TRANSFORMATION OF PROGRAMMING PARADIGM

Branislav Lacko
Vysoké učení technické v Brně, branislav.lacko@seznam.cz

ABSTRACT:
The paper deals with the development and changes in programming paradigm in the 2nd half of the 20th century. In conclusion the paper characterizes a probable development early in the third millennium. The word "paradigm" is considered a style approach to solve a certain problem. The paper is a continuation of a professional block of papers which dealt, at the conference Creation of software 2009 with the past and future development of programming.

KEY WORDS:
Programming, paradigma, second literacy

1 INTRODUCTION

The paper is a continuation of discussions on the past, the present time and the expected development of programming which were dealt with at the jubilee 35th Conference Creation of software 2009 (see the conference lectures 16, 17, 18 and discussions on them).

In particular changes in programming in the past years were discussed, the position of programming today, expected development trends in programming, problems of teaching programming at schools etc.

The present paper is aimed, by means of an analysis of programming paradigma development, at a possible prediction of the change in programming approach within next ten to twenty years.

2 GRADUAL CHANGES IN PROGRAMMING PARADIGM

2.1 Programming as an art

The monograph by Donald Knuth (1) on programming issued in 1968, i.e. at the time, when transistor computers started being unexpectedly used in operation. This fact brought a problem, how to find and create complicated algorithms of various tasks, which were necessary to be written in programming languages so as to be able to be run on automatic computers to control their operation. It was a fully new problem being not solved before. No wonder that first steps of people in the sphere of programming required a high creativeness, invention and original approach to solve the PC problems of programming. A logical consequence was a comparison of this new creation activity with art. Let us mention that basic algorithms dealt with in the book, are topical also today and for this reason the Brno publishing house Computer Press published in 2008 a Czech translation (9) of the D. Knuth publication.

2.2 Programming as a discipline

A successful utilization of thousands of 3rd generation computers, making use of the invention of electronic integrated circuits, required, early in the seventies, to increase the number of programmers and also the productivity of their work so that enormous and complicated operation systems could be worked out, which would ensure their effective operation and meet the demand for application programs of various sorts. Programming could

not be any more a privilege of a small number of originally thinking individuals. Just in these years the expression “software crisis” appeared, characterizing the fact that the productivity of programs started lag behind the productivity of a mass production of electronic industry computers. It was necessary to educate tens of thousands of programmers, capable to produce professionally the requested software. First N. Wirth unveiled the secret of the word “program” by describing the program as a mutual symbiosis of exact algorithms and numerical data structures (2) in 1973. Then E. Dijkstra presented programming as a new professional discipline (3) to both trade and non-professional public.

2.3 Programming as a systematic and structured procedure

A programming of ever more complicated and enormous programs revealed that these programs could not be successfully and correctly built, if they were not based on a system approach and if not progressive rationalistic methods for their creation were used. As early as in 1972 three pioneers of a structured approach to programming, i.e. O.J. Dahl, E.W. Dijkstra, C. A. R. Hoare published a paper (7) which , at the turn of the seventies and eighties, launched an era of a structured programming. As a result of this approach, methods of a standardized programming, logical programming and functional programming, reflecting a technological approach to program creation were subsequently extended. These approaches initiated an introduction of a number of new programming languages, e.g. Pascal, LISP, PROLOG, Ada etc. The principles and ideas of the structured programming were developed by other promoters of this principle, i.e. Yourdon, Jackson and others. At. that time a launching of a thorough structured analysis and a program structure design prior to programming (14,15) revealed necessary. In addition to publications which dealt with general principles of a structured approach to analysis, very thorough and concrete methods for analysis and the SSADM – Structured System Analysis and Design Method (GB), SDM – Structured Development Method (NL) and others started being developed.

