Adaptable Processing: Stage Development

Dumitru Todoroi

Academy of Economic Studies of Moldova, Chisinau, Republic of Moldova

The development and evolution of Microsoft Office and Microsoft Windows shells are based in general on the special methodology of software creation and implementation such as macros, subroutine, custom commands, and specialized features. Microsoft Office for Mac has for long been criticized for its lack of support of Unicode and BiDi languages, notably Arabic and Hebrew. This has not changed in the Office 2008 version. Microsoft Office 2010 (also called Office 2010 and Office 14) is the current version of the Microsoft Office productivity suite for Microsoft Windows, and the successor to Microsoft Office 2007. With Office 2010, users are in control, getting things done and producing amazing results however and wherever they work best. Ms Office 2010 is the last version of Microsoft Office with support for Windows XP, Windows Server 2003, Windows Vista, and Windows Server 2008 due to Office 2013 requiring Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, or Windows RT. Adaptive hardware (Ugurdag, 2006) reflects the capability of a system to maintain or improve its performance in the context of internal or external changes. Adaptation at hardware levels increases the system capabilities beyond what is possible with software-only solutions. Algorithms, techniques, and their implementation in hardware are developed over a diverse variety of applications. The methodology of the On-Off-Line adaptable processors (Todoroi & Micusha, 2012) support development of adaptable software and hardware. Automatic creation of the Off-Line adaptable processors are proved. Development of the On-Line and On-Off-Line adaptable processors based on Off-Line processing creation method is proposed (Todoroi, Micușă, & Todoroi, 2009; Todoroi, 2008a, 2008b).

Keywords: adaptability, adaptive, processor, language, adaptable processor, adaptive hardware, adaptable software

Introduction

The development and evolution of Microsoft Office and Microsoft Windows shells is based in general on the special methodology of software creation and implementation such as macros, subroutine, custom commands, and specialized features. A major feature of the last decade of Ms Office suite development consists in the ability for users and third party companies to write add-ins that extend the capabilities of an application by adding custom commands and specialized features.

Developers Microsoft Office have underlined that in Ms Office 2007:

... The entire user interface has been redesigned to be more intuitive, easier to navigate, and better suited to the task at hand... Now the command you need come to you, depending on the type of object you select and the application you are using. (Ugurdag, 2006, p. 1)

Adaptive hardware (Ugurdag, 2006) reflects the capability of a system to maintain or improve its performance in the context of internal or external changes. Adaptation at hardware levels increases the system capabilities beyond what is possible with software-only solutions. Algorithms, techniques, and their implementation in hardware are developed over a diverse variety of applications.

The methodology of the On-Off-Line adaptable processors supports development of adaptable software (Todoroi, 2008a, 2008b). Automatic creation of the Off-Line adaptable processors has been proved (Todoroi, Micușă, Todoroi, & Spataru, 2007). The adaptable tools as base for creation, application, and development of adaptable software are characterized by a set of advanced linguistic' and processors' features (Todoroi, Micușă, & Todoroi, 2006). Adaptable languages, as part of adaptable software and hardware, integrate such linguistics' features as: multilanguageability, universality, speciality, extensibility, dialect-ability, compatibility of basis, reductability, effectivity of modification, continuity of human, machine experience, and touchability to the formal natural language level of human-machine interactions. Adaptable processors as part of adaptable software and hardware integrate such translation' features as: universality, mobility, transferability, cognisability, specializability, minimizing of processors' quantity, and raising the level of adaptable software to the level of problem formulation.

Difficulties in Porting Office

Microsoft develops Office for Windows and Mac platforms. Beginning with Mac Office 4.2, the Mac and Windows versions of Office share the same file format. Consequently, any Mac with Office 4.2 or later can read documents created with Windows Office 4.2 or later, and vice-versa. Microsoft Office 2008 for the Mac drops VBA support (MS Office, 2008, 2010). Microsoft has replaced VBA (visual basic for applications) with support for AppleScript. As a result, macros created with Office for Windows will not run on Office for the Mac, and vice versa. In addition, Microsoft has also ceased development on Microsoft Virtual PC (personal computer) (MS Office, 2007). There were efforts in the mid 1990s to port Office to RISC (Reduced Instruction Set Computing) processors such as NEC (Nippon Electric Company)/MIPS (Microprocessor Without Interlocked Pipeline Stages) and IBM (International Business Machines Corporation)/PowerPC, but they met problems such as memory access being hampered by data structure alignment requirements. Difficulties in porting Office may have been a factor in discontinuing Windows NT (New Technology) on non-Intel platforms.

Criticisms of Microsoft Office

Microsoft Office is commonly criticized for its security issues and infections from macro viruses. Secunia reports [Wikipedia] that out of the 15 vulnerabilities reported in 2006 for Microsoft Office 2003 (standard edition), 20% are unpatched, 33% are marked as extremely critical, and 53% are marked as highly critical.

