How to find a Software Patent?

Jan A. Bergstra♣ Paul Klint♠

♣ *Informatics Institute, University of Amsterdam*
and
Faculty of Philosophy, University of Utrecht
`www.science.uva.nl/~janb`

♠ *Centrum voor Wiskunde en Informatica (CWI), Software Engineering Department*
and
Informatics Institute, University of Amsterdam
`www.cwi.nl/~paulk`

DRAFT May 27, 2007

Abstract

Many software patents have become known by simply reading the patents in the patent databases. This is, however, a Gargantuan task and a more systematic and automatic method is needed when performing large scale legal and economic studies. Usually, software patents are identified as those patents that satisfy a simple query on the title and the specification of the patent itself. Typically, the query includes words like “software”, “computer”, and “program” but excludes words like “chip”, “semiconductor” and some others.

How does a software engineering judge the software content of patents selected via this method? In order to answer this question we studied a random sample of such “software patents” and come to the conclusion that only a small minority of them is really describing a software-related invention.

1 Background

As part of a three year European Commission (EC) study[1] on the effects of software patents on innovation we are involved in a multi-disciplinary effort to understand the effects of software patents. These effects are studied from legal, economical, and computer science perspectives.

In previous papers we have studied trivial patents [1] and software inventions [2]. In the present paper we discuss the quality of automatic searches for software patents.

In a much-cited study by Bessen and Hunt [3] software patents are selected by the following query:

```
((‘‘software’’ in specification) OR
 (‘‘computer’’ AND ‘‘program’’ in specification))
AND NOT (‘‘chip’’ OR ‘‘semiconductor’’ OR ‘‘bus’’ OR
        ‘‘circuit’’ OR ‘‘circuitry’’ in title)
AND NOT (‘‘antigen’’ OR ‘‘antigenic’’ OR
        ‘‘chromatography’’ in specification)
```

Since then, various refinements to this selection procedure have been proposed and many economic studies have been and are being based on comparable search methods. Here we want to address the question: are the patents selected via such methods describing software-related inventions?

[1] *Study of the effects of allowing patent claims for computer-implemented inventions*, a joint study by MERIT (University of Maastricht, Netherlands), Centre of Intellectual Property Law CIER (University of Utrecht, Netherlands), Centrum voor Wiskunde en Informatica (Amsterdam, Netherlands), Telecommunication Engineering School at the Universidad Politécnica de Madrid (UPM), Spain and Centre for Research on Innovation and Internationalization (CESPRI) at Bocconi University, Milan, Italy.

