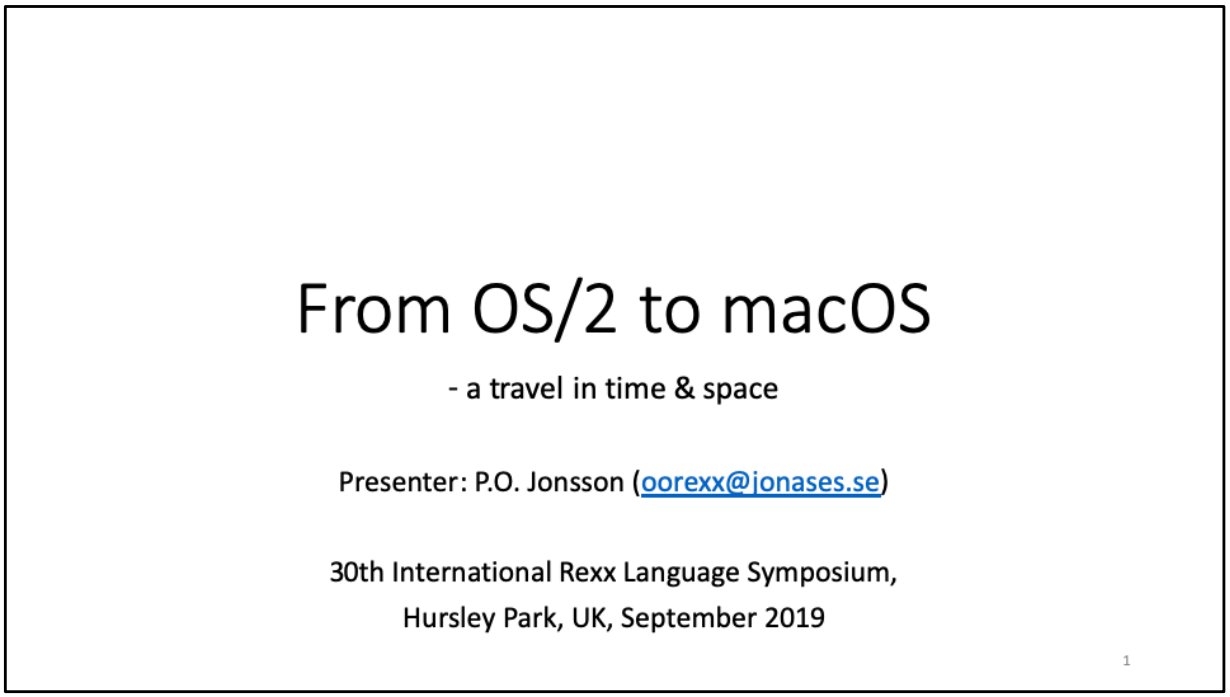


Presenter P.O. Jonsson, oorexx@jonases.se

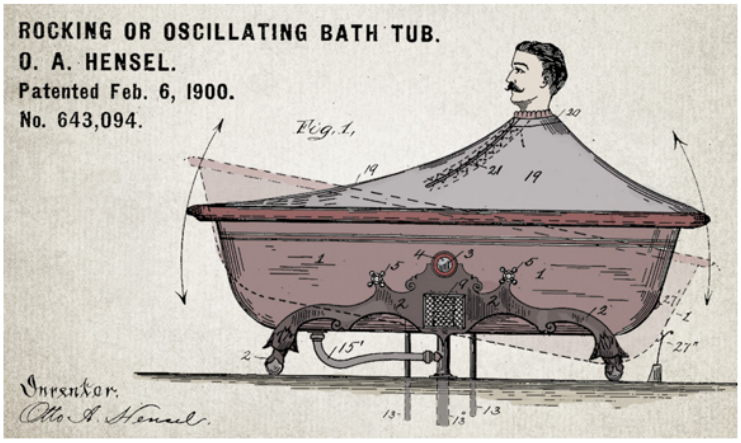
Session title: "From OS/2 to macOS - a travel in time & space“

Session abstract:
Showcased is a rexx project that started as a simple data entry script in 2000, a project that eventually evolved into a fully automated data mining ooRexx program in 2019.

The presentation is, I think, proof of the resilience of Rexx in moving between operating systems and platforms.

Credentials:
P.O. Jonsson have a MSc in engineering physics and worked as a development engineer before taking up a post as a patent examiner at the European Patent Office (EPO) some 30 years ago. At the EPO he came into contact with Rexx as it was used to automate search procedures. In his spare time he used Rexx and later ooRexx to create in-house search tools that are now part of the toolbox for all examiners at the EPO.

Context – Patents



2

I have worked for almost 30 years as a patent examiner and to understand my project you need to get familiar with some patent basics.

I will explain with the help of one example:

I will use a European Patent application rather than this one.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 231 716 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
14.08.2002 Bulletin 2002/33

(51) Int Cl.[7]: **H03M 7/00**

(21) Application number: **01308086.6**

(22) Date of filing: **24.09.2001**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **27.01.2001 GB 0102154**

(71) Applicant: **International Business Machines Corporation**
Armonk, NY 10504 (US)

(72) Inventor: **Cowlishaw, Michael Frederic,**
c/o IBM UK Ltd.
Winchester, Hampshire SO21 2JN (GB)

(74) Representative: **Davies, Simon Robert**
IBM,
United Kingdom Limited,
Intellectual Property Law,
Hursley Park
Winchester, Hampshire SO21 2JN (GB)

(54) **Decimal to binary coder/decoder**

(57) A system and method are provided for encoding from decimal to binary and back again. The coding is based on representing 3 decimal digits as 10 binary bits and is a development of the Chen-Ho algorithm. This provides a storage efficiency of > 99%, yet still allows decimal arithmetic to be readily performed. The decimal input is typically first converted to binary coded decimal (4 bits per decimal digit), before compression to 10 bits. Adopting the encoding of the present invention, if the leading (most significant) decimal digit is zero, then the first three bits of the binary output are zero; and if the first two decimal digits are zero, then the first six bits of the binary output are zero. Accordingly, the same coding can be flexibly used to code 2 decimal digits to 7 binary bits, and 1 decimal digit to 4 binary bits. This makes it particularly suitable for standard computer architectures which are based on a power of two (16-bit, 32-bit, 64-bit, etc), and therefore not directly divisible by 7 or by 10.

