Associative Semantic Computers for Intelligent Computer Systems of a New Generation

Vladimir Golenkov and Daniil Shunkevich and Natalia Gulyakina
Valerian Ivashenko and Vadim Zahariev
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
Email: golen, shunkevich, guliakina, ivashenko, zahariev@bsuir.by

***Abstract*—The paper considers the shortcomings of the currently dominant von Neumann architecture of computer systems as the basis for building intelligent computer systems of a new generation, analyzes modern approaches to the development of hardware architectures that eliminate some of these shortcomings, substantiates the need to develop fundamentally new hardware architectures, which are hardware version of the implementation of the interpretation platform for systems built on the basis of OSTIS Technology, — associative semantic computers.**

***Keywords*—OSTIS Technology, ostis-platform, platform independence, ontology, associative semantic computer.**

I. INTRODUCTION

For the development of *ostis-systems*, the use of modern software and hardware platforms focused on address access to data stored in memory is not always effective, since when developing intelligent systems, it is actually necessary to model nonlinear memory based on linear one. Improving the efficiency of problem solving by intelligent systems requires the development of specialized platforms, including hardware ones, focused on unified semantic models of information representation and processing. Thus, the main purpose of creating *associative* semantic *computers* is to increase the performance of ostis-systems.

II. CURRENT STATE OF DEVELOPING COMPUTERS FOR INTELLIGENT SYSTEMS

The vast majority of modern software and hardware platforms used in the development of modern computer systems and, in particular, intelligent computer systems are based on the principles of the abstract von Neumann machine, or "von Neumann architecture" (see [1], [2]). Let us consider the basic principles underlying the von Neumann machine.

von Neumann machine
:= [abstract von Neumann machine]
∈ *abstract information processing machine*
⇒ *underlying principles**:

⟨

- [The information in memory is represented as a sequence of strings of characters in a binary alphabet ("0" or "1").]
- [The machine memory is a sequence of addressable memory cells.]
- [Any string of characters in the binary alphabet can be recorded to each cell. At the same time, the length of the lines for all addressable cells is the same (in the current standard of cells, called bytes, it is equal to 8 bits).]
- [Each memory cell uniquely corresponds to a bit string that denotes this cell and represents its address.]
- [To each type of elementary actions (operations) performed in the memory of the von Neumann machine, its identifier is uniquely assigned, which is also represented in memory as a bit string.]
- [Each specific operation (command) per- formed in memory is represented (specified) in memory as a string consisting of
    - the code of the corresponding type of operation;
    - the sequence of addresses of memory fragments containing operands on which operations are performed – the source arguments and results. Any such fragment is specified by the address of the first byte and the number of bytes. The number of operands is unambiguously set by the operation type code.

  ]
- [A program running in memory is stored in memory as a sequence of specifications of particular operations (commands).]
- [Thus, both the processed data and the programs for

processing this data are stored in the same memory (unlike, for example, the Harvard architecture) and are encoded in the same way.]

⟩

Let us consider in more detail the features of the logical organization of the traditional (von Neumann) architecture of computer systems, which significantly complicate the effective implementation of *ostis-systems* based on it:

- sequential processing that limits the efficiency of computers by the physical capabilities of the element base;
- low level of access to memory, i.e. complexity and awkwardness of performing the procedure of associative search for the necessary fragment of knowledge;
- linear memory organization and an extremely simple view of constructive objects directly stored in memory. This leads to the fact that in intelligent systems built on the basis of modern computers, the manipulation of knowledge is carried out with great difficulty. Firstly, it is necessary to operate not with the structures themselves but with their inconvenient linear representations (lists, adjacency matrices, incidence matrices); secondly, the linearization of complex structures destroys the locality of their transformations;
- the representation of information in the memory of modern computers has a level very far from the semantic one, which makes the processing of knowledge rather awkward, requiring consideration of a large number of details concerning not the meaning of the processed information but the way it is represented in memory;
- in modern computers, there is a very low level of hardware-implemented operations on non-numeric data and there is no hardware support for logical operations on knowledge fragments with a complex structure, which makes manipulating such fragments very difficult.

Attempts to overcome the limitations of traditional von Neumann computers have led to the appearance of many approaches associated with particular changes in the principles of logical organization of computers, primarily depending on the classes of problems and subject domains that a particular class of computers focuses on. All these tendencies, considered together, allow outlining some key principles of the logical organization of computers focused on the knowledge processing (knowledge processing machines – KPM). Let us list the main of these tendencies:

- transition to nonlinear memory organization (see [3], [4]) and hardware interpretation of complex data structures (see [5], [6], [7]);
- hardware implementation of associative access to information (see [3], [8], [9], [10], [11], [12], [13]);
- implementation of parallel asynchronous processes over memory (see [3], [14]), and, in particular, development of computing machines controlled by data flow (see [15], [16], [17], [18]);
- hardware interpretation of high-level languages (see [19], [20], [21]);
- development of database management hardware tools (database processors) (see [22], [23], [24]).

