A Systematic Approach to Generation of New Ideas for PhD Research in Computing

Vladimir Blagojević[†], Dragan Bojić, Miroslav Bojović, Miloš Cvetanović,
Jovan Đorđević, Đorđe Đurđević, Bojan Furlan, Slavko Gajin, Zoran Jovanović,
Dragan Milićev, Veljko Milutinović, Boško Nikolić, Jelica Protić, Marija Punt,
Zaharije Radivojević, Žarko Stanisavljević, Saša Stojanović[1], Igor Tartalja,
Milo Tomašević, and Pavle Vuletić

Email: {name.surname}@etf.bg.ac.rs
University of Belgrade - School of Electrical Engineering,
Department of Computer Engineering and Informatics,
Bulevar kralja Aleksandra 73
RS-11120 Belgrade, Serbia

Keywords

PhD Research, Idea Generation, Research Methodology, Research Idea Classification, Critical Thinking

Abstract

This article represents an effort to help PhD students in computer science and engineering to generate good original ideas for their PhD research. Our effort is motivated by the fact that most PhD programs nowadays include several courses, as well as the research component, that should result in journal publications and the PhD thesis, all in a timeframe of three to six years. In order to help PhD students in computing disciplines to get focused on generating ideas and finding appropriate subject for their PhD research, we have analyzed some state-of-the art inventions in the area of computing, as well as the PhD thesis research of faculty members of our department, and came up with a proposal of ten methods that could be implemented to derive new ideas, based on the existing body of knowledge in the research field. This systematic approach provides guidance for PhD students, in order to improve their efficiency and reduce the drop-out rate, especially in the area of computing.

---

† Deceased 22 January 2015.

[1]Corresponding author. E-mail address: sasa.stojanovic@etf.bg.ac.rs
Tel.: +381 3218385; fax: +381 11 3248681.

Introduction

During the past decade, in many European countries, the process of restructuring higher education system according to the Bologna process, brought transition to three-level (bachelor, master, PhD) degrees. This restructuring imposes serious changes in the practice of doctoral studies: instead of awarding the PhD degree based exclusively on academic research, with practically unlimited duration, the new doctoral programs additionally require successful completion of a regimen of coursework. The workload of PhD students is evaluated by at least 180 *European Credit Transfer and Accumulation System* (ECTS) credits beyond the master's level, and should be finished in three to six years, according to higher education regulations.

In the previous higher education system, nominal duration of the first-level engineering studies was 5 years, which is equivalent to the current total duration of bachelor-plus master-level studies. Our calculation of time period between finishing first-level studies and achieving doctoral degree, for faculty members of our department in the area of computing, shows that the average duration of this period was 10.75 years, including 6 master courses, master thesis and the PhD thesis (the average period for PhD, after obtaining the master degree, was 5.79 years). This calculation applies to 18 faculty members, who graduated according to the previous higher education rules, based on the research-only PhD (without the coursework). The goal to get to the same point in three to six years for present doctoral students, with the requirement to take 9 courses, publish at least an article in a journal from the Thomson Reuters Journal Citation Reports (JCR) list, and write the PhD thesis, has proved to be extremely demanding. At our school, the first generation of PhD students were enrolled in the new PhD program in computing in 2007, and since then, 60 new students were enrolled each year. However, only about 50 candidates have graduated so far, including only two members of our department. Although the program nominally lasts three years only, most of the candidates finish it in six years, since they are employed in the industry on the regular basis. In addition to economic and logistic reasons, one of the main obstacles in achieving this goal in a predetermined timeframe may be found in the high latency of inventing sufficiently profound research topics and generating results of scientific value. Therefore, most of PhD students express their eager desire to be guided by an appropriate methodology, which makes the motivation for this work.

The area of computing worldwide has some specific characteristics, which may result in longer actual duration of PhD studies. The opportunities to get employed with master degree are currently better in computing than in similar technical disciplines. On the other hand, the nature of PhD research, which is based not only on theoretical mathematical models or measurements, but also on system implementation and programming, often takes more time and efforts to be done. Finally, publication habits are different in computing compared to other scientific and technical disciplines, while the formal requirements for PhD, regarding published articles are typically the same. For example, analysis presented in [Filipi2011] shows that for each citation that an article receives in the area of computing, an article in general engineering receives 4.64 citations, an article in physics receives 11.9 citations, while an article in molecular biology receives 32.89 citations, according to the analysis applied to all papers indexed by the ISI Thomson Reuters, 1996-2006. These results show the focus of publishing in archive journals in

other disciplines, while the focus in computing, caused by rapid changes, is on conferences, project reports and other less formal publishing forms on the Internet. For this reason, Google Scholar as a less formal source than Web of Science (WoS), shows significantly higher indicators' scores than WoS for computer scientists, roughly five times for article-based indicators and eight times for citation-based indicators [Franceshet2010]. Having in mind our goal to bring more efficiency in generating ideas for PhD, publishing the research results and thus reducing the number of PhD students who drop out in this process, we have started this work as a follow up of four previous articles on different aspects of research conducting methodology [Milutinovic1996, Milutinovic2008, and Milutinovic2015] and research presentation methodology [Milutinovic1997 and Omerovic2010].

There are numerous studies related to systematic creativity methods in engineering design. [Ogot2006] explored three dominant learning styles in the engineering education literature. [Yoram2012] conducted analysis of Advanced Systematic Inventive Thinking (ASIT) using the C–K theory, a design theory that offers a formal model of creative thinking. [Horowitz1999] was dealing with a set of sufficient conditions that characterize creative engineering solutions and their empirical and psychological validation. Several learning and teaching styles in engineering education were explored in [Felder2002]. Similarly, many studies tackle methodologies for research innovation in science and engineering [Faulkner1999, Linn1987, Prost2009], as well as the process of authoring a PhD thesis [Dunleavy2003, Kearns2008]. Finally, understanding and explanation of creativity from different perspectives is an important issue [Stierand2011, Dorfler2010].

This article builds on the top of these representative studies, especially [Milutinovic2015], and tries to systemize existing methods of innovation into a set of 10 different methodological approaches to innovation in computer science and engineering. Our aim is to identify and classify various methods of innovation that led to well-known research contributions in computer science and engineering in the past, in order to provide PhD students with some potentially useful methodological guidelines and encouragement for their research. As a case study, we will show how the PhD thesis research of the members of our department (and authors of the article) fits into the proposed classification.

Classification of innovation methods

Generally speaking, scientific innovations may be classified in two basic categories: (1) Revolutionary, for paradigm-shifting breakthroughs, and (2) Evolutionary, for non-paradigm-shifting improvements of existing solutions.

In the category of Revolutionary innovations, all ideas, in their essence, have only one basic characteristic: Creation of a genius inspired by an undeterminable cause and realized through a thinking process that is extremely difficult to define and classify. In the category of Evolutionary innovations, however, ideas may belong to various patterns, and this article observes 10 different classes of ideas (methods for generating ideas), leading to important evolutionary innovations in the past.