3

Here is how a European, or EP patent application looks like (using a coder/decoder system as an example)

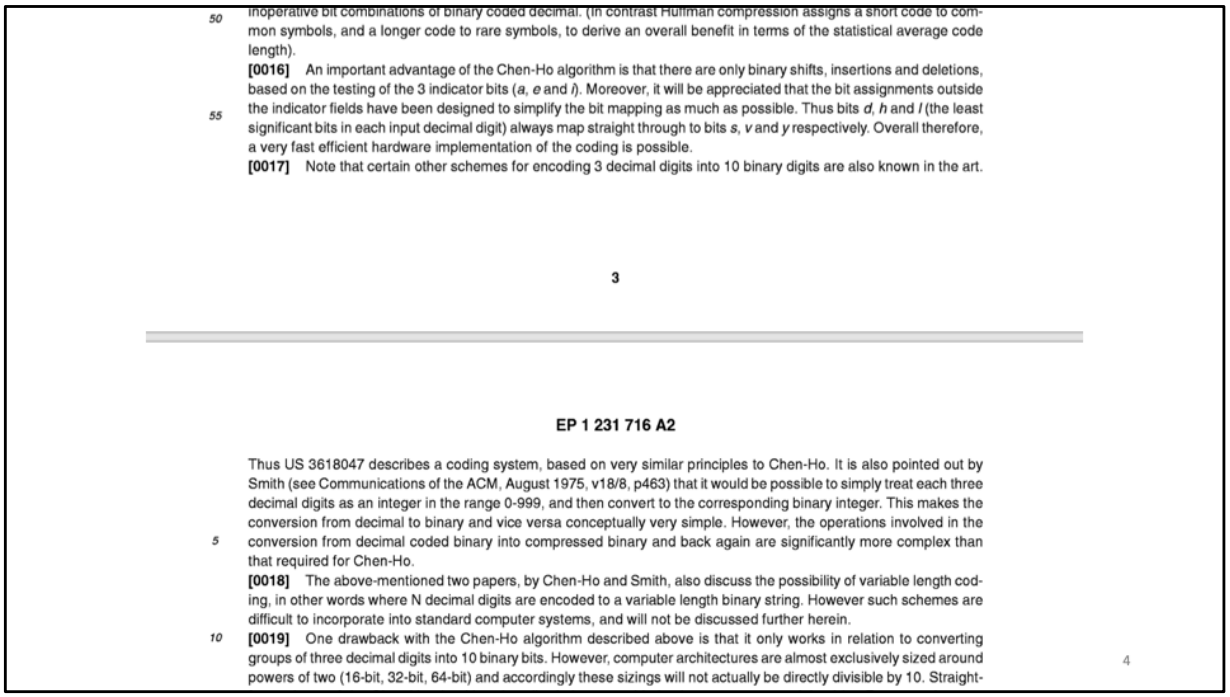
The Applicant is known to us (IBM)

We also know the inventor, I think? (Mike Cowlishaw)

And the representatives location is also familiar (Hursley Part).

Now, for a patent examiner the work starts by reading the application, trying to understand the inventive idea, and what distinguish it from the prior art.

When reading the application you will typically see something like this:



inoperative bit combinations of binary coded decimal. (In contrast Huffman compression assigns a short code to common symbols, and a longer code to rare symbols, to derive an overall benefit in terms of the statistical average code length).

[0016] An important advantage of the Chen-Ho algorithm is that there are only binary shifts, insertions and deletions, based on the testing of the 3 indicator bits (*a, e* and *i*). Moreover, it will be appreciated that the bit assignments outside the indicator fields have been designed to simplify the bit mapping as much as possible. Thus bits *d, h* and *l* (the least significant bits in each input decimal digit) always map straight through to bits *s, v* and *y* respectively. Overall therefore, a very fast efficient hardware implementation of the coding is possible.

[0017] Note that certain other schemes for encoding 3 decimal digits into 10 binary digits are also known in the art.

3

EP 1 231 716 A2

Thus US 3618047 describes a coding system, based on very similar principles to Chen-Ho. It is also pointed out by Smith (see Communications of the ACM, August 1975, v18/8, p463) that it would be possible to simply treat each three decimal digits as an integer in the range 0-999, and then convert to the corresponding binary integer. This makes the conversion from decimal to binary and vice versa conceptually very simple. However, the operations involved in the conversion from decimal coded binary into compressed binary and back again are significantly more complex than that required for Chen-Ho.

[0018] The above-mentioned two papers, by Chen-Ho and Smith, also discuss the possibility of variable length coding, in other words where N decimal digits are encoded to a variable length binary string. However such schemes are difficult to incorporate into standard computer systems, and will not be discussed further herein.

