PROCEEDINGS

of the Union of Scientists - Ruse

Book 5

Mathematics, Informatics and Physics

Volume 10, 2013

RUSE

The Ruse Branch of the Union of Scientists in Bulgaria was founded in 1956. Its first Chairman was Prof. Stoyan Petrov. He was followed by Prof. Trifon Georgiev, Prof. Kolyo Vasilev, Prof. Georgi Popov, Prof. Mityo Kanev, Assoc. Prof. Boris Borisov, Prof. Emil Marinov, Prof. Hristo Beloev. The individual members number nearly 300 recognized scientists from Ruse, organized in 13 scientific sections. There are several collective members too – organizations and companies from Ruse, known for their success in the field of science and higher education, or their applied research activities. The activities of the Union of Scientists – Ruse are numerous: scientific, educational and other humanitarian events directly related to hot issues in the development of Ruse region, including its infrastructure, environment, history and future development; commitment to the development of the scientific organizations in Ruse, the professional development and growth of the scientists and the protection of their individual rights.

The Union of Scientists – Ruse (US – Ruse) organizes publishing of scientific and popular informative literature, and since 1998 – the "Proceedings of the Union of Scientists- Ruse".

BOOK 5

"MATHEMATICS, INFORMATICS AND PHYSICS"

VOLUME 10

CONTENTS

BOOK 5

"MATHEMATICS, INFORMATICS AND PHYSICS"

VOLUME 10

This is the jubilee 10-th volume of book 5 Mathematics, Informatics and Physics. The beginning was in Spring, 2001, when the colleagues of the former section Mathematics and Physics decided to start publishing our own book of the Proceedings of the Union of Scientists – Ruse. The first volume included 24 papers. Through the years there have been authors not only from the Angel Kanchev University of Ruse but as well as from universities of Gabrovo, Varna, Veliko Tarnovo and abroad – Russia, Greece and USA.

Since the 6-th volume the preparation and publishing of the papers began to be done in English.

The new 10-th volume of book 5 Mathematics, Informatics and Physics includes papers in Mathematics, Informatics and Information Technologies, Physics and materials from the Scientific Conference 'Information Technologies in Education' (ITE), held at the University of Ruse in November 2012 in the frame of Project 2012-FNSE-02.

web: suruse.uni-ruse.bg

AUTOMATIC PROGRAM GENERATION WITHOUT INTERNAL MACHINE REPRESENTATION

Valentin Velikov

Angel Kanchev University of Ruse

Abstract: *This paper describes some of the problems with automatic program generation, using or not internal machine representation.*

Keywords: *program generation, internal machine representation.*

INTRODUCTION

Internal machine representation [16] has many advantages: high-speed, the developer is confident in his system (it is not open source code, so that is not accessible to manipulate by all), generation is allowed in different programming languages and so on. But there is a significant drawback - presumably if has to be a functionally complete system. That means to be expensive, with a long period of development and testing, because it has finished all necessary modules have to be in a complete form: those for import and export, dialog editors of different nature for task descriptions [12] (text, image, natural-languages, etc.), carefully designed internal machine representation, generators to relevant program languages and so on. It is imperative a service team for system development and maintaining to be kept available.

DETAIL DESCRIPTION

Let consider such a functionally complete system, like as **KATIA 3D** or **IBM Web Sphere Lombardi** [7] (Fig. 1):

An internal machine representation (different structures) of the considered object is created. Usually one already knows the subject areas in which we operate. The end users (engineers, designers and other specialists) put their task using different dialog resources to describe the job: it can be different types of graphical interfaces, dialog interfaces, menu systems and others (depending on the specific needs for each type of interface can be a generator creating it).

After the job has been described, the system converts it to the internal machine representation, appropriate to the corresponding subject area [9]. Usually different internal machine representations are maintained, for different subject areas. If necessary, the support team can create a new internal machine representation or update the existing one. It can also be requested a supplement / specification of the job.

Next the necessary application should be created. This can be a running application, documentation, drawings, description (some of the created applications can be used to create others). To create them various generators are used. Again, if necessary, the support team can create a new generator or update an existing one.

