

A NO-IPR MODEL AS SOLUTION TO REUSE AND UNDERSTANDING OF INFORMATION SYSTEMS

Kai K. Kimppa
University of Turku, Finland

Abstract This work in progress paper argues that a reason why reuse of software components in information systems development is not more common and why users do not and cannot understand the systems they use partly depends on the implementation of the current IPR¹ system and the repercussions it has for the proprietary software model. A full blown anti-IPR system, modified from the GNU GPL and CC Attribution-ShareAlike licenses, is offered in its place. Furthermore, reasons why both reuse of software components and understanding of software would be enhanced are given. Also, some business models viable in that environment are tentatively suggested. Finally, how to—at least partially—implement this in the current IPR environment is explored.

Keywords: Information systems, information systems development, social informatics, work informatics, IPRs, free source software, GNU GPL, object-oriented development

1. Introduction

The potential of the object-oriented development approach has largely not come to fruition. The idea of reusing objects easily as modules for further development has only been realised within organisations, if even there. This results in reinventing the wheel again and again, even for very similar purposes. The reason for this is obvious: it is not a fault of the object-oriented programming paradigm, on the contrary, object-oriented development strongly supports the idea, but rather it is due to the Intellectual Property Rights (IPRs) system in use in previously Western societies but now globally due to World Intellectual Property Organization (WIPO) negotiations and World Trade Organization (WTO)/ TRIPS (trade-related aspects of intellectual property rights) treaties.

On top of this, as many (e.g., [1-3]) have proposed, the users of information systems typically do not understand how the information system works. The user interface does not use the language of the users, nor does it inform the users of the

¹ IPR is a loaded term which presupposes that immaterial things can be owned. In this paper immaterial is often used in place of IPR.

Please use the following format when citing this chapter:

Kimppa, K.K., 2008, in IFIP International Federation for Information Processing, Volume 282; *Social Dimensions of Information and Communication Technology Policy*; Chrisanthi Avgerou, Matthew L. Smith, Peter van den Besselaar; (Boston: Springer), pp. 319–325.

system logic which is used to give the users an understanding of the way the system works.

This further complicates the situation by transferring control of the information system from the user to an unidentifiable party, often the information system itself which then seemingly is in control [4-5]. This is visible through expressions such as “the system told me to do this”, which of course is true in a sense, but the users often actually do not know who or what is behind the requirements of the system.

At the organisational level this is visible through lack of user confidence, unnecessary work being done, and work being done wrong. Even though participatory design and other more user-centred design methodologies (such as SSM, socio-technical method(s), trade unionist approaches, etc., for a comparison of some of these see [6]) suggest heavy user participation in the design of the systems, the participation has not actualised except in isolated cases and where it has, the lower level understanding of the now typically wide variety of information systems in use remains impossible. There are various reasons for this, of course, one of which being the inability to actually view and modify the specifications and the code itself. It is clear, however, that if problems remain, a knowledgeable worker in an organisation could understand both the working system as well as the information system better than an external developer – and in the current situation this kind of modification remains extremely rare and hard to accomplish.

In work informatics research, an operationalisation of the social informatics research methodology (or methodologies) [7] made famous by Rob Kling², the main idea has been to use what works to solve the issues raised. One potential direction to take to solve some of the problems presented is adaptation of the no-IPRs model proposed by Kimppa [8], which would change the current information systems development situation dramatically.

The current proprietary model, based on the IPR structure in place can be seen as a cause of why the potential of reuse of software components is not happening. The current IPR model encourages closing source code and its parts from both the public and competition. This, naturally, leads to “inventing the wheel” again in a situation where someone other than the original creator of the system needs a similar part.

The current proprietary model does not encourage user participation. Users are not privy to the internal functioning of the information systems they use as the proprietary model keeps the source code a trade secret and thus deprive the users from both understanding of the system and the possibility to adapt it better to their and their colleagues use.