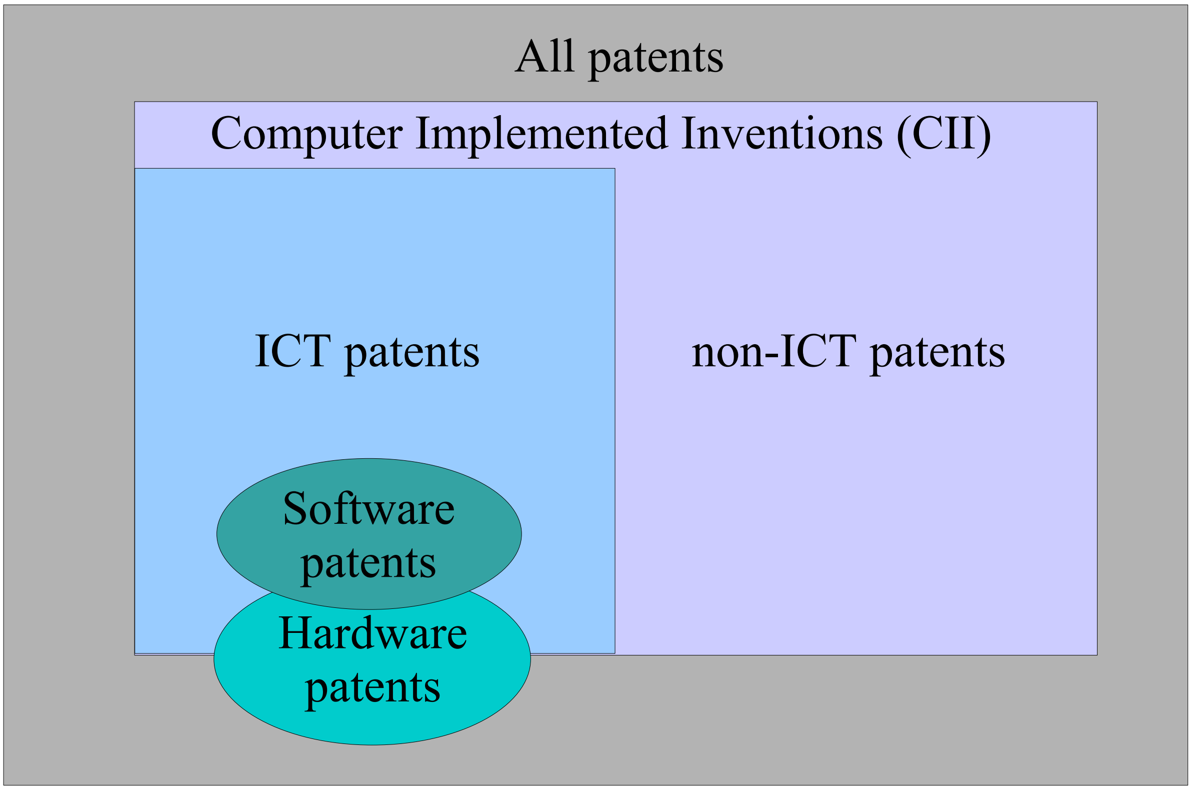

Figure 1: Inclusion relation between patent classes

2 Patent classes

The subject matter in patent specifications can vary widely over nearly all areas of science and technology. Since we are primarily interested in "software patents", it is worthwhile to position software patents in the enclosing space of all patents. This is sketched in Figure 1.

We start with all patents (the outermost rectangle) which includes the class of Computer Implemented Inventions (CII), a phrase coined by the EC. A brochure of the European Patent Office (EPO) defines an CII "as an invention that works by using a computer, a computer network or other programmable apparatus. To qualify, the invention also needs to have one or more features which are realised wholly or partly by means of a computer program" [5]. We are aware of the fact that taking this definition as point of departure makes our findings very EC-centric. However, we prefer to apply the EC terminology consistently, rather than now embarking on an attempt to use more international terminology.

Within CII two subclasses can be distinguished: *ICT patent*s that deal with ICT-related subjects like making backups, data compression, concurrency control and *non-ICT patents* that deal with other subjects like cutting textile or cleaning furnaces. An ICT patent describes an ICT invention and a non-ICT patent only *uses* ICT to achieve an invention in a non-ICT area.

Two further subclasses can be distinguished: *software patents* and *hardware patents*. Software patents are strictly included in ICT patents while pure hardware patents fall outside CII. A software patent describes a software invention and a hardware patent describes a hardware invention. The most complex ICT patents involve both software and hardware inventions.

Although much debate is possible (and necessary!) about the precise definition and inclusion relations between these patent classes, one thing is clear: all Computer Implemented Inventions use software but probably only minority will describe a software invention (also see [2] for a characterization of software inventions).

3 The sample

The sample of patents has been selected by a method described in [4] which is a refinement of the original method in [3]. The result is shown in Table 1 and consists of 32 EU and US patents amounting to over 600 pages of patent descriptions.

The table contains the following columns:

- **Id**: An identifier (used for cross-referencing with later tables).
- **Patent**: A patent number.
- **Area**: An area indication (EU or US).
- **Publ**: Publication date of the patent.
- **App**: Application date of the patent.
- **Company**: The company that has filed the patent application.
- **Topic**: A short description of the topic of the patent.

4 Method

All patents in the sample have been read and classified independently by both authors. For each patent the following aspects have been recorded:

- **P**: Number of pages of the patent.
- **A**: Patent describes an explicit application of Information and Communication Technology (ICT). The application may consist of a mixture of physical hardware, electronics, or software. It may consist of separate components that are connected via a local bus or a local or wide area network. Typically, the patent sketches an application domain and describes how ICT can be applied to solve a specific problem in that domain.
- **M**: Patent describes an ICT method that can be applied in different application domains.
- **E**: Patent describes electronic circuits. Typically, the patent describes an electronic device and contains diagrams of electronic circuits.
- **H**: Patent describes other (non-electronic) hardware. Typically, the patents contains drawings of physical (non-electronic) devices or constructions.
- **N**: Patent describes a network of physically separated computers or components. The described invention uses a local or wide area network to connect certain components.
- **AR**: Patent describes a system architecture. A verbal description or a diagram describe the overall structure of a system and its components.
- **F**: Patent contains flow charts. A flow chart is a graphical notation to describe consecutive steps and choices in an algorithm and describes some algorithmic aspect of the invention.
- **C**: Patent contains source code. A patent may contain a complete listing of an implementation of the invention in some programming language like C or assembly language. It is more common that a patent only contains fragments of source code that highlight certain aspects of the implementation.
- **MA**: Patent gives a mathematical description of an algorithm. This means an unambiguous, mathematical, description of some computational method. The notation and precision correspond with the conventions used in text books about data structures and algorithms.