2.4 Programming as a second literacy

Early in the eighties, progress in semi-conductor technology enabled a mass production of cheap and efficient microprocessors,. By a design of a PC, making use of a 16-bit microprocessor Intel 8086 in 1981, the firm IBM launched an era of a mass production and utilization of PC. The microcomputer became a necessary helper in all human activities, which aroused a great interest of the general public in their programming. TV companies introduced programs on programming in BASIC language for 8-bit computers, a number of periodicals started launching serials on programming, tens of popular computer periodicals issued in high numbers of copies and bookmakers offered popular publications on microcomputers and programming, clubs of microcomputer users of the brands ZX Spectrum, Amstrad, Commodore, Amiga etc. (16) were established. It was obvious that the operation of PC and the acquaintance with their programming would soon become a necessary part of knowledge of every intellectual. A. Jeshov (5) mentioned it at the world conference of computer education in the Swiss town of Lausanne. His expression that programming will become “second literary” (in addition to reading and writing) for a modern man has immediately become symbolic for that time approach to programming.

2.5 Programming as a specialized and specific profession, based on scientific knowledge

At the end of the eighties it was necessary to create theoretic bases for programming and to start educating specialized operators who will be prepared for executing programmer profession. The book by D. Gries (4) formulated the bases of science on programming and soon after specialized fields called computer science, software engineering, informatics etc. started being established. At the same time a number of technical colleges and universities

launched corresponding studying programs and established institutes and faculties, aimed at educating in this respect. Subjects in which the students acquired basic knowledge of programming and computers were on the teaching programs of both technical universities and universities. The study was further extended by modern information and communication technologies which were the result of an intensive worldwide research and development.

2.6 Programming with object-oriented paradigma

Problems, caused by necessary separately carried out analyzes and specifications of functions, data and events which had to precede a structured programming, resulted in an object-oriented paradigma (8) early in the seventies together with the programming languages such as SIMULA and SMALLTALK, but became a predominating paradigma as late as at the end of the nineties. It is of interest that the designation object-oriented paradigma was used.
It is satisfying that the object-oriented approach has been well dealt with by the Czech original and quality professional literature which can be used for acquainting with this problem. The publications by R. Pecinovský (19) and M. Virius (20) are involved.. (Both authors wrote a number of books on the problems of object-oriented programming incl. a common two-volume publication (21), used for launching an object-oriented approach in our country). The problems of the object-oriented paradigma have been discussed at regular conferences Objekty (see e.g. the pages of the 15th run of this conference at the Ostrava University http://konference.osu.cz/objekty2010).

The utilization of the object-oriented programming also requires, in the present programming languages such as Java, Visual Basic and others, a transition to new methods of work in teams which formulate object-oriented products. In particular the application of efficient methods for organizing work of development software information systems teams (22, 10) and risk analysis (risk analysis in Czech SW projects is missing (23)) are involved, but also a reality modelling by means of an object model (12) during an object-oriented analysis and object-oriented application design (11) and an arrangement of individual phases of the whole creation project of an object-oriented application e.g. the method BORM (13) as the Czech alternative to the RUP method by the firm IBM-Rational.

3 CONCLUSION

Computer programming will obviuously long remain inseparatedly associated with the existence of human society. From the point of view of software engineering and the development of programming paradigma , the following trends can be, however, presumed:

- The number of professional system programmers in applying scientific-technical calculations and mass processing who at the end of the 20th century prevailed, will be relatively lower, whereas the number of programmers dealing with automatized system in real-time applications will be growing due to a high number of built-in microprocessors of various products of all sorts.
- Also the number of programmers dealing with programming specialized systems, e.g. CNC machine tools models or various programmable PLC automatic machines will be higher. The number of these machines has been continuously growing. It is of interest that whilst in the eighties of the 20th century competitions in computer programming were staged, after the year of 2000, competitions of CNC machine tools programmers (www.techtyden.cz) have been held within the “Academy of CNC machining” on the pages of the periodical “Technical Weekly” since 2009 and within the event “Automation PANASONIC for schools” pupils compete in solving various PLC tasks (see www.panasonic-electric-works.cz).