Another common criticism of Microsoft Office is its preference of proprietary formats over open standards to store data, which is often intended to be shared with other users, hence forcing them into adoption of the same software platform (Stallman, 2008). However, Office Open XML (Extensible Markup Language), the document format for the latest versions of Office for Windows and Mac, is an ECMA (European Computer Manufacturers Association) standard and open for implementation to anyone. Microsoft has freely published the complete format documentation under the Open Specification Promise (MS Office, 2010) and has made available free

downloadable converters for previous versions of Microsoft Office including Office 2003, Office XP, Office 2000, and Office 2004 for the Mac. Implementations of Office Open XML exist on the Mac platform and Linux. Microsoft Office for Mac has for long been criticized for its lack of support of Unicode and BiDi languages, notably Arabic and Hebrew. This has not changed in the 2008 version.

AEH (Adaptive and Evolvable Hardware)

Adaptive hardware reflects the capability of a system to maintain or improve its performance in the context of internal or external changes, such as uncertainties and variations during fabrication, faults and degradations, modifications in the operational environment, incidental or intentional interference, different users and preferences, modifications of standards and requirements, trade-offs between performance and resources, etc.

The first NASA/ESA (National Aeronautics and Space Administration/European Space Agency) Conference on Adaptive Hardware and Systems (Ugurdag, 2006) was to bring together leading researchers from the adaptive hardware and systems community to exchange experiences and share new ideas in the field. The conference expands the topics addressed by the precursor annual series of NASA/DoD (National Aeronautics and Space Administration/ Department of Defense) Conference on Evolvable Hardware held between 1999 and 2005. This meeting was provided a forum for discussion on the generic techniques of adaptive hardware and systems, with a focus on communications and space applications, with view to its expansion and exploitation in other applications such as consumer, medical, defense and security, etc.

The paper (Stoica & Andrei, 2007) was one of the most important papers at the Second NASA/ESA Conference on Adaptive Hardware and Systems. It explores AEH solutions from three essential perspectives: (1) technology; (2) economics; and (3) system architecture. The growing field of mobile robotics produces various types of robot systems for different applications (Merten & Gross, 2008). What was underlined that the development of new robots often requires the design of new hardware systems which is time consuming and susceptible to errors.

Adaptation at hardware levels increases the system capabilities beyond what is possible with software-only solutions, and a large number of adaptation features employing both analog and digital adjustments are becoming increasingly present in the most elementary system components. Algorithms, techniques, and their implementation in hardware are developed over a diverse variety of applications, such as adaptive communications (adapting to changing environment and interferences), reconfigurable systems on a chip and portable wireless devices (adapting to power limitations) or survivable spacecraft (adapting to extreme environments and mission unknowns).

Adaptable Software

Adaptable tools represent a set of meta-system methods, models, algorithms, and procedures (Todoroi, 1992) used in the process of the software and hardware systems creation and its implementation. They support human-machine interaction processes to be developed by various kinds of software and hardware systems at different stages of information, knowledge, and conscience based Societies ascending evolution.

Adaptors as adaptable meta-system tools represent the union of methods, models, algorithms, and procedures to be used for adaptable languages and processors creation and application. They are based on

definition and usage of new or modified data, operators, statements, and controls. Adaptable tools are represented by the set of adaptors of different types (see Example 1):

Example (1) Adaptable language	AD	Adaptable processor
New data	AP	New actions (operators, statements,
	T	controls)
Extension definition	OR	Extension call

The adaptor as a meta-system tool supports adaptable software (language and processors) and hardware flexibility (extension and reduction). Language adaptor as part of adaptable language which is composed from the pragmatic, syntactic, semantic, environment, and examples of new or modified element's component parts (see Example 2):

Example (2) _BL_ <element's pragmatics>
SY <element's syntax>
SE <element's semantics>
CO <element's usage context>
EX <element's examples call>
EL

Adaptor's component parts support flexibility of languages and of processors as component parts of adaptable systems. Adaptors permit the process of software and hardware adaptation to the home-machine interface needs.

Adaptor is represented by the corresponding extender and reducer. The adaptors permit the bottom-up, top-down, and horizontal adaptable (flexible) software's and hardware's development.

On-Off-Line Adaptable Software

The On-Off-Line adaptable software is composed from adaptable language and from corresponding On-Off-Line adaptable processors. Adaptable language is composed from the adaptable basic language, language's adaptors, and derivative (adaptable) elements (extensions). On-Off-Line adaptable processors are represented by the set of Off-Line, On-Line, and On-Off-Line adaptable processors, which implement adaptable languages. Evolution of On-Off-Line adaptable processors is based on the evolution of Off-Line adaptable software (Todoroi & Micusha, 2012).