[0019] One drawback with the Chen-Ho algorithm described above is that it only works in relation to converting groups of three decimal digits into 10 binary bits. However, computer architectures are almost exclusively sized around powers of two (16-bit, 32-bit, 64-bit) and accordingly these sizings will not actually be directly divisible by 10. Straight-

4

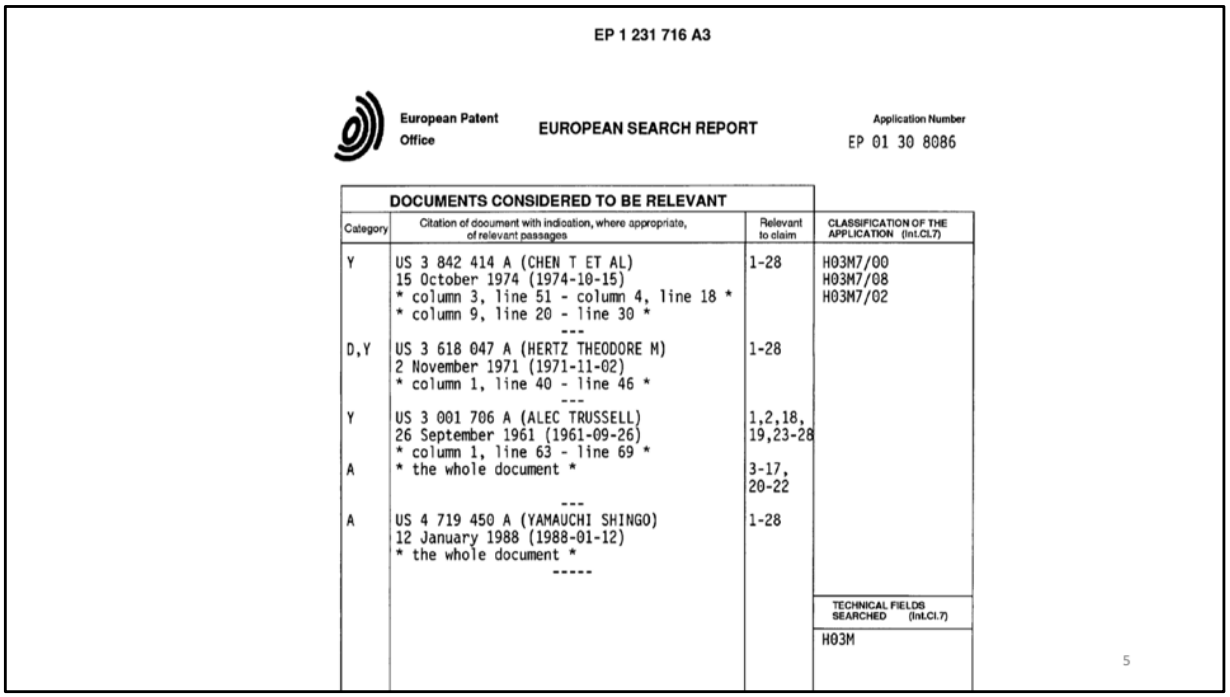
1st line of page 4 contains a cited docuement

I will enlarge the piece I am interested in (poining to the US document in line 1).

Now, since the inventor indicates that the invention builds on ideas presented in this document we need to make a detour and find that document.

THIS is where my little project kicks in. It might seem to be a trivial task but beleive me it is not, at least not every time.

Once the application is understood, a novelty search is performed and a search report drawn up



EP 1 231 716 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 01 30 8086

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
Y	US 3 842 414 A (CHEN T ET AL) 15 October 1974 (1974-10-15) * column 3, line 51 - column 4, line 18 * * column 9, line 20 - line 30 * ---	1-28	H03M7/00 H03M7/08 H03M7/02
D,Y	US 3 618 047 A (HERTZ THEODORE M) 2 November 1971 (1971-11-02) * column 1, line 40 - line 46 * ---	1-28	
Y A	US 3 001 706 A (ALEC TRUSSELL) 26 September 1961 (1961-09-26) * column 1, line 63 - line 69 * * the whole document * ---	1,2,18, 19,23-28 3-17, 20-22	
A	US 4 719 450 A (YAMAUCHI SHINGO) 12 January 1988 (1988-01-12) * the whole document * -----	1-28	

TECHNICAL FIELDS SEARCHED (Int.Cl.7)