- In the near future, obviously within 2015 – 2020, the problems of programming intelligent automatic machines, in particular industrial and humanoid robots, and other automated systems with artificial intelligence will dominate.

In particular the programming of artificial intelligence tools will substantially change programming paradigma. This expected approach to programming could be perhaps best characterized as a programming of aims and development of systems with intelligent behaviour – i.e. affecting the machines of artificial intelligence, not computers of von Neuman's concept. The best for this paradigm will be obviously problem-oriented (DSL - Domain Specific Language) and declarative languages, dialogue languages, making use of spoken word and spatial visualization tools of 3D graphics in combination with teach-in procedures.

LITERATURE

1 Knuth, E. D.: The art of computer programming (vol. 1 & 2). Addison-Wesley, Reading Mass., 1968
2 Wirth, N.: Algorithm + Data Structures = Programs. Prentice Hall 1973 Englewood Cliffs (překlad - Wirth N.: Algoritmy a štruktúry údajov, Alfa 1988)
3 Dijkstra W. E.: Discipline of programming. Prentice Hall 1976 Englewood Cliffs
4 Gries, D.: The science of programming. Springer-Verlag 1981
5 Ershov, A: Programming, the second literacy. In: Proceedings of 3rd IFIP World Conference on Computers in Education (WCCE81), Lausanne, 1981
6 Virius,M.: Programování dnes a zítra. Softwarové noviny, roč.IX.,(1998),č.5, str.28
7 O.-J. Dahl, E. W. Dijkstra, C. A. R. Hoare: Structured Programming. Academic Press, London, 1972
8 Abadi, M. - Cardelli, L.: A Theory of Objects. Springer-Verlag New York 1996
9 Donald E. Knuth: Umění programování. Computer Press 2008 Brno
10 Lacko,B.: Řízení projektů v metodě BORM. Konference OBJEKTY 2005, VŠB – TU Ostrava 2005, str. 112 - 121
11 Duben, J,: Objektové modely podniku. Grada Publishing Praha 1996
12 Merunka, V.: Objektové modelování Nakladatelství ALFA 2008 Praha
13 Polák, J.-Merunka,V.-Carda,A.: Umění systémového návrhu. Grada 2003 Praha
14 Langefors,B.: Theoretical Analysis of Information Systems, Studentliteratur AB 1973
15 Chandor,A.-Graham,J.-Williamson,R.: PracticalSystems Analysis. Hart-Davis Educational Publishing 1972 London
16 Pecinovský, R.: Historie mimoškolní výuky programování u nás. Sborník konference Tvorba softwaru 2009. VŠB-TU Ostrava 2009 Ostrava, str. 129 – 133
17 Virius, M.: Výuka informačních technologií na vysokých školách u nás. Sborník z konference Tvorba softwaru 2009. VŠB- TU Ostrava 2009 Ostrava, str. 156 - 161
18 Lacko, B.: Programování a informační gramotnost ve znalostní společnosti. Sborník Konference Tvorba softwaru 2009. VŠB-TU Ostrava 2009 Ostrava, str. 58 - 66
19 Pecinovský, R.: OOP – Naučte se myslet a programovat objektově. Computer Press
20 M. Virius: Pasti a propasti jazyka C++, Grada, Praha 1997
21 R. Pecinovský, M. Virius: Objektové programováni (I. a II.), Grada Praha 1996
22 Buchalcevová, A.: Metodiky vývoje a údržby informačních systémů. Grada 2005 Praha
23 Adásková,P.-Balcar.J.: Průzkum současného stavu řízení rizik v podnicích a organizacích v ČR (2009). RPIC-VIP červenec 2009, Ostrava