Adaptable translation methods and models (Todoroi, 2008a, 2008b) are used to demonstrate Off-Line adaptable processors' automatic creation. The last one is composed from the Off-Line adaptable software's basis and Off-Line adaptable software' levels.

Adaptable Software's Basis

Adaptable software's basis is represented by the adaptable basic language, language's adaptors, and derivative elements in community with the adaptable basic system. The last one is represented by the definition, fixation, calling, and reduction adaptable sub-systems. The definition sub-system implements the extension definition. The fixation sub-system fixates the extension definition in the adaptable software. The calling sub-system implements the extension call in the adaptable software. The reduction sub-system creates the individual adaptable software.

The First Level of Adaptable Software

There are distinguished three different types of the first level of adaptable software (the first level of translation complexity On-Line, Off-Line, and On-Off-Line adaptable processors). They are based on the next three types of invented adaptable software's creation methods: (1) the E-T-I-M (Extension's Time Implementation Method); (2) the E-L-I-M (Extension's Level Implementation Method); and (3) the P-T-I-M (Processor's Type Implementation Model).

With the help of E-T-I-M were created adaptable pre-, inter-, and post-processors. The E-L-I-M was used to create L-L (level-level), L-D (level-direct), and L-L-D (level-level-direct) adaptable processors. Were demonstrated the theorems of automatically creation of adaptable processors by the help of the E-T-I-M and E-L-I-M processor's creation methods.

On-Line and Off-On-Line first level adaptable processors can be created by the help of the P-T-I-M using Off-Line adaptable processors developed on the base of E-T-I-M and E-L-I-M processor's creation methods.

The Second Level of Adaptable Software

There are distinguished three different types of the second level of adaptable software: the ELIM-PTIM type, the ETIM-PTIM type, and the ELIM-ETIM type.

The second level ELIM-PTIM type of adaptable software, for example, is represented by the L-L-Preprocessors, L-D-Preprocessors, and L-L-D-Preprocessors. The second level ELIM-PTIM type of adaptable software is created on the base of Extension's Level Implementation and of Processor's Type Implementation Methods.

The second level ETIM-PTIM type of adaptable software is created on the base of Extension's Time Implementation and of Processor's Type Implementation Methods.

The second level ELIM-PTIM type of adaptable software is created on the base of Extension's Level Implementation and of Processor's Type Implementation Methods. The adaptable processors of the ELIM-ETIM type, for example, is represented by the L-L-Preprocessors, L-D-Preprocessors, and L-L-D-Preprocessors. Was demonstrated the possibility of automatically creation of Off-Line adaptable processors of the second level of translation complexity: Off-Line Pre-, Off-Line Inter-, and Off-Line Post-processors and of Off-Line: Off-Line L-L-, Off-Line L-D-, and Off-Line L-L-D-processors.

It is demonstrated (Todoroi & Micusha, 2012) the possibility to realize the second level's adaptable software on the base of translation interactions of the first level's adaptable software.

It is need to demonstrate the process of automatically creation of On-Line and On-Off-Line adaptable processors of the second level of translation complexity.

The Third Level of Adaptable Software

Third level of adaptable software is represented by such types of Off-Line adaptable software as Off-L-L-pre-processors, Off-L-D-inter-processors, and Off-L-L-D-post-processors. They are represented by such types of adaptable processors as Off-Line-L-L-Pre-, Off-Line-L-L-Inter-, Off-Line-L-L-Post-, Off-Line-L-D-Pre-, Off-Line-L-D-Inter-, Off-Line-L-D-Post-, Off-Line-L-L-D-Pre-, Off-Line-L-L-D-Inter-, and Off-Line-L-L-D-Post-processors.

It was demonstrated (Todoroi et al., 2007) the process of automatically creation of these Off-Line adaptable

processors of the third level of translation complexity. The demonstrations of automatically creation of adaptable software of the third level is based on the corresponding demonstrations of automatically creation of the adaptable software of the first and of the second levels of translation complexity.

It is need to demonstrate the automatically creation of On-Line and On-Off-Line adaptable software of the third level of translation complexity.

Adaptable Software Advantage

It is demonstrated (Todoroi & Micusha, 2012) that adaptable tools as basis for creation, application, and development of adaptable software are characterized by a set of advanced linguistic' and processors' features.

It is demonstrated that adaptable languages as part of adaptable software integrate such linguistics' features as (see Example 3):

Example (3) -Multilanguageability

- Universality
- Speciality
- Exensibility
- Dialectability
- Compactibility of Basis
- Reducability
- Effectivity of modification
- Continuity of Human and Machine experience
- Touchability to the Formal Natural Language level of Human-Machine Interactions

It is demonstrated that adaptable processors as part of adaptable software integrate such translation' features as (see Example 4):

Example (4) -Universality

- Mobility
- Transferability
- Cognisability
- Specializability
- Minimizing of Processors' Quantity
- Raising the level of adaptable software to the level of Problem Formulation

Conclusions

The Ms Office and Ms Windows Systems are developed by Software' shell methodology. Microsoft Office for Mac has for long been criticized. Adaptation at hardware levels increases the system capabilities beyond what is possible with software-only solutions. The methodology of the On-Off-Line adaptable processors support development of adaptable software and hardware.

