SOFTWARE REENGINEERING PIRACY ISSUES OF CODE TRANSLATION BY END USER

Indravadan S. Patel*

Vandana A. Verma*

ABSTRACT

The reengineering of software was described by Chikofsky and Cross in their 1990 paper as "The examination and alteration of a system to reconstitute it in a new form". Less formally, reengineering is the modification of a software system that takes place after it has been reverse engineered, generally to add new functionality, or to correct errors. There are different reengineering techniques are available such as Source code translation, Reverse engineering, Program structure improvement, Program modularization, Data re-engineering. The problems of code translation techniques are converting binary code to original source code without knowing the owner of that software, redistribute software, and make pirated copy of that software.

Keywords*: Software Reengineering, Software Piracy, Reverse Engineering, Code Translation.*

*Assistant Professor, Department of Computer Science, Ganpat University, Kherva, Mehsana, Gujarat, India.

1. INTRODUCTION

The paper deals with software piracy and measures for prevention of piracy. The paper deals with questions such as what is software, what is software piracy, what is the need to protect software against piracy here the discussed concept of copyright and copy left, how can the software be protected and what is the preferred method of protecting software and why, what international instruments and other bodies that protect software and how, what are the other methods of protecting software, what are the enforcement mechanisms for protection of software. There are certain defenses to infringing copyright in case of software, however for the purpose of this paper; the defenses shall be out of the scope of the paper. Also the topic of source code and object code will be touched upon and not explained in detail. For the purpose of this paper, it shall be assumed that the software is already copyrighted and hence the tests for obtaining software copyright shall not be discussed. Software's can be protected by Patents or Copyright, for the purpose of this paper that has only considered copyright as a mode of protection.

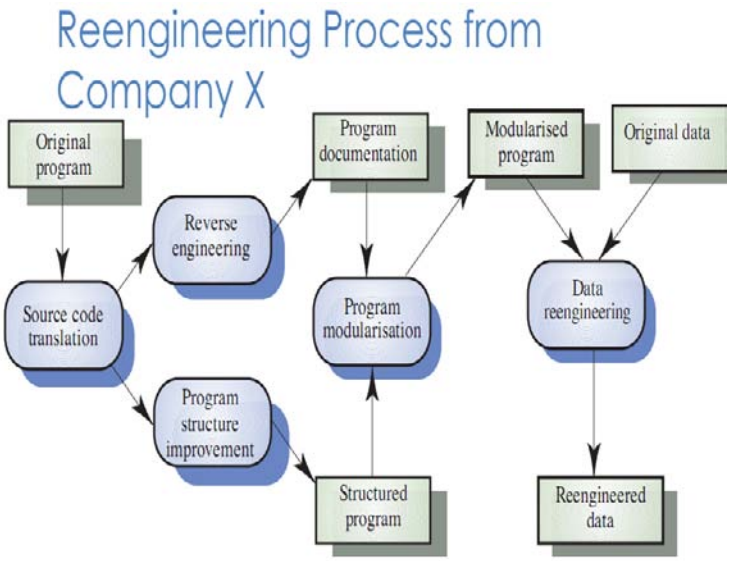
The danger to the software sector is from unauthorized production or piracy. The protection has become increasingly important because every day the economy loses out on staggering amount of money [1]. Apart from statutory protection, there is need for protection through other methods as mere statutory protection has proven to be ineffective in curbing software piracy.

Piracy is an increasing menace. A study conducted last year showed that by reducing piracy by 10 % over four years will lead to 4, 35,000 new jobs, over US$40 billion in economic growth, and over US$5 billion in tax revenues.

2. SOFTWARE RE-ENGINEERING

Software re-engineering is concerned with re-implementing legacy systems to make them more maintainable. Re-engineering may involve re-documenting the system, organizing and restructuring the system, translating the system to a more modern programming language and modifying and updating the structure and values of the system's data. The functionality of the software is not changed and, normally, the system architecture also remains the same.