Id	Patent	Region	Publ.	App.	Company	Topic
1	613302	EU	2000	1994	Philips	Video decoder
2	624495	EU	1997	1994	VALEO	Lighting/signalling on motor vehicle
3	632622	EU	1998	1994	France Telecom	Cell transmission
4	650122	EU	1998	1994	France Telecom	Backup method
5	676418	EU	2000	1995	BP	Polymerisation
6	694242	EU	1999	1994	Thomson	Selecting video programs
7	705006	EU	1999	1994	Siemens	ATM communication
8	706333	EU	2004	1994	Lectra	Sizing of garments
9	711221	EU	1997	1994	Bayer	Polycarbide in rolling mill
10	743579	EU	1998	1996	Siemens	Machine tool control
11	754389	EU	2000	1996	Philips	Video mosaic and teletext
12	766863	EU	2002	1996	Thomson	Reading magnetic or optic information
13	813701	EU	1999	1996	Siemens	Control system for processing plant
14	859979	EU	2000	1996	Wincor Nixdorf	Energy saving
15	922254	EU	2000	1997	Siemens	Error recognition in storage system
16	630846	EU	1998	1994	Savio	Bobine winding
17	651313	EU	1995	1994	SEP	Backup/restore method
18	672998	EU	2000	1995	Interflex	ID cards
19	701513	EU	1997	1994	Sartem	Installation of train contact lines
20	857228	EU	2000	1996	Jimtex	Textile fabrication
21	891588	EU	2001	1997	Imagination Techn.	Multi-media data processing
22	915795	EU	2001	1997	TRW Lucas	Gear boxes
23	974137	EU	2001	1998	Webraska	Vehicle navigation
24	632625	US	1996	1993	IBM	Packet transmission
25	694856	US	1996	1994	IBM	Serial connection of neuron circuits
26	738356	US	1998	1995	American Standard	Automatic water tap
27	743602	US	2002	1995	HP	Function usage control for security
28	755442	US	2002	1995	Genencor	Modified lipases in detergents
29	784813	US	1998	1994	HP	User-interface technique
30	831477	US	2001	1996	HP	Data storage technique for tape
31	839351	US	2001	1996	Novell	Synchronization of disconnectable computer
32	946677	US	2002	1997	Dow Chemical	Control of cracking furnaces

Table 1: Summary of sample patents

Abbreviation	**Meaning**
P	Number of **P**ages
A	ICT **A**pplication
M	ICT **M**ethod
E	**E**lectronic circuits
H	**H**ardware
N	**N**etwork
AR	System **AR**chitecture
F	**F**low charts
C	Source **C**ode
MA	**M**athematical **A**lgorithm
VA	**V**erbal **A**lgorithm
PR	**P**u**R**e software
EMB	**EMB**edded software
CII	**C**omputer **I**mplemented **I**nvention
SwI	**S**oft**w**are **I**nvention

Table 2: List of abbreviations used in Table 3

- **VA**: Patent gives a verbal description of an algorithm. This means a natural language description of the steps involved in a computational method. The readability may be higher, but the precision is less than in the previous case.
- **PR**: Pure software. The described invention is only about software and does not involve and other physical or electronic device.
- **EMB**: Embedded software. The patent describes software that is intended to be part of a electronic or physical device and control the behavior of that device.
- **CII**: The patent describes a Computer Implemented Invention (see Section 2).
- **SwI**: The patent describes a pure software invention (see Section 2).

These aspects will be used in Table 3 in the next section. For easy reference, these aspects are summarized in Table 2.

5 Findings

Our detailed findings are presented in Table 3. See the next section for the limitations and validity of our findings. Tentatively, we can draw the following conclusions from them:

- The majority (20 out of 32) of the “software patents” in the sample are Computer Implemented Inventions.
- Only a small minority (3 out of 32) of the patents are true software patents.
- The majority of patents (2 out of 3) that contain source code are a software patent.
- All software patents contain a verbal description of an algorithm.
- All software patents were issued in the US.
- Most patents involve electronics or other hardware.
- Most patents contain a description of some system architecture.
- None of the patents contains a mathematical description of an algorithm.
- One third of the patents contains a verbal description of an algorithm.