On the basis of adaptable processors of the first level of translation complexity Off-Line adaptable processors it is possible to demonstrate the process of automatically creation of the first and the second levels of translation complexity On-Line and On-Off-Line adaptable processors. The demonstrations of automatically

creation of On-Line and On-Off-Line adaptable software of the third level of translation complexity can be obtained on the base of first and second levels of translation complexity of adaptable processors.

It was demonstrated that adaptable tools as the base of creation, application, and development of adaptable software are characterized by a set of advanced linguistic' and processors' features.

Human social and economic demand and supply for adaptable software in the Information and Knowledge Based Societies is too important.

Adaptable methodology and technology in creation and application of adaptable software permit to develop in the future the research process of applicability of each of the first, second, and third levels of adaptable processors.

Different types of adaptable software will have different domains of its applicability in the process of computerized human-machine intelligent interaction. This process conducts to develop Natural Language Processing Adaptable Software of human-machine interaction.

The adaptable software forms new industry branch of informational technologies of the information and knowledge based societies.

References

- Merten, M., & Gross, H.-M. (2008). Highly Adaptable Hardware architecture for scientific and industrial mobile robots (pp. 1130-1135). *IEEE Conference on Robotics, Automation and Mechatronics*. Chengdu, China.
- Microsoft Kills Virtual PC, VB for Mac. (2007). *Microsoft kills virtual PC, VB for Mac*. Retrieved from <http://www.macnn.com/articles/06/08/07/ms.kills.virtualpc/>
- Microsoft Office 2010. (2010). Ms office 2010. Retrieved from http://en.wikipedia.org/wiki/Microsoft_Office_2010
- Microsoft Office. (2012). *Ms office*. Retrieved from http://en.wikipedia.org/wiki/Microsoft_Office#_note-13#_note-13
- Microsoft Support Lifecycle Policy FAQ. (2010). *Microsoft: Help and support*. Retrieved from <http://support.microsoft.com/gp/lifepolicy>
- Office 365 Home Premium. (2010). Ms office 2010 features and benefits. Retrieved from <http://office.microsoft.com/en-us/products/office-2010-features-and-benefits-HA101812297.aspx>
- Stallman, R. M. (2008). *We can put an end to word attachments*. Retrieved from <http://www.gnu.org/philosophy/no-word-attachments.html>
- Stoica, A., & Andrei, R. (2007). Adaptive and evolvable hardware—A multifaceted analysis (pp.486-498). *Adaptive Hardware and Systems, Second NASA/ESA Conf.*, 5-8.
- Todoroi, D. (1992). *Computer science: The adaptable programming: The basic conceptions*. Chisinau: ASEM Press.
- Todoroi, D. (2008a). Adaptable software shells versus Microsoft Software shells (pp. 84-96). *Journal of Applied Quantitative Methods (JAQM)*, 3(1). Retrieved from <http://jaqm.ro/volume-3,issue-1.php>
- Todoroi, D. (2008b). The software's adaptable shells creation models (pp. 50-53). *Proc. of 32nd Annual Congress of ARA*. Boston, M.A., USA.
- Todoroi, D., & Micusha, D. (2012). *Adaptable software*. Chisinev: ASEM Press.
- Todoroi, D., Micușă, D., & Todoroi, Z. (2006). Adaptable systems linguistic and processing properties (pp. 14-24). *Informatica Economica*, 4(40). Bucharest: ASE Printing House.
- Todoroi, D., Micușă, D., & Todoroi, Z. (2009). Some problems of Ms Office, Adaptive Hardware and On-Off-line Adaptable Processor's development (pp. 615-621). *Proceedings of the 9th Int. Conf. on Informatics in Economy "Education, Research & Business Technologies"*. Bucharest: ASE Printing House.
- Todoroi, D., Micușă, D., Todoroi, Z., & Spataru, Ș. (2007). From recursive programs to the adaptable processors construction (pp. 445-448). *Proc. of 31th An. Congr. of ARA "Pro-Active Partnership in Creativity for the Next Generation"*. Quebec, Canada: Presses Internationales Polytechnique.
- Ugurdag, H. F. (2006). 1st Conference on Adaptive Hardware and Systems—AHS' 2006 June 16-18, 2006. *The First NASA/ESA Conference on Adaptive Hardware and Systems*. Retrieved from <http://ehw.jpl.nasa.gov/events/ahs2006/ahs2006-1st-call4participations.pdf>