

Gruselkabinett der Europäischen Softwarepatente

<http://swpat.ffii.org/patente/index.de.html>

Arbeitsgruppe

swpatag@ffii.org

deutsche Version 2003/11/15 von FFII*

2005-01-14

Eine Datenbank der Monopole auf Programmieraufgaben, die das Europäischen Patentamt gegen den Buchstaben und Geist der geltenden Gesetze massenweise gewährt hat, und über die es die Öffentlichkeit nur häppchenweise informiert. Die Softwarepatent-Arbeitsgruppe des FFII versucht, die Softwarepatente herauszusuchen, besser zugänglich zu machen, und ihre Wirkungen auf die Softwareentwicklung zu zeigen.

Inhaltsverzeichnis

1 Übersicht und Beispiele	1
2 Wie man Patentschriften liest	3
3 Weitere Lektüre	3

1 Übersicht und Beispiele

- Tabellen und Zahlen zu den Europäischen Softwarepatenten¹

*<http://lists.ffii.org/mailman/listinfo/traduk>

¹<http://localhost/swpat/patente/zahlen/index.de.html>

Wir wollen die Patente auf Probleme im Bereich der Programmierung von Datenverarbeitungsanlagen (d.h. des Neumannschen Universalrechners in seiner abstraktesten Form) auflisten und klassifizieren, die das Europäische Patentamt gegen den Buchstaben und Geist der geltenden Gesetze erteilt hat. Aus unten erläuterten Gründen ist es schwierig, die genaue Zahl herauszufinden, aber es dürften mehr als 30.000 sein.

- **Europäische Softwarepatente: Umfassende Dokumentation²**

In den letzten Jahren haben das Europäische Patentamt (EPA) und nationale Patentämter zahlreiche Patente auf computerimplementierte Organisations- und Rechenregeln, d.h. Programme für Datenverarbeitungsanlagen [als solche] erteilt. Wir sammeln diese Patente und machen sie zugänglich.

- **Europäische Softwarepatente: Einige Mustereispiele³**

Auf folgende eingängige Beispiele stießen wir beim ersten Durchstöbern unserer Softwarepatente-Tabellen. Sie wurden fast zufällig ausgesucht und können somit als repräsentativ für die Maßstäbe des Europäischen Patentamtes hinsichtlich Technizität und Erfindungshöhe gelten. Wenn sie in irgendeinem Punkt vom Mittelmaß abweichen, dann darin, dass sie relativ leicht einem breiten Publikum verständlich gemacht werden können.

- **Softwarepatente in Aktion⁴**

In den letzten Jahren sind einige Streitfälle um Softwarepatente durch die Medien bekannt geworden. Es handelt sich hierbei um die Spitze des Eisbergs. Die meisten Entwickler und Firmen werden nur außerhalb der Gerichte mit Patentforderungen konfrontiert, und Schweigen liegt im beiderseitigen Interesse. Viele Projekte werden zurückgeschraubt oder aufgegeben. Es ist schwer, verhinderte Entwicklungen zu dokumentieren. Hier wollen wir diesen Versuch unternehmen.

²<http://localhost/swpat/patente/txt/index.de.html>

³<http://localhost/swpat/patente/muster/index.de.html>

⁴<http://localhost/swpat/patente/wirkungen/index.de.html>

2 Wie man Patentschriften liest

Eine Patentschrift besteht aus

- Anspruchsbereich
- Beschreibung

In den Ansprüchen steht zu lesen, was Sie nicht tun dürfen. Jeder Anspruch beschreibt eine Klasse verbotener Gegenstände. Die Beschreibung hilft bei der Auslegung der Ansprüche. Sie soll *den durchschnittlichen Fachmann befähigen*, die “Erfindung” nachzuarbeiten, ohne weiterhin erfinderisch tätig werden zu müssen. Die EPA-Beschreibungen enthalten jedoch im allgemeinen keine Software-Referenzimplementation, und die eigentlich schwierige Arbeit bleibt dem Programmierer überlassen. Daher könnte man auch vom Gesichtspunkt der *Befähigung des Fachmanns* her gesehen die Gültigkeit der EPA-Softwarepatente in Zweifel ziehen.

In Patentbeschreibungen ist häufig wortreich von Selbstverständlichkeiten wie der “Zuweisung eines Blocks in einem Speichergerät zur Speicherung einer Variablen” u.ä. die Rede. Dies lässt die “Erfindung” besonders “technisch” aussehen, kann aber auch notwendig sein, um Verteidigungslinien für Rückzugsgefechte im Fall von Rechtsstreit aufzubauen. In jedem Falle muss der weniger juristisch interessierte Leser lernen, dieses Wortgeklingel zu überlesen.

3 Weitere Lektüre

- **Richard Stallman – The Anatomy of a Trivial Patent**⁵

Der Artikel zeigt anschaulich, wie man das Wortgeklingel in Softwarepatentschriften überliest und sie im Geiste auf das wesentliche reduziert. Allerdings führt Stallmans Herangehensweise nicht zum Verständnis der juristischen Zusammenhänge, aus denen heraus eine Patentschrift entsteht.

- **Kiesewetter-Köbinger 2000: Über die Patentprüfung von Programmen für Datenverarbeitungsanlagen**⁶

Probleme und Ungereimtheiten der Softwarepatentierung aus der Sicht eines Prüfers am Deutschen Patent- und Markenamt – zeigt sehr scharfsinnig warum Softwarepatente so sind wie sie sind und welche weiteren Gefahren eine Änderung des Art 52 bringen würde.

⁵http://linuxtoday.com/news_story.php3?ltsn=2000-05-26-004-04-0P-LF

⁶<http://localhost/swpat/papiere/grur-skk01/index.de.html>

- **Warum sind Softwarepatente so trivial?**⁷

Die meisten Softwarepatente sind trivial und breit. Die Ursachen hierfür ist im Patentsystem selbst und nicht etwa in Unzulänglichkeiten bei der Patentprüfung zu suchen. Einerseits lässt das Erfordernis der Erfindungshöhe (Nichtnaheliegen) schwer fassen, und bisherige Versuche einer Formalisierung dieses Erfordernisses haben lediglich zu seiner tendenziellen Abschaffung beigetragen. Andererseits haben die Gerichte mit der Einführung von Softwarepatenten den Marsch in die totale Trivialität freigegeben, indem sie das einzige verbleibende Bollwerk niederrissen: das Erfordernis der Technizität.

- **BountyQuest: We Offer Rewards For Knowledge**⁸

Ein amerikanisches Patent-Gruselkabinett, das Preise für diejenigen ausschreibt, die ärgerlich gewordene Patente mit neuheitsschädigenden Dokumenten zu Fall bringen.

⁷<http://localhost/swpat/stidi/frili/index.de.html>

⁸<http://www.bountyquest.com>