Id	**P**	**A**	**M**	**E**	**H**	**N**	**AR**	**F**	**C**	**MA**	**VA**	**PR**	**EMB**	**CII**	**SwI**
1	9	Y	Y	Y										N	N
2	17	N	Y	Y	Y									N	N
3	18	Y	Y	Y										N	N
4	11	Y	Y			Y	Y				Y			Y	N
5	13	N	N								Y			N	N
6	12	N	Y		Y									N	N
7	8	Y	Y				Y							N	N
8	40	N	Y				Y				Y		Y	Y	N
9	13	N	N		Y		Y							N	N
10	8	N	Y		Y		Y						Y	N	N
11	11	Y	Y			Y								Y	N
12	22	Y	Y	Y	Y						Y			Y	N
13	20	Y	Y				Y							Y	N
14	6	Y	Y	Y		Y	Y	Y			Y			Y	N
15	15	Y	Y		Y		Y				Y			N	N
16	20	Y	N		Y		Y	Y						Y	N
17	6	Y	Y			Y	Y							Y	N
18	14	N	Y			Y		Y			Y			Y	N
19	14	Y	N	Y	Y									Y	N
20	17	Y	N		Y		Y							Y	N
21	18	Y	Y	Y			Y							Y	N
22	29	N	Y		Y		Y		Y	Y				Y	N
23	20	Y	Y			Y	Y	Y						Y	N
24	41	Y	Y	Y		Y	Y				Y			Y	**Y**
25	84	Y	Y	Y		Y	Y	Y						Y	N
26	7	N	Y	Y	Y		Y							N	N
27	11	Y	Y	Y			Y							Y	N
28	28	N	N								Y			N	N
29	8	Y	Y						Y		Y	Y	Y	Y	**Y**
30	13	Y	Y	Y	Y		Y				Y		Y	Y	N
31	43	Y	Y			Y	Y		Y		Y	Y		Y	**Y**
32	12	N	Y		Y		Y							N	N

Table 3: Summary of findings for sample patents

6 Discussion

We started this exercise with a list of aspects that was considerably smaller than the one presented in Section 4. However, it turned out that answering questions like “Is this a CII?” or “Is this a software patent?” is easier said than done. There is always a subjective judgment involved while trying to answer such a question.

Out of need we have expanded the list of aspects and hoped to find more objective classification criteria. For some aspects this is easy (**P**, **E**, **H**, **N**, **AR**, **F**, **C**, **MA**, **VA**) but for others this remains difficult (**A**, **M**, **PR**, **EMB**, **CII**, **SwI**). One should keep this in mind when interpreting our findings. As a consequence, the conclusions presented in the previous section are a preliminary indication that there might be a problem with the classification of true software patents. Certainly, more study is needed to get further insight this issue.

6.1 An improved assessment method

We can only speculate how to improve our assessment method. Since completing the research described in this paper we have proposed the Software Invention Cube (SWIC), see [2], that is intended to classify software inventions. This same methodology could be used for patent classification. Typically, aspects like:

- Software requirements.
- Software design.
- Software implementation.
- Software testing.
- Software capabilities.
- Software process.
- Software tools.
- Software deliverables.

should play a role during classification. An ideal research set-up would then be as follows:

- Identify the complete list of criteria to be used for classification.
- Explicitly define each criterion and give a list of operational indicators that signal a *low*, *medium*, or *high* value of that criterion.
- Select a sample of “software patents” according to the commonly used textual search methods.
- Let several human classifiers assess each patent according to the above criteria.
- A statistical analysis of the results may reveal which criteria most strongly correlate with true software patents.
- Based on the above, automation of the most promising criteria may be attempted.

6.2 Unanswered questions

Other questions that need further investigation are:

- We have used a sample of 32 patents. Do our results still hold for larger sample sizes?
- Which search methods can be used to automate the classification method sketched above?
- Our study has only looked at *false positives*: patents that were wrongly classified as software patent. What about *false negatives*: patents that are not classified as software patent according methods but do contain a software invention?

6.3 Outlook

Many quantitative studies investigate economic, financial or other aspects of software patents and rely on a solid identification method for such patents. The conclusions of the qualitative study presented in this paper challenge the validity of the commonly used query-based selection method for software patents: manual inspection of the selected patents reveals that only a small minority of them is really about software. The importance of this observation seems to warrant further study as sketched above.

Acknowledgments

Bronwyn Hall, Grid Thoma and Salvatore Torrisi kindly provided the sample of patents that is used in this paper.

References

[1] J.A. Bergstra and P. Klint. About ”trivial” software patents: The IsNot case. *Science of Computer Programming*, 16(3):264–285, 2007.

[2] J.A. Bergstra and P. Klint. The software invention cube: a classification scheme for software inventions. Submitted for Publication, 2007.

[3] J. Bessen and R.M. Hunt. An emperical look at software patents. Working Paper 03-17/R, Boston University School of Law, 2004. `http://www.researchoninnovation.org/swpat.pdf`.

[4] B.W. Hall, G. Thoma, and S. Torrisi. The market value of patents and r&d: Evidence from european firsm. Technical Report WP n.186, Centre for Research and Internationalization (CESPRI), Bocconi University, October 2006.

[5] European Patent Office. Computer implemented inventions and patents. Brochure. `http://cii.european-patent-office.org/_pdf/cii_brochure_en.pdf`.