

Example Prior Art search

aus PriorArt, der freien Wissensdatenbank

<< back

Copyright (c) 2006 Jan O. Kechel
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A link to the license is included at the end of this document.

Here we try to verify our HowTo by using it ourself on real patent-applications.

Contents

- 1 Example Prior Art search for Patent WO2004097666
 - 1.1 Abstract
 - 1.2 Basic Steps
 - 1.2.1 Identify the Patent Priority Date
 - 1.2.2 Try to Understand the Patent (find the nut of the invention)
 - 1.2.2.1 Replace Unnecessary Complex Terms
 - 1.2.2.2 Remove Descriptions That do not Describe the Invention Itself
 - 1.2.2.3 Structure
 - 1.2.2.4 Pseudo-Code
 - 1.2.2.5 Summarize With Own Words (the point of novelty)
 - 1.3 Identify Related Projects
 - 1.3.1 Try to find out which kinds of systems might use similar methods
 - 1.3.2 Create a List With All Programs/Projects That Might Use Similar Methods
 - 1.3.3 Select Those That You Expect to Fit Best
 - 1.3.4 Determine in a first step from which date those (best rated) programs are
 - 1.3.4.1 ftp-clients - chance 1
 - 1.3.4.2 version control systems - chance 1
 - 1.3.4.3 mirror software - chance 1
 - 1.4 Verify selected Projects
 - 1.4.1 Verify SCCS
 - 1.4.1.1 Is the description older than the Priority-Date?
 - 1.4.1.2 Does it really implement or describe the patent?
 - 1.4.1.2.1 verify the summarization
 - 1.4.1.2.2 verify the original claim
 - 1.4.2 Verify CVS
 - 1.4.2.1 Is CVS older than the Priority-Date?
 - 1.4.2.2 Does CVS really implement or describe the patent?
 - 1.4.2.2.1 verify the summarization
 - 1.4.2.2.2 verify the original claim
 - 1.5 Conclusion
- 2 changelog
 - 2.1 31-MAR-2006
 - 2.2 19-MAR-2006

Example Prior Art search for Patent WO2004097666

Applicant
SIEMENS AG (DE)

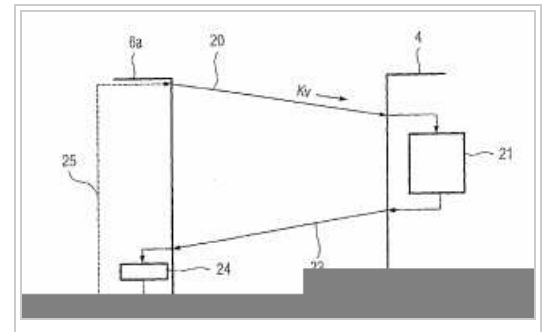
Documents

DE10319887 (<http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=DE10319887&F=0>) (B4 - granted patent)
WO2004097666 (<http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=WO2004097666&F=0>) (A1 - PCT)
EP1618494 (<http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=EP1618494&F=0>) (A1 - european application)
patent family
(<http://v3.espacenet.com/family?DB=EPODOC&IDX=EP1618494&F=8&QPN=EP1618494&OREQ=0&textdoc=T>)

Abstract

Abstract of WO2004097666

Disclosed is a method for adapting a database provided on a client data processing unit to a source database stored on a server data processing unit. In order to further develop said method in such a way that only comparatively small amounts of data have to be transmitted, the following steps are carried out: - a comparative characteristic value (KV) indicating the time status of the momentary database of the client data processing unit (6a) is transmitted from the client data processing unit (6a) to the server data processing unit (4); - a part of the source database of the server data processing unit (4), which has a more recent time status than the time status of the momentary database of the client data processing unit (6a) indicated by the comparative characteristic value (KV), is determined; - the determined part of the source database is transmitted to the client data processing unit (6a); the part of the source database, which has been transmitted from the server data processing unit (4), is received in the database of the client data processing unit (6a) so as to form an updated database of the client data processing unit (6a).



