefront Corporation to develop Framework. In July of that year, they approached Ashton-Tate to provide the venture capital and to later market the product. Together with a team of six other individuals, Carr and company released the original Framework. The product proved successful enough that in 1985, Ashton-Tate bought Forefront, a year sooner than planned [7].

Ashton-Tate continued to enhance the product by producing Framework II, Framework III, and finally Framework IV in 1989. Beginning with Framework III, the company also produced Framework III Runtime and Framework III Developer's Toolkit. These products allowed application developers to create business applications using the internal FRED programming language, and to hide the user interface from the end-users [7].

While it remains a DOS program, Framework also works on most versions of Windows. More specifically, Framework 7 (which is still supported) was the last

[7] http://www.framework.com/

version which can be run on Windows 95/98/ME or on DOS. Framework 8 and 9 only run on Windows XP although they still run in a separate box within WinXP and provide access to DOS functionality such as batch commands [7].

2.4 Enabling applications

Tomcat 5

Apache Tomcat[8] is a web container developed at the Apache Software Foundation (ASF). Tomcat implements the servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Tomcat includes its own internal HTTP server [8].

Hexadecimal editor

A hex editor (or binary file editor or byte editor) is a type of computer program that allows a user to manipulate binary (normally non-plain text) computer files [9].

By using a hex editor, a user can see or edit the raw and exact contents of a file as opposed to the interpretation of the same content that other, higher level application software may associate with the file format. For example, this could be raw image data, in contrast to the way image editing software would interpret the same file [9].

In most hex editor applications the data of the computer file is represented as hexadecimal values grouped in two groups of 8 bytes and one group of 16 ASCII characters, nonprintable characters normally represented by a dot (".") in the ASCII part [9].

2.5 Emulators

VisualBoyAdvance

The VisualBoyAdvance (also known as VBA for short) is a free software emulator distributed under the terms of the GNU General Public License. It emulates software targeted for the Game Boy, Super Game Boy, Game Boy Color, and Game Boy Advance handheld game consoles sold by Nintendo. As of 2006, the

[8] http://tomcat.apache.org/

VisualBoyAdvance is the most popular such emulator for Microsoft Windows, and is one of the most widely-distributed Game Boy-compatible emulators [10].

Besides the DirectX version for the Windows plattform there is also one that is based on the free platform independent graphics library Simple DirectMedia Layer (SDL). This is available for a variety of operating systems like Linux, BSD, Mac OS X, Windows older BeOS [10].

DOSBox

DOSBox is an x86 emulator which creates a DOS-like environment intended for running MS-DOS-based IBM PC compatible programs, especially computer games, which may not run properly on newer PCs and may not run at all on non-IBM PC compatibles (e.g. PowerPC Macintosh). It also allows such games to be run on other operating systems that do not normally support DOS programs or run them too fast with normal compatibility layers. DOSBox is open source and available for many operating systems, such as Linux, OpenBSD, FreeBSD, Windows 9x, Windows NT 4.0, Windows 2000, Windows XP (32-bit and 64-bit), Windows 2003, Windows Vista, Mac OS X, OS/2, Palm OS, RISC OS, and BeOS [11].

3 Preservation strategies

This section describes the most cited strategies for digital preservation: emulation, encapsulation and migration.

3.1 Emulation

The emulation strategy consists in the use of special software to reproduce the behaviour of a given hardware/software platform in a different technological environment [12]. This strategy allows the user to interpret digital objects by running the same software that was used in their creation [12, 13]. Emulation plays an important role in preservation, especially in the interpretation of highly dynamic and/or interactive objects, such as software [14, 15].

3.2 Encapsulation

The encapsulation strategy aspires to solve the problem of digital preservation by storing together with the object, enough information about how it should be interpreted [16]. This information may consist of a formal specification of the object's format which will enable the future development of viewers, emulators or converters. Lorie argues that this formal specification could be expressed in some sort of machine language compiled for a virtual machine [17, 18]. If the virtual machine is sufficiently well documented, it should be possible to recreate it in a future platform. The encapsulation strategy is usually applied to collections of objects which are expected to remain unused for long periods of time. Using encapsulation on such collections may reduce the number of preservation interventions leading to a generalised reduction in preservation costs [15].

3.3 Migration

The migration strategy consists of a "(…) set of organized tasks designed to achieve the periodic transfer of digital materials from one hardware/software configuration to another, or from one generation of computer technology to a subsequent generation." [19]. Contrary to other preservation strategies, migration does not intend to maintain the digital object in its original format. Alternatively, it converts the object from a near obsolete format to a format that users are able to interpret with their up-to-date computers. The main disadvantage of this approach relies on the fact that when an object is migrated, some of its properties may not be completely or adequately transferred to the target format, i.e. some sort of data loss may be experienced. The reason for this is twofold: there may be structural incompatibilities between the source and the target formats or the converter may be faulty and therefore incapable of performing its task appropriately.

There are a considerable number of migration variants such as normalisation or migration on-request. Normalisation consists in the conversion of digital objects from a wide range of distinct formats to a smaller and more manageable set of formats. The rationale behind it is based on the idea that by reducing the number of formats in custody, subsequent preservation interventions will eventually be simplified as the same strategy may be applied to a higher number of objects. Normalisation is usually performed by the digital archive during ingest of new material or by the material's producer before any archival process is initiated [20, 21].

Migration on-request aims at reducing the data loss problem usually associated with migration strategies by applying all the necessary transformations to the original object instead of converting any of its derivatives that have resulted from previous migrations [22]. In this strategy the focus is not on the preservation of the digital object but in making sure that the migration tool remains usable over time [13, 15].

3.4 Refreshment

Refreshment consists of the transference of information from its original physical medium to another one, usually more recent, before the first reaches a state of irreparable deterioration or obsolescence. If the medium becomes corrupt or obsolete, the information inscribed on it will be lost forever [19, 23, 24].

Refreshment should not be understood as a preservation strategy *per se*. Instead, we should see it as a fundamental requisite to enable any of the other previously described preservation strategies to be applied [25].

4 Task descriptions and solution walkthroughs

The following sections include detailed descriptions of all the steps taken to solve the problems depicted on each of the preservation scenarios of this challenge. The sections are organised as follows: first, a short description of the scenario is given. Each description is then followed by a list of tasks that the applicant is required to address. For each preservation scenario there is a detailed report on all the actions undertaken.

4.1 File Format Identification

Scenario description

You have met someone online and instant messaging has been your only form of contact. You agree to meet and your friend sends you a sound file using instant messaging to provide you with the details of where and when this meeting will occur. Unfortunately your computer crashes before the file transfer is complete and most of your hard drive is destroyed. You find the sound file that was being transferred in a temporary folder but all of the contact information you had for your friend is lost. The only way you can contact your friend again is to recover this file and meet them in person as agreed - can you do it?

Task

1. Identify the file format. 2. Fix its header. 3. Show that you are able to play/decode the file by stating when and where the meeting should take place. 4. Decide if the processing of such files could be automated and if so whether this processing would also be applicable for hundreds of files. 5. Explain why the file format chosen was appropriate or inappropriate for the task of sending this information online. 6. If you decide the file format was inappropriate explain which format would have been more suitable.

Solution walkthrough

To identify format of the transmitted message we used the UK's National Archives Droid application. However, Droid was unable to identify the object's format, probably due to fact that the file header had been corrupted during the file transfer (i.e. the magic number embedded in the file was ruined). We attempted to use the

Unix *file* command as well, but again we got unsuccessful results - it would only suggest that the file was of *data* type.

As last resort, a hexadecimal editor was used to open file and attempt to manually identify its format by looking for clues within the file itself. Fortunately, the format revelled it self quite easy to identify. Several references to the *Ogg* format and the *Speex* encoder could easily be distinguished from the overall data. For example, the following string of text could be seen very close to the beginning of the file: *"[…] OggS […] Encoded with Speex 1.1.7 […] OggS […]"*

There were 4 bytes of data at the beginning of the file that looked quite suspicious. These bytes were represented by the string of text "XXXX". In order to reconstruct the header, we searched the Web for the common Ogg header. We found that information on Wikipedia as follows [26]:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1| Byte
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| capture_pattern: Magic number for page start "OggS"           | 0-3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| version       | header_type   | granule_position              | 4-7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               | 8-11
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               | bitstream_serial_number       | 12-15
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               | page_sequence_number          | 16-19
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               | CRC_checksum                  | 20-23
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               | page_segments | segment_table | 24-27
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ...                                                           | 28-
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

> *Capture Pattern - 32 bits: The capture pattern or sync code is a magic number used to ensure synchronisation when parsing ogg files. Every page starts with the four byte sequence 'OggS'. This assists in resynchronising a parser in cases where data has been lost or is corrupted, and is a sanity check before commencing parsing the page structure.* [26]

The following sequence of data "`0x4F 0x67 0x67 0x53`" ("OggS" in ASCII) was therefore placed at the beginning of the file, replacing the original "XXXX" string. This proved effective in making the file valid and interpretable.

After restoring the file, a common music player (i.e. iTunes) was used to render the audio file and decipher the recorded message: "**I'm gonna meet you in London on Tuesday at ten. See you then. Bye!**"

Its curious to notice that after restoring the file, Droid was still unable to detect the object's format, while Unix *file* command would clearly identify it as a "Ogg data, Speex audio".

If one would have to process/restore in an automated fashion hundreds of files with the same issue as the file presented in this scenario, one would have to make sure that the following two conditions were in place: first, all the files must be of the same format, i.e. an Ogg file, and secondly the corrupted sequence of bytes should be in the same place as they were in the presented message file.

Detecting the format of a corrupted file, especially when the corruption takes place in the header section (where all the relevant information for detecting the file format is usually placed) is an extremely complex task. Some formats include information about its encoding (such as magic numbers) both at the beginning and at the end of the file. This makes file format identification easier in the event of file corruption.

In the particular case of the Ogg type, there was information about its format in more than one place within the file, which helped us to identify it. One could try to automatically determine if a given file is an Ogg file by running the following Unix command: `grep "OggS" <filename>.` However, this would not attest, at least in an irrefutable way, that the file format was in fact an Ogg, it would just provide a very naïve method of trying to *guess* if the digital object was in fact an Ogg. More complex format identification methods such as the one provided by the JHove [27] tool could eventually be used.

In the event of a successful format identification, one could pass on to the file recovery procedure. This task is surely more complex. Corruption of the file could occur any place within the file. Distinguishing the corrupted bytes from the correct data is actually a very difficult task. In some cases this could be considered impossible. Some applications might attempt to fix an audio file by forcing the portions of the file that were not corrupted to be playable, however, the corrupted parts could never be appropriately restored. Nevertheless, depending on the level of corruption, the message could eventually be grasped by a human being. Humans have incredible brains that are "ebla to rd prty mch anthng".

In what concerns the aptness of the Ogg format for transmitting the information of the time and place were both friends were supposed to meet, we would have to say that since they were communicating via instant messaging why didn't they agree on the *rendez vous* place by solely using the instant messaging tool itself, i.e. by exchanging a line of text? This way the receiver would not have to save the file, find an appropriate player, decode the message and interpret it. He or she could easily read the message right off the screen. The message was short enough for the receiver to memorize it.

Still, if the message to be transmitted had, in fact, to be an audio file (e.g. if the two friends were to exchange an audio song or if both friends were visually impaired) we would have to argue that the Ogg format comes with both pros and cons. Ogg is an open standard for digital multimedia, unrestricted by software patents and designed for efficient streaming and manipulation [26]. This means that finding a free player for a particular operating system is fairly easy[9].

Additionally, Ogg is a compressed file format. Compression is usually a good asset when exchanging information in a networked environment as it shortens the time necessary to transmit the digital information. On the other hand, a compressed file is composed by an additional interpretation layer. A human willing to interpret a compressed file faces an additional step before getting contact with the actual information carried within the object. First, the object must be decompressed, and then decoded into something intelligible to the human receiver (Figure 1).

Additionally, compressed files that have been corrupted are more difficult to fix than uncompressed ones. If a single bit of information in a compressed file gets scrambled, the whole file can become unreadable, whilst in an uncompressed format, if some bytes get accidentally changed, that doesn't necessarily mean that the information will be imprisoned in it. Usually, only the portions affected by corruption will become unreadable.

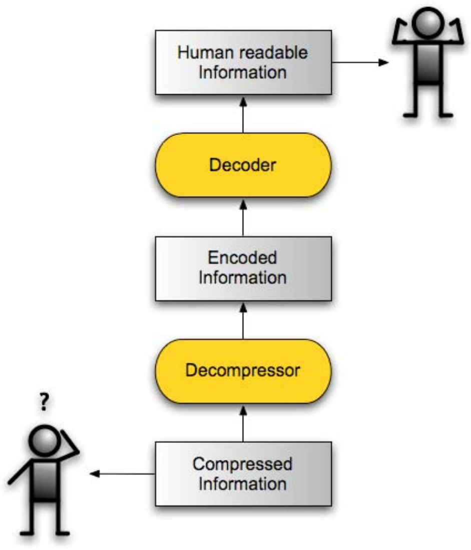


Figure 1 - Steps involved in the interpretation of a compressed digital object.

We have tested this hypothesis with two different files in distinct formats: the provided Ogg and a Wave version of it. We took the two files and deliberately scrambled four bytes of information (using an hexadecimal editor) at approximately half the length of the file. We then used the iTunes application to play both of files and assess the amount of damage we had imposed on to them.

In the case of the Ogg file, the audio file would play up to the scrambled bytes. Silence would come out of the speakers from that point on. In the case of the Wave

[9] The following Web site contains links for tens of applications capable of reading and producing Ogg files: http://vorbis.com/software/

file, a barely noticeable high pitched glitch could be heard half way through the playing the file. The file would continue to play correctly after the event.

In sum, even though the Ogg format comes some disadvantages we are led to believe that it would be a good choice in the given context. The compressed audio file is small enough to be transmitted online and the fact that the Ogg is open and patent free makes it a good candidate for an interchange format as the receiver would not have to go through a great deal of trouble to find an appropriate player. Furthermore, the audio stream inside the Ogg was encoded in Speex, a codec whose design goals have been to make it optimized for high quality speech at low bit rates. Additionally, Speex was designed to cope with lost packets although not corrupted ones (i.e. adequate for UDP transmissions) [28].