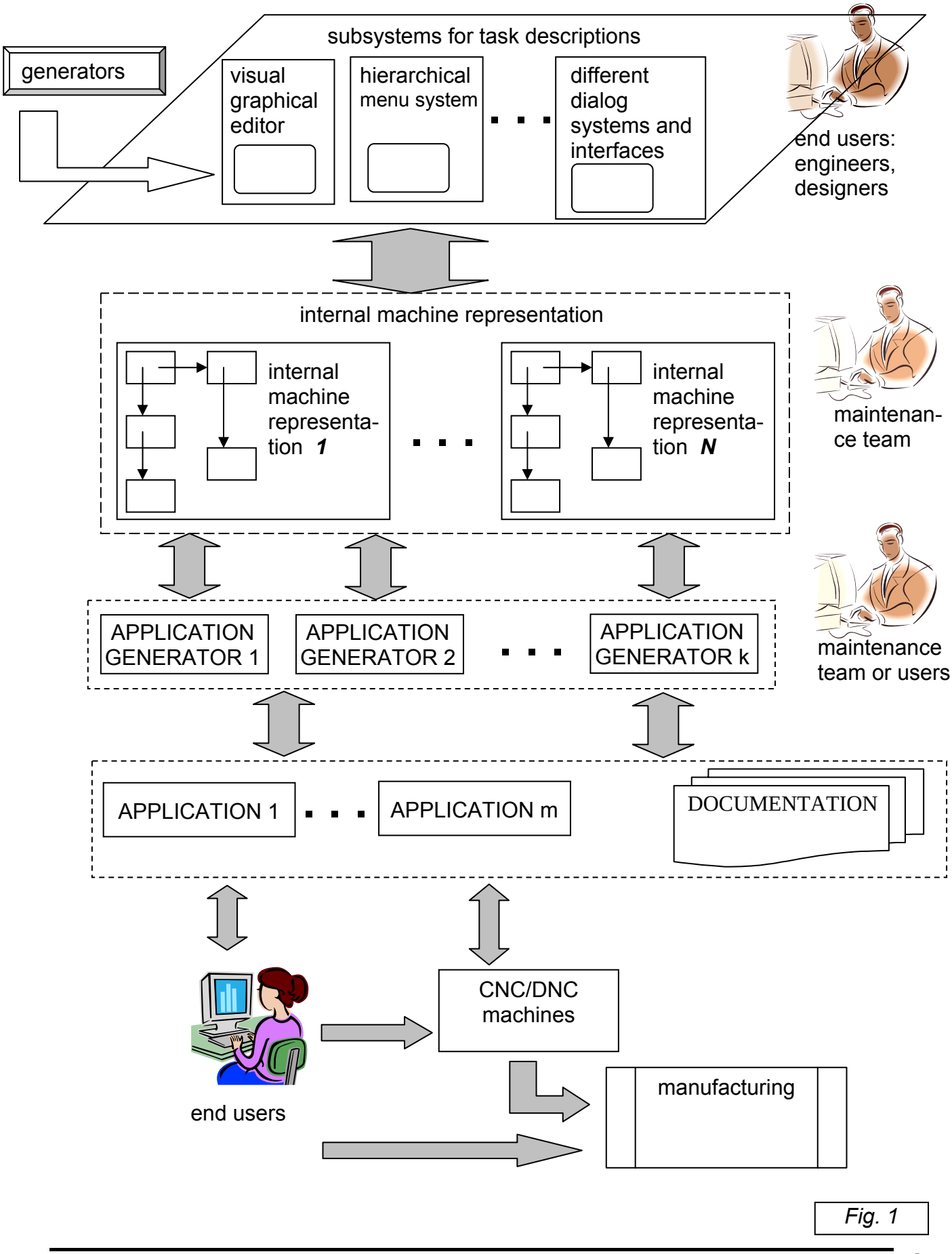
subsystems for task descriptions
generators
visual graphical editor
hierarchical menu system
different dialog systems and interfaces
end users: engineers, designers
internal machine representation
internal machine representa-tion 1
internal machine representa-tion N
maintenan-ce team
APPLICATION GENERATOR 1
APPLICATION GENERATOR 2
APPLICATION GENERATOR k
maintenance team or users
APPLICATION 1
APPLICATION m
DOCUMENTATION
CNC/DNC machines
manufacturing
end users

Fig. 1

It is essential that it is necessary a great support and development team of such a system. Minor modifications of the subject area, adding new ones, creating a new generator - in all cases of manipulation of the system that can not be done by hired for the purpose team without his prior education and training. This is related to time and money again.