Basic Steps

Identify the Patent Priority Date

Priority date for this patent is **25-APR-2003**

Try to Understand the Patent (find the nut of the invention)

Replace Unnecessary Complex Terms

Disclosed is a method for adapting a database provided on a client to a source database stored on a server. In order to further develop said method in such a way that only comparatively small amounts of data have to be transmitted, the following steps are carried out: - a value (KV) indicating the time status of the momentary database of the client is transmitted from the client to the server; - a part of the source database of the server, which has a more recent time status than the time status of the momentary database of the client indicated by the value (KV), is determined; - the determined part of the source database is transmitted to the client; the part of the source database, which has been transmitted from the server, is received in the database of the client so as to form an updated database of the client.

- I replaced 'client data processing unit' with 'client' and 'server data processing unit' with 'server'
- I also removed the references to the (useless) diagram included in this *invention*

Remove Descriptions That do not Describe the Invention Itself

Method for adapting a database provided on a client to a source database stored on a server. - a value (KV) indicating the time status of the momentary database of the client is transmitted from the client to the server; - a part of the source database of the server, which has a more recent time status than the time status of the momentary database of the client

indicated by the value (KV), is determined; - the determined part of the source database is transmitted to the client; the part of the source database, which has been transmitted from the server, is received in the database of the client so as to form an updated database of the client.

- *In order to further develop said method in such a way that only comparatively small amounts of data have to be transmitted, the following steps are carried out:*

this is just a description of what the invention is intended do, but not part of the invention itself, so I removed it.

Structure

Method for adapting a database provided on a client to a source database stored on a server.

1. a value (KV) indicating the time status of the momentary database of the client is transmitted from the client to the server;
2. a part of the source database of the server, which has a more recent time status than the time status of the momentary database of the client indicated by the value (KV), is determined;
3. the determined part of the source database is transmitted to the client;
4. the part of the source database, which has been transmitted from the server, is received in the database of the client so as to form an updated database of the client.

Pseudo-Code

Method for adapting a database provided on a client to a source database stored on a server.

1. `KV = timestamp(client_database) // must be done on the client`
2. `part = identify_newer_parts(KV, server_database) // must be done on the server`
3. `send_to_client(part) // client <-> server`
4. `update_client_database(part) // must be done on the client`

Summarize With Own Words (the point of novelty)

This *invention* compares server data to client data using timestamps and updates the client data with any newer data from the server.

Identify Related Projects

Try to find out which kinds of systems might use similar methods

ftp-clients, http-clients, gopher-clients, ways-clients, mail-clients, irc-clients, version control systems, instant-messengers, but also each application that can just copy files in combination with network-filestems and also systems that can open files from network-drives or backup-systems

Create a List With All Programs/Projects That Might Use Similar Methods

Actually I just **try to find categories of software** here. I'll select specific projects later.

ftp-clients

http-clients

gopher-clients

ways-clients

mail-clients

irc-clients

version control systems

instant-messengers

icq, aim and msn and many many more

applications that can just copy files in combination with network-filestems

MS-Explorer, cp, copy, mv, move, xcopy and many many more

programs that can open files from network-drives and save those on a local drive

cat, Gimp, Word, OpenOffice, type and many many more

backup programs

mirror software

Select Those That You Expect to Fit Best

To rate my expectation of a successful Prior Art I use marks from 1 (very good chance) to 10 (no chance). I made heavy use of Google and Wikipedia (searching for 'ftp client list' etc.)

ftp-clients

- chance 1 (I know that some have options to 'overwrite only older versions')

<http://de.wikipedia.org/wiki/Kategorie:FTP-Client> - A list of FTP-Clients on Wikipedia

http-clients

- chance 3 (I'm not sure if these actually compare timestamps -- maybe for caching)

<http://www.blooberry.com/indexdot/history/browsers.htm> - A Browser-Timeline

gopher-clients

- chance 5 (I don't know anything about those, but they are really old and might be used for older patents than this one)

