Item Modeling Concept Based on Multimedia Authoring

Janez Stergar, Andrija Šulić, Marko Brvar
Faculty of Electrical Engineering and Computer Science, University of Maribor

***Abstract*—In this paper a modern item design framework for computer based assessment based on Flash authoring environment will be introduced. Question design will be discussed as well as the multimedia authoring environment used for item modeling emphasized. Item type templates are a structured means of collecting and storing item information that can be used to improve the efficiency and security of the innovative item design process. Templates can modernize the item design, enhance and speed up the development process. Along with content creation, multimedia has vast potential for use in innovative testing. The introduced item design template is based on taxonomy of innovative items which have great potential for expanding the content areas and construct coverage of an assessment. The presented item design approach is based on GUI's – one for question design based on implemented item design templates and one for user interaction tracking/retrieval. The concept of user interfaces based on Flash technology will be discussed as well as implementation of the innovative approach of the item design forms with multimedia authoring. Also an innovative method for user interaction storage/retrieval based on PHP extending Flash capabilities in the proposed framework will be introduced.**

***Index Terms*—Computer Testing, Computer Based Assessment, Multimedia Authoring, Innovative Item Design, GUI, MySQL**

I. INTRODUCTION

With the wide availability of course management software the World Wide Web is becoming increasingly attractive as a mechanism for the delivery of assessment. Many students now make use of such a system their day-to-day studies. Integrating the technology into the assessment brings the assessment regime closer to the learning environment of the students. Computer-based assessment not only automates routine tasks like marking multiple-choice questions, but can enrich student's learning experiences [1]. Therefore for the last decade major companies such as Cisco and Microsoft complement their e-training with on-line learning assessments and certifications.

There are a number of perceived benefits in using computers in assessment, e.g. student response can be monitored, assessment can be offered in an open access environment, assessments can be stored and reused, immediate feedback can be given, assessment items can be randomly selected to provide different exam questions to each student, etc. Thus test-based assessment is essential to achieve an optimal learning process [2].

There is a tendency to associate computer-based assessment (CBA) with automated multiple-choice questions, possibly because it was one of the earliest uses of computer technologies. The restricted range of task types should make only a small contribution to any balanced assessment. The major challenges in delivering a wider diversity of assessment via computer include exploiting the potential of multimedia and interactivity, providing a rich and natural working environment for the student to work on a complex task, collecting richer more open forms of response from the student [3].

One potential limitation for realizing the benefits of CBA in both instructional assessment and large scale testing comes in designing questions and tasks with which computers can effectively interface while still gathering meaningful measurement evidence [5].

Another area concerning CBA are innovations regarding the "observation" space and automatic item development. Automatic item generation involves approaches such as templates for items or "item shells" used to create many items from one or the so called item modeling, in which descriptions of task aspects, or facets, are devised and then implemented across a variety of contexts, or scenarios [5].

II. QUESTION DESIGN IN COMPUTER-BASED ASSESSMENT

The major focus of online testing has been on multiple-choice questions (MCQ). MCQ have a long association with CBA and are relatively easy to convert to an online format. As with all MCQ it is the question design that makes multiple-choice tests effective. An important but time consuming procedure in MCQ examinations is to produce a test that differentiates between levels of intellectual ability. These questions take more time to create but once produced, they offer flexibility in the delivery of examinations and efficiency can be gained through a computer calculating and recording student marks.

As with all assessment, MCQ should test clearly identified learning outcomes and be integrated into the course rather than treated as an afterthought. An advantage of using MCQ online is the ease in scheduling an examination once you have a pool of questions. Known as "question banks", these are databases of large numbers of questions in one or more subject area [6].

For any particular exam questions can be selected from the database at random. Questions in the database are organized by level of difficulty or different values within a standard question randomized to create slightly different questions. Having randomized questions with the same

structure and level but different content is considered to also deter cheating as there is no advantage by seeing another student's exam paper [8]. For this to be effective the database needs to be large enough to prevent high levels of repetition.

Even with careful question design it is difficult to assess the highest level outcomes, like synthesis and evaluation, by using multiple choice responses [7]. Nevertheless if e-learning designers adopt only MCQ format alone as the focus of assessment formats in the emerging field of computer based assessment, much of the computer platform's potential for rich and embedded assessment could be sacrificed [5]. Our intent therefore was to extend the MCQ-forms with a different kind of questionnaire using item shells.