In a research process that should lead to a PhD thesis research, the following phases may be observed: (1) precise problem specification, (2) studying of related work and existing

approaches, (3) generating an idea for a new solution, (4) formulating an essence of the approach, (5) qualitative analysis, order of complexity estimation, and comparison with the state-of-the-art approaches, (6) analysis of valid assumptions and conditions for the solution, (7) formulating details of the solution, (8) quantitative analysis in spatial and temporal domain and comparison with state-of-the-art solutions, (9) implementation analysis, and (10) determining drawbacks of the solution and proposal for future research avenues. Definitely, this scenario is typical for a PhD thesis research which is based on a hardware or software engineering innovation, which overcomes existing solutions in quality and/or performance. For theses based on comprehensive surveys and comparison of existing solutions or theses that focus on theoretical contributions, this scenario would not be appropriate. Generally, we have no intention here to give a prescription for complete PhD thesis research development process. We just want to point to the step in the typical process for the targeted thesis research type, where our methodology may help both the candidate and the supervisor. Obviously, the point in this process where our classes of methods for generating ideas for innovations could serve as a road sign for PhD candidate is phase 3.

The 10 classes presented in this article should be considered only as idea generation guidelines, not as orthogonal classes such that each and every idea belongs to only one of them. In other words, an idea may belong to a number of classes, i.e., it may be characterized with properties of several classes presented here. Also, we do not consider that our classification is closed, meaning that it does not cover all possible ways of generating new ideas for PhD research. Referring to the UML 2 terminology [UML], specifically to the *generalization set* notation, our classification may be described as: *Overlapping* and *Incomplete*. Each class name in the classification has a unique initial, so single-letter class description is unambiguous.

The rest of this section respects the following template for each particular idea generation class: (a) description of the idea generation method, (b) a figure that illustrates the method, and (c) one or more examples that illustrate the method. All used examples consider well known innovations, because our criterion to include an example was that it (i.e., the innovation based on the related idea generation class) or its crucial elements are taught in the curriculum of computer engineering, software engineering, information systems, information technologies, and computer science and well described both in formal literature and informal sources such as the Wikipedia web site[2].

Mendeleyevization (M)
Description. If one of the categories in an existing taxonomy of a problem domain includes no examples, it first has to be checked why is that so. If it is so because it makes no sense, an appropriate explanation is in place. If it is so because the technology or the applications are not yet ready for such an approach, one could act in the same way as the famous chemists Mendeleyev: Empty positions in any classification are potential avenues leading to new inventions. We refer to such an approach as Mendeleyevization (M). Precisely, any real innovation inherently fits to an empty place in some classification. However, a researcher is sometimes unaware of the classification, or the classification does not exist at the moment when the new approach is being invented. We consider

---

[2] http://www.wikipedia.org/

Mendeleyevization as a method of generating a research idea only in the case when a researcher is explicitly aware of the classification and the empty places in the classification, which influences an idea for the innovation. This class represents a top-down approach of idea generation.
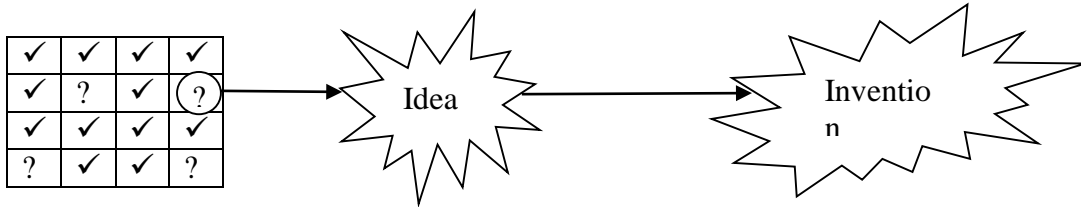

Fig. 1 The existing classification is represented with the table were some cells contain existing solutions (✓), while the others are empty (?); empty-cell analysis may lead to an idea for innovation

Examples. The famous taxonomy of computer systems by Mike Flynn (SISD, SIMD, MISD, MIMD) [Flynn1966] initially included no examples of the MISD (Many Instructions Single Data) type. Systolic arrays [Kung1979], which may be classified as MISD computers, as well as the space shuttle flight computer, which works on this principle to achieve fault tolerance [Spector1984], appeared years later. We could assume that Flynn's taxonomy influenced the inventions.

Generalization (G).
Description. Frequently, there are many versatile concrete solutions to a problem, but there is no common model that could encompass all of the existing solutions. Somebody may catch the important common properties of the existing solutions and could make an abstraction that presents a common model or a language (notation and semantics) for describing each particular solution. Such an abstraction may help in producing a number of new solutions of the problem. We refer to such an approach as Generalization (G). Contrary to Mendeleyevization, this class represents a bottom-up approach to idea generation, based on inductive reasoning.
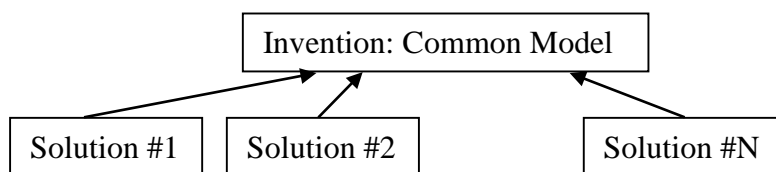

Fig. 2 Generalization of particular concrete solutions to a common model.

Examples. The Harel's state-charts [Harel1987] that represent an abstraction of finite-state automata, and Petri nets [Petri1966] that represent an abstraction of concurrent flows. Also UML [UML] that combines specific properties of a number of previously invented modeling languages.

Specialization (S).
Description. Starting from a well-established general approach, someone could derive a specific knowledge/technology for a specific domain. We refer to such approach as Specialization (S). Similarly to Mendeleyevization, and contrary to Generalization, this

class represents a top-down approach to idea generation, based on deductive way of thinking.
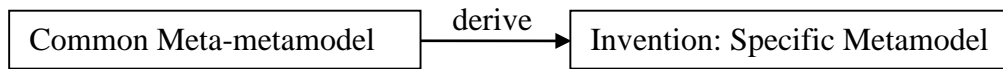

Fig. 3 Deriving specific metamodel from common meta-metamodel.

Examples. Starting from a common meta-metamodel, someone could derive a specific metamodel (e.g., , language) for specific domain. Examples are development of the domain-specific metamodel BPMN (Business Process Model and Notation) [BPMN] based on the common meta-metamodel MOF (Meta-Object Facility) [MOF]. Another example of specialization is partial evaluation, which is used as a technique for different types of program optimization [Jones1993]. The main goal of optimization is to produce new programs which run faster than the originals while being guaranteed to behave in the same way.

Revitalization (R).
Description. Sometimes, there is some theoretical invention that is practically dead, since the technology is not ready to support it, and the invention becomes forgotten. In the meantime, the technology upgrades, but nobody is aware that the existing theoretical invention may revive, until a new idea is born to apply the new technology on the old invention. We refer to such approach as Revitalization (R).
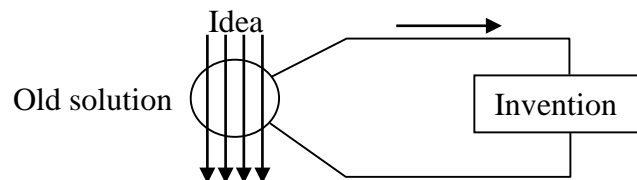

Fig. 4 A dead theoretical invention (resistor) becomes alive when a new idea (variable magnetic field) uses a new technology (inductor).