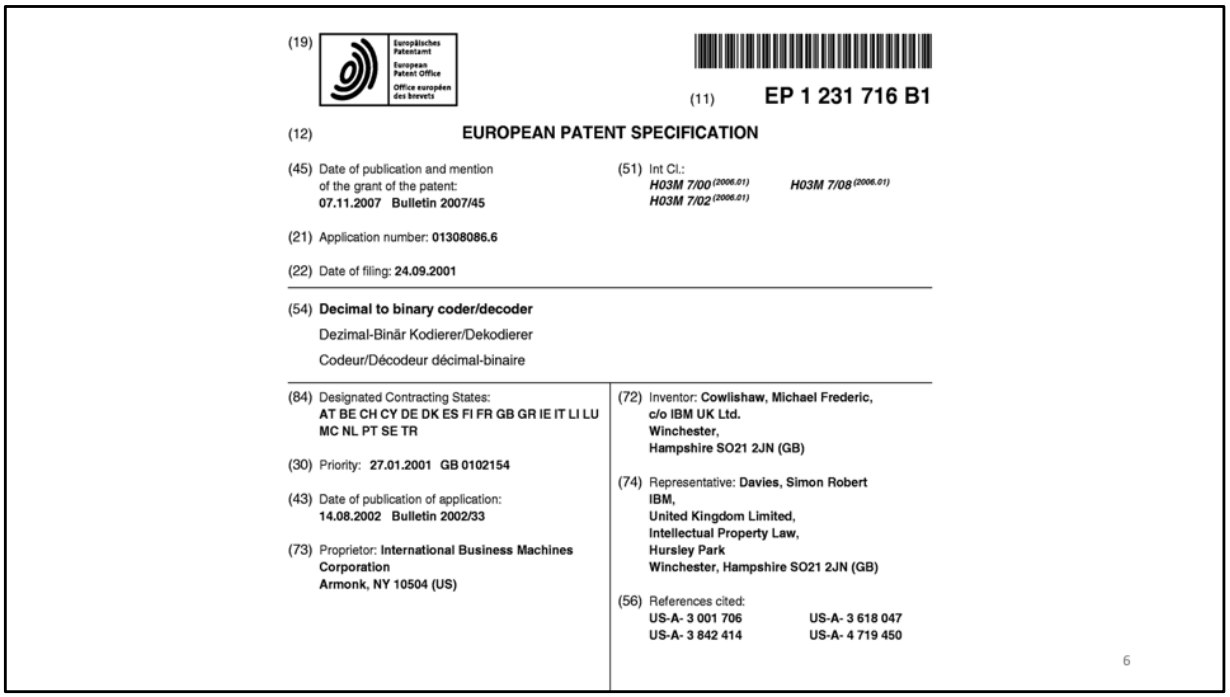
H03M

5

Here the examiner cited four documents, the second one being the one we just saw.

The „D“ in the search report indicates that this document come from the description itself

It is thus a „D-document“ in the lingua of patent examiners, and also the name of my project, Ddoc.



(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 231 716 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
07.11.2007 Bulletin 2007/45

(51) Int Cl.:
H03M 7/00 *(2006.01)* ***H03M 7/08*** *(2006.01)*
H03M 7/02 *(2006.01)*

(21) Application number: **01308086.6**

(22) Date of filing: **24.09.2001**

(54) **Decimal to binary coder/decoder**
Dezimal-Binär Kodierer/Dekodierer
Codeur/Décodeur décimal-binaire

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR

(30) Priority: **27.01.2001 GB 0102154**

(43) Date of publication of application:
14.08.2002 Bulletin 2002/33

(73) Proprietor: **International Business Machines Corporation**
Armonk, NY 10504 (US)

(72) Inventor: **Cowlishaw, Michael Frederic,**
c/o IBM UK Ltd.
Winchester,
Hampshire SO21 2JN (GB)

(74) Representative: **Davies, Simon Robert**
IBM,
United Kingdom Limited,
Intellectual Property Law,
Hursley Park
Winchester, Hampshire SO21 2JN (GB)

(56) References cited:
US-A- 3 001 706
US-A- 3 618 047
US-A- 3 842 414
US-A- 4 719 450

6

After an examination phase a patent is eventually handed down by the EPO

The EP patent can be validated in the Designated countries.

My project DDOC thus aims at retrieving the documents cited by the Applicant

Please note that US3618047, the document cited by the Applicant, is cited in three different ways in this single document, which makes it a non-obvious task to extract such documents in general.

Year 2000

- Version 1 of the program
- Under OS/2
- Using IBM REXX (or IBM Object REXX?)

7

The original data entry script contained around 300 lines of code

Another script was used by he examiners

The system was used by around 40 examiners

```
/* REXX */

/*
Command file ddocbns.cmd version 1.01
(c) P.O. Jonsson  15.6.2000
*/

IF ( RxFuncQuery( "SysLoadFuncs" ) ) THEN
DO
  CALL RxFuncAdd "SysLoadFuncs" , "RexxUtil" , "SysLoadFuncs"
  CALL SysLoadFuncs
END

PARSE ARG userid crossfile

IF userid = ' ' THEN
DO
  '@echo Please enter a valid userid'
  EXIT
END

IF crossfile = ' ' THEN
DO
  '@echo Please enter a valid cross file name'
  EXIT
END

'@copy t:\general\%1\ddoc\%2.xft  d:\epoque2\epxfil\vi_t1.xft'
'@copy t:\general\%1\ddoc\%2.cmt  d:\epoque2\epxfil\vi_t1.cmt'
EXIT
```

8

This is a representative script for this time,
just to give an idea of the complexity (joke)

Early Year 2003

- Version 2 of the program
- Still using OS/2 (possibly OS2/Warp)
- Still using IBM REXX (or IBM Object REXX)