According to the taxonomy of item types of the "Multi-faceted Organization Scheme" in [9] and item type dissemination in [5], six major item categories including MCQ item type can be emphasized: Selection/Identification type, Reordering/Rearrangement type, Substitution/Correction type, Completion type, and Construction (Fig. 1). We deliberately excluded the Presentation/Portfolio item type as it is currently not foreseen for implementation into our proprietary CBA system.

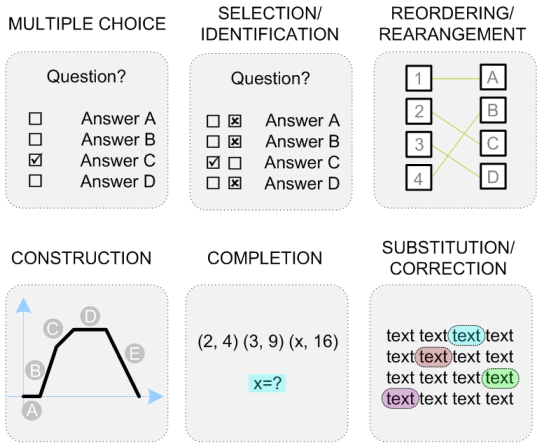

Figure 1. The implemented item types.

Multimedia is very promising in content design and use in innovative testing [4]. Hence the construction of the dynamic GUI environment supporting multimedia content for implementation of different dynamic item types was realized with Flash authoring environment.

III. THE ON-LINE ASSESSMENT ENVIRONMENT

The used experimental framework for CBA including dynamically driven GUIs is a PHP script based framework developed in Laboratory for Digital and Information Systems (LDIS) at the Faculty of EE and Computer Sciences University of Maribor.

Currently, the main use of the LDIS proprietary on-line assessment framework is for summative rather than formative testing. This is achieved through on-line quizzes. As issues of security, access and equity are not currently resolved, onsite invigilated summative computer assisted testing is performed (personal ID required).

The CBA system is in experimental phase, therefore our main concern is towards question design. As suggested by [7] concern hast to be devoted to learning outcomes that have to be identified for the courses and evaluated if the tests address those learning outcomes. Currently three types of tests are supported in our CBA environment:

- Chapter exams which devote the attention to a particular topic as a reflection of its importance in the course,
- Topic exams which address chapters with common topics and
- Final exams which assess topics which comprehend the whole course.

Assessment is integrated into the course in the form of on-line quizzes on a regular basis, also additional quizzes are generated for preparation (students appreciate the opportunity to try out sample questions). Randomized selection/generation of questions/quizzes as suggested in [8] is implemented. Additionally a randomization of MCQ answers was implemented to achieve a different structure of content to deter cheating as there is no instant advantage by seeing another student's exam page. Currently our database for three courses includes over 1000 questions supporting six item types (only tree of them are used). The topics examined are from the area of Multimedia Systems, Digital Structures and Computer Networks.

IV. THE ITEM TYPE MULTIMEDIA AUTHORING ENVIRONMENT

In the today's world where every digital interaction – whether in the classroom, the office, the living room, the airport, or the car – is a powerful, simple, efficient, and engaging experience, Flash Player is widely used to deliver these experiences. Flash content can be viewed on nearly all computers. The Flash technology, in a general sense, may be one of the most widely available technologies used on the WWW. For developers, the ability to program one multimedia presentation that can be viewed the same on nearly all computer platforms makes the technology very appealing. Hence our item design framework is based on the Flash platform.

A. Flash Platform Architecture

The Flash multimedia authoring environment has evolved into a complete platform across browsers, operating systems, and devices. It provides a robust, end-to-end architecture for delivering Rich Internet Applications (RIAs), content, and communications across multiple platforms and devices [10].

The Flash Platform has a layered architecture that encompasses the following key elements:

- *Rich client*: The core of the platform is Flash Player, which is deployed in browsers on personal computers, on devices as Flash Lite, and on the desktop in the developer release of Macromedia Central™, which supports occasionally connected computing.
- *Programming model*: A consistent model for developers that combines ActionScript (an ECMAScript procedural language just like JavaScript) and MXML, which is an XML-based declarative language for rapidly developing user interfaces and data binding. The programming model extends across rich clients and the experience servers.