4.2 Proprietary File Format

Scenario description

A friend contacts you asking for your help to access two files which she has been given by her PhD supervisor. She urgently needs to access these files so that she can complete her dissertation. She has been unable to open the files as they appear to have been created using a proprietary format that is not recognised by her operating system. She tried to contact her supervisor for additional information but unfortunately her supervisor is attending a conference and is not contactable for a week. Your friend needs these files before then to meet her deadline - can you help her?

Task

1. Recover the content of the files and describe it. 2. Find out which software they were created with. 3. Restore the files. 4. Find a free viewer and open the files. Show that you are able to do this by providing a Screenshot.

Solution walkthrough

The proposed problem includes two files: “wchwom04.cbh” and “problem”. Our first action consisted in identifying the format of both files in order to determine which application was use in their creation.

For this purpose, we used Unix *file* command and Droid. However, both applications were unable to determine the format of the files. We then used the extension of the first file (i.e. cbh) and searched the Pronom database to check if it was a known extension. Unfortunately, Pronom did not recognise the extension. Our final attempt was to look for other format registries. We found the FILExt web site[10]. Searching its

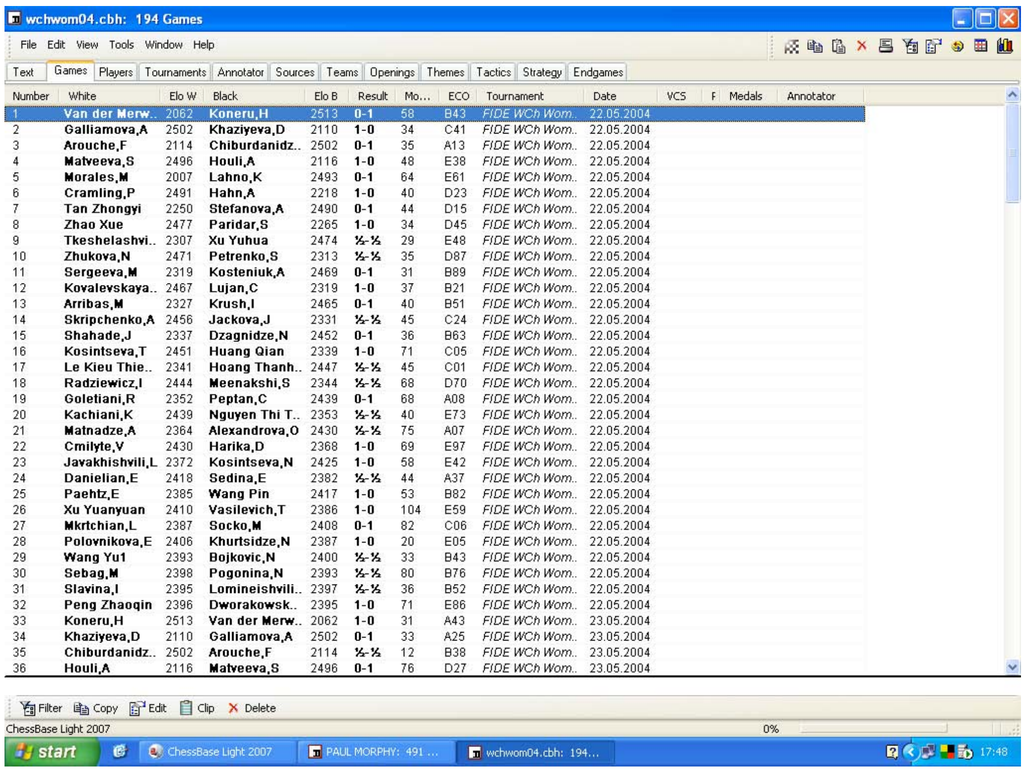
[10] http://www.filext.com

database enabled us to realise that the file was associated with a software application called ChessBase.

The next step consisted in downloading ChessBase Light 2007, a free version of the application to see if we could render the files. However, ChessBase could not open the files as they were. It displayed the error message "*filename* is not a database".

After browsing through the application menu we discovered that there was Backup database functionality. We have tested this option and it produced a file called "CB Light Database.cbv". We edited this file with a hex editor and realised that its structure was very similar to the one of the "wchwom04.cbh" file. Our next attempt to open the file was to change its extension to .cbv and see if the application could now open the file. To our surprise, it did!

Double-clicking the file icon lead the application to open and restore the backed up chess database. After the restoration process 11 new files were created in the directory were the backup was. One of those files was called "wschwom04.cbh". This file could now be adequately opened in ChessBase (Figure 2).

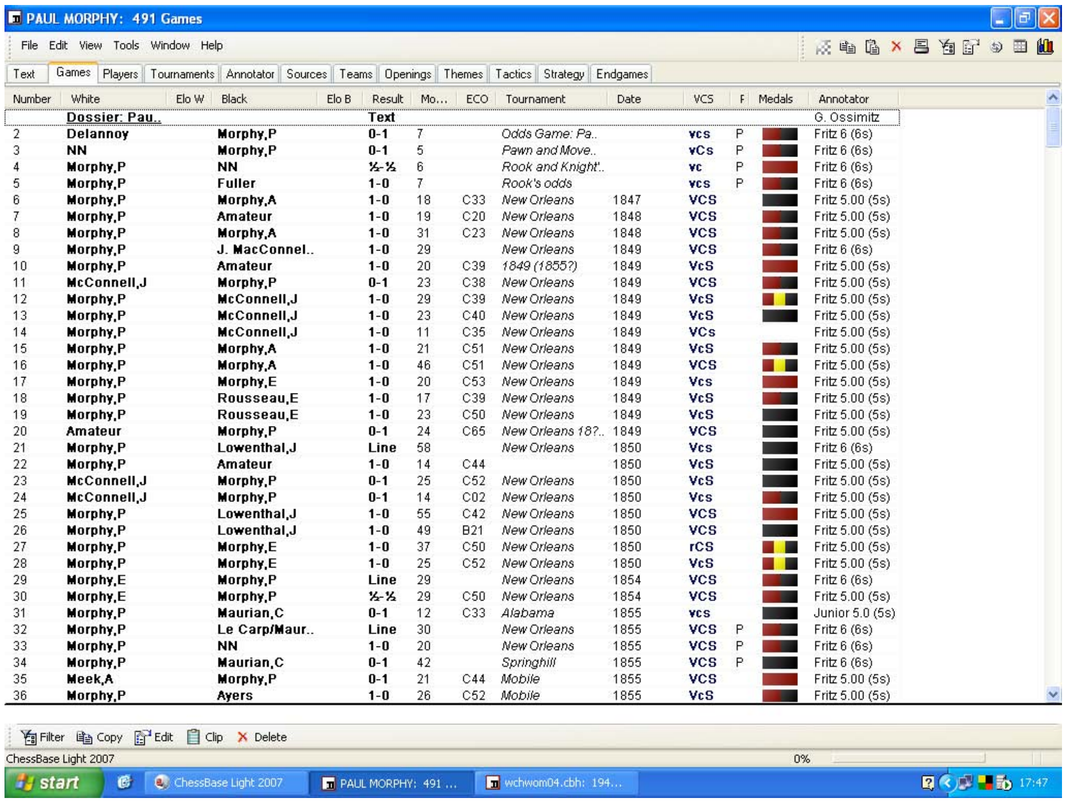

wchwom04.cbh: 194 Games
File Edit View Tools Window Help
Text Games Players Tournaments Annotator Sources Teams Openings Themes Tactics Strategy Endgames

Number	White	Elo W	Black	Elo B	Result	Mo...	ECO	Tournament	Date	VCS	F	Medals	Annotator
1	Van der Merw..	2062	Koneru,H	2513	0-1	58	B43	FIDE WCh Wom..	22.05.2004				
2	Galliamova,A	2502	Khaziyeva,D	2110	1-0	34	C41	FIDE WCh Wom..	22.05.2004				
3	Arouche,F	2114	Chiburdanidz..	2502	0-1	35	A13	FIDE WCh Wom..	22.05.2004				
4	Matveeva,S	2496	Houli,A	2116	1-0	48	E38	FIDE WCh Wom..	22.05.2004				
5	Morales,M	2007	Lahno,K	2493	0-1	64	E61	FIDE WCh Wom..	22.05.2004				
6	Cramling,P	2491	Hahn,A	2218	1-0	40	D23	FIDE WCh Wom..	22.05.2004				
7	Tan Zhongyi	2250	Stefanova,A	2490	0-1	44	D15	FIDE WCh Wom..	22.05.2004				
8	Zhao Xue	2477	Paridar,S	2265	1-0	34	D45	FIDE WCh Wom..	22.05.2004				
9	Tkeshelashvi..	2307	Xu Yuhua	2474	½-½	29	E48	FIDE WCh Wom..	22.05.2004				
10	Zhukova,N	2471	Petrenko,S	2313	½-½	35	D87	FIDE WCh Wom..	22.05.2004				
11	Sergeeva,M	2319	Kosteniuk,A	2469	0-1	31	B89	FIDE WCh Wom..	22.05.2004				
12	Kovalevskaya..	2467	Lujan,C	2319	1-0	37	B21	FIDE WCh Wom..	22.05.2004				
13	Arribas,M	2327	Krush,I	2465	0-1	40	B51	FIDE WCh Wom..	22.05.2004				
14	Skripchenko,A	2456	Jackova,J	2331	½-½	45	C24	FIDE WCh Wom..	22.05.2004				
15	Shahade,J	2337	Dzagnidze,N	2452	0-1	36	B63	FIDE WCh Wom..	22.05.2004				
16	Kosintseva,T	2451	Huang Qian	2339	1-0	71	C05	FIDE WCh Wom..	22.05.2004				
17	Le Kieu Thie..	2341	Hoang Thanh..	2447	½-½	45	C01	FIDE WCh Wom..	22.05.2004				
18	Radziewicz,I	2444	Meenakshi,S	2344	½-½	68	D70	FIDE WCh Wom..	22.05.2004				
19	Goletiani,R	2352	Peptan,C	2439	0-1	68	A08	FIDE WCh Wom..	22.05.2004				
20	Kachiani,K	2439	Nguyen Thi T..	2353	½-½	40	E73	FIDE WCh Wom..	22.05.2004				
21	Matnadze,A	2364	Alexandrova,O	2430	½-½	75	A07	FIDE WCh Wom..	22.05.2004				
22	Cmilyte,V	2430	Harika,D	2368	1-0	69	E97	FIDE WCh Wom..	22.05.2004				
23	Javakhishvili,L	2372	Kosintseva,N	2425	1-0	58	E42	FIDE WCh Wom..	22.05.2004				
24	Danielian,E	2418	Sedina,E	2382	½-½	44	A37	FIDE WCh Wom..	22.05.2004				
25	Paehtz,E	2385	Wang Pin	2417	1-0	53	B82	FIDE WCh Wom..	22.05.2004				
26	Xu Yuanyuan	2410	Vasilevich,T	2386	1-0	104	E59	FIDE WCh Wom..	22.05.2004				
27	Mkrtchian,L	2387	Socko,M	2408	0-1	82	C06	FIDE WCh Wom..	22.05.2004				
28	Polovnikova,E	2406	Khurtsidze,N	2387	1-0	20	E05	FIDE WCh Wom..	22.05.2004				
29	Wang Yu1	2393	Bojkovic,N	2400	½-½	33	B43	FIDE WCh Wom..	22.05.2004				
30	Sebag,M	2398	Pogonina,N	2393	½-½	80	B76	FIDE WCh Wom..	22.05.2004				
31	Slavina,I	2395	Lomineishvili..	2397	½-½	36	B52	FIDE WCh Wom..	22.05.2004				
32	Peng Zhaoqin	2396	Dworakowsk..	2395	1-0	71	E86	FIDE WCh Wom..	22.05.2004				
33	Koneru,H	2513	Van der Merw..	2062	1-0	31	A43	FIDE WCh Wom..	23.05.2004				
34	Khaziyeva,D	2110	Galliamova,A	2502	0-1	33	A25	FIDE WCh Wom..	23.05.2004				
35	Chiburdanidz..	2502	Arouche,F	2114	½-½	12	B38	FIDE WCh Wom..	23.05.2004				
36	Houli,A	2116	Matveeva,S	2496	0-1	76	D27	FIDE WCh Wom..	23.05.2004				

Figure 2 - Screen shot of the ChessBase application showing the contents of the "wchom04.cbh" file.

A similar procedure was tried for the "problem" file. We begun by adding a .cbv extension to the file, and see if it restored appropriately. However, during restoration the following error was displayed by the application: "c:\…\pm is not a database".

"pm" was in fact one of the files created during the restoration. We attempt to open the "pm" database by adding a ".cbh" extension to it. This allowed the application to appropriately load the "pm" database (Figure 3).