Example. Computational model of artificial neural networks (ANN) had been invented in early 1940s [McCulloch1943], but it was revitalized and research in the field exploded in early 1980s [Hopfield1982], when the technology of parallel processing matured.

Crossdisciplinarization (C).
Description. Many times, good new ideas appear if some solutions (models, algorithms, mechanisms – not only in computer science) are ported from one field to another field, along the lines of cross-disciplinary research methodologies and applied analogies (Crossdisciplinarization). The degree of the solution modification during cross-disciplinarization may vary. On the first end of scale, the solution may be ported directly, and only the interpretation of related variables is different. On the other end of the scale, just an analogy is used to generate a new idea from some existing solution in a different field, so the new solution has almost nothing to do with the initial solution.
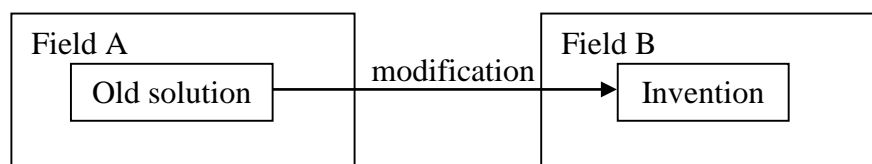
Examples. Popular examples include introduction of mathematical neural networks inspired by biological neural networks [Hopfield1982], or introduction of genetic algorithms based on principles of evolution of live organisms [Holland1975].

Implantation (I).

Description. A new solution is invented by implanting a resource into an existing solution. The characteristics of the new solution overcome the mere sum of the characteristics of the old solution and implanted resources; it brings a new quality or a significant performance gain.
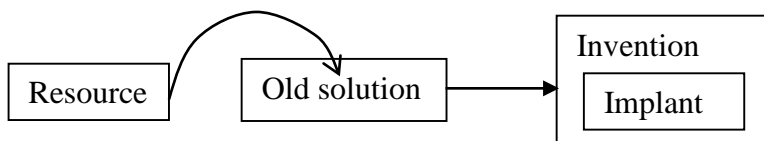

Fig. 6 A resource implanted into an existing solution creates a new solution.

Example. Translation lookaside buffer (TLB) [Couleur1968, Case1978] is a specific cache memory that represents an implant in the virtual memory mechanism. Virtual memory may work without TLB, but TLB considerably improves the mechanism.

Adaptation (A).

Description. The assumption here is that one solution is better under one set of conditions, and the other solution is better under another set of conditions. The idea is in dynamic combination of different solutions, thus adapting a new solution to work the best way in different conditions. Consequently, the complexity of a new solution method is always higher than the complexity of each existing solution used to generate the solution.

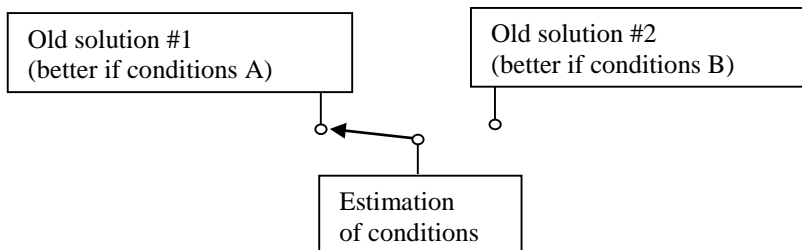**Fig. 7** Selecting among existing solutions to adapt a new solution to the conditions.

Example. Adaptive switching mechanism in computer networking [Intel1997] changes port-running mode from cut-through switching mode, which is normal switching mode that achieves better performance in the case of moderate error rate, via fragment-free mode, to store-and-forward mode when error rate becomes too high.

Hybridization (H).

Description. Sometimes elements of two or more existing solutions, or two or more complete solutions, could be combined, in order to obtain a hybrid solution. Although the Adaptation method already presents a kind of dynamically combining existing solutions, we will use the term hybridization only for referring to the method of static combination of resources from existing solutions in the new solution. The aim is to select elements from a set of existing solutions in a way to outperform each one of the existing solutions.
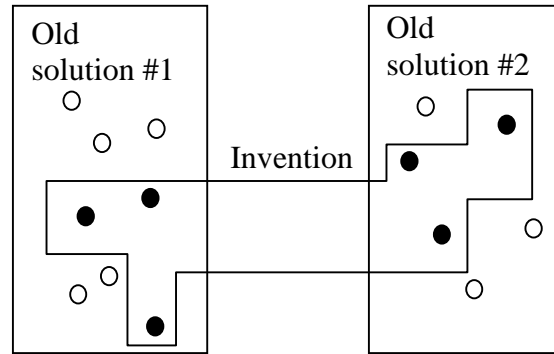

Fig. 8 A new hybrid solution combines parts of the existing solutions.

Example. Combination of the back-face culling algorithm with the z-buffer algorithm [Catmull1974] improves performance of the hidden surface detection and removal algorithm, since back-face culling is much more efficient than z-buffer, but not as general as z-buffer.

Transgranularization (T).
Description. Sometimes a similar algorithm or mechanism may be applied with a different level of granularity, solving a problem never solved before. The direction of transformation may be to a coarser or to a finer granularity from the existing solution. Such an approach we will call Transgranularization.
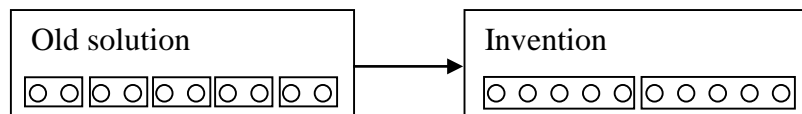
Fig. 9 Elements of an old solution are fine-grained, and invention is achieved by changing granulation of the solution elements.

Examples. One example is applying the well-known virtual memory [Fotheringham1961] principles to the lower-level of memory hierarchy - processor cache [Wilkes1965]. Similar principles may be applied to a web browser, proxy or server caches. The essence of innovation is in changed granularity of data that the mechanism manipulates with: a data block is a cache-line of a few words in processor caches, a page in a virtual memory system or data file (and possible folder) in web caching.

Extraparametrization (E).
Description. An existing solution is based on a simpler model that depends on a relatively small set of parameters. By adding new parameters to the model, it becomes more complex and leads to a new, more precise, efficient, and sophisticated solution to the problem. Of course, it is possible to start from a more complex model and to downgrade

it to a simpler model by ignoring some of the negligible (under some conditions) parameters, but introducing extra parameters to the model is a more frequent requirement for new model inventions.
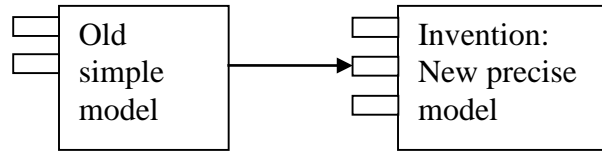
Fig. 10 Adding an extra parameter transform an existing simple model to a more complex, but more precise or more efficient one.