– *Experience servers*: A set of server technologies that work with existing back-end systems built on J2EE, .NET, or standard web servers. These server technologies include Flex Enterprise Services, Flash Media Server, and FlashCast™ server. The Flex enterprise services enable the integration of rich user interfaces with back-end services in application servers. The Flash Media Server enables streaming, scalable audio-video delivery. The FlashCast server enables telecom carriers to deliver a better, more responsive user experience through background delivery of data to mobile devices.

In addition the Flash authoring tool offers comprehensive features for creating sophisticated interactive content, and RIAs accelerating development with productivity enhancements that simplify the creation and incorporation of animation, interactivity and rich media assets.

The elements of the Flash Platform architecture include the following [11]:

– High-performance servers and data services that operate seamlessly in existing J2EE and .NET environments to efficiently deliver Flash-based applications, content, and communications without increasing administrative burden or infrastructure requirements.
– A robust, standards-based programming model grounded in industry best practices.
– Comprehensive development tools that support a team approach, as well as a variety of third-party independent software vendors.
– A client runtime that delivers a consistent user experience across the widest range of platforms and devices.
– A flexible communication and collaboration solution that addresses the increasingly complex communications issues of today's organizations.
– An innovative, tested method of delivering applications, content and on-demand data services to mobile devices.

B. Flash and CBA Forms

The suitability of the Flash environment for on-line assessment evolves with learning interactions. Each individual Flash assessment form can send tracking information to a server-side learning management system (LMS) that complies with the Aviation Industry CBT Committee protocol (AICC). In our case this is the experimental assessment platform developed in our laboratory. Additionally, the assessment templates track cumulative results from a sequence of interactions and can pass them along to the LMS using an enhanced data tracking functionality that conforms to either AICC or Shareable Content Object Reference Model (SCORM) standards [12], [16], [17].

With Flash interactive online assessment item forms can be dynamically driven. Using the Flash as the construction environment for item form design has numerous benefits:

– Anyone with a Flash-enabled web browser can use the instructional content.
– The interface can be customized to meet the user needs.
– Because of using Flash, high-quality interfaces that load quickly can be created and look the same on different platforms.
– Interactions to online courses can be easily added with the Flash learning interaction components, which provide a simple interface for entering data without writing code.
– Each individual Flash learning interaction can send tracking information to a server-side LMS that complies with the AICC protocol.

Additionally, the quiz templates track cumulative results from a sequence of interactions and can pass them along to the LMS using an enhanced data tracking functionality that conforms to either AICC or SCORM standards.

C. Action Script

The implemented ActionScript language is a procedural programming language similar to JavaScript. ActionScript supports object-oriented development and strong data typing of variables. Flash Player contains a virtual machine that executes compiled ActionScript code in exactly the same way across operating systems and devices. Flash Player is event-driven – it supports asynchronous events as they occur, which lends itself well to building responsive Internet applications. ActionScript also provides a robust set of APIs and components for building content and applications.

There are two ActionScript versions. The first, ActionScript 1.0, is a simple scripting language targeting interactions, such as a user clicking a button. The second, ActionScript 2.0, is an object-oriented programming language developed for complex programming tasks, such as content and data manipulation. ActionScript is an ECMAScript6 language; it is therefore a close relative of JavaScript, with many syntax similarities. However, the ActionScript client model is based on movie clips, text fields, and sounds; and the JavaScript client model is based on windows, documents, and forms [14].

V. ITEM FORM TEMPLATES

The assessment templates of the LDIS CBA are designed for scenarios in which interaction-based assessment is required and tracking is necessary. The assessment learning interactions are graphically designed to fit into the assessment format. The assessment templates contain a mechanism that counts a cumulative score and starts and stops the necessary tracking.

Computer operating systems do not consistently or securely protect one application, such as Flash Player, from another (except to some limited degree when multiple user IDs are used). Other software running on the platform might include malevolent software, such as viruses and worms. A Flash application should protect against using such malevolent pieces of software in its execution [15]. Thus security is an important matter in CBA systems.