PAUL MORPHY: 491 Games

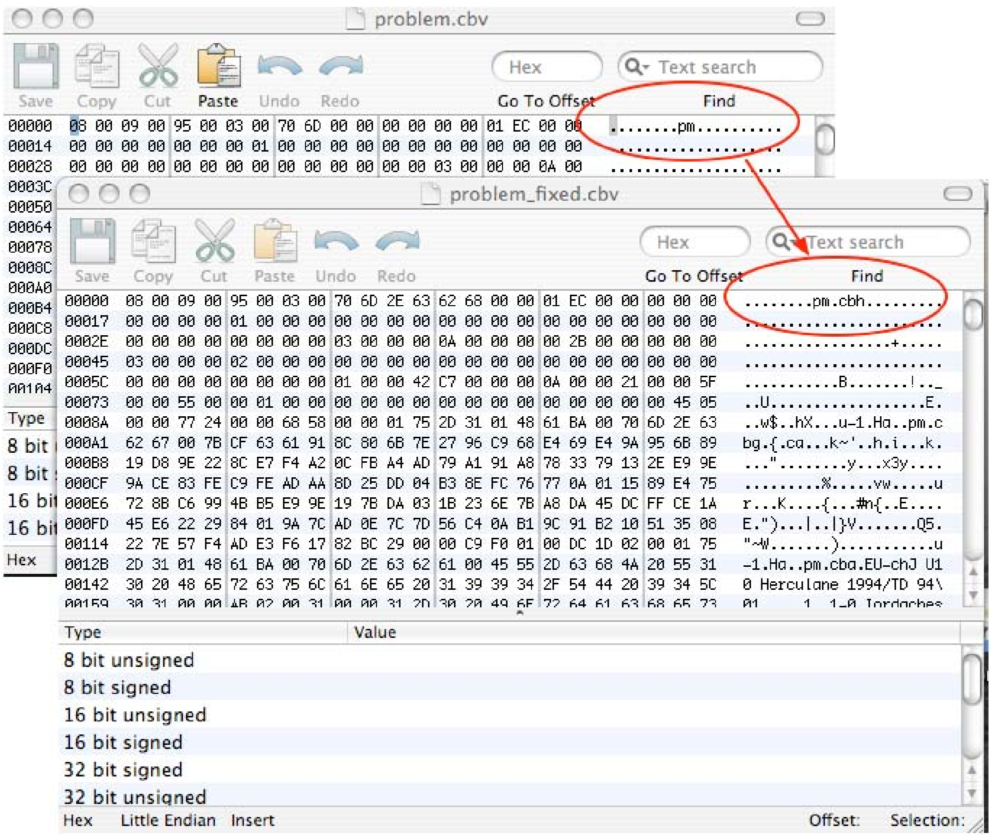
Number	White	Elo W	Black	Elo B	Result	Mo...	ECO	Tournament	Date	VCS	F	Medals	Annotator
	Dossier: Pau..				Text								G. Ossimitz
2	Delannoy		Morphy,P		0-1	7		Odds Game: Pa..		vcs	P	🟥⬛	Fritz 6 (6s)
3	NN		Morphy,P		0-1	5		Pawn and Move..		vCs	P	🟥⬛	Fritz 6 (6s)
4	Morphy,P		NN		½-½	6		Rook and Knight'..		vc	P	🟥	Fritz 6 (6s)
5	Morphy,P		Fuller		1-0	7		Rook's odds		vcs	P	🟥⬛	Fritz 6 (6s)
6	Morphy,P		Morphy,A		1-0	18	C33	New Orleans	1847	VCS		⬛	Fritz 5.00 (5s)
7	Morphy,P		Amateur		1-0	19	C20	New Orleans	1848	VCS		🟥⬛	Fritz 5.00 (5s)
8	Morphy,P		Morphy,A		1-0	31	C23	New Orleans	1848	VCS		🟥⬛	Fritz 5.00 (5s)
9	Morphy,P		J. MacConnel..		1-0	29		New Orleans	1849	VCS		🟥⬛	Fritz 6 (6s)
10	Morphy,P		Amateur		1-0	20	C39	1849 (1855?)	1849	VcS		🟥	Fritz 5.00 (5s)
11	McConnell,J		Morphy,P		0-1	23	C38	New Orleans	1849	VCS		🟥⬛	Fritz 5.00 (5s)
12	Morphy,P		McConnell,J		1-0	29	C39	New Orleans	1849	VcS		🟥🟨⬛	Fritz 5.00 (5s)
13	Morphy,P		McConnell,J		1-0	23	C40	New Orleans	1849	VcS		⬛	Fritz 5.00 (5s)
14	Morphy,P		McConnell,J		1-0	11	C35	New Orleans	1849	VCs			Fritz 5.00 (5s)
15	Morphy,P		Morphy,A		1-0	21	C51	New Orleans	1849	VcS		🟥⬛	Fritz 5.00 (5s)
16	Morphy,P		Morphy,A		1-0	46	C51	New Orleans	1849	VCS		🟥🟨⬛	Fritz 5.00 (5s)
17	Morphy,P		Morphy,E		1-0	20	C53	New Orleans	1849	Vcs		🟥	Fritz 5.00 (5s)
18	Morphy,P		Rousseau,E		1-0	17	C39	New Orleans	1849	VcS		🟥⬛	Fritz 5.00 (5s)
19	Morphy,P		Rousseau,E		1-0	23	C50	New Orleans	1849	VcS		⬛	Fritz 5.00 (5s)
20	Amateur		Morphy,P		0-1	24	C65	New Orleans 18?..	1849	VCS		⬛	Fritz 5.00 (5s)
21	Morphy,P		Lowenthal,J		Line	58		New Orleans	1850	Vcs		⬛	Fritz 6 (6s)
22	Morphy,P		Amateur		1-0	14	C44		1850	VcS		⬛	Fritz 5.00 (5s)
23	McConnell,J		Morphy,P		0-1	25	C52	New Orleans	1850	VcS		⬛	Fritz 5.00 (5s)
24	McConnell,J		Morphy,P		0-1	14	C02	New Orleans	1850	Vcs		🟥⬛	Fritz 5.00 (5s)
25	Morphy,P		Lowenthal,J		1-0	55	C42	New Orleans	1850	VCS		🟥	Fritz 5.00 (5s)
26	Morphy,P		Lowenthal,J		1-0	49	B21	New Orleans	1850	VCS		⬛	Fritz 5.00 (5s)
27	Morphy,P		Morphy,E		1-0	37	C50	New Orleans	1850	rCS		🟥🟨⬛	Fritz 5.00 (5s)
28	Morphy,P		Morphy,E		1-0	25	C52	New Orleans	1850	VcS		🟥🟨⬛	Fritz 5.00 (5s)
29	Morphy,E		Morphy,P		Line	29		New Orleans	1854	VCS		🟥⬛	Fritz 6 (6s)
30	Morphy,E		Morphy,P		½-½	29	C50	New Orleans	1854	VCS		🟥⬛	Fritz 5.00 (5s)
31	Morphy,P		Maurian,C		0-1	12	C33	Alabama	1855	vcs		⬛	Junior 5.0 (5s)
32	Morphy,P		Le Carp/Maur..		Line	30		New Orleans	1855	VCS	P	🟥⬛	Fritz 6 (6s)
33	Morphy,P		NN		1-0	20		New Orleans	1855	VCS	P	🟥⬛	Fritz 6 (6s)
34	Morphy,P		Maurian,C		0-1	42		Springhill	1855	VCS	P	⬛	Fritz 6 (6s)
35	Meek,A		Morphy,P		0-1	21	C44	Mobile	1855	VCS		🟥	Fritz 5.00 (5s)
36	Morphy,P		Ayers		1-0	26	C52	Mobile	1855	VcS		🟥⬛	Fritz 5.00 (5s)

File Edit View Tools Window Help
Text Games Players Tournaments Annotator Sources Teams Openings Themes Tactics Strategy Endgames
Filter Copy Edit Clip Delete
ChessBase Light 2007
0%
start
ChessBase Light 2007
PAUL MORPHY: 491 ...
wchwom04.cbh: 194...
17:47

Figure 3 - Screenshot of the ChessBase application showing the contents of the "problem" file.

However, the last procedure seemed strange. Why didn't the restore functionality work correctly? Something was obviously wrong with the "problem.cbv" file. Using a hex editor we compared "problem.cbv" with "wschwom04.cbv".

By inspecting the file we realised that the name of the files created during the restoration procedure were clearly visible within the file. We then decided to include the correct extension close to the "pm" statement at the beginning of the "problem.cbv" file (Figure 4). We tested the restoration procedure again and everything worked correctly. Twelve files were created in the same directory as the backup file.



problem.cbv

```
00000  08 00 09 00 95 00 03 00 70 6D 00 00 00 00 00 00 01 EC 00 00   ........pm..........
00014  00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00   ....................
00028  00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 00 00 00 0A 00   ....................
```

problem_fixed.cbv

```
00000  08 00 09 00 95 00 03 00 70 6D 2E 63 62 68 00 00 01 EC 00 00 00 00 00   ........pm.cbh.........
00017  00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .......................
0002E  00 00 00 00 00 00 00 00 03 00 00 00 0A 00 00 00 00 2B 00 00 00 00 00   .................+.....
00045  03 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .......................
0005C  00 00 00 00 00 00 00 00 01 00 00 42 C7 00 00 00 0A 00 00 21 00 00 5F   ...........B.......!.._
00073  00 00 55 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 45 05   ..U..................E.
0008A  00 00 77 24 00 00 68 58 00 00 01 75 2D 31 01 48 61 BA 00 70 6D 2E 63   ..w$..hX...u-1.Ha..pm.c
000A1  62 67 00 7B CF 63 61 91 8C 80 6B 7E 27 96 C9 68 E4 69 E4 9A 95 6B 89   bg.{.ca...k~'..h.i...k.
000B8  19 D8 9E 22 8C E7 F4 A2 0C FB A4 AD 79 A1 91 A8 78 33 79 13 2E E9 9E   ..."........y...x3y....
000CF  9A CE 83 FE C9 FE AD AA 8D 25 DD 04 B3 8E FC 76 77 0A 01 15 89 E4 75   .........%.....vw.....u
000E6  72 8B C6 99 4B B5 E9 9E 19 7B DA 03 1B 23 6E 7B A8 DA 45 DC FF CE 1A   r...K....{...#n{..E....
000FD  45 E6 22 29 84 01 9A 7C AD 0E 7C 7D 56 C4 0A B1 9C 91 B2 10 51 35 08   E.")...|..|}V.......Q5.
00114  22 7E 57 F4 AD E3 F6 17 82 BC 29 00 00 C9 F0 01 00 DC 1D 02 00 01 75   "~W.......)...........u
0012B  2D 31 01 48 61 BA 00 70 6D 2E 63 62 61 00 45 55 2D 63 68 4A 20 55 31   -1.Ha..pm.cba.EU-chJ U1
00142  30 20 48 65 72 63 75 6C 61 6E 65 20 31 39 39 34 2F 54 44 20 39 34 5C   0 Herculane 1994/TD 94\
00159  30 31 00 00 AB 02 00 31 00 00 31 2D 30 20 49 6F 72 64 61 63 68 65 73   01    1  1-0 Iordaches
```

Type	Value
8 bit unsigned	
8 bit signed	
16 bit unsigned	
16 bit signed	
32 bit signed	
32 bit unsigned	

Figure 4 - Change done to the "problem.cbv" file.

4.3 Client Server Database Application

Scenario description

You work for a small company that manufactures grand pianos. The company is currently facing financial difficulties due to low sales. A customer, named Mr. Miller, calls you asking for a quote for a large order. As a previous customer of your company, Mr Miller assumes that you already have all his contact details on record, and therefore he hangs up without providing any.

When you check the company's records however, you cannot find these details. He bought his piano during the month that your company trialled a new database application for managing customer relations. However, you still have a file from this trial and an accompanying note stating that the application has to be 'set up' and contains the 'current database'. You must recover Mr. Miller's contact details in order to make this important sale - can you obtain them?

Task

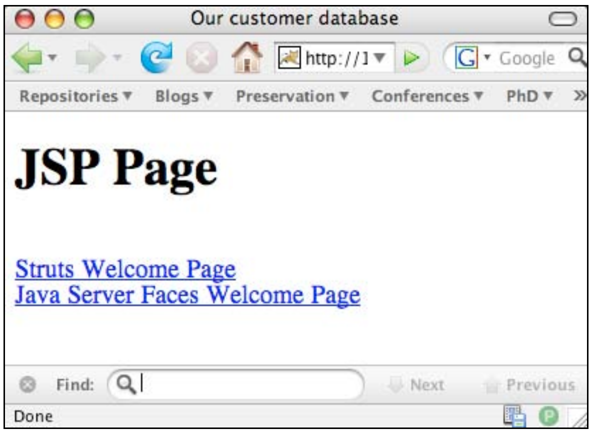
1. Identify the file format. 2. Set up a feasible environment to handle the file. 3. Perform any steps necessary to properly access and set up the provided object. 4. Identify Mr. Miller's address. 5. Migrate the data records to a relational database system (include the sources of your migration system in your submission – in zipped form).

Solution walkthrough

The third scenario included a file called "dpc.ace". Using Unix *file* application, we could identify the file format as being a Zip archive file[11]. Following that, we used the Unix *unzip* command to decompress the object and gain access to its contents.

For someone who has already worked with the Servlet/JSP technology is quite easy to recognise the folder structure that was created during the decompression step. The contents of the "dpc.ace" file consisted of a Servlet application.

We have installed Apache Tomcat 5[12], an open-source application server, in order to run the servlet application. By copying the entire content of "dpc.ace" to `TOMCAT_HOME/webapps/dpc` and pointing our browser to `http://localhost:8080/dpc` we have managed to get a glimpse of the web application (Figure 5).



JSP Page

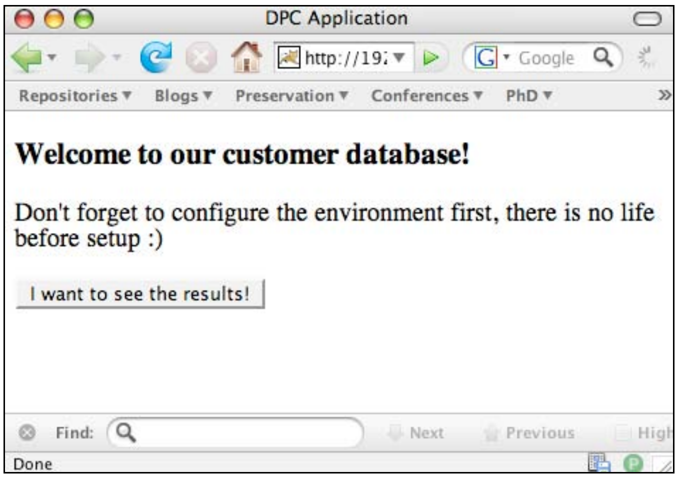
Struts Welcome Page
Java Server Faces Welcome Page

Figure 5 - Screenshot of the opening Web page produced by the Servlet.

Following the first link on the opening web page, we found a second screen, which displayed the following message: "Welcome to our customer database! Don't forget to configure the environment first, there is no life before setup :)" (Figure 6).

[11] `file`'s output was "Zip archive data, at least v2.0 to extract".

[12] http://tomcat.apache.org



Welcome to our customer database!

Don't forget to configure the environment first, there is no life before setup :)

I want to see the results!

Figure 6 - Screenshot of the Struts Welcome page.

Examining the files that comprised the Servlet application, we could detect a file called "setup.jar". Executing the file, by running the command "java –jar setup.jar", would render the following exception "*Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0" at setup.Setup.main (Setup.java:59)*".