There are, of course, current alternatives, such as Free Libre Open Source Software (FLOSS) and Creative Commons (CC) Attribution ShareAlike models. Unfortunately they have not spread widely, and the current IPR systems do not

² See e.g., “Social Informatics: An Information Society for All? In Remembrance of Rob Kling” for the various ways in which social informatics has been developed [20].

encourage their use and at times even actively prohibit it. The situation is contradictory to their use, as copyright for any “literary” work such as software is the default position and the creator of software needs to actively seek to bypass this to enable free (or open) use of the software they create. Thus, the playing field is by default heavily tilted towards the proprietary model.

There are also plenty of examples of clearly using IPRs to prevent FLOSS or CC usage even in situations which previously would have been possible, typically due to the strengthening of copyright via methods such as DMCA (Digital Millennium Copyright Act), its European counterpart, DRM (Digital Rights Management) systems, etc. Some illustrative cases include Vivendi vs. BnetD (see e.g. [9]) where DCMA was used to stop reverse-engineering. Another example provided by Lessig [10] was the creation of the the Grey album by mixing the White and the Black albums.

2. A suggested solution

Kimppa [8] has proposed a no-IPRs model that would closely resemble the Free Software Foundation’s GNU GPL model (e.g., [11]) to completely replace the current IPR legislation. The idea is not to offer a new model, but to replace the current model with the default being no IPRs instead of strong protection. In the model any published material would be free to be used and further developed by any other, be it an organisation or an individual user. This does not mean that information systems would magically become available at no cost. Development of systems would still be necessary for various reasons. Of course, it would mean rethinking of current models of software business. The benefits and draw backs of the system would undoubtedly be various; some are presented in this work in progress paper.

2.1 *Benefits and draw backs from organisational perspective*

First, any modules written for any published information system would be reusable for any other system needing them. That would significantly cut back in the need to recode same or similar parts (e.g., [12]). The main benefit of the object-oriented thinking in software development could be fully utilised by extensive object libraries as well as larger parts of systems available for the basis of new or further developed systems. As good systems are designed to strengthen the business strategy of an organisation and is thus dependant on the aims and working practices of that particular organisation; almost no information system is directly applicable to another organisation. This helps in keeping the benefit from internal/out sourced development alive even in an environment where parts of systems could easily be copied or adapted to another organisation.

Management of information system projects would fall more squarely within the companies needing the software. This would mean inside development for integration combined with selecting from the existing what can be integrated to

the needed product. This would, however, mean more selection possibilities since there would be no need to take fully outside developed systems, which would translate to more choice than currently exist. This would also offer a possibility to integrate other modules than the ones offered by the proprietary company (i.e., any existing or any that are wanted and can be created either through buying from an outside supplier or through inside development). A possibility to tie information system development more firmly to the strategic goals of the company would also emerge.

CIOs (Central Information Officers) and IT Managers would be more capable of supporting the business strategy if they were able to make more concrete choices between modules they choose for the organisations information systems (for a call for CIOs to understand business strategy and be able to better support it see [13]). With the possibility to use previous modules and the need to understand them creating information systems which would support the general strategy of the organisation would be more important than the capability “to make the best deal” for the software obtained. New skills in the managerial portfolio of an information system development manager would become valuable and this would likely also change the education required for the position.

Cooperation instead of competition would become a more viable option if others could access the software in any case. Thus, the costs of implementing a system could be shared with others operating in the same field. This would be especially beneficial for SMEs (Small and Medium Sized Enterprises) which now are largely dependent on certain distributors for their internal software and cannot afford internal development.

Of course, for off-the shelf software making end user products this would mean hard times—at least with the current business models. Luckily other models exist already. Models such as offering “handholding” services [14] such as help desk services from the producer of the software (as is done by Red Hat or other major GNU/Linux distributors), or further development of the software package, where the customer should expect to be able to get better service than from an external competitor.

Trying to hold the current model of many copies at low price would likely not be a profitable model, however. The model presented by Kimppa [8] would likely be closer to the mark (figure 1).

As noted in the figure 1, the first sale of a new system would be the main profit maker. This would entail finding new ways for producing systems, such as finding partners willing of cooperation to finance a new system instead of attempting to compete with each other.

