TECHNICS AND INFORMATICS IN EDUCATION
6th International Conference, Faculty of Technical Sciences, Čačak, Serbia, 28–29th May 2016
TEHNIKA I INFORMATIKA U OBRAZOVANJU
6. međunarodna konferencija, Fakultet tehničkih nauka, Čačak, Srbija, 28–29. maj 2016.

UDK: 004.3::62 Plenary Session

Where the Computing Ends, and Begins …?

Siniša Ranđić[1]
[1] Faculty of Technical Sciences Čačak, University of Kragujevac, Čačak, Serbia
e-mail sinisa.randjic@ftn.kg.ac.rs

Abstract: *The development of computer science at the beginning of this century has seen great momentum. Further development of semiconductor technology offer faster components with greater procedural possibilities. This is particularly evident through the emergence and increasing dominance of mobile computing devices. At the same time, the possibilities for users in terms of independent development of application software have significantly increased. It was created as a result of the emergence of powerful software development tools, which fixed the boundaries of computing. Regular users can now develop a wide range of applications without the need for a deeper understanding of the essence of the computer, both hardware and system software. In order to achieve this transparency of computers operations, in the development of application software a major work of computer specialists is required. Therefore, it came to the creation of a fairly sharp boundaries between computer engineering and what is colloquially treated as information technology.*

Keywords: *computing, system software, application software, software development*

1. INTRODUCTION

Computing, what most people are familiar with, is quite a young technique. However, in these few decades, although perhaps computing did not experienced significant conceptual changes, technological certainly did. The most significant consequence of technological change is certainly a significant expansion of the field of computer applications. However, we should point out another change that has occurred, and is related to the design and implementation of computer. Although since the beginning of the era of computing there have been considerations that computers are as adapted as possible to the specific application, they are primarily designed as a general-purpose devices. The end of the twentieth century and the rapid development of semiconductor technology have enabled the development of computer-based highly integrated electronic circuits. This caused the computers to be increasingly designed according to the specific application requirements. In this way, computing has become a kind of "slave" of other areas of human activity where they have found their wide application.

Unlike the past period in which the originality of computing could be seen, today it is significantly lost in the multitude of other areas in which computers find their application. As a result, the boundaries between design and implementation of a computer are not as sharp as they used to be. Such a condition was highly contributed by the phenomenon of highly integrated circuits such as microprocessors, microcontrollers, and graphics processors on the hardware, or software development tools. Thanks to that, the illusion that for their use and

implementation of computers and appropriate software the fundamental computer skills are not necessary is created.

Within the computing there have been significant conceptual changes, which caused that under the same terms today are meant different things in relation to the not-so-distant past. This is a common misunderstanding within the computing community. Simultaneously with these changes in computer practice, especially in the field of computer applications, the concept of information technology was introduced. This term in some way was a replacement for the previously domesticated concept of informatics. Although the term Information Technology primarily relates to the use of computers, there are tendencies to extend the domain of meaning to the field of design of computers and computer equipment.

Without neglecting the dynamics of the future of computing and the increasing adaptation to specific computer applications, the question of boundary of computing as a technique must be risen. This is not an academic question, but more of a determination of the framework for the acquisition of knowledges needed to practice the computer technology.

This paper will only indicate some structural changes within computing. In doing so, emphasis is on the software subsystem, because it has gone through significant changes, especially with the advent of mobile computing devices. Although of the provocative title, the paper does not pretend to offer a definitive answer to the question. It just opens up a space for discussion, which may be the subject of a special conference dedicated to this issue.

2. STRUCTURE OF COMPUTING

For the widest range of people who have contact with computers, they consist of hardware and software. In fact, even if it is an intuitive understanding, most of them have relatively good sense of what constitutes the hardware and software of the computer. However, when trying to parse these terms the doubts arise, in terms of concepts such as:

- Architecture and organization of computers;
- System software;
- Operating Systems.

In order to give the correct answer to the question what represent the aforementioned, and also the other terms related to the computer, it is necessary to start from the possible views to the computer as a device. Elemental approach to this problem is shown in Figure 1.