A. Client Run Time Environment

The client runtime environment (Flash Player) runs Flash applications (SWF files). Flash content is delivered

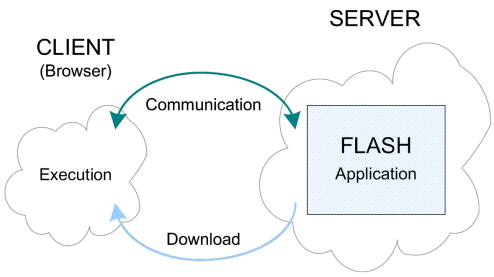

Figure 2. The client/server data-flow framework.

as a series of instructions in binary format to Flash Player over web protocols in the precisely described SWF (.swf) file format. The SWF files themselves are typically hosted on a server and then downloaded to, and displayed on the client computer when requested (Fig. 2). SWF files consist of multimedia content (vectors, bitmaps, sound, video) and binary ActionScript instructions.

B. *The adaptable Item Form Framework*

The main Item Form application designed with Flash is executed at client-side. Thus when the user visits a HTML page, where the Flash application is embedded, the whole page, including the application executes in the client's browser (Fig. 2, Phase 2 - Execution). That excludes any direct execution of server side scripts and leaves a gap between the exam and the database used for tracking.

Prior to client side execution security parameters are exchanged. As mentioned, Flash is AICC/SCORM compliant supporting client side (local) user interaction tracking/saving. Nevertheless there is no support for server side data storage e.g. MySQL interaction. This gap can be filled with PHP, which provides data integration.

Flash supports loading and sending of variables through the *LoadVars* class in ActionScript. This class provides essential mechanisms for communication between Flash and PHP. Data is sent though a channel in a HTTP form. The LoadVars class supports GET and POST methods, which have the same function as in the HTTP protocol, including the support for input parameters. Variables can be sent/retrieved from a PHP file with the *LoadVars.sendAndLoad()* function.

Data is passed prior the script execution and retrieved based on script return values (Fig. 2, Phase 3 - Communication).

The designed application includes three PHP scripts that function independently (Fig. 3, Fig. 4):

– *load.php*, which is used for loading Item Forms (question data) into the GUI,

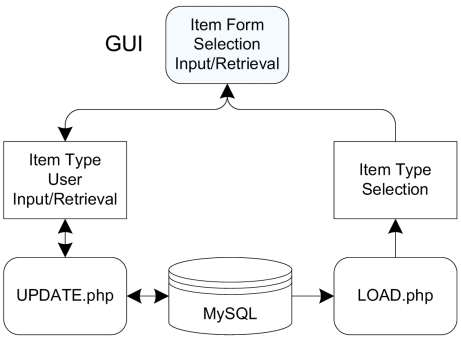

Figure 3. A diagram of the I/O Item Type input/selection procedure.

– *update.php*, which is used for tracking and statistics and

– *save.php*, which is used by the question creation mechanism to save item forms to the database.

Each of the scripts contains a set of SQL queries and methods that function independently and universally for each question type (Fig. 4). When a script gets executed with *sendAndLoad()* from the client's Flash application, with needed input variables, it first parses the variables to ensure their validity and then, based on the input, executes a set of SQL queries that modify or retrieve data from the database. The retrieved data is then sent back to the Flash application, which is in standby for its retrieval.

C. *Current Item Forms*

In our experimental CBA environment currently two forms are used in addition to a MCQ form. As discussed in previous chapters the user has the ability for interactive creation of items. This is achieved through purposely designed GUI (Fig. 5). The adaptable GUI is constructed of 4 subfields. In the upper left subfield – the item form field (Fig. 5– field 1.) the item type is chosen. In the

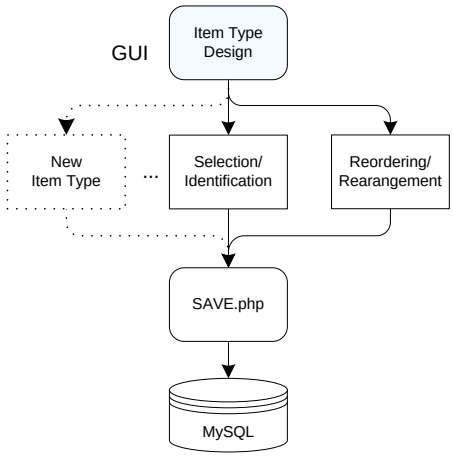

Figure 4. Item Type design flowchart.

middle left field – the text field (Fig. 5 – field 2.) the item text is added. The lower left field – the graphic field (Fig. 5 – field 3.) is intended for graphical input. JPEG, PNG and other common image formats are supported. The lower right field (Fig. 5 – field 4.) is the central field. It is intended for source and target specification. The input is universal. The final form is created through selection in the item type field. This field is intended for future expansion allowing other more sophisticated forms (not yet implemented).