2.2 *Benefits for user understanding of systems*

As previously used modules could be reused widely, some functions of the information systems would already be widely tested with other users, thus making it simpler for the next users to take the system in use. Also, similar functions

would be more likely to function in a same manner in other systems if the same generic pattern of design would be reused instead of reinvented—a feature commonly found in various usability heuristics (e.g., [15]).

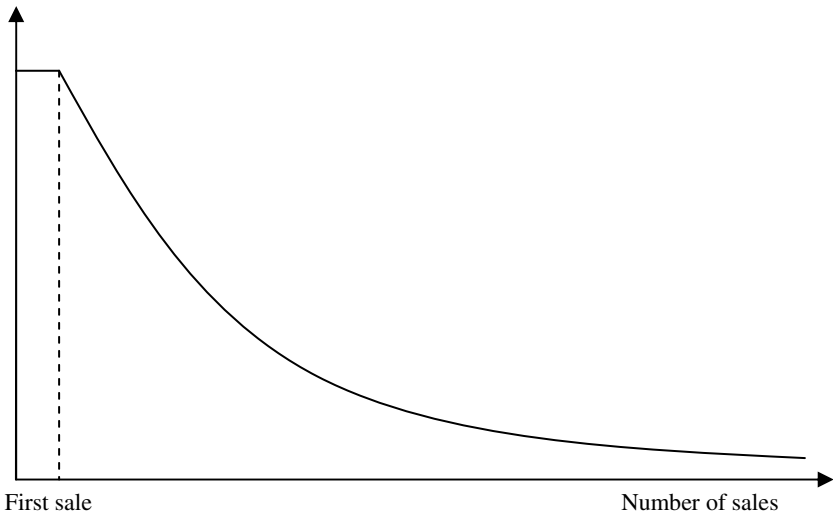


Figure 1: The new profit model for regular sales. Note: other functions that would add profit such as modifications and support are excluded [8].

For the more ambitious user even the internal functioning of the system would be—at least theoretically—available. As “[u]sers [would] no longer be at the mercy of one programmer or company which owns the sources and is in sole position to make changes” [15] they could actually make relevant changes to the information systems they need or want to use. This would give them a wholly new understanding of the systems they use.

With the return of more internal development and organisation directed development the needs of the users could be taken into account more in the development process. As the directors of information system development processes would be internal to the organisations, their understanding of the needs of the organisation and its employees would be greater than in the current out sourced system development model.

3. Current possibilities to adapt the model

Entirely internal development is typically not an option in the current systems development climate. Most organisations are not mainly systems development organisations and thus do not have the competence needed for extensive system development from within the organisation nor is it plausible to expect this to change.

Entirely external development would likely encounter problems as well. It would essentially be the same situation as exists with the current proprietary system development models: understanding of the system would remain mainly in the hands of the external organisation while the understanding of the work processes would remain internal to the organisation needing the information system.

Free Libre Open Source Software development offers possibilities even in the current IPR environment. For this to work, however, cooperation between organisations would seem to be the key factor. This would provide both the internal understanding of work processes, external help for the coding available from 1) external companies integrating existing FLOSS and coding the necessary parts for the system, 2) enthusiasts in the field, and 3) even “competition” in the field who need the same or similar information system.

This would be especially beneficial for SMEs, however public sector organisations such as hospitals could also benefit by cooperation instead of competition. Unfortunately, the current proprietary system and its unfortunate side-effects, such as requirements for competition from both EU and national decision makers, do not encourage this.

4. Discussion

In this work in progress paper an alternative to the current IPR regime was briefly introduced (for more details, see [8, 16-19]). It was argued that the alternative could bring benefits through enabling the reuse of software components and systems as well as increasing the understanding of information systems by the users and organisations. Some reasons, such as the openness of the source code as well as the encouragement to cooperation, were discussed. Unfortunately, in the current environment, this model is unlikely to be largely adapted, although some signs of its usability for SMEs are evident. Actual examples in the field of the current situation are clearly needed for verification/falsification of the proposals made.