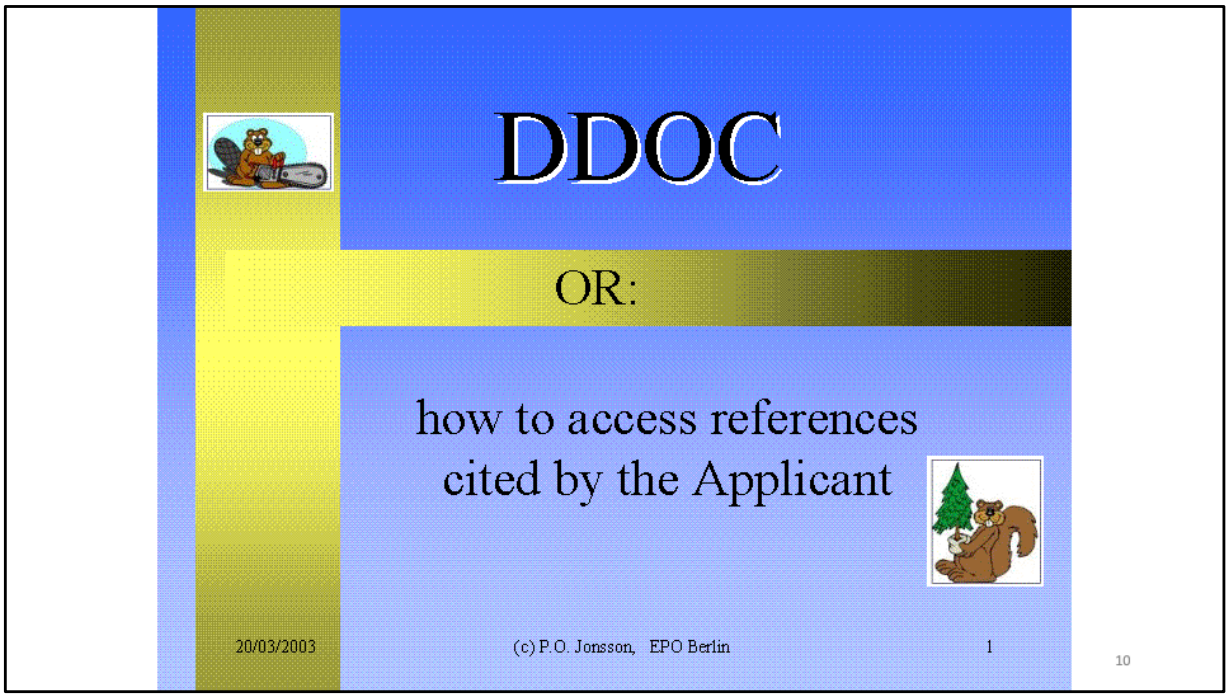
9

Used in 5 directorates ~140 Users

6500 files were processed between 2000 and 2003

Corresponding to 528 000 pages of paper not printed

A saving of 4.5 Pallets of paper, or 2.6 tons.



This is from a presentation I held in 2003
(probably the last time I held a presentation ☺)

DDOC System Layout I:

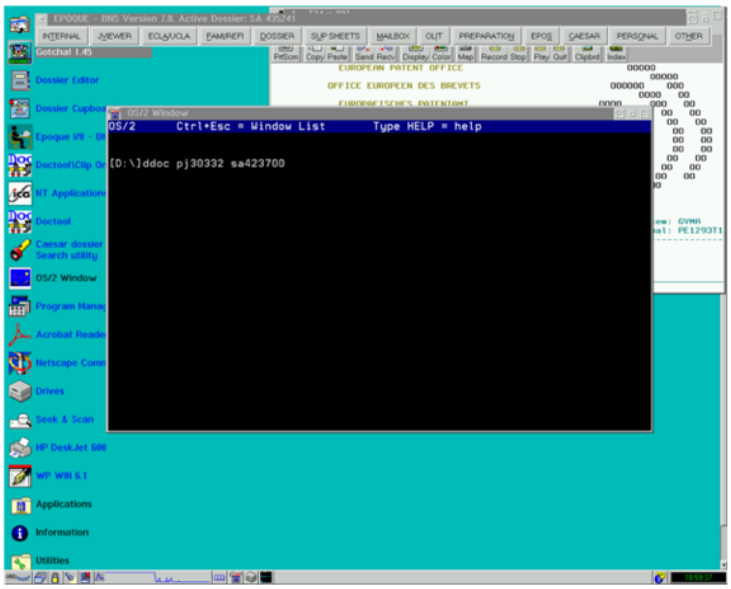
- **Clerk enter the references and DDOC generates a crossfile on the T: drive**

  T:\general\ddoc\pj30332\12345678.xft

DDOC System Layout II:

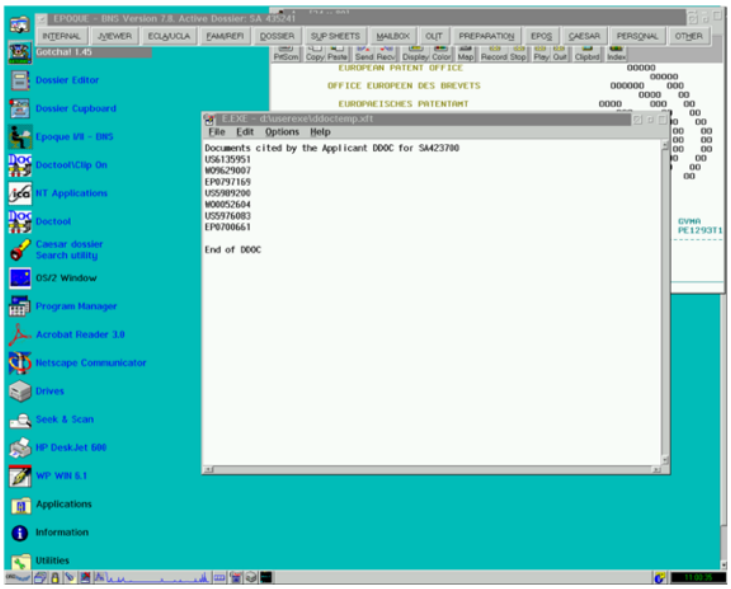
- **Crossfile is automatically downloaded to the Examiners workstation for direct use in EPOQUE (Jviewer, Internal, Caesar etc)**