It is very important that different generators (for different applications) use the same internal machine representation [10]. Very often one subsystem describes the task, it is fed to the next - to create a project, then it is submitted to the next - to create software and so on. For normal operation of this chain of reciprocity-associated modules it is needed a general (universal) internal machine representation. If an external developer wants to create a new generator and new application or modify an existing one - this is impossible due to the unknown of the internal machine representation.

IS IT POSSIBLE TO SUGGEST SOME OTHER APPROACH?

For a variety of subject areas there are small (relatively simple) systems, making small, well-defined things to automate human activity. Even more - some of them without claim for functional completeness are open source or free. For example: I want to create an information system for library service. It is a need a project to be created with a defined functionality, to describe subject area, to define the database, to create / generate software etc. A system such as ***Enterprise Architect*** [3] or ***IBM Web Sphere Lombardi*** [6], [8] will make a lot of this (or anything) but the price is not low and the maintenance of trained staff will cost as well.

An idea arises: to use multiple unrelated systems [1] (if possible open source or free), each of which makes some of the needful activities. One such a system is the project graphic description (for example: MS Visio for UML-diagrams or [5] - BPM), another - to define the database, a third - to describe data structures (classes [2]) in the program and so on. How different systems to exchange information with each other (i.e. - the result of the work of the first to be used as input for the second)? Different options may be offered, one of which is a common, universal format for information describing a variety of text files, such as XML (eXtensible Markup Language) [19]. Many modern systems support functions for Export and Import of text files and/or variants thereof, including XML-files (Fig. 2).

Thus, developing a problem, business processes can be described in UML diagrams [14], [15] using MS Visio, which comes in handy for drawing but it does not make any logical control over the generated algorithm.

The created (drawing) diagram can be exported as a file in XML-format [4], [17], [18], which is applied as input to the next system (commercial or free) for further processing. And so on, using a lot of internally independent products communicating between by XML-files, we can create a chain of tools that lead us to the final result or bring us closer to it.