Examples. Introduction of extra parameters could lead to a refined model. Contrary to intuitive expectations, new parameters and variables could make it easier for programming implementation, which enables experiments with a broader range of basic parameters. The experiments could result in an optimal solution for reconfiguration of the system. An example could be found in the performance analysis of multiprogrammed systems represented by closed queuing networks, based on state probabilities. The computation of state probabilities has been made easier by introduction of the Buzen's algorithm [Buzen1973], based on the Gordon-Newell theorem [Gordon1967]. This algorithm introduces a new parameter called the normalization constant $G(K)$ and implements its computation in a simple nested-loops programming structure. Once G is computed, the probability distributions for the network could be found. On the other hand, introduction of new parameters on the conceptual level could result in new solutions in all areas of computer science. For example, introduction of compression techniques that do not decompress data back to 100% of the original, known as lossy methods, provide high degrees of compression suitable for images that have to be transferred over the Internet. JPEG file format is one of such solutions, which parameterizes image quality, contrary to its optional lossless mode.

Representative Examples from the Authors' PhD Theses

This section presents the essence of past innovations of the authors of this article and classifies them into the 10 idea generation methods introduced in this article.

The PhD thesis research of Vladimir Blagojević entitled "Analysis of anharmonicity of ZnSFe monocrystal reflectivity spectra" was an interdisciplinary one, covering both material science and software science. The field of research is either "Computational materials science" or "Computational physics". The essence of the material science part of the thesis research is in developing a new physical model for monocrystal reflectivity spectra fitting in two variants - classical (additive) and semiquantum (factorized), as well as introducing a generalized hybrid physical model. Within the software science part of the thesis research, a general high performance and extensible software system was formally specified, designed and implemented. Finally, the software system developed was successfully applied for analyzing the monocrystal reflectivity spectra of ZnSFe that could not be treated by using previously existing physical models, and the results were published in [Blagojevic1990]. This thesis research introduces an innovation predominantly along the lines of two methods: *M* and *G*.

The PhD thesis research of Dragan Bojić was in the domain of reverse engineering behavioral elements of the UML software model, and introduces an innovation predominantly along the lines of the method *T*. Previous approaches to feature interaction problem in mapping features to code, considered either a single feature or a pair of them.

By using a novel representation, the concept lattice, a full set of features is considered at once. Results were published in [Bojic2000], [Bojic2004].

The PhD research of Miroslav Bojović covered the field of synchronization and communication mechanisms in fault tolerant multiprocessor systems, and introduced an approach which minimizes the latency and the number of messages used in order to accomplish secure and consistent data exchange, along the method *M*. All approaches existing in the open literature till that moment utilized some form of strict consistency maintenance, so the classification developed by the author included no examples based on loose consistency. In order to be able to generate a novel solution which is based on loose consistency, the author introduced three different mechanisms that acted as implants according to method *I*: the event mechanism, the resource usage synchronization mechanism, and the mutual exclusion mechanism. As indicated in [Bojovic1988], the approach introduced by this PhD thesis research enabled that the worst case latency be N+1 (rather than (N(N+1)/2)-1, which was the best of the open literature till the moment when the thesis was published) and that the worst case message count be N-1 (rather than 2(N-1)).

The PhD research of Miloš Cvetanović explored the automated comparison of relational database models and led to the development of an educational system that helps students to bridge the gap between database management system theory and practice. The system permits active tutoring of students by providing interactive feedback by comparing the answer given by a student with the correct solution. This research introduced an innovation predominantly along the lines of the method M in the case of conceptual database models, and the method C in the case of logical database models. Results were published in [Cvetanovic2011].

In name-space architectures, which are the subject of the PhD thesis research by Jovan Đorđević, the mapping of names onto fast registers is hardware, rather than software, function. The MU5 computer is an example of such an architecture, having a single-address instruction format. Two-store-address and three-store-address architectures developed from MU5 concepts are proposed, using the method *S*. ISPS descriptions of all three architectures have been written, verified and used in a series of experiments [Djordjevic1980], conducted at Carnegie-Mellon University, Pittsburgh, from Manchester University, England, using the ARPA Network. Results for a number of benchmark programs run on the ISPS simulation model of MU5 are first related to actual results obtained by hardware monitoring of the MU5 processor, and some comment is included on the validity of this type of architectural evaluation. Results of measurements of static and dynamic code usage for the same benchmark programs run on the ISPS simulation models of these systems are then presented, and comparisons between the three architectures are made on the basis of these results.

The PhD research of Đorđe Đurđević was in the domain of parallel compression of regular height fields (matrices of elevations of 3D points), along the method M, with elements of method R. Previous methods were mostly sequential, parallel only on coarse granularity (batches of points). On the other hand, the proposed method is per-point parallel, suitable for implementation on modern highly parallel GPUs (Graphics Processing Unit), which became widely available in the recent years. The essential innovation is that the previous methods predominantly compress data by predicting the

elevation of a point from elevations of previously compressed points, while the proposed method approximates elevations of a set of points by a mathematical function. Only a few previous methods, dating from the pre-GPU or pre-multicore CPU era, considered approximation, but did not consider parallelization. The proposed method was published in [Djurdjevic2013].

The PhD thesis research of Bojan Furlan described a methodology for Intelligent Question Routing Systems (IQRS) design. The key functionality of IQRS is that for a question addressed by a user, to provide a good answer by searching the list of all (other) available users. The answer is provided by selecting a certain number of competent users and forwarding the question to them. The thesis research first surveys existing solutions and introduces an original presentation paradigm that generalizes the essence of these solutions (method *G*). Existing solutions were then analyzed and the outcome was a proposal for a new approach that tackles identified problems (method *M*). The rest of the thesis research describes an IQRS prototype that implements the proposed ideas. The ideas relate to the three major problems of system implementation: question analysis, question forwarding, and users' knowledge profiling. The question analysis module enables question visualization and combines a fully automatic text processing with manual correction of the results, giving a user ability to increase the accuracy of the output (method *E*). For the question forwarding phase a new algorithm – LInSTSS, was proposed. It calculates the semantic similarity between two short texts, by including, among other parameters, the specificity of words that these texts contain (method *E*). The specialized version of this algorithm – P2Q, was designed to compare questions with available user profiles (method *S*). Finally, the text processing module, used both for question and answer analysis, combines two complementary approaches for information extraction from text (method *E*). Results were published in [Furlan2013a] and [Furlan2013b].

The PhD of Slavko Gajin focused on analytical modeling for performance evaluation of routing in multicomputer systems, along the method *G*, with elements of the method *H*. Solutions existing at the time of the PhD research of Gajin were using different models for different interconnection network topologies. The research of Gajin created a look from above and introduced a performance model that applies to all possible interconnection network topologies. The first ideas were published in [Gajin2006], and detailed research results in [Gajin2012].

The PhD thesis research of Zoran Jovanović was in the area of simulation and development of a VLIW (Very Large Instruction Word machine) used for digital signal processing. The simulator developed with a high-level language was used both for the machine design and testing of highly parallel programs developed by using a machine specific parallel assembler. At the time the machine was developed, the name of such machines was Peripheral Array Processors and when compiler technology became mature for aggresive instruction level parallelization, the name was changed to VLIW. In the available literature, there was only limited information about the leading processors of that type produced by Floating Point Systems (AP-120). The new design developed through simulation was done by using the first integrated floating point ALU pipelines that had a specific 22-bit floating-point format. The developed machine was faster and logicaly more complex than the existing machines known in the open literature, but due

to the reduced floating point precision and range, it was applicable only in digital signal processing. This approach could be classified as *Specialization (S)*.