T:\general\ddoc\pj30332\12345678.xft



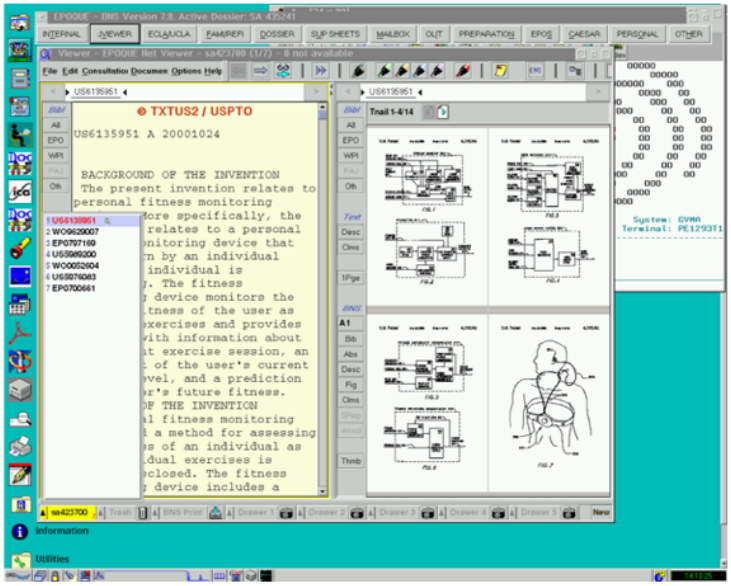
13

Please note the (familiar) IBM OS/2 desktop



14

Editor “E” was used for data input



15

The examiner´s view of the entered documents

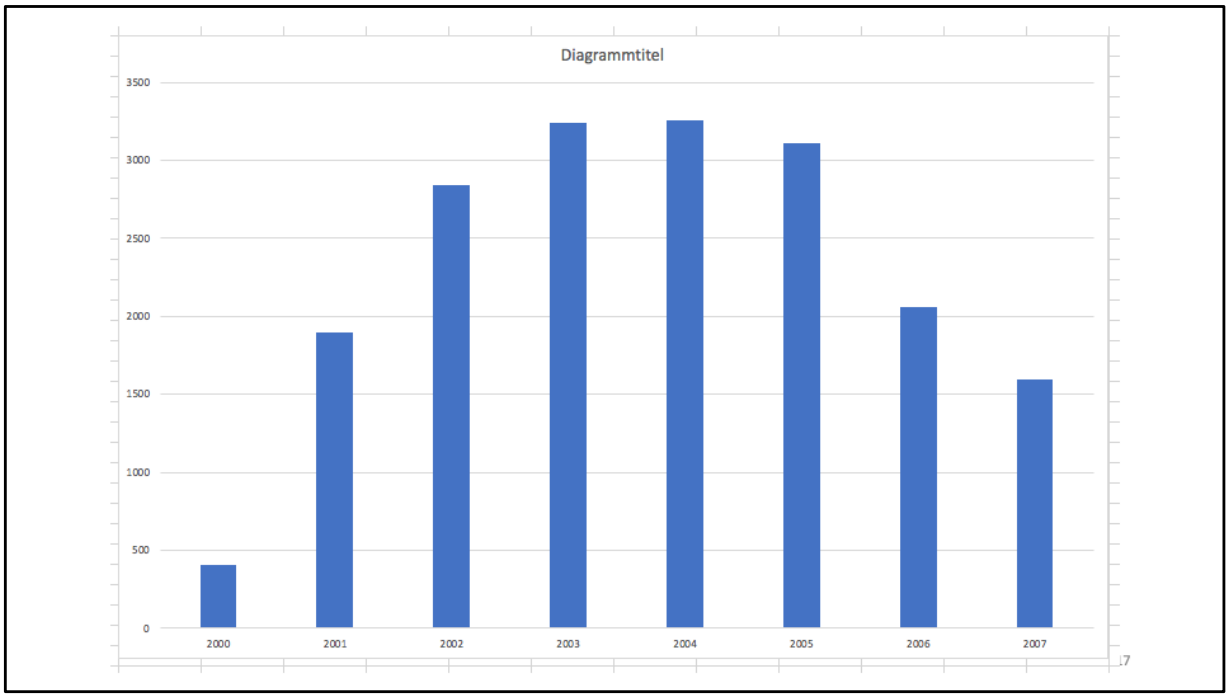
Late Year 2003

- Version 3 of the program
- Ported to Windows 2000
- Using IBM Object REXX V2.1 (most likely)

16

IBM Object Rexx was apparently delivered with W2k in 2003

In general the version number was bumped with one whenever the OS was changed



Stats on use

The decreasing use 2006-2007 and the possibility of obtaining the fulltext made me ponder over an automated tool.

Year 2008

- Ddoc version 4
- Windows XP replaces Windows 2000
- Still using IBM Object REXX V2.1

18

Ddoc is now using automatic extraction of data from fulltext

First tests with BSF4Rexx

First tests with ooRexx (4.0) in parallel to IBM Object Rexx

Year 2010

- Ddoc version 5
- Windows XP
- ooRexx (4.1) and BSF4ooRexx

19

The use of ooRexx + BSF4ooRexx made it possible to establish an interface between rexx and JAVA