At the intersection of these tendencies, different classes of computing devices have appeared at different times. Let us list some of them:

- machines with hardware interpretation of complex data structures (see [6], [25], [26]);
- machines with developed associative memory (see [10], [27], [28]);
- associative parallel matrix processors (see [29]);
- homogeneous parallel structures for solving combinatorial logic problems on graphs and hypergraphs (see [30]);
- various graph processing devices(see[31],[32],[33],[34]), in particular, based on FPGA (see [35], [36], [37]) and vector processors (see [38]);
- systems that process information directly in memory by evenly distributing functional means in memory and, in particular, the processor-memory proposed by M. Weinzweig, focused on solving artificial intelligence problems (see [39], [40]);
- machines controlled by data flow (see [15], [18], [29]) and, in particular, processors, that are reconfigurable taking into account the semantics of the input data flow (see [41]);
- recursive computing machines (see [3]);
- relational database processors (see [9], [18], [22]);
- computers with restructurable memory (see [42], [43], [44], [45]);
- active semantic networks (M-networks) (see [46]);
- associative homogeneous environments (see [47]);
- neural-like structures (see [48], [49]). In recent years, the active development of the theory of artificial neural networks has led to the development of various approaches to the building-up of high- performance computers designed for training and interpretation of artificial neural networks (see [50], [51], [52]) and their implementation in various software and hardware complexes. In a separate direction, the so-called neuromorphic processors (see [53]) are distinguished by high performance and low power consumption.
- machines for interpreting logical rules (see [54]).

In addition, the development of graphics processors (graphics processing unit, GPU) has led to the possibility of organizing parallel computing directly on the GPU, for which specialized software and hardware architectures are being developed, for example, CUDA (see [55]) and OpenCL (see [56]). The advantage of the GPU in this case is the presence of a large number of cores within one GPU (compared to the central processor), which makes it possible to effectively solve problems with natural parallelism on such an architecture (for example, operations with matrices). Works dedicated to the principles of processing graph structures on the GPU are also being developed (see [57], [58], [59]).

In general, it can be said that due to the increase in the performance of modern computers, the number of developments of specialized hardware solutions has decreased in recent decades, since many complex computing problems can now be solved on traditional universal architectures in an acceptable time. As shown above, the exception is mainly specialized computers for processing artificial neural networks and other graph models, which is conditioned by the high demand for such models and their complexity.

At the same time, most of these approaches (even if they deviate far enough from the basic principles of computer organization proposed by von Neumann) implicitly retain the point of view of the computer as a large arithmometer and thereby retain its orientation to numerical problems. Works aimed at developing hardware architectures, designed to process information represented in more complex forms than in traditional architectures, have not been widely distributed and used due, firstly, to the specifics of the proposed solutions and secondly, due to the lack of a common universal and unified coding language for any information, in the role of which within the *OSTIS Technology*, the *SC-code* acts, as well as the appropriate proven technology for developing software systems for such hardware architectures. Thus, developers of such architectures often face the need to develop specialized software for these architectures, which ultimately leads to a strong limitation in the scope of application of such architectures, since their use turns out to be reasonable only if the complexity of developing specialized software proves itself, taking into account the low efficiency of solving the corresponding problems on more traditional architectures.

The SC-code, which is the formal basis of the *OSTIS Technology*, was originally developed as a language for encoding information in memory of *associative semantic computers*, so it originally contains principles such as universality (the ability to represent knowledge of any kind) and unification (uniformity) of representation, as well as minimization of the *Alphabet of the SC-code*, which, in turn, makes it easier to create a hardware platform that allows storing and processing texts of the *SC-code*.

The main methodological feature of the proposed approach to the development of hardware implementation tools for intelligent systems support is that such tools should be developed not before but after the main terms of the corresponding technology for the design and operation of intelligent systems will be tested on modern technical means. Moreover, within the *OSTIS Technology*, the methodology of transition to new hardware tools has been clearly thought out, which affects only the lowest level of the technology – the level of implementation of the basic semantic network processing machine (interpreter of the *SCP Language*).

The project of the *associative semantic computer* has a long history, the main stages of which are the following ones:

- 1984 – at the Moscow Institute of Electronic Technology, V. Golenkov defended the PhD dissertation on the topic “Structural organization and processing of information in electronic mathematical machines controlled by the flow of complex structured data”, in which the basic principles of semantic associative computers were formulated and considered (see [60]).
- 1993 – the Goskomprom Commission carried out successful tests for the prototype of the *associative semantic computer* developed on the basis of transputers within the research project “Parallel graph computing system focused on solving artificial intelligence problems” (see [61], [62]).
- 1996 – V. Golenkov defended the doctoral dissertation on the topic “Graphodynamic models and methods of parallel asynchronous processing of information in intelligent systems” (see [63]).
- 2000 – at the Institute of Management Problems of the Russian Academy of Sciences, P. Gaponov defended the PhD dissertation on the topic “Models and methods of parallel asynchronous processing of information in graphodynamic associative memory” (see [64]).
- 2000 – at the Institute of Software Systems of the Russian Academy of Sciences, V. Kuzmitsky defended the PhD dissertation on the topic “Princi- ples of building a graphodynamic parallel computer focused on solving artificial intelligence problems” (see [65]).
- 2004 – at the Belarusian State University of Informatics and Radioelectronics, R. Serdyukov defended the PhD dissertation on the topic “Basic algorithms and tools for information processing in graphodynamic associative machines”, in which the basic software of semantic associative computers was considered (see [66]).

At the same time, despite the presence of a working prototype of the *associative semantic computer*, based on transputers, the main attention within the corresponding project and other listed works was paid to the principles of organizing distributed parallel processing of SC-code constructions, in particular, the SCD Language (Semantic Code Distributed) was developed for distributed storage of SC-code constructions and the SCPD Language for their distributed parallel processing. However, the general principles of information storage and the general architecture of each of the processor elements (of transputers) remained behind von Neumann. In particular, to encode SC-code constructions in traditional address memory, appropriate data structures have been developed, close to those described in [67].