The PhD thesis research of Dragan Milićev [Milićev2002] was in the field of model transformations in modeling tools. It was a pioneering work in what was later shaped into the Model-Driven Architecture (MDA). Its idea emerged from analyzing the problem of generating ultimate output forms (e.g., code) in specific semantic domains (e.g., textual programming languages) from highly abstract models (e.g., UML or domain-specific modeling languages). For such conceptually distant domains, defining transformations is tedious and error-prone, and the thesis research proposed introducing one or more intermediate modeling domains (defined in terms of metamodels in MOF) and defining multiple successive transformations between the source, intermediate, and target models. Because the intermediate domains are semantically closer, the transformations are easier to define and implement. In that sense, the proposed method could be classified as *implantation* (*I*). Second, the thesis research also proposed a novel modeling technique for defining transformations by using UML object models with a specific, object creational semantics. The core UML object diagrams were extended with additional constructs for conditional, iterative, recursive, and polymorphic creation. By this, the method had reused some well established concepts and notations from other software engineering disciplines in a new context, but with a different interpretation and semantics, and thus could be classified as *crossdisciplinarization* (*C*).

The PhD research of Veljko Milutinović covered the field of suboptimal detection of data signals, and introduced a method that eliminates both the A/D converter in the input stage and the sample memory (SM) at the processing stage of the system, for a minimal performance degradation, along the methods *M+I*. Existing approaches eliminated either only the A/D converter or only the sample memory, but not both. Consequently, when all four possibilities were combined, a classification was obtained in which one class (neither A/D, nor SM) was not covered by examples from the open literature. Results were published in [Milutinovic1988].

The PhD research of Boško Nikolić covered the field of Web-based visual simulation, designed to help teaching and learning computer architecture and organization courses. Simulation offers a unique environment that exposes students to both the programmer and the designer's perspective of the computer system. The Web-based simulator features an interactive animation of program execution and allows students to navigate through different levels of the educational computer system's hierarchy—starting from the top level with block representation down to the implementation level with standard sequential and combinational logic blocks. This work introduces an innovation predominantly along the lines of the method *T*. Results were published in [Nikolic2005].

The PhD thesis research of Jelica Protić explored the consistency maintenance of shared data in a distributed shared memory systems, and examined the potential for performance improvements of the protocol based on entry consistency, using different techniques, some of which were inspired by the lazy release consistency implementation, predominantly along the lines of the method *H*. At the time this research was conducted, the two most sophisticated relaxed consistency models were: entry consistency (EC) implemented in Midway and lazy release consistency (LRC) implemented in TreadMarks. The main goal of this research was to combine the advantages of LRC and

EC, taking into account communication and computation costs of the memory consistency protocol, as well as synchronization costs, which makes it a representative example of *H* research method, with the elements of *E* introduced by including new parameters in the analytical modeling. The first ideas were published in [Protic1996], and detailed analysis in [Protic2000].

The PhD research of Marija Punt was in the field of human-computer interaction (HCI), exploring user interaction scenarios in an integrated environment of digital TV, mobile devices, and the Internet. In the beginning of the research, existing HCI systems targeting a living room environment were explored. Using methods M and H, a list of HCI scenarios that applications, developed for an integrated environment, could offer, were identified: sharing content on multiple screens (private content showing on the mobile device screen and public content showing on the central screen), using the mobile device to control actions on the central screen, motion detection using sensors, supplemental sensory feedback (haptic feedback, background lighting), combining passive viewing of television with parallel interaction, enriching the user experience with access to various information services (such as DVB, social media, and metadata databases), combining interaction between co-located and non-co-located people. The research spanned multiple fields as in method C. After identifying the scenarios, the SHARP development framework was designed and implemented providing the building blocks to quickly construct distributed applications offering different combinations of the aforementioned HCI scenarios. A series of different games was developed using the SHARP framework showing its flexibility and demonstrating novel applications that offer new combinations of scenarios to the users. Results were published in [Punt2015].

The PhD research of Zaharije Radivojević involved defining methodological approach that should help students connect theory and practice in the domain of computer architecture and organization simulator design, and to design simulators capable to work in a concurrent and distributed environment. In this manner, knowledge from two domains was interchanged as in method C. The approach is based on a multi-layer design, where each layer is responsible for different type of processing and communication, which is done in accordance with method M. A referent simulator implementation created according to the methodology was demonstrated. Results were published in [Radivojevic2011].

The PhD research of Žarko Stanisavljević was in the field of cryptographic algorithms visual representation, intended for use in an elearning tool for teaching data security course. The result is COALA (CryptOgraphic Algorithms visuaL simulAtion) system that supports five different types of cryptographic algorithms and enables detailed analysis of their execution. Method C and method G were used to acquire knowledge from elearning tools design principles in various areas of use. Then systems for visual representation of algorithms were analyzed and method G was used to define important characteristics of these systems. Finally, systems for visual representation of cryptographic algorithms were analyzed and using method S and method M the new approach was developed. Results were published in [Stanisavljevic2014].

The PhD research of Saša Stojanović is related to the process of revealing whether a software library available in source code is used in a binary code. The essence of the proposed approach is to speedup searching for procedures originating from a software

library, by comparing procedures resulting in an estimated measure of similarity between the searched procedure originating from a software library and each one of the procedures in binary code; all that is done in order to rank procedures from binary code according to decreased similarity with the searched procedure. Procedure comparison is a technique that could be used in many different domains, and there is a significant possibility to apply it in the domain of software violations that have elements of method M and method C. Moreover, new software metrics are proposed, meaning that the approach also includes elements of method E. Along with the proposed approach, five techniques are proposed based on an analysis of the points where information is lost; this corresponds to method M. Some of the proposed techniques are inspired by existing approaches where comparison of code is used, leading to the conclusion that the research includes elements of method C. Results are published in [Stojanović2015, Radivojević2016].

The PhD research of Igor Tartalja covered the field of software methods for cache coherence maintenance, where the author proposed a dynamic method for conditional invalidation of shared data segments, along the methods $M+H+C$. Existing approach suffered from unnecessary invalidations, resulting in performance degradation. Eager consistency model of the existing dynamic (run-time) software method mutated to the lazy consistency model, by applying a version control mechanism similar to the one proposed in a static (compile-time) cache coherence scheme. First results were published in [Tartalja1992] and later in [Tartalja1997].

The PhD research of Milo Tomašević was focused on the hardware methods for preserving the cache coherence in shared memory multiprocessors and proposed the principle of partial block invalidation, being alike both to the H and A methods. The contemporary solutions at that time followed the principle of full block invalidation which could incur a significant overhead in conditions of increased false sharing. The proposed WIP protocol starts with partial, word-based invalidations trying to preserve the valid block contents and switches to full block invalidation when a treshold that signals its excessive pollution is reached. The proposed protocol and its evaluation analysis were published in [Tomasevic1996].

