PROYECTOS Y LEGISLACIÓN

PATENTES SOFTWARE

¿POR QUÉ NO DEBERÍAN CONCEDERSE PATENTES SOBRE EL SOFTWARE?

- Carlos Aguilar de la Morena
- Ángela Martina Padrón Pol
- José Tomás Rodríguez Díaz

Grado en Ingeniería de Software
2017, Universidad de Málaga

Introducción

Una patente se define por la RAE como el "*documento en que oficialmente se le reconoce a alguien una invención y los derechos que de ella se derivan*". Es común confundir este término con la propiedad intelectual y el copyright. La propiedad intelectual referencia a los derechos sobre la obra que tiene el autor original, aparecen desde el momento que se crea la obra; mientras que el copyright se refiere a sus derechos de explotación y aparecen a raíz de la inscripción de dicha obra en un registro. Por tanto y para que no nos lleve a confusión, podemos decir que la propiedad intelectual y las patentes son diferentes, pertenecen a ramas y leyes distintas y no se pueden aunar en un mismo saco. Además son concedidas por entidades diferentes.

Actualmente en España no se considera la existencia de una patente de software per se, de hecho el uso de esta expresión es erróneo. En realidad, lo que se puede patentar son las invenciones implementadas en ordenador, cualquiera que implique el uso de ordenadores, redes informáticas u otros aparatos programables para los cuales un programa de ordenador realiza alguna función. Por tanto, cuando nos referimos a patentes de software hablamos de la trasposición al software de la patente tal y como la legislación la define, con sus ventajas e inconvenientes.

¿Por qué no deben concederse patentes sobre programas de ordenador?

Patentado

El hecho de poder patentar un programa de ordenador otorgaría al propietario el uso exclusivo de la idea tras el programa, dificultando así el desarrollo de programas debido a la falta de recursos comunes a todos ellos. Pilares tan básicos para un programa como las estructuras de datos, algoritmos de ordenación, patrones de arquitectura, diseño, etc. se convertirían en componentes patentables, impidiendo así su uso a posteriori por terceros. Es parte del bien común que las tecnologías puedan ser desarrolladas y usadas por terceros sin trabas para evitar un retraso en el desarrollo de la técnica.

Un problema que presentan este tipo de patentes es que lo patentado es una funcionalidad y la única manera de evitar incluir una funcionalidad patentada es no implementándola, lo cual puede privar al sistema de parte o la totalidad de su valor potencial. Por ejemplo: imaginemos que en una herramienta usada por millones de usuarios a diario, como Word, tuviese una funcionalidad patentada —como poner abreviaturas— y que el autor de dicha patente decidiese que no quiere que nadie use dicha funcionalidad y no quisiera vender lincencias; probablemente para muchas personas esta herramienta dejaría de ser útil o la usarían menos. Con este ejemplo no sólo vemos el problema de que se patente una funcionalidad, también el de que alguien que, por intereses propios, lo patente con el mero objetivo de dañar a la competencia sin querer sacarla al mercado, lo cual afecta directamente a los usuarios de los programas.

Desde la entrada del software en nuestro mundo, éste se ha diversificado muchísimo. Está evolucionando de manera continua. En parte, esta evolución constante hace que el software, como tal, avance y mejore. Por tanto, conceder una patente en este entorno podría implicar que durante 20 años el software con dicha funcionalidad no estuviese disponible o se vendiese en licencias con un valor elevado, dado que la tendencia en el mundo de las patentes es la rentabilización de las mismas. Los usuarios y pequeñas o medianas empresas podrían quedarse sin la opción de adquirirlo por no tener fondos. Además, al estar patentado tampoco tendrían la opción de implementarlo ellos mismos. Dado que el derecho de explotación de una patente en nuestra legislación dura 20 años, esto provocaría un desgaste en el desarrollo del software —como técnica— de igual magnitud. Con el patentado se puede haber privado a la sociedad de un desarrollo o mejora al no haberse podido usar dicha funcionalidad. Es decir, estamos evitando la mejora y evolución del software.

Otro problema de las patentes de software es que no solo imponen restricciones a los desarrolladores, sino que también se la imponen a cualquier usuario de ordenadores, ya que limitan lo que se puede hacer con un ordenador. Este problema no está presente en otras pantente, por ejemplo: las patentes de motores de coches solo afectan a los fabricantes de coches y las restricciones no se le imponen a las personas que lo compran.

Si bien las patentes surgieron con el objetivo de incentivar la actividad inventiva, en el caso del software —que podría considerarse como el conjunto de técnicas aplicadas para solventar un problema— realmente generan un cuello de botella que obstruye su desarrollo. Esto es debido a que el software tiene un fuerte componente colaborativo y de reutilización que no está presente en otras invenciones que constituyen objetos más mundanos.

Guerra de patentes

En otras legislaciones sí que está permitido el patentado del software. Si observamos un poco la historia reciente, hemos vivido auténticas guerras de empresas en las que se demandaban mutuamente con intención de mermar al contrario mientras seguían patentando y usando tecnología del contrario, muchas veces sin saberlo. Cuando son grandes empresas muchas veces quedan en un *quid pro quo*, pero una pequeña empresa podría no sobrevivir al proceso. Estos casos son sólo los sintomas que podríamos considerar más leves del patentado. Si se llegaran a patentar programas o funcionalidades software, la empresa que empezara apostando más fuerte por realizar patentes de los componentes más basicos del software podría conseguir, de facto, un monopolio durante 20 años en nuestro país. Por suerte, nuestra legislación actual sobre patentes prohibe patentar invenciones que ya sean de dominio público. No obstante, si contemplamos la historia del software, se puede observar la tendencia a que el software —y sobre todo la idea tras éste— que en un momento de la historia parece puntero, tienda a terminar siendo un componente de sistemas software más sofisticados, y así sucesivamente hasta quedar relegado al estatus de componente útil para el todo. Podría parecer que 20 años es un margen holgado para que eso ocurra, pero se ha observado que el desarrollo del software como técnica es exponencial. Un buen ejemplo es que hace 20 años era impensable el software de reconocimiento de voz del que disponemos hoy en día. Intentos tales como *via voice* —un famoso software de reconocimiento del habla, desarrollado por IBM en los 90— eran proyectos que, si bien eran considerados punteros, sus capacidades son una versión nímia de lo que podemos encontrar a día de hoy. Necesitaban entrenarse, tanto el software como el hablante, para poder usarlo correctamente. Por lo que eran *mono-usuario*. Pero más aún, lo impactante es que ni siquiera hace falta mirar 20 años adelante desde tal evento para ver software claramente superior en ese aspecto.

Patentes cruzadas

Recientemente, algunas de las grandes empresas han discutido sobre el uso cruzado de sus patentes. Esto ha frenado un poco la guerra de patentes, pero ha surgido la problemática de que las grandes empresas puedan decidir que las pequeñas empresas —sin patentes, o con patentes de poco interés para ellos— no tienen por que usar las suyas —en mucho caso críticas para ciertas partes del desarrollo— y, como consecuencia, podrían fracasar.

Conclusiones

- Las patentes de software evitan el desarrollo de la técnica. Un retraso de 20 años en la técnica evitaría que la sociedad pudiese disfrutar de mejoras en su calidad de vida.
- Limitación de los programas que no pueden incluir funciones patentadas. Condenándolos al fracaso.
- La intención de las patentes es estimular la creación de nuevas invenciones, pero esto no pasa en el software debido a su naturaleza colaborativa y reutilizable.
- Estimulan las guerras de patentes y favorece los monopolios. También dificultan la supervivencia de empresas pequeñas.