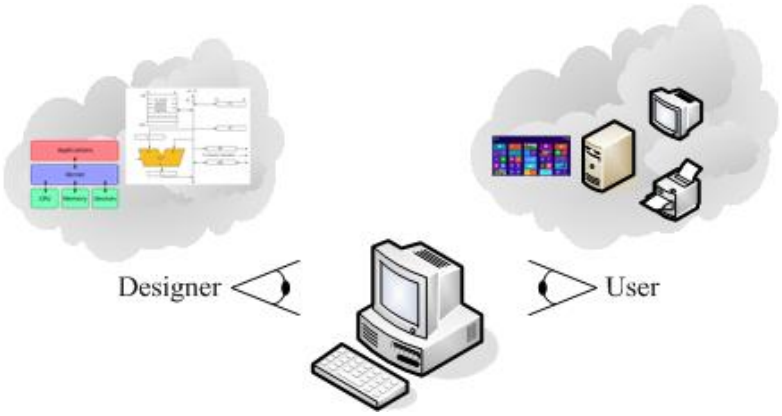

Figure 1. *Possible views to the computer as a device*

The designer sees the computer architecture and organization as hardware features that affect the development of the upper levels of a computer system, such as software compilers and operating system, having the compiler software directly depended on the computer architecture. Program compiler has the task of mapping the program written in a suitable high level programming language to the architecture of the computer [5], [6]. This is precisely served as a basis for defining computer architecture as a set of attributes that designer of computer program compiler sees [1], [2]. Unlike architecture, computer organization assumes the flow and control of data, logical design, and physical realization of the computer [3]. Through the operating system, designers provide control of computer hardware, the interface between computer and user, and a framework for developing application programs [4].

On the other hand, a user usually thinks of architecture as the structure of a computer system, i.e. a set of components from which the computer consists. In doing so, this approach is often linked to the physical realization of the computer and to the computer that is most commonly used. The ad hoc question of what are the parts of the computer, the following answer is given - chassis, motherboard, monitor, keyboard, etc. It is obvious that most standard users see the computer through the PC.

As for the operating system, users usually reduce them to the level of user interface. When it comes to program compilers, vast number of ordinary users is not aware of their existence.

In line with the development and the level of applications of computers, at the beginning of the computer era not much attention was given to differentiating computers particularly to designers, and particularly to customers, through proper education, especially because the user is required to have a high degree of knowledge of the computer. Also, it was influenced by the fact that the development of the user or application software was closely associated with designing computer.

The broad commercial application of computers has influenced that in the context of computing a clear differentiation of the areas in which the designers of various subsystems of the computer was made. These areas are shown in Figure 2.

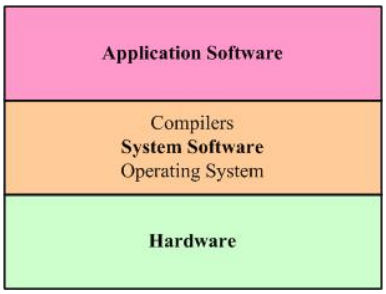

Figure 2. *The subsystems within the computer system*

The basis of every computer makes its hardware, on to which architecture the software system or software leans. This part of the computer system consists of system software and application software [7]. This division of the software is initiated by the need of facilitating the development of programs to meet the needs of their users. This idea will be followed by the development of a software subsystem elements to the present day. This is what will lead to a situation in which the development of modern application programs significantly exceeds the scope of computing.

In addition to the other, one of the main goals of computer science was that the development

of application programs is as much as possible released from the necessity of knowing the characteristics of the hardware. The development of operating systems and high level programming languages made it possible to achieve this goal considerably [8]. The operating system allowed the hardware characteristics to be hidden from the programmer. Rather than directly programming the computer hardware, the programmer worked in the so-called virtual machine [9]. Therefore, although the same programming language was used in programming, the executable program was not portable to computers with different operating systems [10].

The next step in the development of computing was the development of programming languages in which the resulting programs were transferable from computer to computer, regardless of the operating system used. This practically means that such programming languages have their own virtual machine. The corresponding virtual machine is hiding from developers all the features of the computer that are below it. This meant that such programs could be transferred from system to system. In doing so, the destination computer has to have and enforce appropriate virtual machine on which portable version of the program is performed. A typical example of this is the JAVA programming language [11].

Simple development of modern application software is provided by the lower levels of the software that include the necessary software modules from which the desired application is built. This generally remains hidden from the developer of application software, because the software modules are transparent to him. The set of modules and their location within the computer system can be represented as in Figure 3.