Fig 1: Re-Engineering Process



From a technical perspective, software re-engineering may appear to be a second-class solution to the problems of system evolution. The software architecture is not updated so distributing centralised systems is difficult. It is not usually possible to radically change the system programming language so old systems cannot be converted to object-oriented programming languages such as Java or C++. Inherent limitations in the system are maintained because the software functionality is unchanged.

The activities in this re-engineering process are:

Source code translation: The program is converted from an old programming language to a more modern version of the same language or to a different language.

Reverse engineering: The program is analyzed and information extracted from it which helps to document its organization and functionality.

Program structure improvement: the control structure of the program is analyzed and modified to make it easier to read and understand.

Program modularization: Related parts of the program are grouped together and, where appropriate, redundancy is removed. In some cases, this stage may involve architectural transformation.

Data re-engineering: the data processed by the program is changed to reflect program changes.

3. WHAT IS SOURCE CODE?

Source code is text written using the format and syntax of the programming language that it is being written in. Such a language is specially designed to facilitate the work of computer

programmers, who specify the actions to be performed by a computer mostly by writing source code, which can then be automatically translated to binary machine code that the computer can directly read and execute. An interpreter translates to machine code and executes it on the fly, while a compiler only translates to machine code that it stores as executable files these can then be executed as a separate step.

4. SOURCE CODE TRANSLATION

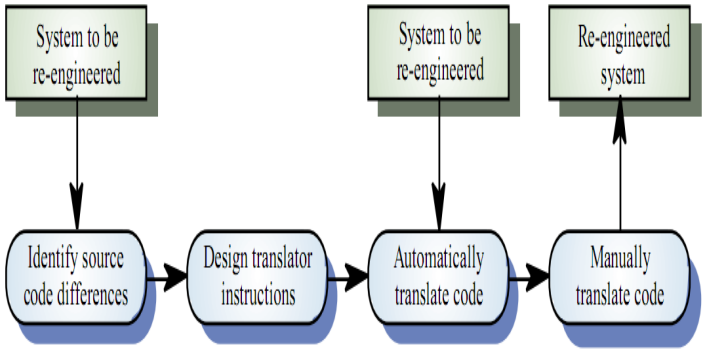
Converting the code from one language (or language version) to another e.g. FORTRAN to C

It may be necessary because of:

- Hardware platform update
- Staff skill shortages
- Organisational policy changes

This is possible if an automatic translator is available.

Fig 2: Code Translation Process.



The emulation of of one instruction set by another through translation of code is called binary translation. Sequences of instructions are translated from the source to the target instruction set. In some cases such as instruction set simulation, the target instruction set may be the same as the source instruction set, providing testing and debugging features such as instruction trace, conditional breakpoints and hot spot detection.

There are two types of binary translations

Static binary translation:

A translator using static binary translation aims to convert all of the code of an executable file into code that runs on the target architecture without having to run the code first, as is done in dynamic binary translation. This is very difficult to do correctly, since not all the code can be discovered by the translator. For example, some parts of the executable may be reachable only through indirect branches, whose value is known only at run-time.

One such static binary translator uses universal super optimizer peephole technology (developed by Sorav Bansal, and Alex Aiken from Stanford University) to perform efficient translation between possibly many source and target pairs, with considerably low development costs and high performance of the target binary. In experiments of PowerPC-to-x86 translations, some binaries even outperformed native versions, but on average they ran at two-thirds of native speed.

Dynamic binary translation

Dynamic binary translation looks at a short sequence of code—typically on the order of a single basic block—then translates it and caches the resulting sequence. Code is only translated as it is discovered and when possible, and branch instructions are made to point to already translated and saved code (memorization).

Dynamic binary translation differs from simple emulation (eliminating the emulator's main read-decode-execute loop—a major performance bottleneck), paying for this by large overhead during translation time. This overhead is hopefully amortized as translated code sequences are executed multiple times.