D. *CBA Item Form Security*

Hypertext Transfer Protocol (HTTP) defines how messages are formatted and transmitted, and what actions web servers and browsers should take in response to various commands. HTTP is called a stateless protocol, because each command is executed independently, without any knowledge of the commands that came before it. HTTP is an insecure protocol subject to a variety of security weaknesses, so it is not appropriate applications that transmit or provide access to sensitive data.

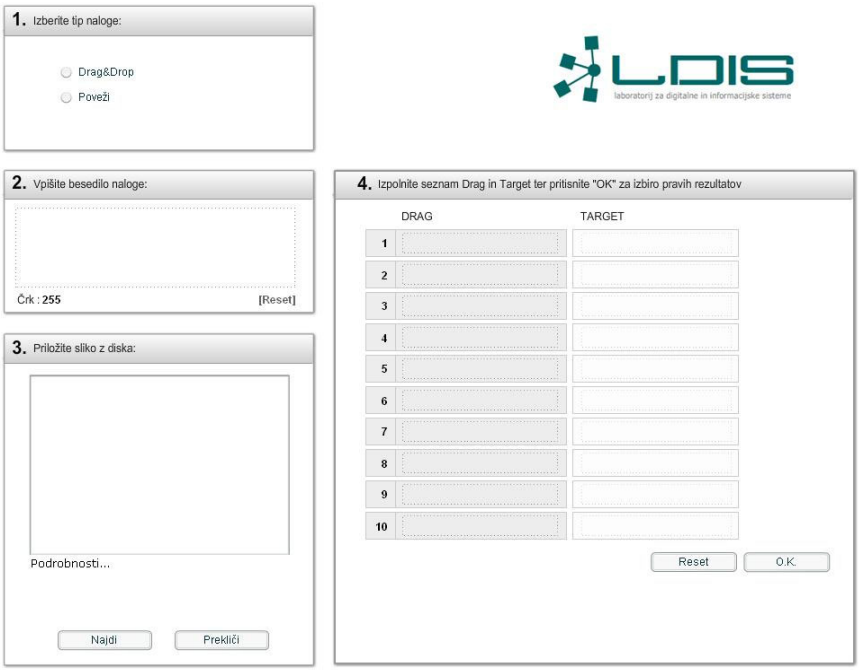

Figure 5. The item type item shell GUI.

Flash Player is designed to provide a robust environment to ensure security and privacy for the author, user, host institutions, and any of their respective data. The Flash Player client runtime provides a comprehensive security architecture that allows content to access functionality only if that content has been explicitly granted permission to use that functionality. Generally, these permissions are of the type read or send. Permissions are granted to a SWF file by the administrative user, end user, or website based on the origin of that SWF file. Authors may also grant permissions to specific SWF files.

With the Flash Player client runtime comprehensive security architecture potentially security breaches like unauthorized access to data, unauthorized access to end-user information and unauthorized access to host system resources have been taken care with code isolation [15].

A Flash application can potentially express actions like reading files, making connections over the network interface or contacting other SWF files. Each of these operations is, in fact, ultimately performed by routines included in and controlled by the native Flash Player code (not by any third-party or application code), and then only after checking and enforcing all applicable access policies as established by the security model and the current runtime security controls.

VI. CONCLUSION

We introduced a novel item shell design framework based on a multimedia authoring environment. The presented approach of modern question design is based on taxonomy of innovative items. We expanded the current MCQ assessment item type of the proprietary experimental CBA system with innovative item type support and additional enhancements – improvements on quantity and quality of information gathered and improvements on efficiency of gathering/processing information with improved fidelity.

The presented item design framework is based on user friendly GUI for item type selection prior question design with secure user interaction tracking/retrieval. The concept of user interfaces based on state-of-the-art underlying multimedia authoring technology was extended with the open source technology of user interaction backup/retrieval based on MySQL. The introduced framework is independent of the used proprietary CBA application thus having the potential of an open source CBA environment implementation, e.g. Moodle, extending the basic MCQ item type form. Flash Quiz templates are supported in Moodle however as to our knowledge no external storing end result tracking for Flash based item types is provided.