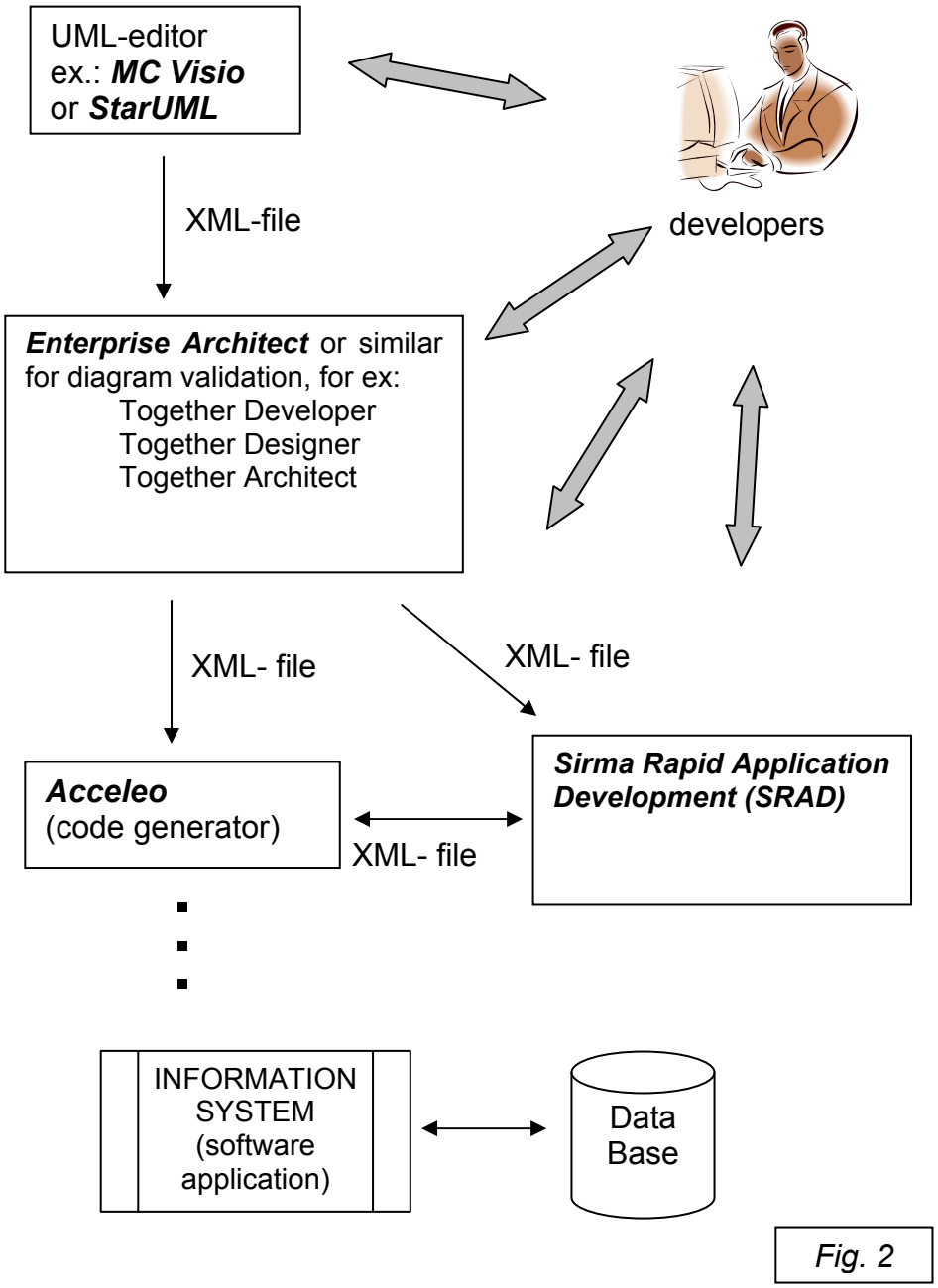

Fig. 2

CONCLUSION

With a good knowledge of the subject area, there are options to select suitable software tools (free or open source [11], [13]) which can automate much of the work in software system creating. Viewed individually, each product works for itself automating "piece" of monotonous work in application development. Passing the results from one system to another (via the corresponding XML-format [4]) we could get the desired end result without using common internal machine representation. Thus, adjusting a chain of open source tools, small companies with insufficient resources can automate some of the activities in the software project implementation.

Moreover: even though we have a commercial functionally complete system of internal machine representation, it is possible to need doing something extra, beyond of the system scope. So in this case we can look for another (external) application and communication between them done in XML-format (Fig. 3).

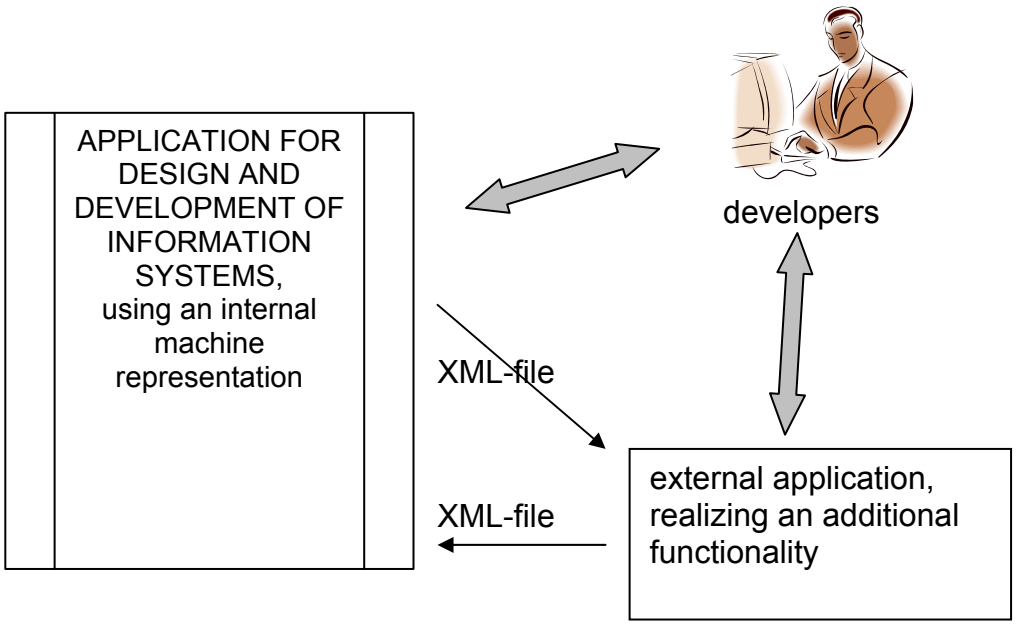

Fig. 3

REFERENCES

[1] AutoSAR, http://www.autosar.org/, 02.2012r.