The PhD research of Pavle Vuletić described in [Vuletic2011] presents the analysis of the statistical nature of the cross-traffic on paths in computer networks as a foundation for choosing among active available bandwidth measurement strategies. Due to the highly variable statistical nature of network traffic, common bandwidth estimation tools and methods are not enough accurate and robust to function in different networking environments. Therefore, this work analyzed theoretical foundations for active available bandwidth measurement strategy through the self-similar process sampling analysis. The results obtained show a relationship between the main parameters in the measurement procedure, such as the number of samples, sample length, and sample distance and their impact on the measurement accuracy. The facts that previous research work in this field did not analyze at all these parameters, or used very simplified models, classify this work as E. Through the analysis of several existing Internet packet traces, it was recommended that the minimum single sample probe stream length must be longer than the average cross traffic interarrival. Following these findings, a new method for available bandwidth estimation, along the lines of the method A was proposed, that has shown significant accuracy under different network setups.

Table 1 summarizes the proposed classification and illustrates classes with examples from above cited PhD research studies.

	Class name	Authors of Example PhD Thesis Research
1	Mendeleyevization	Blagojević, Bojović, Cvetanović, Đurđević, Furlan, Milutinović, Punt, Radivojević, Stanisavljević, Stojanović, Tartalja
2	Generalization	Blagojević, Furlan, Gajin, Stanisavljević
3	Specialization	Đorđević, Furlan, Jovanović, Stanisavljević
4	Revitalization	Đurđević
5	Crossdisciplinarization	Cvetanović, Milićev, Radivojević, Punt, Stanisavljević, Stojanović, Tartalja
6	Implantation	Bojović, Milićev, Milutinović
7	Adaptation	Tomašević, Vuletić
8	Hybridization	Gajin, Protić, Punt, Tartalja, Tomašević
9	Transgranularization	Bojić, Nikolić
10	Extraparameterization	Furlan, Protić, Stojanović, Vuletić

Table 1: The 10 Approaches to Evolutionary Innovations with Examples.

As the experiences are gained during the work with PhD students, a follow up research (maybe a decade from now) could summarize new findings related to the advisory work with young researchers, using the idea generation methodologies advocated in this article. Some of the PhD students at our Department of Computer Engineering and Informatics have recently published the papers or articles in Thomson Reuters Journal Citation Reports (JCR) publications [Kojic2012, Sustran2015, Misic2016, Draskovic2016]. Their younger colleagues currently explore some challenging research areas and they have presented their work in progress at national and international conferences [Milenkovic2014, Tubic2015, Prodanov2016, Puric2016]. Finally, the youngest generation of associates is continuing their research efforts towards master degree, after successful completion of their diploma thesis [Delcev2015, Hadzic2015]. Some of the PhD candidates have already recognized the methodological classes of their ideas. Marko Mišić is intensifying his research on plagiarism detection in programming assignments [Misic2016], by exploring the social network analysis along the methods $I+R+C$. Sofija Purić and Živojin Šuštran demonstrated the method S, by using a standard parser in the process of automatic generation of exam questions about syntax analysis for programming courses [Puric2016]. We assume that the other young researchers will also recognize the benefits of introducing methodological classes to their research, in order to speed up the process of shaping their PhD research.

Conclusions

This article introduces and explains 10 different methods that one could use to generate ideas for PhD research in the area of computing. It also provides a case study based on the examples of the PhD research of the authors of this article, and shows how they fit into the proposed classification.

The presented methodology implies that a PhD student is first asked to create a survey of existing solutions to the problem attacked by his/her PhD research, and to classify them. The classification may include classes without existing examples, which opens doors for *Mendeleyevization*. In this process, one could also catch the important common properties of the existing solutions and make an abstraction, which leads to *Generalization*. On the other hand, if a well-established common meta-metamodel is identified, it could be used for development of a metamodel for the specific domain, following the method of *Specialization*. During the survey process, some theoretical inventions that are practically dead, could be revisited and applied using the technology upgrades, which results in the approach that we refer to as *Revitalization*. *Crossdisciplinarization* occurs when one finds the way to port some good ideas from one field to another. If a new solution is invented by implanting a (relatively small) resource into an existing solution, we follow the path of *Implantation*. Algorithms/approaches inherent to various solutions could be combined based on conditions in which one of them performs better, which opens doors for a method that we named *Adaptation*. On the other hand, by recombining parts of existing solutions, we sometimes could create a good new solution along the lines of *Hybridization*. Further on, taking the direction of transformation to coarser or to finer granularity from the existing solution, with less or more modifications, leads to *Transgranularization*. Finally, by adding new parameters to the model, it becomes more complex and may lead to a new, more precise solution to the problem, so one could perform *Extraparameterization*. Although the set of the proposed methods is not closed (playing with anagrams made of first letters of the proposed methods - *M, G, S, R, C, I, A, H, T,* and *E*), this set of methods may be considered as *CHARMInGEST*, but PhD candidates have to keep open their minds and take care not to be caught in the *TRAGIC MESH*.

Future work on this subject should examine more well-known examples of innovations in computer science and engineering, as well as the on-going research of our PhD students. Since we have qualified the proposed classification as *Incomplete,* some new categories may be added in the future.

References

Methodology Related References
[Dorfler2010] Dorfler, V., Baracskai, Z., and Velencei, J., "Understanding Creativity," *Transactions on Advanced Research*, Vol. 6, No. 2, 2010, pp. 17-25.
[Dunleavy2003] Dunleavy, P., "Authoring a PhD: How to Plan, Draft, Write and Finish a Doctoral Thesis or Dissertation," *Palgrave Macmillan*, 2003.
[Faulkner1994] Faulkner, W., "Conceptualizing Knowledge Used in Innovation: A Second Look at the Science-Technology Distinction and Industrial Innovation," *Science, Technology and Human Values*, Vol. 19, No. 4, October 1994, pp. 425-458.
[Felder2002] Felder,R., Silverman, L., "Learning and Teaching Styles in Engineering Education," *Engr. Education*, 78(7), 674-681, 2002.
[Filipi2011] Filipi-Matutinovic, S., "Scientific Information in Serbia," University Library Svetozar Markovic, Belgrade, 2011 (in Serbian).

[Franceshet2010] Franceshet, M., "A Comparison of Bibliometric Indicators for Computer Science Scholars and Journals on Web of Science and Google Scholar," *Scientometrics*, Vol. 83, No. 1, April 2010, pp. 243-258.

[Horowitz1999] Hporowitz, R.,"Creative Problem Solving in Engineering Design," PHD Thesis, Tel Aviv University, Israel, 1999.

[Kearns2008] Kearns, H., Gardiner, M., Marshall, K., "Innovation in PhD Completion: The Hardy Shall Succeed (and Be Happy!)," *Higher Education Research & Development*, Vol. 27, No 1, pp. 77-89.

[Linn1987] Linn, M. C., "Establishing a Research Base for Science Education: Challenges, Trends, and Recommendations," *Journal of Research in Science Teaching*, Vol. 24, No. 3, March 1987, pp. 191–216.

[Milutinovic1996] Milutinovic, V., "The Best Method for Presentation of Research Results," *IEEE TCCA*, Vol. 214, September 1996, pp. 1-6.

[Milutinovic1997] Milutinovic, V., "A Good Method to Prepare and Use Transparencies for Research Presentations," *IEEE TCCA,* March 1997, pp. 1-8.

[Milutinovic2008] Milutinovic, V., Tomazic, S., "How to Ruin the Career of a PhD Student in Computer Science and Engineering: Precise Guidelines," *The IPSI BgD Transactions on Internet Research*, Vol. 4, No. 2, July 2008, pp. 24-26.