Additionally we proposed a scheme for server side data storage support not supported by Flash authoring environment by using MySQL also applicable in Moodle. This gap of unsupported secure server side storage was solved with PHP, which provided data integration.

REFERENCES

[1] S. Brown, J. Bull, P. Race, Eds., *Computer-assisted assessment in higher education*, Kogan Page, London, 1999.

[2] J. R. Anderson, F. G. Conrad, and A. T. Corbett, "Skill acquisition and the lisp tutor," *Cognition Sciences*, vol. 13 (pp. 467–505), 1989.

[3] H. Bukhard, D. Pead, "Computer-based assessment: a platform for better tests?" in *Whither Assessment*, C. Richardson, Eds., Qualifications and Curriculum Authority, 2003.

[4] I. Cheng, A, Basu, "Improving Multimedia Innovative Item Types for Computer Based Testing," in *Proceedings of the Eighth IEEE International Symposium on Multimedia* (ISM'06), IEEE Computer Society, 2006.

[5] K. Scalise, B. Gifford, Computer-Based Assessment in E-Learning: A Framework for Constructing "Intermediate Constraint" Questions and Tasks for Technology Platforms. Journal of Technology, Learning, and Assessment, 4(6), 2006.

[6] M. Thelwall, "Computer-based assessment: a versatile educational tool," in *Computers & Education*, vol. 34, Nr. 1 (pp. 37-49), 2000.

[7] N. Pritchett, "Effective question design," in *Computer-Assisted Assessment in Higher Education* eds. S. Brown; P. Race and J. Bull (pp. 29-37), 1999.

[8] J. Harvey, N. Mogey, "Pragmatic issues when integrating technology into the assessment of students," in *Computer-Assisted Assessment in Higher Education* eds. S. Brown; P. Race and J. Bull, pp. 7-20, 1999.

[9] R. E. Bennett, "On the Meaning of Constructed Response," in *Construction versus Choice in Cognitive Measurement: Issues in Constructed Response, Performance Testing, and Portfolio Assessment* R. E. Bennett, Ward, W. C. (Ed.), (pp. 1–27). Hillsdale, NJ: Lawrence Erlbaum Associates, 1993.

[10] K. Lynch, "The Flash Platform - Delivering Effective User Experiences Across Browsers, Operating Systems, and Devices," white paper, Adobe (Macromedia), 2005.

[11] V. Karvir, "Delivering Enterprise Applications, Content, and Communications with the Flash Platform," white paper, Adobe (Macromedia), 2005.

[12] B. Nelson, "Using Macromedia Flash MX learning interactions," *Macromedia Flash Support Center*, http://www.adobe.com /support/flash/applications/learning_interactions/, Doc ID 9906, 2002.

[13] S. Gowin, "Using a Macromedia Flash MX quiz template," *Flash Support Center*, http://www.adobe.com/support/flash/applications /quiz_tutorial/, Doc ID 9921, 2002.

[14] B. M. Michelson, "Macromedia's Flash Platform – Bringing Rich Experiences to the Masses," Patricia Seybold Group, Report DOI: 10.1571/psgp6-16-05cc, 2005.

[15] A. Ludwig, "Macromedia Flash Player 8 Security," white paper, http://www.adobe.com/devnet/flashplayer/articles/flash_player_8_ security.pdf, 2005.

[16] AICC Independent Test Lab Subcommittee, AICC/Web-Based CMI Certification Testing Procedures, rev. 1.5, 2000.

[17] Advanced Distributed Learning (ADL), Sharable Content Object Reference Model (SCORM) 2004 3rd Edition Overview, 2006.

AUTHORS

Janez Stergar is with the Laboratory of Digital and Information Systems, Institute of Electronics and Telecommunications, University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova ulica 17, 2000 Maribor, Slovenia, (e-mail: janez.stergar@uni-mb.si).

Andrija Šulić is with the Laboratory of Digital and Information Systems, Institute of Electronics and Telecommunications, University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova ulica 17, 2000 Maribor, Slovenia (e-mail: andrija.sulic@uni-mb.si).

Marko Brvar is with the Laboratory of Digital and Information Systems, Institute of Electronics and Telecommunications, University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova ulica 17, 2000 Maribor, Slovenia (e-mail: marko.brvar@uni-mb.si).

Manuscript received 7 January2008. Published as submitted by the authors.