<http://www.yale.edu/pclt/WINWORLD/GOPHER.HTM> - Gopher Clients

ways-clients

- chance 9

I'm actually not even sure what those did, I just remembered em from the old days (well I'm only 26 yet ;-)

mail-clients

- chance 2 (Many mail-clients have the ability to leave messages on the server and only download those that are new)

http://en.wikipedia.org/wiki/List_of_e-mail_clients - Long list of many different E-Mail clients

irc-clients

- chance 5 (I know they can transfer files, but probably they won't compare timestamps)

http://en.wikipedia.org/wiki/List_of_IRC_clients

version control systems

- chance 1 (All of those use timestamps in some way and there are hell a lot of those out there)

http://en.wikipedia.org/wiki/Revision_control

http://en.wikipedia.org/wiki/List_of_revision_control_software

http://en.wikipedia.org/wiki/Source_Code_Control_System

instant-messengers

- chance 5 (same as on irc-clients)

icq, aim and msn and many many more

applications that can just copy files in combination with network-filestems

- chance 2 (i know that e.g. xcopy can use archive-bits, probably some of them can use timestamps)

MS-Explorer, cp, copy, mv, move, xcopy and many many more

programs that can open files from network-drives and save those on a local drive

- chance 5 (some of those warn you if you overwrite a file, maybe do some check if the file is newer ..)

cat, Gimp, Word, OpenOffice, type and many many more

backup programs

- chance 3 (most will be able to do backups newer than some date, but i don't know any backup-program that does backups from one computer to another but i'm sure there are some)

mirror software

- chance 1 (as e.g. sourceforge.net uses for it's mirrors, they definitely use timestamps)

<http://www.lyra.org/sitecopy/>

<http://lftp.yar.ru/> - this is actually an ftp-client, but it has also mirror-capabilities

installation software

- chance 3 (I know that many say 'This system already has a newer DLL, do you want to overwrite?)

Many Microsoft Windows Installation Routines, I'm sure many did it even back on Win 3.1

Determine in a first step from which date those (best rated) programs are

ftp-clients - chance 1

<http://de.wikipedia.org/wiki/Kategorie:FTP-Client> - A list of FTP-Clients on Wikipedia

<http://curl.haxx.se>

first publication **20. March 1998** as cURL 4 (cited (<http://de.wikipedia.org/wiki/CURL>))

<http://www.gpsoft.com.au>

Since **1994** it has been developed and published by the Australian-based GP Software. cited

(http://en.wikipedia.org/wiki/Directory_Opus)

<http://www.ftpvoyager.com/>

Rhino Software, Inc., in **1997** as one of the first FTP clients using a graphical user interface. cited

(http://en.wikipedia.org/wiki/FTP_Voyager)

<http://filezilla.sourceforge.net/>

Registered : **2001-02-27 00:04** cited (<http://sourceforge.net/projects/filezilla>)

<http://en.wikipedia.org/wiki/FlashFXP>

no publication date found

http://en.wikipedia.org/wiki/Internet_Explorer

Internet Explorer 3 was the first major browser with CSS support released in **August, 1996** cited

(http://en.wikipedia.org/wiki/Internet_Explorer#History)

<http://sourceforge.net/projects/kbear>

Registered **2001-01-05 12:38** cited (<http://sourceforge.net/projects/kbear>)

<http://konqueror.kde.org/>

List kfm-devel, Subject: Re: Gotta get this thing to work **1997-05-28 20:16:17** cited

(<http://lists.kde.org/?l=kfm-devel&r=1&w=2>)

<http://krusader.sourceforge.net/>

Das Projekt Krusader wurde im **Mai 2000** von Shie Erlich und Rafi Yanai begründet, cited

(<http://de.wikipedia.org/wiki/Krusader>)

<http://www.smartftp.com/>

no first publication date found, seems to be pretty jung though

<http://www.speedproject.de/speedcommander/index.html>

no first publication date found, seems to be pretty jung though

<http://www.ghisler.com/>

Die erste öffentliche Version des Programms erschien am **29. September 1993**. cited