Rexx used in the frontend by examiners to script database queries

and with the backend with middleware written in JAVA

A first ooRexx and BSF Workshop was held with Rony Flatscher 2011

Year 2014

- Ddoc version 6
- Windows 7
- Still ooRexx (4.2.0) and BSF4ooRexx

20

An entirely new database structure requires rewrite

Windows 7 triggers a new main version

Year 2016

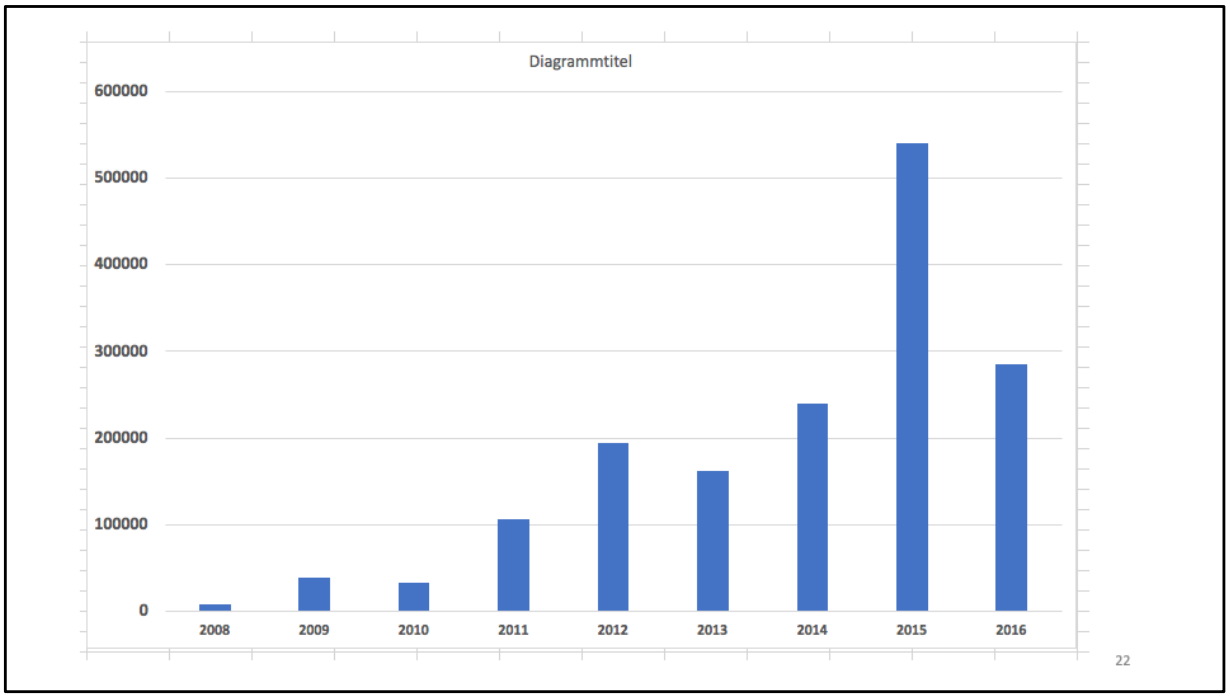
- Development frozen
- Ddoc handed over to the EPO
- migrated to JAVA (by the EPO)

21

Ddoc Migrated to JAVA 2016 by the EPO

I supported the 1:1 port from ooRexx to JAVA of 15 000 lines of code.

The port was so good that even bugs (as I detected later) had been ported!

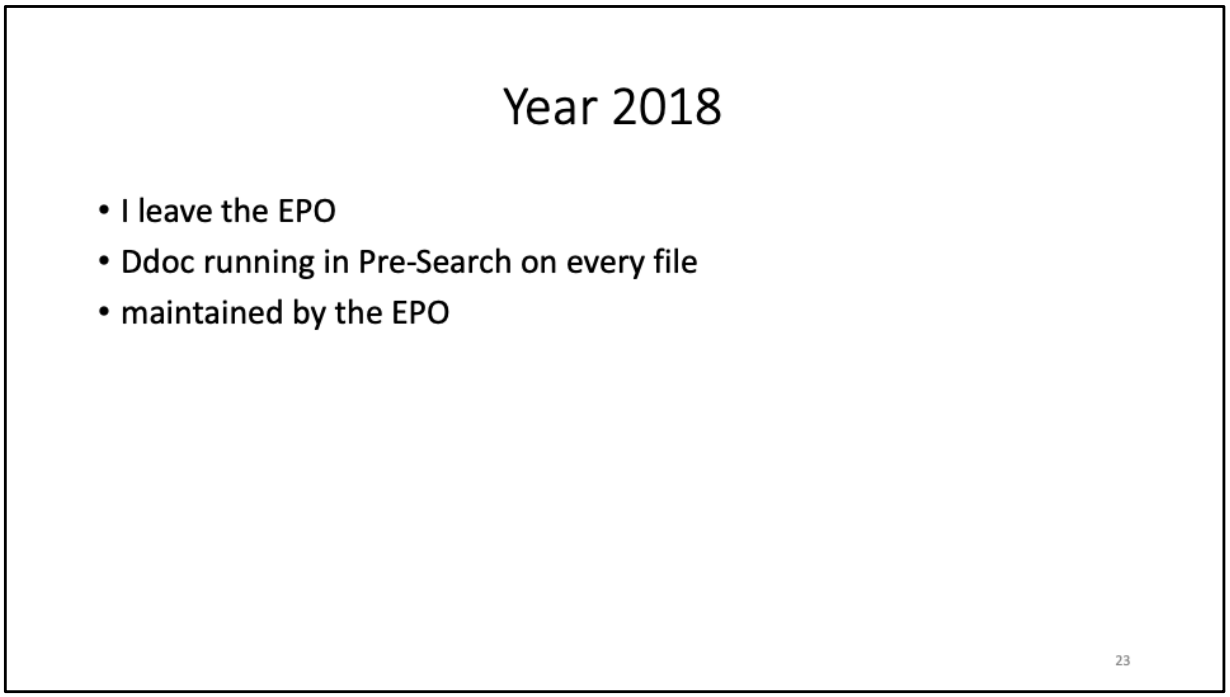


The statistics on use show a peak of half a million uses of Ddoc in 2015, or 2000 uses per day or 2-3 times per minute.

The reason that 2016 show lower number is because the JAVA version have taken over part of the work and stores the data differently than the ooRexx version.

What is seen here is for the ooRexx version only

After 2016 no further statistics on use was collected (by me)



In 2018 the JAVA version have taken over all tasks and the ooRexx version is only kept as a control

Year 2019

- Ddoc version 7
- macOS (High Sierra and Mojave)
- ooRexx 5.0.0

24

In 2019 I wanted to revive my project Ddoc and make it available to the public as a service.