More advanced dynamic translators employ dynamic recompilation where the translated code is instrumented to find out what portions are executed a large number of times, and these portions are optimized aggressively. This technique is reminiscent of a JIT compiler, and in fact such compilers (e.g. Sun's Hotspot technology) can be viewed as dynamic translators from a virtual instruction set (the byte code) to a real one [2].

5. SOFTWARE PIRACY

The term piracy is not defined under the Act. Software piracy is a term which means that the software is used without the licensor’s permission. Software piracy is the unauthorized copying or distribution of copyrighted software. This can be done by copying, downloading, sharing, selling, or installing multiple copies onto personal or work computers. It is a misconception that software is bought, the transaction actually is in nature of a license. Simply put, making or downloading unauthorized copies of software is breaking the law and committing copyright infringement, also known as software piracy [3].

During the code translating from original source code to another language code there are some risk of software or source code piracy. In that the any computer programmer who have good knowledge of that language as well as some recompilation techniques of that software can easily pirate that software.

6. PROTECTION OF SOFTWARE

There are arguments against giving legal protection because it stifles competition and allows the owner of the invention to charge exorbitant prices and to make a profit out of proportion to the investment and risks undertaken. The idea behind Copy left is to ensure that an individual cannot take advantage of being able to modify a free software program and then sell the resulting modified program as a new work. In some cases, any modifications made to a Copy left program must be made freely available to all parties interested in using them. Copy left licensing evolved from the open source movement which is based on the idea that by making source code widely available and freely modifiable, higher quality software is developed and the resulting product is relatively inexpensive as compared with some commercially available software. Copy left licensing of open source software is based on principles of unencumbered redistribution, self-perpetuating terms right to create modifications and derivative works, lack of warranties, source code, distinguishable modifications [4],

It can be safely said that the copyright is needed to protect the interest of the owner however, it should be seen that it does not lead to unfair competition.

7. CONCLUSION

Apart from statutory protection, other methods should be focused on, this is because in most cases, it is not known as to whom the perpetrator is and hence it is almost impossible to bring such people to court. One of the measures that should be focused on should be the gain that consumer will get if he or she buys an original software. This would also help the piracy to substantially reduce [5].

Software piracy needs to be looked at not only violation of IP rights of the owner but also loss to the society and loss of revenue to the government. Over the years the piracy rates have gone down but by and large it is still rampant. This project has attempted to show that apart from the statutory protection, other modes of protection should also be focused on. Also shown is that the statutory protection needs to be updated periodically and suggestions of the C EAC should be implemented. Like any other legislation, enforcement is a very important factor for the success of the legislation and hence special focus should be given on better enforcement mechanism. Introduction of stringent criminal liability and fast track courts to deal with these would go a great length to curb piracy. Therefore in conclusion, software piracy should not only be looked as loss of revenue or violation of IP rights but as a loss to the world economy at large.

8. REFERENCES

[1] B.W. Husted, "The Impact of Natural Culture on Software Piracy," Journal of Business Ethics, Vol. 26 (3), (2000) 205.

[2] G. Cronin, "A Taxonomy of Methods for Software Piracy Prevention", www.croninsolutions.com/writing/piracytaxonomy.pdf

[3] G. Talwar, "Cyberpirates the Internet and the protection of Software Copyrights", The Law Relating to Computers, Internet and E-Commerce: A Guide to Cyber Laws & The Information Technology Act 2000 (2nd Edn., N.Kamath Ed., Universal Publishing Company: New Delhi, 2000).

[4] S. Kochar and other, "Protection of Software Programs via Patents- the Thorn Within", *All India Reporter Journal*, Vol. 11, (2007).

[5] P. Wehner et al, "Software Piracy in Developing Nations- A Mounting Problems and Proposed Solutions" www.softwareceo.com/files/white_papers/IP_Research.pdf