(http://de.wikipedia.org/wiki/Total_Commander)

<http://www.gnu.org/software/wget/wget.html>

Geturl 1.0, released **January 1996**, was the first publically available release. cited

(<http://en.wikipedia.org/wiki/Wget#History>)

```
This list of ftp-clients took me about half an hour:
```

```
# 13:55, 15. Mär 2006 Kechel (→ftp-clients)
# 13:53, 15. Mär 2006 Kechel (→ftp-clients)
# 13:28, 15. Mär 2006 Kechel (→Determine in a first step from which date those (best rated) programs are)
# 13:27, 15. Mär 2006 Kechel (→Determine in a first step from which date those (best rated) programs are)
```

version control systems - chance 1

http://en.wikipedia.org/wiki/Revision_control

http://en.wikipedia.org/wiki/List_of_revision_control_software

http://en.wikipedia.org/wiki/Source_Code_Control_System

As above links contain hundreds of Links I only take one example which (hopefully) can be used as Prior Art, but which is really very old:

SCCS (Source Code Control System) - <http://en.wikipedia.org/wiki/SCCS>

It was originally developed at Bell Labs in **1972** by Marc Rochkind for an IBM System/370 computer running OS/MVT. (cited (<http://en.wikipedia.org/wiki/SCCS>))

M. J. Rochkind: The Source Code Control System (<http://basepath.com/aup/talks/SCCS-Slideshow.pdf>) . In *IEEE Transactions on Software Engineering* SE-1:4 (**Dec. 1975**), pages 364–370.

- This Paper also contains some references which might also be useful as Prior Art, remember to come back to those if we can't find anything here
- CiteSeer is very helpful in finding Documents that refer to this one, so remember to come back to those if we can't find anything here: <http://citeseer.ist.psu.edu/context/7732/0>

mirror software - chance 1

<http://www.lyra.org/sitecopy/>

Version 0.1, **September 23rd, 1998** First release. (cited (<http://www.lyra.org/sitecopy/history.html>))

<http://lftp.yar.ru/> - this is actually an ftp-client, but it has also mirror-capabilities

Version 0.10.7 - **1997-03-09** (cited (<http://lftp.yar.ru/NEWS>))

Verify selected Projects

We now have 13 Projects that were (yet only probably) published prior to the Priority Date and which might be used as Prior Art.

I just start with one of the remaining projects I really like. But it actually doesn't matter at all, you will be finished as soon as you found **one single** Prior Art and you *must not* finish without a single Prior Art before you verified **all** of them!

Verify SCCS

Is the description older than the Priority-Date?

This is fairly easy here, because we found this *tiny nice and shiny* paper that was actually printed in 1975: The Source Code Control System (<http://basepath.com/aup/talks/SCCS-Slideshow.pdf>)

As I couldn't find any digital text of this document, I'm going to transcribe interesting passages:

Page 1

Abstract - The Source Code Control System (SCCS) is a software tool designed to help programming projects control changes to source code. It provides facilities for storing, updating, and retrieving all versions of modules, for controlling updating privileges, for identifying load modules by version number, and for recording who made each software change, when and where it was made, and why. This paper discusses the SCCS approach to source code control, shows how it is used and explains how it is implemented.

[...]

Identification: The system automatically stamps load modules with information such as version number, date, time, etc. The source code that was used to make the load module may later be retrieved from this information alone.

[...]

Each time the module is changed ([...]) the change is stored as a discrete *delta*.

[...]

Each delta is applied to the source code as it existed just prior to that delta. Similarly, the source code as it was just before the last change is accessed by applying only the first three deltas, and so on.

Page 3

[...]

SCCS incorporates the delta documentation into several reports. The most common is issued whenever a module is accessed. It is a chart giving, for each delta, the release, level, option letter (if any), date and time of

Page 4

creation, who added it, why it was added, whether it was applied in this particular access, and why it was or was not applied ([...]).

Does it really implement or describe the patent?