[Milutinovic2015] Milutinovic, V.,"A Structured Approach to Research for PHD Students in Computer Science and Engineering: "How to Create Ideas, Conduct Research, and Write Papers," IPSI Transactions on Internet Research, July 2015.

[Ogot2006] Ogot, M., Okudan, G., "Systematic Creativity Methods in Engineering Education: A Learning Styles Perspective," *Int'l Journal of Engineering Education*, Vol. 22, No. 3, pp. 566-576, 2006.

[Omerovic2010] Omerovic, S., Tomazic, S., Milutinovic, M., Milutinovic, V., "Methodology for Written and Oral Presentation of Research Results," *Journal of Professional Issues in Engineering Education and Practice*, Vol. 136, No. 2, 2010, pp.112-117.

[Prost2009] Prost O'Leary, D., "Graduate Study in the Computer and Mathematical Sciences: A Survival Manual,"
http://www.cs.umd.edu/~oleary/gradstudy/gradstudy.html. Accessed 25 June 2014.

[Stierand2011] Stierand, M. and Dorfler, V., "Methods Against Methods," Mesquita, A. (Editor), *Technology for Creativity and Innovation: Tools, Techniques and Applications*, IGI Global, Hershey, PA, 2011, pp. 121-134.

[UML] *OMG Unified Modeling Language™ (OMG UML), Superstructure*, Object Management Group, 2011,
http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/. Accessed 25 June 2014.

[Yoram2012] Yoram, R., Hatchnel, A., Shai, O., Subrahmanian, E.,"A Theorethical Analysis of Creativity Methods in Engineering Design:Casting and Improving ASIT within C-K Theory," *Journal of Engineering Design*, 23(2), 2012.

Common examples

[BPMN] *Business Process Model and Notation (BPMN)*, Object Management Group, 2011, http://www.omg.org/spec/BPMN/2.0/PDF/. Accessed 25 June 2014.

[Buzen1973] Buzen, J., "Computational Algorithms for Closed Queueing Networks with Exponential Servers," *Communications of the ACM*, Vol. 16, No. 9, September 1973, pp. 527-531.

[Case1978] Case, R.P. and Padegs, A., "The Architecture of the IBM System/370," *Communications of the ACM*, Vol. 21, No. 1, 1978, pp. 73-96.

[Catmull1974] Catmull, E., *A Subdivision Algorithm for Computer Display of Curved Surfaces*, Ph.D. Thesis, Report UTEC-CSc-74-133, Computer Science Department, University of Utah, Salt Lake City, UT, December 1974.

[Couleur1968] John F., Couleur, J.F., and Glaser, E.L., "Shared-access Data Processing System," Patent 3412382, November 1968.

[Flynn1966] Flynn, M., "Very High-Speed Computing Systems," Proceedings of the IEEE, Vol. 54, No. 12, 1966, pp. 1901–1909.

[Fotheringham1961] Fotheringham, J., "Dynamic Storage Allocation in the Atlas Computer, Including an Automatic Use of a Backing Store," *Communications of the ACM*, Vol. 4 , No. 10, October 1961, pp. 435 – 436.

[Gordon1967] Gordon, W.J. and Newell, G.F., "Closed Queueing Systems With Exponential Servers," *Operations Research*, Vol. 15, No. 2, 1967, pp. 254–65.

[Harel1987] Harel, D., "Statecharts: A Visual Formalism for Complex Systems," *Science of Computer Programming*, Vol. 8, No.3, 1987, pp. 231–274.

[Holland1975] Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.

[Hopfield1982] Hopfield, J.J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proc. of National Academy of Sciences of the USA*, Vol. 79, No.8, April 1982, pp. 2554-2558.

[Intel1997] Intel Corporation, *Intel Adaptive Technology Optimizing Network Performance*, 1997.

[Jones1993] Jones, N. D., Gomard, C.K., and Sestoft, P., *Partial Evaluation and Automatic Program Generation* (with chapters by L.O. Andersen and T. Mogensen) Prentice Hall International, June 1993, ISBN 0-13-020249-5

[Kung1979] Kung, H.T. and Leiserson, C. E., "Algorithms for VLSI Processor Arrays," in: Mead, C., Conway, L. (eds.), *Introduction to VLSI Systems*, Addison-Wesley, 1979.

[McCulloch1943] McCulloch, W. S., and Pitts, W., "A Logical Calculus of Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biology*, Vol. 5, No. 4, 1943, pp. 115-133.

[MOF] *OMG Meta Object Facility (MOF) Core Specification*, Object Management Group, 2011, http://www.omg.org/spec/MOF/2.4.1/PDF/. Accessed 25 June 2014.

[Petri1966] Petri, C.A., "Communication with Automata," *New York: Griffiss Air Force Base, Technical Report, RADC TR-65-377-vol-1-suppl-1 Applied Data Research*, Princeton, NJ, Contract AF 30(602)-3324, 1966.

[Spector1984] Spector, A., and Gifford, D., "The Space Shuttle Primary Computer System," *Communications of the ACM*, Vol. 27, No. 9, 1984, pp. 872–900.

[Wilkes1965] Wilkes, M.V., "Slave Memories and Dynamic Storage Allocation," *IEEE Transactions on Electronic Computers*, Vol. EC-14, No. 2, 1965, pp. 270 – 271.

Author's PhD related references

[Blagojevic1990] Blagojevic, V., Gledhil, G.A., Hamilton, A., Upadhuay, S.B., Nikolic, P.M., Pavlovic, M.B., and Rakovic, D.I.: "Far Infrared Optical Properties of ZnS Highly Doped with Fe," *Infrared Physics*, Vol. 31, No. 4, 1991, pp. 387-393.

[Bojic2000] Bojic, D. and Velasevic, D., "Reverse Engineering of Use Case Realizations in UML, " *ACM SIGSOFT Software Engineering Notes*, Vol. 25, No. 4, July 2000, pp. 56-61.

[Bojic2004] D. Bojic, T. Eisenbart, R. Koschke, D. Simon, and D. Velasevic, Addendum to "Locating Features in Source Code," *IEEE Transactions on Software Engineering*, Vol. 30, No. 2, February 2004, pp. 140.

[Bojovic1988] Bojovic, M., Konstantinovic, Z., "Synchronization and Communication Mechanism for a Highly Reliable Multiprocessor System," *The 15th IFIP Conference on Real Time Programming*, May 1988, Valencia, Spain.

[Cvetanovic2011] Cvetanovic, M., Radivojevic, Z., Blagojevic, V., and Bojovic, M., "ADVICE—Educational System for Teaching Database Courses," *IEEE Transactions on Education,* Vol. 54, No. 3, August 2011, pp. 398-409.

[Djordjevic1980] Djordjevic, J., Ibbett, R.N., Sumner, F.H., "Evaluation of Some Proposed Name-Space Architectures Using ISPS," *IEE Proceedings E (Computers and Digital Techniques),* Vol. 127, No. 4, July 1980, pp. 120–125.

[Djurdjevic2013] Đurđević, Đ., Tartalja, I. "HFPaC: GPU Friendly Height Field Parallel Compression," *GeoInformatica*, Vol. 17, No. 1, January 2013, pp. 207-233.