This usually happens when we do not provide all the necessary parameters to the application. So we tried to append a parameter to the previously executed command (e.g. "java –jar setup.jar test". This forced the application to print out a syntax instructions message:

```
"Usage: setup.Foo writeyap <path>"
```

We then executed the following command-line:

```
"java -jar setup.jar writeyap $TOMCAT_HOME/WEB-INF/dpc/dpc.yap"
```

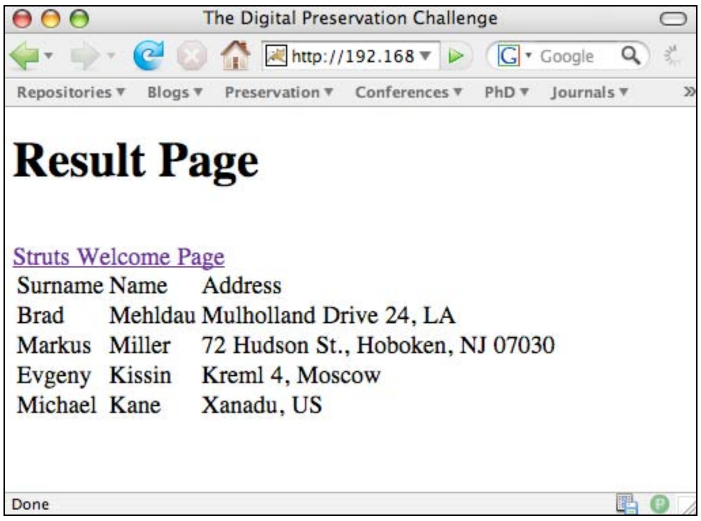
This action produced the message "*Don't forget to put all files in the correct(!) locations to be able to run the web application!*" and created a file called "db4o.config".

Rerunning the Web application generated an exception that suggested that the application was not able to find the configuration file. After some experimentation in moving the configuration file around the application folder structure, we finally realised where it should be present: $TOMCAT_HOME/webapps/dpc/WEB-INF/classes/at/tuwien/dpc.

Moving the configuration file to the appropriate folder rendered an additional exception: "*java.lang.NoClassDefFoundError: com/db4o/Db4o*". Searching the web for "Db4o", we found that it was an open-source object database[13]. At the Db4o Web

[13] db4o is the open source object database that enables Java and .NET developers to slash development time and costs and achieve unprecedented levels of performance. The unique design of db4o's native object database engine makes it the ideal choice to be embedded in equipment and devices, in packaged software running on mobile or desktop platforms, or in real-time control systems - in short: in all Java and .NET environments, where no database administrator (DBA) is present.

site we could find some developer tools[14], including the library that was missing in our application. After downloading the development tools, we copied the db4o-6.1-java5.jar[15] library to the appropriate folder in our Web application (i.e. $TOMCAT_HOME/webapps/dpc/WEB-INF/lib) and managed to correctly present the Result page where Mr Miller's address could be visualised: "**72 Hudson St., Hoboken, NJ 07030**" (Figure 7).



Result Page

Struts Welcome Page

Surname	Name	Address
Brad	Mehldau	Mulholland Drive 24, LA
Markus	Miller	72 Hudson St., Hoboken, NJ 07030
Evgeny	Kissin	Kreml 4, Moscow
Michael	Kane	Xanadu, US

Figure 7 - Screenshot of the database listing.

In order to export the Db4o database to a relational database we have adapted the "result.jsp" JSP to reproduce the entire contents of the object database as SQL statements that could be used to regenerate the database in a standard Relational Database Management System (RDBMS) (e.g. MySQL). The source code of the adapted java server page is depicted next:

```
<%@page contentType="text/plain"%>
<%@page pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

CREATE TABLE customers(givenname varchar(100), surname
varchar(100), address varchar(500));

<c:forEach var="b" items="${sessionScope.GoForm.list}">
INSERT INTO customers(givenname, surname, address) VALUES
('${b.vorname}', '${b.nachname}', '${b.address}');
</c:forEach>
```

Figure 8 - Source code of the exportToSQL.jsp.

[14] http://www.db4o.com/community/

[15] We opt for this file among all the other provided with the developer tools, because we were using JDK 1.5.

To recreate the original database in a relational database, save the resulting Web page to a file called, for example, "customers.sql" and run the following command on a Linux machine with a previously installed version of MySQL:

```
# mysql -u username -p dbname < customers.sql
```

After executing the previous command, if the following SQL statement is executed within the MySQL shell environment, the following output would be produced:

```
mysql> select * from customers;

+-----------+---------+----------------------------------+
| givenname | surname | address                          |
+-----------+---------+----------------------------------+
| Brad      | Mehldau | Mulholland Drive 24, LA          |
| Markus    | Miller  | 72 Hudson St., Hoboken, NJ 07030 |
| Evgeny    | Kissin  | Kreml 4, Moscow                  |
| Michael   | Kane    | Xanadu, US                       |
+-----------+---------+----------------------------------+
4 rows in set (0.00 sec)
```

Additionally, we have developed a more portable and generic solution to tackle the problem of migrating Db4o databases. An accompanying java application called "db4o2xml" (jar and source are provided with this report) is provided that is able to traverse any Db4o database and save its contents as XML.

The structure of the produced document is simple enough so that a stylesheet may easily be developed to transform the XML into SQL statements or any other textual format that one might require (e.g. comma-separated values).

Running the "db4o2xml" application on the provided "dpc.yap" database produces the following output:

Table 1 - Output produced by running the Db4o to XML application.

```
<?xml version="1.0" encoding="UTF-8"?>
<db4o>
    <at.tuwien.dpc.Bean>
        <attribute name="vorname">Brad</attribute>
        <attribute name="nachname">Mehldau</attribute>
        <attribute name="address">Mulholland Drive 24, LA</attribute>
        <attribute name="kinder">0</attribute>
    </at.tuwien.dpc.Bean>
    <at.tuwien.dpc.Bean>
        <attribute name="vorname">Markus</attribute>
        <attribute name="nachname">Miller</attribute>
        <attribute name="address">72 Hudson St., Hoboken, NJ
07030</attribute>
        <attribute name="kinder">0</attribute>
    </at.tuwien.dpc.Bean>
    <at.tuwien.dpc.Bean>
        <attribute name="vorname">Evgeny</attribute>
        <attribute name="nachname">Kissin</attribute>
        <attribute name="address">Kreml 4, Moscow</attribute>
        <attribute name="kinder">0</attribute>
    </at.tuwien.dpc.Bean>
    <at.tuwien.dpc.Bean>
        <attribute name="vorname">Michael</attribute>
        <attribute name="nachname">Kane</attribute>
        <attribute name="address">Xanadu, US</attribute>
        <attribute name="kinder">0</attribute>
    </at.tuwien.dpc.Bean>
</db4o>
```

4.4 Legacy Emulator

Scenario description

You are a first year Computer Science student. For one assignment, you are set the task of recreating a computer game from your childhood for use on mobile platforms. You must incorporate as many of the game's original features as possible within your project including audio samples. Although you cannot remember the exact name of the particular game you have chosen to recreate, you have found a file relating to it on your hard drive. Can you find the application to run the game and retrieve the sound file?

Task

1. Identify the file format. 2. Name the game the file was saved from. 3. Run the emulator and game. 4. Record the sound that is played and save it in an open format that can be embedded into a presentation.

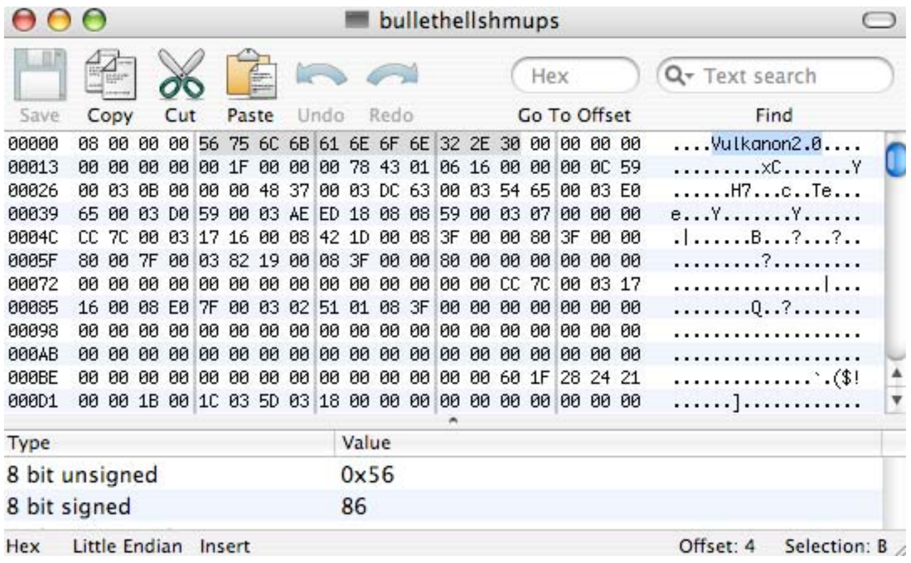
Solution walkthrough

To solve this scenario, we begun by using the Unix *file* command to identify the format of the included digital object. *file* identified the object as being a "rzip compressed data - version 2". We used the Linux command rzip to decompress the object as follows:

```
$ rzip -d bullethellshmups.rz
```

This created a new file called "bullethellshmups". Running the *file* command again would reveal that the recently created file was a "gzip compressed data" file. After decompressing the new file with the Linux command *gunzip* we obtained a file called "bullethellshmups". Now *file* could not identify its format.

We then used a hexadecimal editor to see if the entrails of the file would reveal any clues on application that was used to generate it. Fortunately, it was quite clear that an application called Vulcanon 2.0 was somehow related to the object (Figure 9).



```
00000  08 00 00 00 56 75 6C 6B 61 6E 6F 6E 32 2E 30 00 00 00 00   ....Vulkanon2.0....
00013  00 00 00 00 00 1F 00 00 00 78 43 01 06 16 00 00 00 0C 59   .........xC.......Y
00026  00 03 0B 00 00 00 48 37 00 03 DC 63 00 03 54 65 00 03 E0   ......H7...c..Te...
00039  65 00 03 D0 59 00 03 AE ED 18 08 08 59 00 03 07 00 00 00   e...Y.......Y......
0004C  CC 7C 00 03 17 16 00 08 42 1D 00 08 3F 00 00 80 3F 00 00   .|......B...?...?..
0005F  80 00 7F 00 03 82 19 00 08 3F 00 00 80 00 00 00 00 00 00   .........?.........
00072  00 00 00 00 00 00 00 00 00 00 00 00 00 00 CC 7C 00 03 17   ...............|...
00085  16 00 08 E0 7F 00 03 02 51 01 08 3F 00 00 00 00 00 00 00   ........Q..?.......
00098  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...................
000AB  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...................
000BE  00 00 00 00 00 00 00 00 00 00 00 00 00 00 60 1F 28 24 21   ..............`.($!
000D1  00 00 1B 00 1C 03 5D 03 18 00 00 00 00 00 00 00 00 00 00   ......]............
```

Type	Value
8 bit unsigned	0x56
8 bit signed	86

```
Hex    Little Endian    Insert                Offset: 4    Selection: B
```

Figure 9 - Hexadecimal view of the entrails of the mistery file.

Searching the Web revealed that Vulcanon 2.0 was a game for the Nintendo's Game Boy Advance[16]. A game ROM was available for download at the Web site as well as a Game Boy Advance emulator called "Visual Boy Advance 1.8.0 Beta 3.

We run the game using the emulator and recorded (Figure 10) the initial screen music using the Windows Sound Recorder.

[16] http://gba.pqrs.org/~tekezo/gba/vulkanon/

Figure 10 - Screenshots of Vulcanon 2.0.

The resulting audio file was encoded in the Ogg format and is provided together with this report (see accompanying files on page 39).

4.5 Legacy File Preservation Strategy

Scenario description

Your company is going through a certification process and requires access to business reports for the last ten years. You have discovered that many of these legacy files are partially incompatible with the company's modern applications and that there are many other files like these in the company's archive. You have been provided with representative samples of the files in question and have eventually been able to recover all of the information from the legacy files. However, it has taken a lot of time and effort to access these files and your boss is keen to avoid this kind of scenario in the future. She has asked you to design an appropriate preservation strategy that will facilitate access to such records. How will you do it?

Task

1. Display the given files in their original format, or the closest available format to this. 2. Propose a suitable preservation strategy. 3. Implement a preservation strategy capable of mass handling of files of this type. Your strategy should, as a minimum, be able to handle all the example files. 4. Provide a thorough description of your ideas about preservation strategies and an in-depth explanation of the file format.

Solution walkthrough

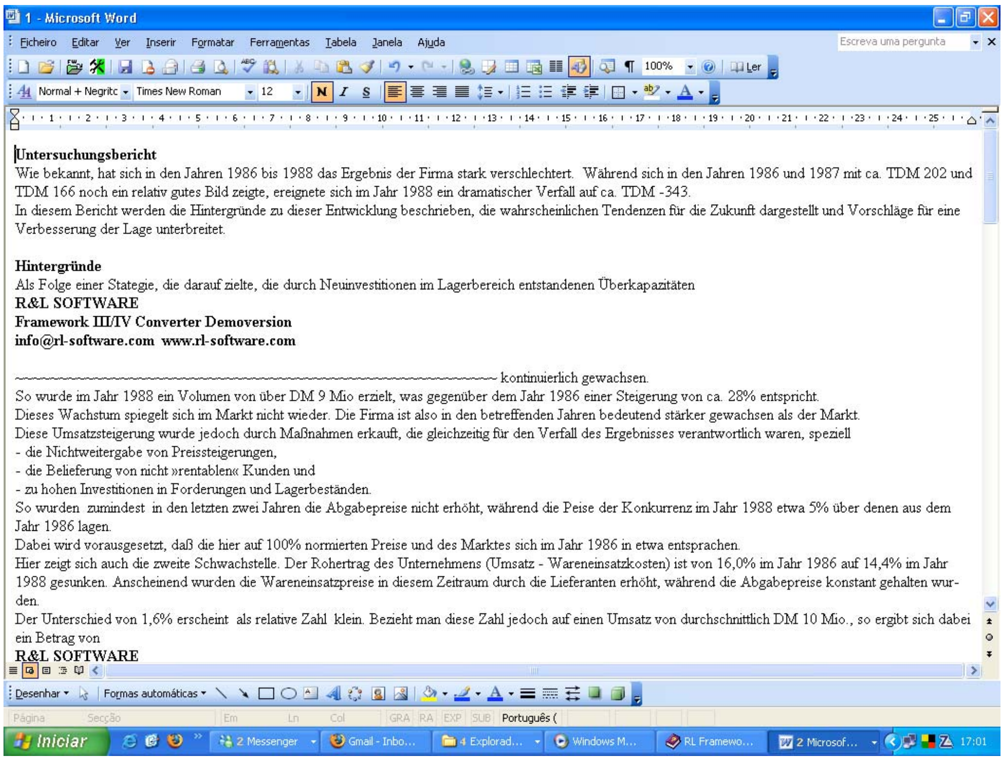
To solve this scenario we begun by querying PRONOM[17] and the FILExt[18] Web site to check if the extension ".fw3" was somehow a known as a common file format extension. It turns out that both Web sites identified the extension as being a Framework III file.