To answer this question we start with our own summarization of the patent, and work upwards until we compare it to the original claim:

verify the summarization

This *invention* compares Server-Data to Client-Data using timestamps and updates the Client-Data with newer data from the server.

I here noticed that I used *timestamp* in my summarization, but the original claim used *time status*. Yes, now we have to pay attention to such small differences. So I correct the summary as follows:

This *invention* compares Server-Data to Client-Data using time status and updates the Client-Data with newer data from the server.

I use the following notation for my verification:

patent citations (actually summary citations yet)

SCCS Paper citations

some comments

And here we go:

updates data

It provides facilities for storing, updating, and retrieving all versions of modules

this clearly describes about the same thing

compares data using time status

get modx causes SCCS to generate the source code corresponding to the very end of the delta chain.

As the end of the delta chain is definitely a time status (which is also written somewhere else in the paper) and `get modx` generates the appropriate file, it updates the Client-Data using a time status.

Attention: So far we don't have hints about using Clients and Servers here.

conclusion

So far, knowing about SCCS my summarization makes the invention obvious if using Clients and Servers is obvious too.

verify the original claim

Disclosed is a method for adapting a database provided on a client data processing unit to a source database stored on a server data processing unit. In order to further develop said method in such a way that only comparatively small amounts of data have to be transmitted, the following steps are carried out: - a comparative characteristic value (KV) indicating the time status of the momentary database of the client data processing unit (6a) is transmitted from the client data processing unit (6a) to the server data processing unit (4); - a part of the source database of the server data processing unit (4), which has a more recent time status than the time status of the momentary database of the client data processing unit (6a) indicated by the comparative characteristic value (KV), is determined; - the determined part of the source database is transmitted to the client data processing unit (6a); the part of the source database, which has been transmitted from the server data processing unit (4), is received in the database of the client data processing unit (6a) so as to form an updated database of the client data processing unit (6a).

Disclosed is a method for adapting a database provided on a client data processing unit to a source database stored on a server data processing unit.

It provides facilities for storing, updating, and retrieving

In order to further develop said method in such a way that only comparatively small amounts of data have to be transmitted, the following steps are carried out

accessing algorithm that allowed many deltas to be kept at a reasonable cost in terms of disk space and processing time. SCCS meets both of these criteria. The space required to store a delta is only slightly greater than the amount of text inserted by that delta.

So we even have the same goals here

- a comparative characteristic value (KV) indicating the time status of the momentary database of the client data processing unit (6a) is transmitted from the client data processing unit (6a) to the server data processing unit (4); with information such as version number, date, time, etc. The source code that was used to make the load module may later be retrieved from this information alone.

This says that the number, date, time can be used to retrieve the information. This includes sending this data to the server if a server is used.

- a part of the source database of the server data processing unit (4), which has a more recent time status than the time status of the momentary database of the client data processing unit (6a) indicated by the comparative characteristic value (KV), is determined;

Each delta is applied to the source code as it existed just prior to that delta.

- the determined part of the source database is transmitted to the client data processing unit (6a);

Each delta is applied to the source code

This includes transmitting, there is no other way of doing this (with and without using a server)

the part of the source database, which has been transmitted from the server data processing unit (4), is received in the database of the client data processing unit (6a) so as to form an updated database of the client data processing unit (6a).

Each delta is applied to the source code

This is the update of the client database

conclusion

If you know SCCS, then the invention is obvious if using Clients and Servers is obvious too.

This might be enough to proof for Prior Art, it might not.

In this case it is good to look for other Prior Arts.

Taking into account that SCCS got really very close, I suggest continuing with work that is derived from SCCS, such as RCS:

RCS

Revision Control System (http://de.wikipedia.org/wiki/Revision_Control_System) tells us, that *later it was replaced by CVS* (http://en.wikipedia.org/wiki/Concurrent_Versions_System), which firstly enhanced RCS with networking capabilities

CVS

So we should check if CVS is also prior the Priority Date, because then we surely have Prior Art for the Client/Server part.