The logical next step for this research would be to look into these issues to see whether the model is actually used (the author has seen some minor examples), and especially whether the proposed benefits present themselves in such cases.

References

- [1] Norman, D. A. (1988-1989). *The design of everyday things*, New York: Doubleday.
- [2] Eriksson, I. & Nurminen, M. I. (1991). Doing by learning: Embedded application systems. *Journal of Organizational Computing*, 1, 4, 323-339.
- [3] Faulkner, C. (1998). *The essence of human-computer interaction*, Essex: Prentice Hall.
- [4] Langefors, B. (1970). *Theoretical analysis of information systems*. Lund: Student Litteratur.

- [5] Kimppa K. K., Lahtiranta J., & Nurminen, M. I. (2007). Agentization in computing: How to ameliorate history today? *History of Nordic computing* - HiNC2.
- [6] Iivari J., Hirschheim R., & Klein K (1998). A paradigmatic analysis contrasting information systems development approaches and methodologies, *Information Systems Research*, 9(2), pp. 164-193.
- [7] Nurminen, M. I. (2006). Work Informatics – An operationalisation of social informatics. In Berleur, J., Nurminen, M. I., and Impagliazzo, J. (Eds.), *Social informatics: An information society for all? In Remembrance of Rob Kling* (pp. 407—416). Boston: Springer.
- [8] Kimppa, K. K. (2007). Problems with the justification of intellectual property rights in relation to software and otherdigitally distributable media. PhD thesis. University of Turku, Finland
- [9] Corante (2004). Major DMCA/EULA Loss - District Court Clueless in BNETD Case, Corante, *Tech News*. October 01, 2004, Available at <http://www.corante.com/importance/archives/026273.php>. Accessed 28 April, 2005.
- [10] Lessig, L. (2004). The black and white about grey Tuesday. <http://www.lessig.org/blog/archives/001754.shtml>. Accessed 25 August, 2006.
- [11] Free Software Foundation (1996-2007). The free software definition, <http://www.gnu.org/philosophy/free-sw.html>. Accessed 30 November, 2007.
- [12] Stallman, R. M. (1992). Why software should be free, <http://www.gnu.org/philosophy/shouldbefree.html>. Accessed 30 November, 2007.
- [13] Sambamurthy, V., Straub, D. W., and Watson, R. T. (2001). Managing IT in the digital era. In Dickson, G. W. and DeSanctis, G. (Eds.), *Information Technology and the Future Enterprise: New Models for Managers* (pp. 282—305). New Jersey: Prentice Hall.
- [14] Stallman, R. M. (1985-2007). The GNU manifesto, <http://www.gnu.org/gnu/manifesto.html>. Accessed 30 November, 2007.
- [15] Nielsen, J. (1993). *Usability engineering*, Boston: Academic Press.
- [16] Kimppa, K. K. (2004). Consequentialist considerations of intellectual property rights in software and other digitally distributable media. Ethicomp 2004, *Challenges for the Citizen of the Information Society*, Apr 2004.
- [17] Kimppa, K. K. (2004). Intellectual property rights – or rights to the immaterial – in digitally distributable media gone all wrong?. In Lee Freeman & A. Graham Peace (Eds.) *Information Ethics: Privacy and Intellectual Property* (pp 53—67). Hershey: Idea Group.
- [18] Kimppa, K. K. (2004). Intellectual property rights in software—Justifiable from a liberalist position? The free software foundations position in comparison to John Locke's concept of property. In Richard A. Spinello, Herman T. Tavani (Eds.), *Intellectual Property Rights in a Networked World: Theory and Practice*. (pp 67—82). Hershey: Idea Group.
- [19] Kimppa, K. K. (2005). Kantian duty ethics compared with current intellectual property rights laws. Ethics of New Information Technology: Proceedings of the Sixth International Conference of Computer Ethics: Philosophical Enquiry (CEPE2005).
- [20] Berleur, J., Nurminen, M. I., & Impagliazzo, J. (Eds.) (2006). *Social informatics: An information society for all? In remembrance of Rob Kling*, Boston: Springer.