[2] Class Diagrams, http://en.wikipedia.org/wiki/Class_diagram, 02.2012r.

[3] Enterprise Architect - Model Driven UML Tools, http://www.sparxsystems.com.au/, 03.2013.

[4] Grigorova K., P. Hristova, G. Atanasova, J. Q. Trelewicz, Extracting Business Structures from XML Documents, Intenational Conferense "Automation and Informatics", Sofia, 2005

[5] IBM - Business Process Management (BPM), http://www-142.ibm.com/software/products/us/en/category/BPM-SOFTWARE, 2011.

[6] IBM - WebSphere Lombardi Edition, http://www-01.ibm.com/software/integration/lombardi-edition/, 2011.

[7] IBM - Lombardi architecture, http://publib.boulder.ibm.com/infocenter/wle/v7r2/index.jsp?topic=%2Fcom.ibm.wle.doc%2Fcommon%2Ftopic%2Fteamworks_architecture.html, 2011.

[8] IBM Rational Rhapsody, http://www-01.ibm.com/software/awdtools/rhapsody/, 02.2012r.

[9] Kratchanov K.D., I.N.Stanev et al. Rule Networks: A Style of Programming and an Expert System Toolkit. Artificial Intelligence III. Methodology, Systems, Applications (T.O.Shea and V.Sgurev - eds.). North-Holland, 1988, 265-272.

[10] Kratcahnov K., T. Golemanov, E. Golemanova, I. Stanev. Control Network Programming in SPIDER: Built-in Search Control Tools. Proceedings. New trends in Artificial Intelligence and Neural Networks Conference. Ankara. Turkey. 1997. Pp. 105-109.

[11] Open Source Software, http://en.wikipedia.org/wiki/Open-source_software, 02.2012r.

[12] Stanev I., Generation of Computer Programs Specified through Natural Language Texts, T.N.Sys8, Systems Engineering Division, University of Wales, Cardiff, UK, 47, Mar 2000. Pp. 47.
[13] StarUML, http://en.wikipedia.org/wiki/StarUML, 02.2012г.
[14] Unified Modeling Language (UML), http://en.wikipedia.org/wiki/Unified_Modeling_Language, 02.2012г.
[15] Unified Modeling Language Tools, http://en.wikipedia.org/wiki/List_of_Unified_Modeling_Language_tools, 02.2012г.
[16] Valentin Velikov - Some possibilities for automatic programs generation, Proceedings of the Union of Scientists – Rousse, book 5, Mathematics, Informatics and Physics, 2011, Vol.8, in English, ISSN 1311-9974, pp. 44 – 49.
[17] http://bg.wikipedia.org/wiki/XML, 03.2013.
[18] http://infoman.musala.com/articles/2006-03-xml/article.html, 03.2013;
[19] http://www.xml.com/programming/, 03.2013.

CONTACT ADDRESS
Pr. Assist.Valentin Velikov,
Department of Informatics and Information Technologies
Faculty of Natural Sciences and Education
Angel Kanchev University of Ruse
8 Studentska Str., 7017 Ruse,Bulgaria
Phone: (++359 82) 888 326,
Cell Phone: (++359) 886 011 544,
E-mail: val @ ami. uni-ruse. bg

АВТОМАТИЗИРАНО ГЕНЕРИРАНЕ НА ПРОГРАМИ БЕЗ ВЪТРЕШНО-МАШИННО ПРЕДСТАВЯНЕ

Валентин Великов

Русенски университет "Ангел Кънчев"

Резюме: *Статията дискутира някои от проблемите при автоматизираното генериране на програми при използването на вътрешномашинно представяне и без него.*

Ключови думи*: генерация на програми, вътрешно-машинно представяне.*

ISSN 1314-3077