Verify CVS

We know that CVS is based upon RCS, which is based upon SCCS and that CVS enhanced RCS with networking capabilities. So i give this a chance of 1+. We'll see how it works out:

Is CVS older than the Priority-Date?

Publication: *The code was publicly released to mod.sources on June 23, 1986* (cited (http://en.wikipedia.org/wiki/Concurrent_Versions_System) / original usenet post (http://groups.google.com/group/mod.sources/msg/2ebab72ac0744fb8?:mod.sources.*=&hl=en&lr=lang_en&ie=UTF-8&...))

I also found a book about CVS prior to the Priority Date (using <http://amazon.com>):

Open Source Development with CVS
ISBN: 1576104907, year **2000**

Another good ressource to look for Prior Art are Universities:

- University Munich **WS 02/03** “Programmierpraktikum”, CVS is part of the lecture
http://www.dbs.informatik.uni-muenchen.de/Lehre/Programmierpraktikum/Skript/ProgPraktWS02_1.pdf
- University Oldenburg, **WS 01/02** “Modul Softwareprojekt incl. Proseminar”:
www-is.informatik.uni-oldenburg.de/~sauer/lehre/swp_01_02/swp_01_02.html
This lecture contains a 174 pages description to CVS 1.10.7 and has a copyright from **1992/1993**:
http://www-is.informatik.uni-oldenburg.de/~sauer/lehre/swp_01_02/cvs.pdf
- University Karlsruhe, **WS 01/02** “Softwareengineering“, contains slides on CVS und RCS
http://www.aifb.uni-karlsruhe.de/Lehrangebot/Winter2001-02/SwEng/PPFolien/Folien_SE09.ppt
- University Bonn, **SS 99** “Softwaretechnologie“, contains in homework 3 extra homework on CVS, especially for the commands checkout, update und commit
<http://www.informatik.uni-bonn.de/III/lehre/vorlesungen/SWT/SS99/>

So I see it as defenitely proofed, that CVS was published prior the Priority Date of the Invention.

And if you find sth. like *Der „MetaCodeGenerator“ arbeitet mit allen markt-üblichen CM-Systemen wie Visual SourceSafe, ClearCase, CVS,*, where the patentee itself says that CVS is one of the major source control systems and which was published prior to the Priority Date, then it even starts to make fun searching for Prior Art ;-) (link to prospect (http://www.industry.siemens.de/isitps/Lsg_images/LsgGrafiken/467d_MetaCodeGenerator_pdf.pdf))

Does CVS really implement or describe the patent?

verify the summarization

This *invention* compares Server-Data to Client-Data using time status and updates the Client-Data with newer data from the server.

[...]

verify the original claim

... and so on.

Conclusion

I hope this gave you an overview on how such a search for Prior Art can be accomplished!

This example was made by Kechel 18:30, 16 March 2006 (PST)

Accordingly, a "closest prior art" standard will work for now.
Once we get everything up and running, we can address the nuances,
if there is a need.

changelog

31-MAR-2006

- Author: Jan O. Kechel
- Correctors: Paul E. McKenney
- License: GNU Free Documentation License (<http://www.gnu.org/copyleft/fdl.html>)
- <http://prevalent.de/priorart/example-prior-art-search-060331.ps>
- <http://prevalent.de/priorart/example-prior-art-search-060331.pdf>

19-MAR-2006

- Author: Jan O. Kechel
- Correctors: Paul E. McKenney
- License: GNU Free Documentation License (<http://www.gnu.org/copyleft/fdl.html>)
- <http://prevalent.de/priorart/example-prior-art-search-060319.ps>
- <http://prevalent.de/priorart/example-prior-art-search-060319.pdf>

<< back

Von "http://developer.osdl.org/dev/priorart/wiki/index.php/Example_Prior_Art_search"

Seitenkategorien: Use already existing publications as Prior Art

-
- Diese Seite wurde zuletzt geändert um 11:49, 31. Mär 2006.
 - Content is available under Attribution-ShareAlike 2.5.