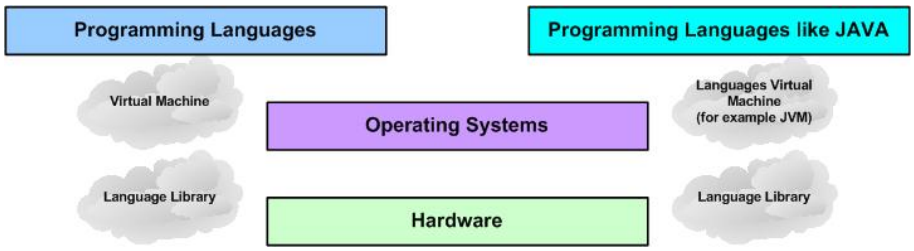

Figure 3. *Distribution of program modules within the software development environment*

Software modules, such as language libraries, virtual machines, and language virtual machines, provide interfaces between the individual computer subsystems. Thanks to them, the developer is released from having to know the characteristics of the subsystem that are below the level of the application program development.

The evolution in the process of software development has led to the emergence of tools for the design of the programs. Thanks to that, the context of computing was introduced with another subsystem, as shown in Figure 4. This subsystem is located between the system and application software, and includes a collection of programs that can be collectively labeled as tools for software development.

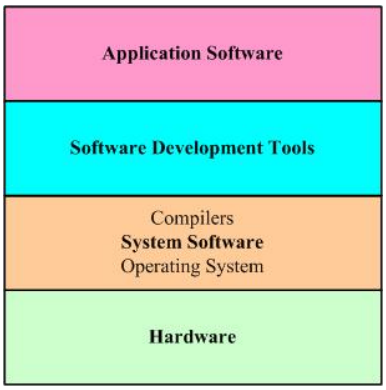

Figure 4. *Modified structure of computer system*

Thanks to the development of the environment for the design and implementation of application software, the computing in the classic sense ends at the level of design of software development tools.

3. CONCLUSION

Development of Computer Science at the beginning of this century occurred in two directions. One followed the developments in the field of semiconductor technology aiming the integration of multiple number of functions in the same circuit board. On the other hand, in the field of software development, the objective was that the approach to software development, especially application one, is much more accessible to the user. This creates the possibility that experts in the specific field can participate in the development of software, even when they are not sufficiently familiar with the computer itself. This is particularly important in case of realization of Internet applications in which software is developed for remote, virtually unknown users. In order to achieve that, the software environment for computers for which the applications are developed, must be significantly expanded. The result of this approach was the expansion of computing framework through the preparation of powerful software development tools. Thus, the boundaries of computing became more specific, particularly in terms of the knowledge required to implement elements of the system software and tools for developing application software.

In the paper, the development of the hardware was not discussed in detail, as it is in terms of development less interesting for a wide range of users. However, the market is offered with "open" hardware solutions that have the possibility of upgrading by users who are not necessarily experts in this field [12]. It should soon be expected to have discussions on the extension of the computing framework in the field of hardware.

REFERENCES

[1] Amdahl, G., Blaauw, G., Brookks, F., "*Architecture of IBM System*", IBM Journal of Research and Development, Volume 8, 2, April 1964, pp. 87 - 101
[2] Hennessy, J. L., Patterson, D. A., "*Computer Architecture: A Quantitative Approach*", The Morgan Kaufmann Series in Computer Architecture and Design, 5th Edition, Morgan Kaufmann, 2011
[3] Patterson, D. A., Hennessy, J. L., "*Computer Organization and Design: Hardware/*

Software Inteface", The Morgan Kaufmann Series in Computer Architecture and Design, 5th Edition, Morgan Kaufmann, 2013
[4] Stallings, W., "*Operating Systems: Internals and Design Principles*", Pearson, 8th Edition, 2014
[5] Aho, A. A., Lam, M., Sethi, R., Ullman, J. D., "*Compilers: Principles, Techniques and Tools*", Addison Wesley, 2nd Edition, 2006
[6] Mak, R., "*Writing Compilers and Interpreters: A Software Engineering Approach*", Wiley, 3th Edition, 2009
[7] Beck, L. L., "*System Software: An Introduction to Systems Programming*", Pearson, 3rd Revised Edition, 1996
[8] Tucker, A., Noonan, R., "*Programming Languages*", McGraw-Hill Education, 2nd Edition, 2009
[9] Smith, J., Nair, R., "*Virtual Machines: Versatile Platforms for Systems and Processes*", The Morgan Kaufmann Series in Computer Architecture and Design, 1st Edition, Morgan Kaufmann, 2005
[10] Denning, P. J., Martell, C. H., "*Great Principles of Computing*", The MIT Press, 2015
[11] Lindholm, T., Yellin, F., Bracha, G., Buckley, A., "*The Java Virtual Machine Specification*", Addison-Wesley Professional, 1st Edition, 2014
[12] Gibb, A., "*Building Open Source Hardware: DIY Manufacturing for Hackers and Makers*", Addison-Wesley Professional, 1st Edition, 2014