Searching the Web for that particular application, we realised that it was one of the earliest office suites and that it was still being maintained by a company called Selections & Functions, Inc.[19]. Unfortunately, a free version of the software could not be found at the company Web site.

We continued to look for an application that would be able to render the contents of the .fw3 files. We found two plausible applications:

- FW3 to TXT converter[20] – is DOS based converter application written in 1999 by an anonymous programmer. However, the converter does not always produce good results as the contents of the fw3 files are not always text based. In 2004, that very same programmer created a version 2.0 of the converter and made it compatible with Win32. It also added the capacity to handle documents in different codepages. We tested this application and the results were still very poor. The text produced by the application was mostly gibberish.
- Framework IV or Framework III to MS Word converter[21] – is an add-on for Microsoft Office Word developed by *R&L Software GmbH* which enables the user to import FW3 files to Microsoft Word. The files can then be saved in any Word compatible format (e.g. Word format, RTF, ASCII text, etc.). We have installed the trial version of this add-on, freely available at the company's Web site, and opened the files provided with this scenario. The results seemed acceptable (text and style seemed to be imported), however the software deliberately replaced some of the text paragraphs due to the fact that it was a trial version (Figure 11).

[17] http://www.nationalarchives.gov.uk/PRONOM
[18] http://filext.com/
[19] http://www.framework.com/
[20] http://visualcpp.net/index.php?qID=56
[21] http://www.rl-software.com/converter/framework-to-word-e.htm



Untersuchungsbericht
Wie bekannt, hat sich in den Jahren 1986 bis 1988 das Ergebnis der Firma stark verschlechtert. Während sich in den Jahren 1986 und 1987 mit ca. TDM 202 und TDM 166 noch ein relativ gutes Bild zeigte, ereignete sich im Jahr 1988 ein dramatischer Verfall auf ca. TDM -343.
In diesem Bericht werden die Hintergründe zu dieser Entwicklung beschrieben, die wahrscheinlichen Tendenzen für die Zukunft dargestellt und Vorschläge für eine Verbesserung der Lage unterbreitet.

Hintergründe
Als Folge einer Stategie, die darauf zielte, die durch Neuinvestitionen im Lagerbereich entstandenen Überkapazitäten
R&L SOFTWARE
Framework III/IV Converter Demoversion
info@rl-software.com www.rl-software.com

~~~~~~ kontinuierlich gewachsen.
So wurde im Jahr 1988 ein Volumen von über DM 9 Mio erzielt, was gegenüber dem Jahr 1986 einer Steigerung von ca. 28% entspricht.
Dieses Wachstum spiegelt sich im Markt nicht wieder. Die Firma ist also in den betreffenden Jahren bedeutend stärker gewachsen als der Markt.
Diese Umsatzsteigerung wurde jedoch durch Maßnahmen erkauft, die gleichzeitig für den Verfall des Ergebnisses verantwortlich waren, speziell
- die Nichtweitergabe von Preissteigerungen,
- die Belieferung von nicht »rentablen« Kunden und
- zu hohen Investitionen in Forderungen und Lagerbeständen.
So wurden zumindest in den letzten zwei Jahren die Abgabepreise nicht erhöht, während die Peise der Konkurrenz im Jahr 1988 etwa 5% über denen aus dem Jahr 1986 lagen.
Dabei wird vorausgesetzt, daß die hier auf 100% normierten Preise und des Marktes sich im Jahr 1986 in etwa entsprachen.
Hier zeigt sich auch die zweite Schwachstelle. Der Rohertrag des Unternehmens (Umsatz - Wareneinsatzkosten) ist von 16,0% im Jahr 1986 auf 14,4% im Jahr 1988 gesunken. Anscheinend wurden die Wareneinsatzpreise in diesem Zeitraum durch die Lieferanten erhöht, während die Abgabepreise konstant gehalten wurden.
Der Unterschied von 1,6% erscheint als relative Zahl klein. Bezieht man diese Zahl jedoch auf einen Umsatz von durchschnittlich DM 10 Mio., so ergibt sich dabei ein Betrag von
R&L SOFTWARE

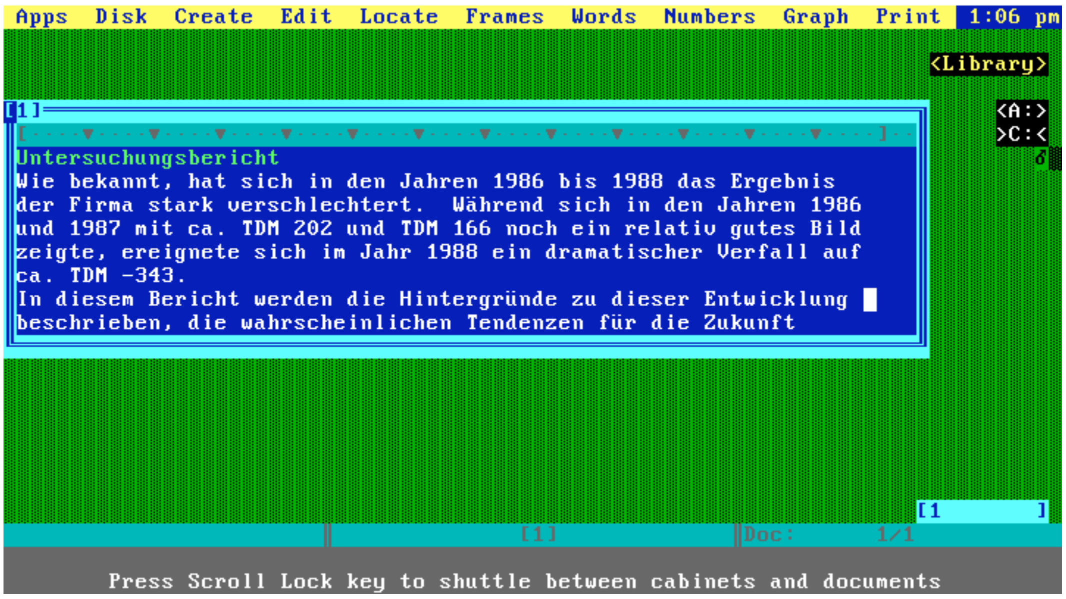
Figure 11 - Screenshot of "1.fw3" file opened on Microsoft Word.

Yet, we were not yet satisfied. We wanted to try the original Framework office suite to make sure that the Word add-on was working correctly, i.e. that the significant properties of the original object were being adequately tranferred. However, in order to identify the significant properties of the objects produced in Framework III we needed to fully understand the possibilities and functionalities offered by the application.

Obtaining a working copy of Framework III was everything but easy. We have established contact with José Maria Fernandes Almeida, the main responsible for the Virtual Museum of Informatics[22], an initiative of the Department of Information Systems of the University of Minho, to see if among all the hardware and software that was being kept at the museum one could find a working copy of Framework III. Unfortunately, the museum did not maintained that particular application. Our next step was to contact the IT manager of our department to see if he, by any change, would have a copy of the application. As it so happens, he did!

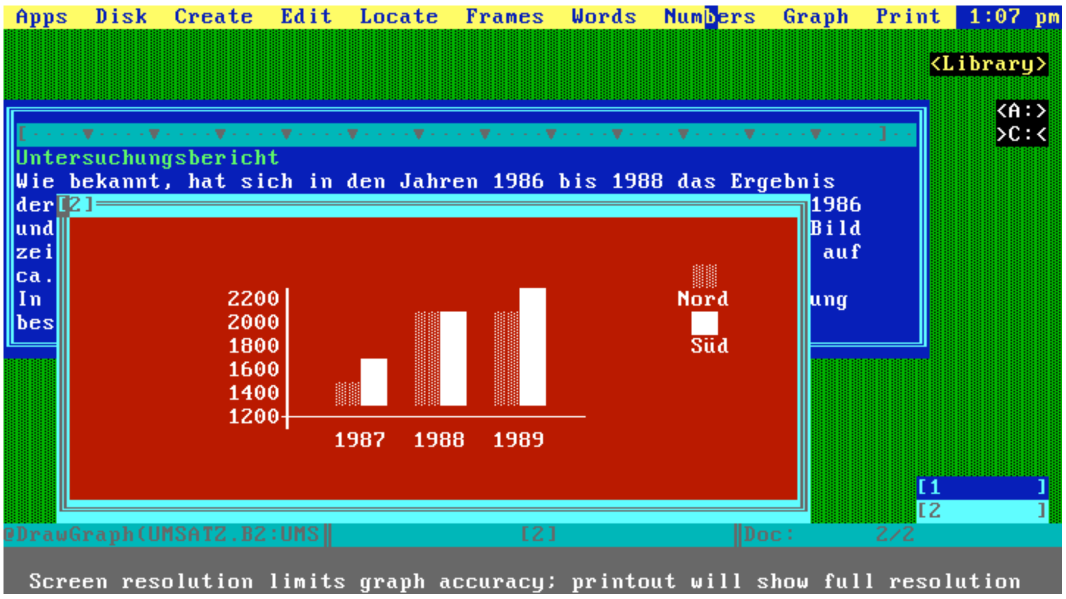
We installed Framework III using a DOS emulator called DOSBox under Mac OS X and opened all the files with the original software (Figure 12 to Figure 19).

[22] http://www.dsi.uminho.pt/museuv
~~~~~~



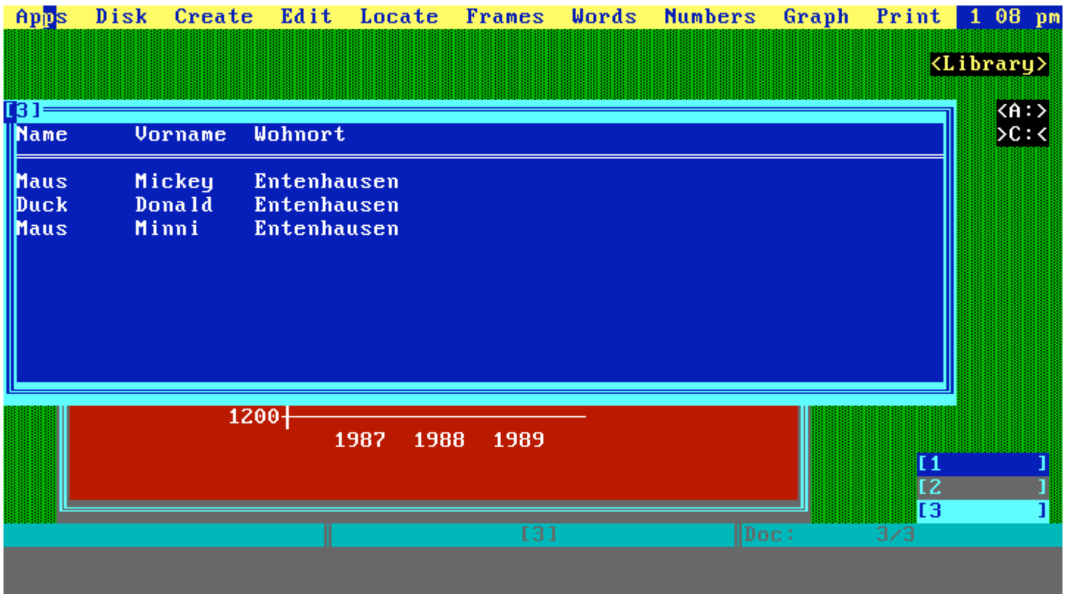
```
Apps  Disk  Create  Edit  Locate  Frames  Words  Numbers  Graph  Print  1:06 pm
                                                                     <Library>
[1]
Untersuchungsbericht
Wie bekannt, hat sich in den Jahren 1986 bis 1988 das Ergebnis
der Firma stark verschlechtert.  Während sich in den Jahren 1986
und 1987 mit ca. TDM 202 und TDM 166 noch ein relativ gutes Bild
zeigte, ereignete sich im Jahr 1988 ein dramatischer Verfall auf
ca. TDM -343.
In diesem Bericht werden die Hintergründe zu dieser Entwicklung
beschrieben, die wahrscheinlichen Tendenzen für die Zukunft

                        [1]                  Doc:       1/1
     Press Scroll Lock key to shuttle between cabinets and documents
```

Figure 12 – Screenshot of “1.fw3” file opened on Framework III original software.



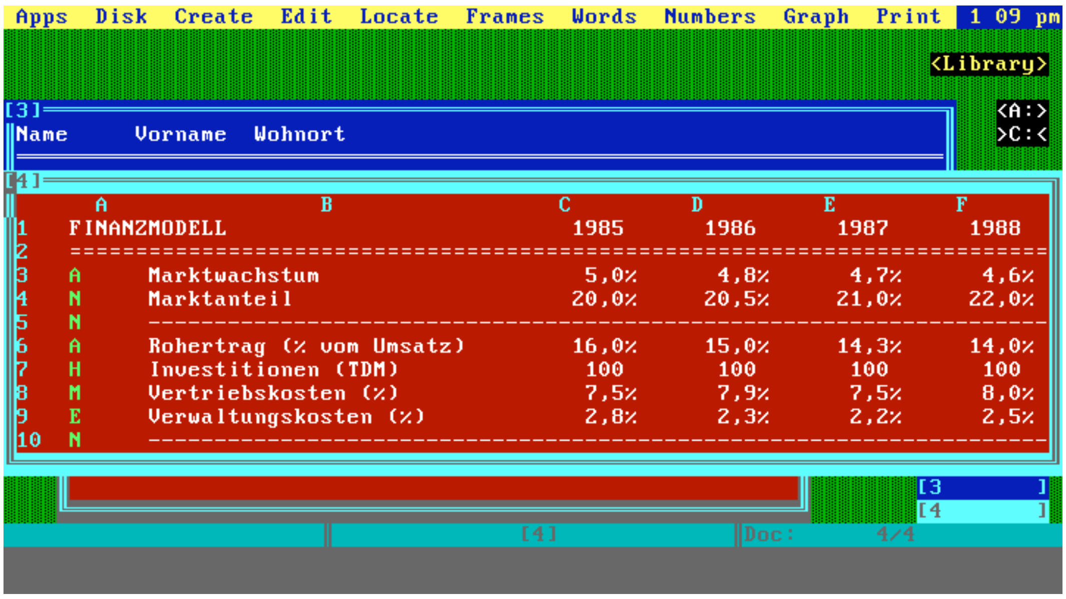
```
Apps  Disk  Create  Edit  Locate  Frames  Words  Numbers  Graph  Print  1:07 pm
                                                                     <Library>
Untersuchungsbericht
Wie bekannt, hat sich in den Jahren 1986 bis 1988 das Ergebnis
der[2]                                                        1986
und                                                           Bild
zei                                                            auf
ca.
In                                                            ung
bes

@DrawGraph(UMSATZ.B2:UMS   [2]                  Doc:       2/2
   Screen resolution limits graph accuracy; printout will show full resolution
```

Figure 13 - Screenshot of “2.fw3” file opened on Framework III original software.