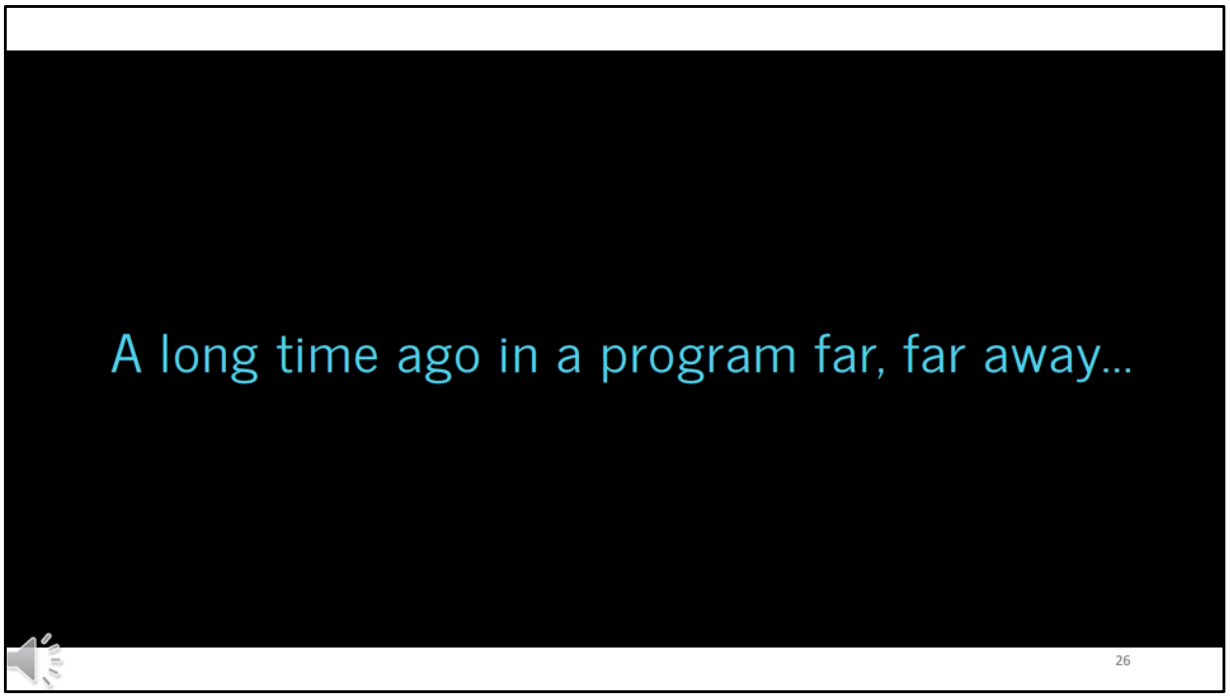
In order to migrate Ddoc to macOS I needed to replace the checking of document numbers.

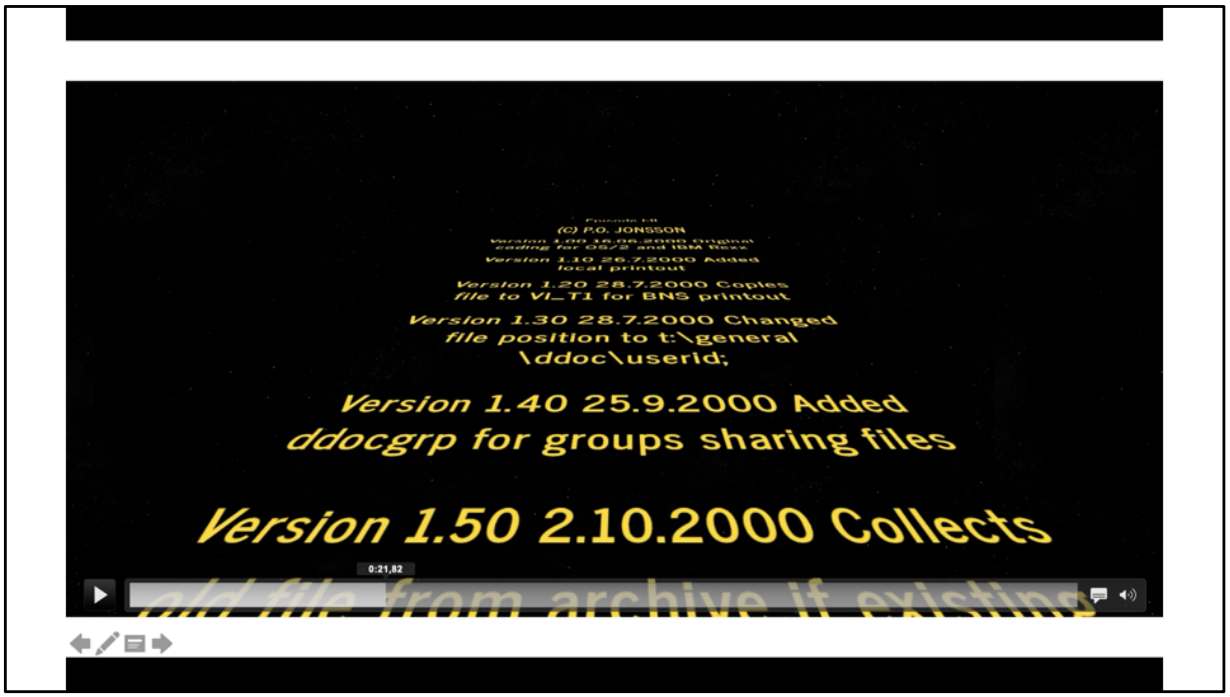
Where previously it was an inhouse database query it is now a https call over the internet to the EPOs API called OPS, Open Patent Service

Less than 10% of the code was rewritten, the client for OPS, the rest was ported to MacOS without any major changes needed to the code.

Thank you for your attention!

P.O. Jonsson (oorexx@jonases.se)


A long time ago in a program far, far away...
26


(c) P.O. JONSSON
Version 1.00 18.06.2000 Original
coding for OS/2 and IBM Rexx
Version 1.10 26.7.2000 Added
local printout
Version 1.20 28.7.2000 Copies
file to VI_T1 for BNS printout
Version 1.30 28.7.2000 Changed
file position to t:\general
\ddoc\userid;
Version 1.40 25.9.2000 Added
ddocgrp for groups sharing files
Version 1.50 2.10.2000 Collects
0:21,82