[Furlan2013a] Furlan B., Nikolic B., and Milutinovic V., "A Survey and Evaluation of State-of-the-Art Intelligent Question Routing Systems," *International Journal of Intelligent Systems*, Vol. 28, No. 7, July 2013, pp. 686–708.

[Furlan2013b] Furlan B., Batanović V., and Nikolić B., "Semantic Similarity of Short Texts in Languages with a Deficient Natural Language Processing Support," *Decision Support Systems*, Vol. 55, No. 3, June 2013, pp. 710–719.

[Gajin2006] Gajin, S. and Jovanović, Z., "Explanation of Performance Degradation in the Turn Model," *The Journal of Supercomputing*, Vol. 37, No. 3, September 2006, pp. 271-295.

[Gajin2012] Gajin, S. and Jovanovic, Z., "An Accurate Performance Model for Network-on-Chip and Multicomputer Interconnection Networks," *Journal of Parallel and Distributed Computing*, Vol. 72, No. 10, October 2012, pp. 1280–1294.

[Milicev2002] Milićev, D., "Automatic Model Transformations Using Extended UML Object Diagrams in Modeling Environments," *IEEE Transactions on Software Engineering*, Vol. 28, No. 4, April 2002, pp. 413-431.

[Milutinovic1988] Milutinovic, V., "A Comparison of Suboptimal Detection Algorithms Applied to the Additive Mix of Orthogonal Sinusoidal Signals," *IEEE Transactions on Communications*, Vol. 36, No. 5, May 1988, pp. 538-543.

[Nikolic2005] Djordjevic, J., Nikolic, B. and Milenkovic, A., "Flexible Web-based Educational System for Teaching Computer Architecture and Organization," *IEEE Transactions on Education*, Vol. 48, No. 2, May 2005, pp. 264-273.

[Protic1996] Protic, J., Tomasevic, M., and Milutinovic, V., "Distributed Shared Memory: Concepts and Systems," *IEEE Parallel and Distributed Technology: Systems and Applications*, Vol. 4, No. 2, Summer 1996, pp. 63-79.

[Protic2000] Protic, J. and Milutinovic, V., "A Comparison of Three Protocols for Entry Consistency Maintenance Based on MVA Algorithm," *Proc. 8th Int'l Symp. MASCOTS 2000*, San Francisco, USA, August 2000, pp. 517-523.

[Punt2015] Punt, M., Bjelica, M. Z., Zdravkovic, V., Teslic, N.," An Integrated Environment and Development Framework for Social Gaming Using Mobile Devices, Digital TV and Internet," *Multimedia Tools and Applications*, Vol. 74 No.18, September 2015, pp. 8137-8169.

[Radivojevic2011] Radivojevic, Z., Cvetanovic, M., and Jovanovic, Z., "Reengineering the SLEEP Simulator in a Concurrent and Distributed Programming Course," *Computer Applications in Engineering Education*, Vol. 19, February 2011, pp. 1-13.

[Stojanović2015] Stojanović, S., Radivojević, Z., and Cvetanović, M., "Approach for Estimating Similarity Between Procedures in Differently Compiled Binaries," *Information and Software Technology*, Vol. 58, No. 1, February 2015, pp. 259-271.

[Radivojevic2016] Radivojević, Z., Cvetanović, M., and Stojanović, S., "Comparison of Binary Procedures: A Set of Techniques for Evading Compiler Transformations," *The Computer Journal*, Vol. 59, No. 1, January 2016, pp. 106-118.

[Stanisavljevic2014] Stanisavljevic, Z., Stanisavljevic, J., Vuletic, P., and Jovanovic, Z., "COALA - System for Visual Representation of Cryptography Algorithms," *IEEE Transactions on Learning Technologies*, Vol. 7, No. 2, April-June 2014, pp. 178-190.

[Tartalja1992] Tartalja, I. and Milutinovic, V., "An Approach to Dynamic Software Cache Consistency Maintenance Based on Conditional Invalidation," *Proc. 25th HICSS*, USA, Vol. 1, January 1992., pp. 457-466.

[Tartalja1997] Tartalja, I. and Milutinovic, V., "Classifying Software-Based Cache Coherence Solutions," *IEEE Software*, Vol. 14, No. 3, May/June 1997, pp. 90-101.

[Tomasevic1996] Tomasevic, M. and Milutinovic, V., "The Word-invalidate Cache Coherence Protocol," *Microprocessors and Microsystems*, Vol. 20, No. 1, March 1996, pp. 3-16.

[Vuletic2011] Vuletic, P. and Protic, J., "Self-similar Cross-traffic Analysis as a Foundation for Choosing Among Active Available Bandwidth Measurement Strategies," *Computer Communications*, Vol. 34, No. 10, July 2011, pp. 1145–1158.

Selected References of Young Researchers
on the Faculty of the Department of Computer Engineering and Informatics, School of Electrical Engineering, University of Belgrade

[Delcev2015] Delcev, S., *Simulator of the superscalar processors with the reservation stations*, diploma thesis (in Serbian), University of Belgrade, School of Electrical Engineering, October 2015.

[Draskovic2016] Drašković, D., Mišić, M., Stanisavljevic, Ž., "Transition from traditional to LMS supported examining: A case study in computer engineering,"

unconditionally accepted, *Computer Applications in Engineering Education*, 2016.

[Hadzic2015] Hadzic, F., *An implementation of superscalar processor on the FPGA chip,* diploma thesis (in Serbian), University of Belgrade, School of Electrical Engineering, October 2015.

[Kojic2012] Vujicic Stankovic, S., Kojic, N., Rakocevic, G., Vitas, D. M., Milutinovic, V. M., "A Classification of Data Mining Algorithms for Wireless Sensor Networks, and Classification Extension to Concept Modeling in System of Wireless Sensor Networks Based on Natural Language Processing," *Advances in Computers,* Vol. 90, 2012, pp. 223-283.

[Milenkovic2014] Milenkovic, K., Draskovic, D., Nikolic, B., "Educational software system for reasoning and decision making using Bayesian networks," IEEE Global Engineering Education Conference (EDUCON) 2014, pp. 1189-1194, Istanbul, Turkey, April 2014.

[Misic2016] Mišić M., Šuštran Ž., Protić J., "A Comparison of Software Tools for Plagiarism Detection in Programming Assignments," *International Journal of Engineering Education*, Vol. 32, No. 2, pp. 738-748, 2016.

[Prodanov2016] Prodanov, M., Drašković, D., "Software Simulation of Intelligent System based on Search Algorithms," (in Serbian), YU INFO 2016, pp. 379-383, Kopaonik, Serbia, February 2016.

[Puric2016] S. Purić, Ž. Šuštran, J. Protić, "Software Tool SinGen for Generating Questions and Answers about Syntax Analysis in Programming Courses," (in Serbian), YU INFO 2016, pp. 351-356, Kopaonik, Serbia, February 2016.

[Sustran2015] Sustran, Z., Rakocevic, G., Milutinovic, V. M., "Dual Data Cache Systems: Architecture and Analysis," *Advances in Computers,* Vol. 96, 2015, pp. 187-233.

[Tubic2015] Tubic, S., Draskovic, D., "Realization of A* Search Algorithm in a Computer Game," (in Serbian), *Proc. of the 59th ETRAN Conference*, Srebrno jezero, Serbia, Jun 2015.