Apps Disk Create Edit Locate Frames Words Numbers Graph Print 1 08 pm
<Library>
<A:>
>C:<
[1]
[2]
[3]
[3]
Doc: 3/3

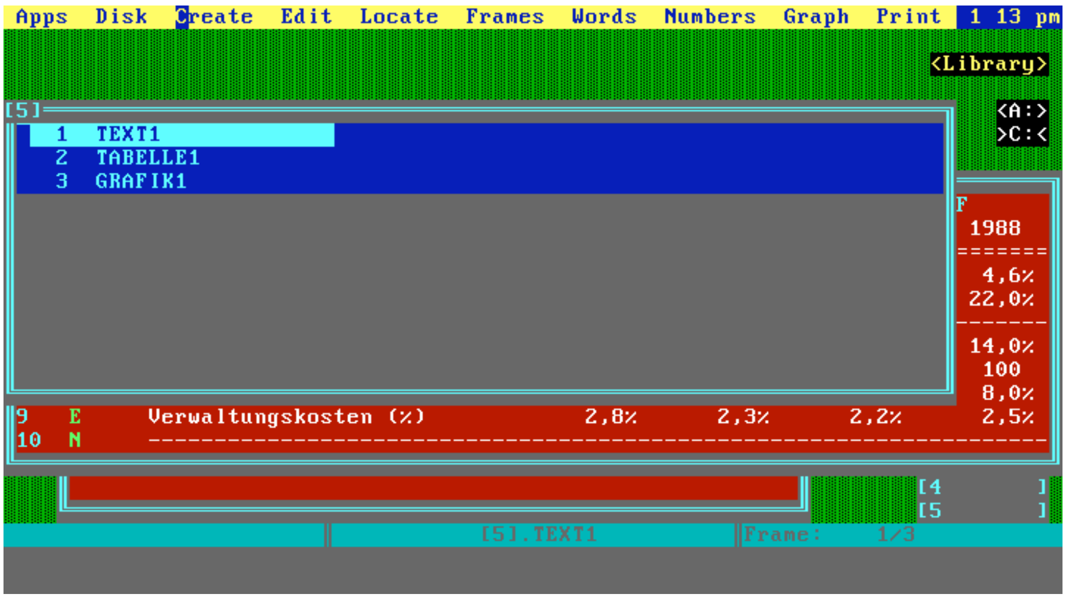
```
[3]
Name       Vorname   Wohnort

Maus       Mickey    Entenhausen
Duck       Donald    Entenhausen
Maus       Minni     Entenhausen

        1200
                 1987   1988   1989
```

Figure 14 - Screenshot of "3.fw3" file opened on Framework III original software.


Apps Disk Create Edit Locate Frames Words Numbers Graph Print 1 09 pm
<Library>
<A:>
>C:<
[3]
[4]
[4]
Doc: 4/4

```
[3]
Name       Vorname   Wohnort
[4]
       A                 B                    C          D          E          F
1   FINANZMODELL                            1985       1986       1987       1988
2   ============================================================================
3   A      Marktwachstum                    5,0%       4,8%       4,7%       4,6%
4   N      Marktanteil                     20,0%      20,5%      21,0%      22,0%
5   N      ---------------------------------------------------------------------
6   A      Rohertrag (% vom Umsatz)        16,0%      15,0%      14,3%      14,0%
7   H      Investitionen (TDM)              100        100        100        100
8   M      Vertriebskosten (%)              7,5%       7,9%       7,5%       8,0%
9   E      Verwaltungskosten (%)            2,8%       2,3%       2,2%       2,5%
10  N      ---------------------------------------------------------------------
```

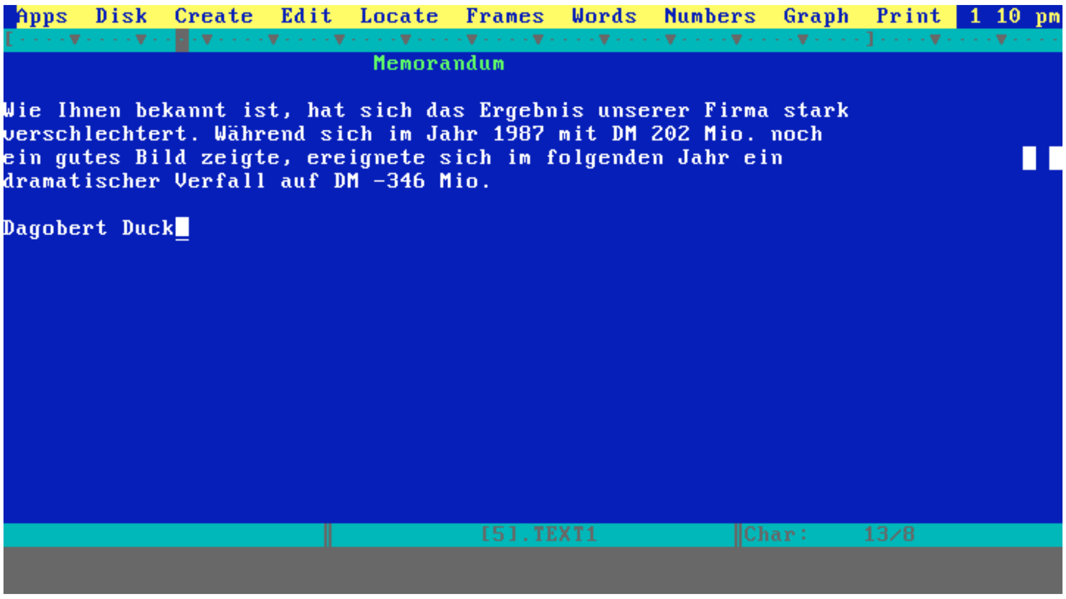
Figure 15 - Screenshot of "4.fw3" file opened on Framework III original software.


Apps Disk Create Edit Locate Frames Words Numbers Graph Print 1 13 pm
<Library>
<A:>
>C:<
[5]
[4]
[5]
[5].TEXT1
Frame: 1/3

```
1   TEXT1
2   TABELLE1
3   GRAFIK1
```

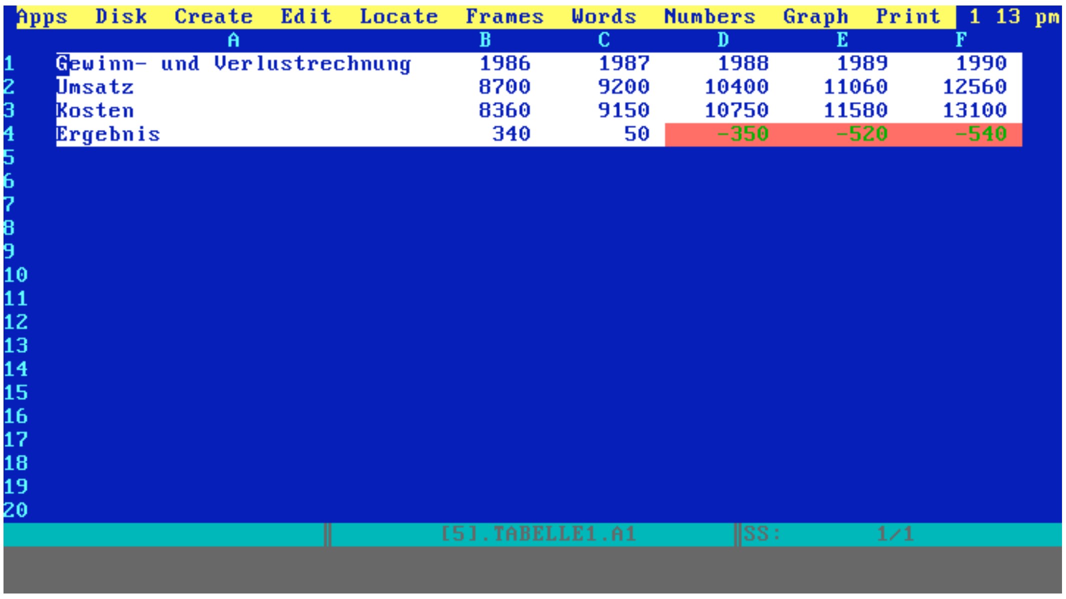
```
                                                              F
                                                               1988
                                                              =======
                                                                4,6%
                                                               22,0%
                                                              -------
                                                               14,0%
                                                                100
                                                                8,0%
9   E     Verwaltungskosten (%)          2,8%      2,3%      2,2%      2,5%
10  N     ------------------------------------------------------------------
```

Figure 16 - Screenshot of “5.fw3” file opened on Framework III original software.


Apps Disk Create Edit Locate Frames Words Numbers Graph Print 1 10 pm
[5].TEXT1
Char: 13/8

```
                            Memorandum

Wie Ihnen bekannt ist, hat sich das Ergebnis unserer Firma stark
verschlechtert. Während sich im Jahr 1987 mit DM 202 Mio. noch
ein gutes Bild zeigte, ereignete sich im folgenden Jahr ein
dramatischer Verfall auf DM -346 Mio.

Dagobert Duck
```

Figure 17 – Screenshot of “5.fw3” sub frame 1 opened on Framework III original software.


Apps Disk Create Edit Locate Frames Words Numbers Graph Print 1 13 pm

	A	B	C	D	E	F
1	Gewinn- und Verlustrechnung	1986	1987	1988	1989	1990
2	Umsatz	8700	9200	10400	11060	12560
3	Kosten	8360	9150	10750	11580	13100
4	Ergebnis	340	50	-350	-520	-540

```
[5].TABELLE1.A1        SS:        1/1
```

Figure 18 - Screenshot of "5.fw3" sub frame 2 opened on Framework III original software.

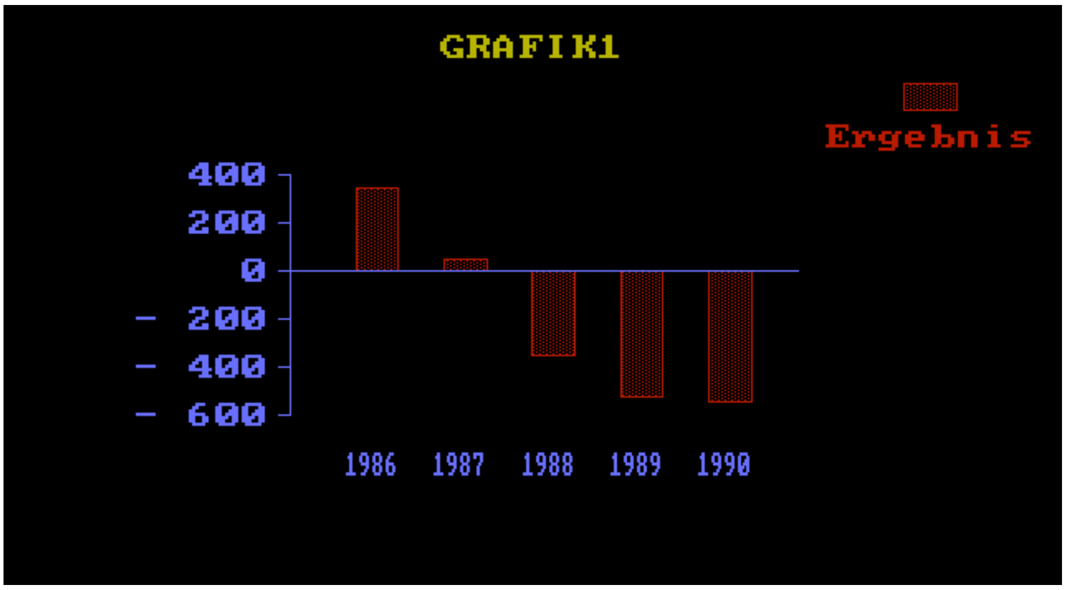


Figure 19 - Screenshot of "5.fw3" sub frame 3 opened on Framework III original software.

Opening the objects using their original software enabled us to better understand what in reality were the contents of those files. Framework supports several types of objects, as an office suite usually does, however, instead of storing each type into a different file format (as current office suites do), Framework stores all sorts of object types in a single file format – the FW3. We could identify at least 5 types of

Framework objects are called frames and are of the following types: word-processing, graph, database, spreadsheet and outline (some sort of index type of frame that can link to other frames). The following table categorizes each of the objects provided in this scenario:

File name	Type	Observations
1.fw3	Word-processing frame	This object was composed of text with embedded style information such as bold, italic or underline. Justification of text, such as centred or right alignment, may also be present in this type of frame. Page breaks are also a possible feature (Figure 12).
2.fw3	Graph frame	This object was constituted by a bar graph. The information carried by this object was mostly graphical (Figure 13).
3.fw3	Database frame	This object contained simple implementation of a database with three attributes (Name, Vorname and Wohnort) and three records (Figure 14).
4.fw3	Spreadsheet frame	This object was a spreadsheet frame. The spreadsheet was composed of several cells with values and formulas (Figure 15).
5.fw3	Outline frame	This object can be considered a compound object composed of 4 different frames: an outline frame that provided links to the rest of the frames, a word-processing frame, a spreadsheet frame and a graph frame (Figure 16 to Figure 19).

Comparing the objects rendered with the original software with the output produced by the Word add-on we would argue that for the word-processing frames, the add-on performed fairly well. The text was converted accordingly, as well as style. However, other types of objects were not property converted by the Word add-on: spreadsheet formulas were not converted correctly, graph frames were not converted at all. The outline frame was not converted, but its constituents were (except the graph frame).