Appendix 1

Knuth, E. D.: The art of computer programming (vol. 1 & 2).
Addison-Wesley, Reading Mass., 1968
Chapter outline of published and unpublished volumes
(http://en.wikipedia.org/wiki/The_Art_of_Computer_Programming
on line -February2011):

- Volume 1 - Fundamental Algorithms
  - Chapter 1 - Basic concepts
  - Chapter 2 - Information structures
- Volume 2 - Seminumerical Algorithms
  - Chapter 3 - Random numbers
  - Chapter 4 - Arithmetic
- Volume 3 - Sorting and Searching
  - Chapter 5 - Sorting
  - Chapter 6 - Searching
- Volume 4 - Combinatorial Algorithms
  - Volume 4A - Enumeration and Backtracking
    - Chapter 7 - Combinatorial searching
  - Volume 4B - Graph and Network Algorithms, in preparation
    - Chapter 7 *continued*
  - Volume 4C and possibly 4D - Optimization and Recursion, in preparation
    - Chapter 7 *continued*
    - Chapter 8 - Recursion
- Volume 5 - Syntactic Algorithms, planned (as of 2011, *estimated* in 2020).
  - Chapter 9 - Lexical scanning
  - Chapter 10 - Parsing techniques (includes also string search and data compression)
- Volume 6 - Theory of Context-Free Languages, planned.
- Volume 7 - Compiler Techniques, planned

Appendix 2

Wirth, N.: Algorithm + Data Structures = Programs. Prentice Hall 1973 Englewood Cliffs
(on line Wikipedia - Niklaus Wirth)
Chapter outline:

- Chapter 1 - Fundamental Data Structures
- Chapter 2 - Sorting
- Chapter 3 - Recursive Algorithms
- Chapter 4 - Dynamic Information Structures
- Chapter 5 - Language Structures and Compilers

Programming Paradigms: Lecture Notes. Table of Contents. 1. âœª What is a "programming paradigm"? 1.1. Definition. 1.2. Paradigms as "ways of organizing thought". 1.3. Paradigms and Languages. 1.4. Fluidity between paradigms. 1.5. âœª A remark on paradigm shift. 2. Prelude: Abstraction and Types. 3. Imperative programming. 3.1. Model: von Neumann computer. 3.2. Example: Sorting. 3.3. Transformation: Loops âŸ¶ Gotos. 3.4. Transformation: If then else âŸ¶ Gotos. 3.5. Reverse transformation: (Gotos âŸ¶ Loops). 3.6. Transformation: inlining procedure calls.Â In fact the above transformation is parametric on the condition and body of the loop. Hence we may just abstract over these parts. We will present the next transformation in this format. The term programming paradigm refers to a style of programming. It does not refer to a specific language, but rather it refers to the way you program. There are lots of programming languages that are well-known but all of them need to follow some strategy when they are implemented. And that strategy is a paradigm. The types of programming paradigms. Describes the different styles of programming (Source: geeksforgeeks.org).Â In an imperative programming paradigm, the order of the steps is crucial, because a given step will have different consequences depending on the current values of variables when the step is executed. To illustrate, let's find the sum of first ten natural numbers in the imperative paradigm approach. Example in C: #include . A programming paradigm is the concept by which the methodology of a programming language adheres to. Paradigms are important because they define a programming language and how it works. A great way to think about a paradigm is as a set of ideas that a programming language can use to perform tasks in terms of machine-code at a much higher level. These different approaches can be better in some cases, and worse in others. A great rule of thumb when exploring paradigms is to understand what they are good at. While it is true that most modern programming languages are general-purpose and can do ju Automata-based programming â€“ a program, or part, is treated as a model of a finite state machine or any other formal automaton. Reactive programming is a declarative programming paradigm concerned with data streams and the propagation of change. The subroutines that implement OOP methods may be ultimately coded in an imperative, functional, or procedural style that may, or may not, directly alter state on behalf of the invoking program. There is some overlap between paradigms, inevitably, but the main features or identifiable differences are summarized in this table