R&L Software GmbH also provides a Framework III to Microsoft Excel Converter, however, our tests led us to conclude that although the results of importing a spreadsheet to Excel produced better results than importing it to Word, the results were not perfect - formulas were imported with many errors such as cyclic references.

In order to implement an appropriate preservation strategy for this type objects we would have to find/develop a better conversion application. In alternative, we could use the original Framework III software under an emulator such as DOSBox to render the objects and get a grasp of its contents. However, we feel that this approach embeds some disadvantages for the common user, such as:

- The need to understand and be able to operate long-gone operating systems and applications.
- The necessity to preserve together with the original object an assortment of applications such as viewer application, its operating system and an emulator capable of setting an adequate environment to run them.
- The ability to reuse information is not guaranteed.

Furthermore, automating an emulation-based preservation strategy in a way that access to the information ca be made easy for all types of users is not a straightforward task.

We feel that the best preservation strategy for this particular scenario would be a migration-based strategy. The objects should be converted to easy to interpret formats, such as ASCII text (or annotated text such as XML), comma separated values (for the spreadsheet objects assuming that the formulas were not a significant property) and raster images for the generated charts (uncompressed TIF for example). However, in order to do that, we would need an appropriate conversion application.

Selections & Functions, Inc. sells a Windows version of Framework suite. In their Web site they state that the Windows version of the office suite (Framework IX) works seamlessly with other windows applications and is able to correctly open objects produced with previous versions of the Framework application suite:

> *Framework is a continuation of Ashton-Tate's Framework (Framework, II, III, and IV) and the FRED computer language. The Framework software and trademark were taken over by Selection & Functions Inc. from Borland after Ashton-Tate was taken over by Borland. Selection & Functions Inc. with the stated goal of protecting the significant investment made by Framework users and FRED programmers in this unique technology and concepts. The current Windows versions of Framework which are available to Framework users are fully compatible with the original Framework methods and user interface, files, macros, FRED programs, libraries and dictionaries, while providing transparent compatibility with Windows programs and data.*
>
> *Copying between Framework and Windows MS Word, Excel, HTML, Internet e-mail, and most other programs preserves text style and format as well as spreadsheet cell and database field values. Images cut by Framework, including pictures opened in Framework (JPG, GIF, BMP, etc.) can be pasted into Windows graphic programs such as MS Paint and Photoshop.*[23]

[23] Text taken from http://www.framework.com.

Additionally, Framework comes with a programming language called FRED. This language can be used to develop small programmes that can interact with any of Framework's internal objects (something we would now call a Macro).

Our suggestion for a wide-scale preservation strategy for Framework III objects would be to develop a conversion Macro using FRED programming language that could open the original ".fw3" object and save it in a format more adequate for long term preservation. The Macro should be able to be executed from the command-line by some sort of batch file, or shell script, that would pass the name of the file to be converted, and eventually the destination directory where the results of conversion should be saved.

The shell script should be able to open Framework and execute the correct Macro script (with the correct parameters). The Macro programme should also be able to determine the type of frame included in the object (e.g. word-processing document, spreadsheet, graph, database or outline) and save then in an appropriate preservation format.

A suggestion for preservation formats would be:

- Word-processing frames - As far as we could tell from the original application, Framework III could not mix graphic data with text. However, text could have style associated with it, such as bold, underline, strike-through or italic. Furthermore, we can have page breaks and text may be aligned in several different ways: left, centred, right and justified. We feel that style and alignment should be preserved with the text it self. Therefore, we would suggest a preservation format such as XML. XML would be both easy to decode and easy to convert to any presentation format. XHTML would be a good candidate for this purpose. Only page breaks could not be represented in this format. Additionally, TEI[24] could be considered as a preservation format.
- Spreadsheet frames – a spreadsheet frame is usually composed of a two-dimensional matrix of cells that typically contain numeric data. Each cell also contains a unique identifier (usually its X and Y co-ordinates) and may contain a formula to execute a mathematical function and formatting for the presentation of the cell content [29]. Some cells contain text that serves as label for the numeric values included. Framework spreadsheets include the possibility to have three-dimensional data, i.e. a single cell may be clicked to open an entirely new spreadsheet. To preserve this type of frame we could export each worksheet to an ASCII or UNICODE delimited text file (tabs or pipe delimiters are preferred over commas). Each additional worksheet (3D effect) could be saved separately and adequately referenced from the original pointing cell. The Formulas used to calculate data values may be extracted as separate objects if their explanatory value is considered worth preserving [30].

[24] http://www.tei-c.org

- Database frames – database frames created with Framework III are composed of simple tabular data. The first row contains the name of the record fields. The subsequent rows constitute the records of the database. No relational information could be found on the originial application. As a preservation format, we would suggest exporting all tabular data to a delimited text file (tabs or pipe delimiters are preferred over commas).
- Graph frames – graph frames are rendered by Framework as graphical entities. We would suggest preserving this type of object in a raster image format such as uncompressed TIF. JPEG 2000 is often cited as the preservation format to be used in the near future for this type of objects, but its adoption is still low. We would advise to wait a little longer and evaluate how the community has embraced this recent format.
- Outline frames – Outline frames are composed by frames of the previously described types. We would suggest storing the index file in a METS file keeping the links to all its constituents and convert each of them to the previously described preservation formats accordingly.

4.6 Digital Preservation of Multimedia Art

Scenario description

A large international art network has provided an interdisciplinary platform for developing digital art works since 1984. The rapid pace of change in software tools and frameworks used for multimedia authoring has meant that its archived artworks are in danger of becoming inaccessible and unusable. You have been asked to preserve two of these historical digital artworks for future generations and to develop appropriate digital preservation strategies. How will you achieve this?

Task

1. Develop preservation strategies for the two pieces of multimedia art provided. 2. Decide which aspects of the artwork to preserve and which to lose. 3. Point out the differences in the strategies between to two artworks. 4. (Optional) Implement the preservation strategies you have developed and submit code for this.

Solution walkthrough

Scenario 6 included two folders. The first contained the installation package of a children's music application called DoremiBox (Figure 20). The other, contained a Windows executable file produced with Macromedia Director 7.0.1G that when run, displayed a simple animation of a wobbling rounded-corner rectangle over a black canvas (Figure 21).

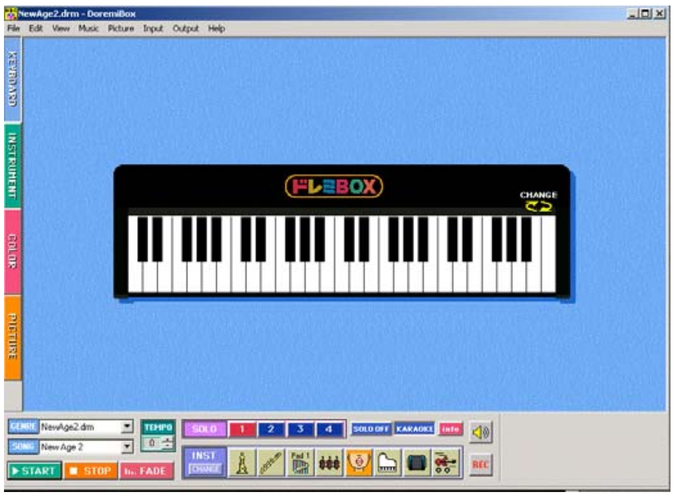


Figure 20 - Screenshot of the "DoremiBox" application.

Both of these objects are dynamic and/or interactive in nature. Furthermore, they are software applications which means that they have embed functionality that does not depend of any particular application to render them besides the operating system. The technological context that is required to render these files is therefore an operating system, which in this particular case is a 32 bits Windows operating system, such as Windows XP or Vista.

DoremiBox, however, possesses some further dependencies. To use this application correctly, one would need a MIDI compatible soundcard.

Since these are software applications, the identification of their significant properties is more complex than for other types of digital objects. These objects are interactive and dynamic. One of them is able to play sound. So "behaviour" and "aesthetics" (or "look and feel") are fundamental properties that should be preserved. The ability to reproduce "sound" is also relevant in the case of the "DoremiBox" application.

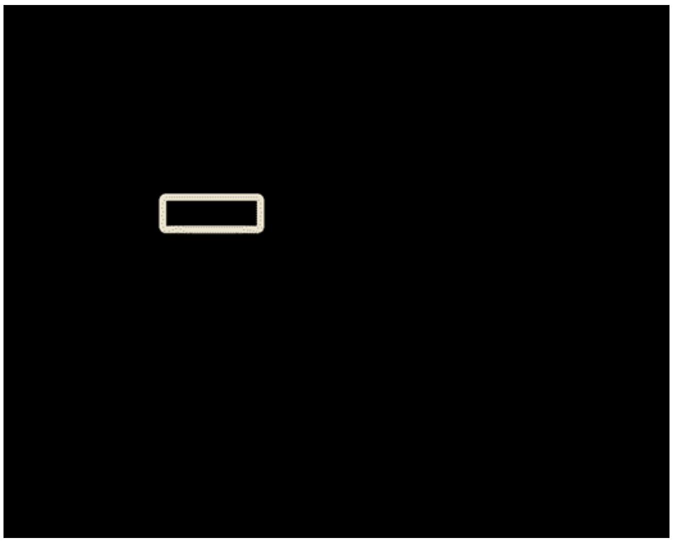


Figure 21 – Screenshot of the "mund.exe" animation.

To preserve these objects we would recommend an emulation strategy (see Emulation on page 8). In order to fully implement this strategy the following steps should be taken:

1. Preserve the original bits – store the original bits of the application in a safe medium (CD-ROM, hard disk, etc.). Make enough replicas of the data to make sure that at least one copy of the object survives in the event of hazard. Refresh the media whenever necessary, i.e. before the media gets corrupted or as soon as a new technology invades the market to replace the current media.
2. Save a working copy of the rendering environment – make a clean installation of all the software and hardware necessary (including the operating system) to render the objects and make a bitwise image of that environment[25]. Preserve the image of your technological environment as it will be used in the future to recreate the original setting.
3. Add sufficient metadata – together with your original object and image of the rendering environment keep enough metadata to describe both the object (descriptive metadata) and the technological setting including both software and hardware (technical metadata). This will be useful later on to discover your original objects and to identify an appropriate emulator to run the preserved image.

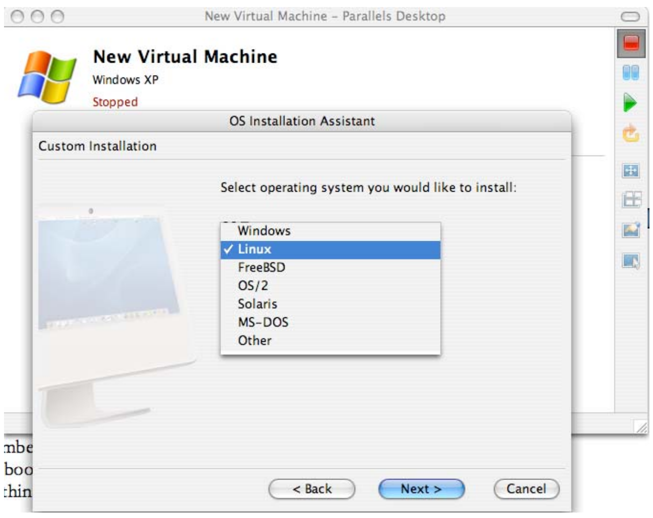
During the time that the hardware is still compatible with the operating system that originally supported the objects under preservation no special treatment has to be

[25] Several tools are available that allow the user to perform this task. Under Unix-based operating systems one may use the *dd* command to accomplish this.

done to gain access to the files. One must only copy the bitwise image of the technological setting to a location were a regular computer will be able to boot from.

When current hardware is no longer compatible with the operating system stored within the image, an emulator should be used. A special type of emulator called virtual machine [31, 32] is becoming trendy nowadays for many purposes other than digital preservation. This type of software has some interesting features such as the capacity to setup your own hardware configuration. For example, you may specify the amount of memory that will be available to your virtual machine, the size of the hard disk, the number of external drives, etc. (here we begin to understand the importance of technical metadata). It will also let you specify an image file to boot from. If the hardware is set correctly, the OS image will boot appropriately within the previously set virtual machine.

There are at least two decent products on the market today: VMWare Workstation [32] and Parallels Desktop [31] (Figure 22). Both offer enough virtual hardware to run most of the operating systems of the last 10 years, e.g. Windows (from version 3.11 to Vista), Linux, FreeBSD, OS/2, Solaris and Ms-DOS.



Select operating system you would like to install:

✓ Linux

Figure 22 - Screenshot of Parallels Desktop.

Figure 23 presents a screenshot of the "DoremiBox" application running on a Mac Os X inside a Windows Vista emulator (i.e. Parallels). The application requires a sound card to play the songs included in the package. The emulator is capable of transferring the audio from the virtual sound card to the physical card on the computer so the user can hear whatever audible information is being produced.

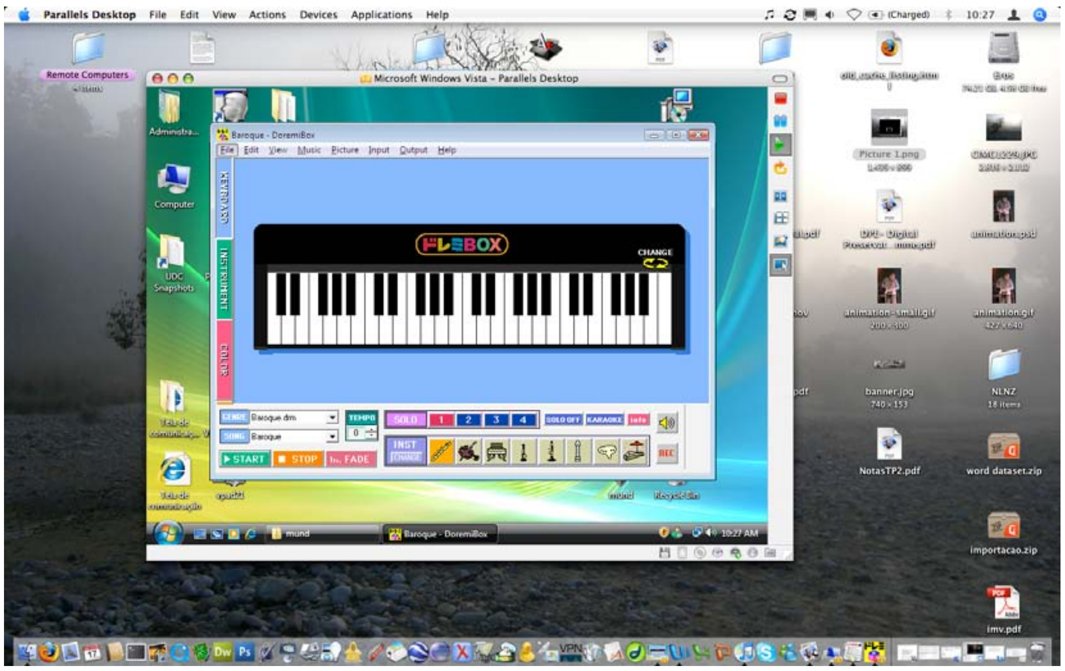


Figure 23 – "DoremiBox" application running on an emulator.

The "mund.exe" application, albeit dynamic, is not interactive. When executed it simply plays an animation of a wobbling rectangle. A shallow inspection of the animation led us to believe that the rectangle changes its shape in a non-repetitive way as if the movement and size of the rectangle were being calculated in real-time. Nevertheless, and given the nature of this particular multimedia object, one could attempt to preserve it by recording a few minutes of its animation and preserve the video file instead of the executable application. Access to the object would be much simpler for the majority of consumers, as no additional artefacts such as emulators would be necessary to render the file.

We are tempted to say that by proceeding with this strategy we are preserving enough significant properties to ensure a correct interpretation of the artwork by future art consumers. The uniqueness of this particular multimedia artwork would be preserved (at least partially) at an acceptable cost, i.e. loosing its randomness and unpredictability. This are, of course, our own thoughts. If possible, the artist should be consulted in order to assess its opinion regarding the importance of the randomness-associated properties.

The video file with the recording should be encoded in a format adequate for long term preservation, e.g. MPEG-4 at a high bit rate [33].

CamStudio[26] application was used to record a 54 seconds video of "mund.exe" activity.

[26] http://sourceforge.net/projects/camstudio/

5 Summary

This section provides a summary of all the digital objects included in this Digital Preservation Challenge together with information regarding its format and a suggestion of a rendering/enabling application. This section also includes a description of all the files that accompany this report.

5.1 Digital objects

Scenario	File name	Format	Rendering application	Observations
1	message	Ogg	Apple iTunes 7	None
2	problem wchwom04.cbh	ChessDatabase Backup	ChessBase Light 2007	None
3	dpc.ace	Servlet	Jakarta Tomcat 5	The provided object was a compressed package with various files inside. After decompression, the resultant file structure was a Servlet application.
4	bullethellshmups.rz	Save game	Visual Boy Advance 1.8.0 Beta 3	The object was compressed. After decompression the object was identified as possibly being a save game from Vulanon 2.0 for Nintendo Game Boy Advance.
5	1.fw3 2.fw3 3.fw3 4.fw3 5.fw3 6.fw3	Fw3	Framework III	None
6	DoremiBox	Windows installation package	Windows Operating System (32 bits)	None
	Mund.exe	Windows application	Windows Operating System (32 bits)	None

5.2 Accompanying files

Folder name	File name	Description
Scenario 1	message-fixed.ogg	The fixed audio message.
	message corruped.ogg	A corrupted ogg file used in our compression experiment.
	message corrupted.wav	A corrupted wav file used in our compression experiment.
Scenario 3	exportToSQL.jsp	A JSP application that can be placed with the Servlet of scenario 3 to convert the original object database to SQL statements.
	db4o2xml	A java application and respective source-code for a generic Db4o to XML converter.
Scenario 4	Vulkanon-audio.ogg	The song of the initial screen of the game Vulkanon 2.0 for the Game Boy Advance.
	vulkanon-2.0.rom.gba	The Vulkanon 2.0 game ROM for the Game Boy Advance.
Scenario 6	Mund.avi	A video recording of the “mund.exe” animation.

6 Summary and conclusions

This section includes a summary of the most common steps involved in each walkthrough of the six preservation scenarios proposed in this challenge. It also outlines some conclusions we have drawn from this experience.

Lets first summarize the steps that composed each of the solution walkthroughs:

1. Format identification – this is fundamental step before any preservation action can be applied. It consists in determining the file format of the object we intend to preserve. Without information about an object's format, it is impossible to make sense of the bytes that compose it. Format identification throughout the challenge was accomplished using three distinct methods:
   a. Internal identification (*file* and Droid) – attempts to determine an object's file format by looking at its internal byte structure, especially, looking for known sequences of bytes usually called *magic numbers*.
   b. External identification (Droid and FILExt) – an attempt to determine an object's file format by looking at its extension.
   c. Manual (hexadecimal editor) – if the previous methods fail to determine the object's format, one may attempt to open the object in a hexadecimal editor and manually attempt to identify a byte sequence that might suggest a file format or a rendering application.
2. Fixing corrupted objects – some objects in the challenge contained errors in its byte stream, which prevented rendering applications from adequately open them. Before we could open such objects, we had to find the corrupted bytes and correct them. This usually involves looking at technical documentation about the given format or comparing the headers of corrupted objects with valid ones. After fixing an object, it is generally good practice to check if our format identification tools are able to determine the object's format.
3. Rendering the objects – getting access to the information embedded in a digital object usually implies using a software application that is capable of rendering the given object. This application can be the original software used in the creation of the object (e.g. opening a Word document in Microsoft Word) or a compatible application (e.g. opening an image produced in Photoshop in Picasa). It can also mean converting the object to a different format that is compatible with other viewing applications (e.g. converting a Word document to PDF to make it readable in Linux).
4. Preserving the object – to preserve a digital object we have fundamentally two approaches at our disposal. We may convert the object from its original encoding to a format more suitable for long-term preservation (i.e. normalisation) or we may preserve the entire technological context that is necessary to render that object in the future and make use of an appropriate emulator to recreate that environment.

In what concerns the preservation tools that were used thoroughly throughout this challenge we are led to draw the following conclusions. Droid, as a format identification tool, has proved to be ineffective in determining most of the file formats included in the challenge. Although we consider it to be a very useful tool for digital preservation, especially in what concerns format naming conventions, the format detection tool must be improved to cope with:

1. a larger spectrum of formats;
2. objects without extensions (a large part of Droids inner workings is dedicated to guessing a file's format based on its extension);
3. handle incomplete information, such as files with broken headers.

PRONOM should cooperate with FILExt web site and probably with some others such as Wotsit's Format[27] to collect format and decoding information.

Additionally, we believe it would be useful to assemble a Web site an assortment of useful applications for digital preservation. Examples of such applications could be hex editors, emulators, free viewers and converters for all kinds of file formats.

An online conversion service such as the ones provided by CRiB [34-37], TOM [38], Panic [39-41] or Media Convert[28] with comprehensive graphical user interfaces, programmable APIs and a large number of converters available would greatly simplify the migration of digital objects from obsolete file formats to more up-to-date encodings.

To conclude, we must congratulate all the people involved in the elaboration of this first Digital Preservation Challenge, as all the proposed problems were indeed quite challenging and very close to real life digital preservation scenarios.

References

[1] Wikipedia contributors, "Mac OS X," in Wikipedia, The Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Mac_OS_X&oldid=137150102. [Accessed 12 June 2007 09:45 UTC]

[2] Wikipedia contributors, "Windows XP," in Wikipedia, The Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Windows_XP&oldid=137418815. [Accessed 12 June 2007 09:53 UTC]

[3] Wikipedia contributors, "Ubuntu (Linux distribution)," in Wikipedia, The Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Ubuntu_%28Linux_distribution%29&oldid=137567263. [Accessed 12 June 2007 09:52 UTC]

[4] Wikipedia contributors, "PRONOM technical registry," in Wikipedia, The Free Encyclopedia. [Online]. Available:

[27] http://www.wotsit.org/

[28] http://media-convert.com/

http://en.wikipedia.org/w/index.php?title=PRONOM_technical_registry&oldid=122499347. [Accessed 12 June 2007 09:55 UTC]

[5] Wikipedia contributors, "File (Unix)," in Wikipedia, The Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/w/index.php?title=File_%28Unix%29&oldid=129554926. [Accessed 12 June 2007 09:56 UTC]

[6] Wikipedia contributors, "ChessBase," in Wikipedia, The Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/w/index.php?title=ChessBase&oldid=137056687. [Accessed 12 June 2007 09:58 UTC]

[7] Wikipedia contributors, "Framework (office suite)," in Wikipedia, The Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Framework_%28office_suite%29&oldid=135287221. [Accessed 12 June 2007 09:59 UTC]

[8] Wikipedia contributors, "Apache Tomcat," in Wikipedia, The Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Apache_Tomcat&oldid=136380369. [Accessed 12 June 2007 10:01 UTC]

[9] Wikipedia contributors, "Hex Editor," in Wikipedia, The Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Hex_editor&oldid=137068890. [Accessed 12 June 2007 10:02 UTC]

[10] Wikipedia contributors, "VisualBoyAdvance," in Wikipedia, The Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/w/index.php?title=VisualBoyAdvance&oldid=136747226. [Accessed 12 June 2007 10:03 UTC]

[11] Wikipedia contributors, "DOSBox," in Wikipedia, The Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/w/index.php?title=DOSBox&oldid=135337263. [Accessed 12 June 2007 10:09 UTC]

[12] J. Rothenberg, Commission on Preservation and Access and Council on Library and Information Resources, *Avoiding technological quicksand: finding a viable technical foundation for digital preservation: a report to the Council on Library and Information Resources*. Washington, DC: Council on Library and Information Resources, 1999.

[13] K.-H. Lee, O. Slattery, R. Lu, X. Tang and V. McCrary, "The State of the Art and Practice in Digital Preservation," *Journal of Research of the National Institute of Standards and Technology*, vol. 107, no. 1, pp. 93-106, 2002.

[14] D. Woodyard, "Digital Preservation: The Australian Experience," presented at Third Conference Digital Library: Positioning the Fountain of Kowledge, Malaysia, 2000.

[15] M. Ferreira, "Automatic Evaluation of Migration Quality in Distributed Networks of Converters," *Bulletin of the IEEE Technical Committee on Digital Libraries (TCDL)*, vol. 2, no. 2, 2006.

[16] Digital Preservation Testbed, "Migration: Context and Current Status," The Hague, White Paper, 2001.

[17] R. A. Lorie, "Long Term Preservation of Digital Information," presented at First ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'01), Roanoke, Virginia, USA, 2001.

[18] R. A. Lorie, "A Methodology and System for Preserving Digital Data," presented at Second ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'02), Portland, Oregon, 2002.

[19] Task Force on Archiving of Digital Information, Commission on Preservation and Access and Research Libraries Group, *Preserving digital information: report of the Task Force on Archiving of Digital Information*. Washington, D.C.: Commission on Preservation and Access, 1996.

[20] G. Hodge and E. Frangakis, "Digital Preservation and Permanent Access to Scientific Information: The State of the Practice," International Council for Scientific and Technical Information & CENDI, Report 2004-3: Rev. 05/04, 2004.

[21] M. Hedstrom, "Digital Preservation: A time bomb for digital libraries," *Computers and the Humanities*, vol. 31, pp. 189-202, 1998.

[22] P. Mellor, P. Wheatley and D. M. Sergeant, "Migration on Request, a Practical Technique for Preservation," presented at ECDL '02: 6th European Conference on Research and Advanced Technology for Digital Libraries, London, UK, 2002.

[23] T. Hendley, "Comparison of Methods & Costs of Digital Preservation," British Library Research and Innovation Center, West Yorkshire 106, 1998.

[24] D. Bearman, "Archival Methods," Archives and Museum Informatics, Pittsburgh, Techical Report 1, 1989.

[25] H. Besser, "Digital Preservation of Moving Image Material?," *The Journal of the Association of Moving Image Archivists*, vol. 1, no. 2, pp. 39-55, 2001.

[26] Wikipedia contributors, "Ogg," in Wikipedia, The Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Ogg&oldid=138051728. [Accessed 14 June 2007 10:55 UTC]

[27] JSTOR and Harvard University Library, "JHove - JSTOR/Harvard Object Validation Environment." [Online]. Available: http://hul.harvard.edu/jhove.

[28] Wikipedia contributors, "Speex," in Wikipedia, The Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Speex&oldid=135706302. [Accessed 22 June 2007 14:51 UTC]

[29] R. Ruusalepp, "The taxonomy of data types and file formats in the AHDS collection," AHDS and Estonian Business Archives, Ltd., 2002.

[30] AHDS History, "Preservation Handbook - Spreadsheets," Arts and Humanities Data Service, London, 2005.

[31] Parallels, "Parallels Desktop Web site," 1995. [Online]. Available: http://www.parallels.com. [Accessed 2006-10-12]

[32] VMWare, "VMWare Workstation Web site," 1998. [Online]. Available: http://www.vmware.com/.

[33] Library of Congress, "Sustainability of Digital Formats - Planning for Library of Congress Collections - Moving Image." [Online]. Available: http://www.digitalpreservation.gov/formats/content/video_preferences.shtml. [Accessed 2007-07-10]

[34] M. Ferreira, "CRiB: Migration Workbench," 2006. [Online]. Available: http://digitarq.di.uminho.pt/MigrationWorkbench.

[35] M. Ferreira, A. A. Baptista and J. C. Ramalho, "CRiB: A service oriented architecture for digital preservation outsourcing," presented at XATA - XML: Aplicações e Tecnologias Associadas, Portalegre, Portugal, 2006.

[36] M. Ferreira, "CRiB - Conversion and Recommendation of Digital Object Formats Web site," 2006. [Online]. Available: http://crib.dsi.uminho.pt.

[37] M. Ferreira, A. A. Baptista and J. C. Ramalho, "An intelligent decision support system for digital preservation," *International Journal on Digital Libraries*, 2007.

[38] J. M. Ockerbloom, "TOM Conversion Service," 2003. [Online]. Available: http://tom.library.upenn.edu/convert/.

[39] J. Hunter and S. Choudhury, "A Semi-Automated Digital Preservation System based on Semantic Web Services," presented at Joint ACM/IEEE Conference on Digital Libraries (JCDL'04), 2004.

[40] J. Hunter and S. Choudhury, "Preservation webservices Architecture for Newmedia and Interactive Collections (PANIC)," 2005. [Online]. Available: http://metadata.net/newmedia/.

[41] J. Hunter and S. Choudhury, "PANIC: an integrated approach to the preservation of composite digital objects using Semantic Web services," *International Journal on Digital Libraries*, vol. 6, no. 2, pp. 174-183, 2006.