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, Slavko Gajin,
Veljko Milutinović, Boško Nikolić, Zaharije Radivojević, Jelica Protić,
Igor Tartalja, Milo Tomašević and Pavle Vuletić

University of Belgrade, School of Electrical Engineering,
Department of Computer Engineering and Informatics, Belgrade, Serbia

Abstract

This paper 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 in 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 PhD thesis of faculty members of our department, and came up with a proposal of ten methods that can be implemented to derive new ideas, based on the existing body of knowledge in the research field. This systematic approach provides useful guidance for PhD students, in order to improve their efficiency and reduce the drop-out rate, especially in the area of computing.

Introduction

During the past decade, in many European countries, the process of restructuring higher education system according to 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 ECTS credits beyond the masters 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 computation 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 10.88 years, including 6 courses, master thesis and the PhD thesis (the period for PhD only was 5.77 years). 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 a paper in a journal from the 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 five years ago, and none of them has graduated yet, although the program nominally lasts three years only. In addition to economic and logistic reasons, one of the main obstacles in achieving this goal in a limited 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 papers are typically the same. For example, analysis presented in [Filipi2011] shows that for each citation that a paper receives in the area of computing, a paper in general engineering receives 4.64 citations, a paper in physics receives 11.9 citations, while a paper in molecular biology receives 32.89 citations, according to the analysis for ISI Thompson, 1996-2006. This 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 reasons, Google Scholar as a less formal source then Web of Science (WoS), shows significantly higher indicators' scores than WoS for computer scientists, roughly five times for paper-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 papers on different aspects of research conducting methodology [Milutinovic1996 and Milutinovic2008] and research presentation methodology [Milutinovic1997 and Omerovic2010]. In addition, this paper builds on the top of other representative studies related to methodologies for research innovation in science and engineering [Dorfler2010, Faulkner1994, Linn87, Prost2009, Stierand2011], and tries to systemize existing methods of innovation into an original 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 of members of our department fit 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 paper observes 10 different classes of ideas (methods for generating ideas) that led to important evolutionary innovations in the past.

In a research process that should lead to a PhD thesis, the following phases may be observed: (1) precise problem specification, (2) studying of related work and existing approaches, (3) generating an idea for 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 which is based on a hardware or software engineering innovation, which overcomes existing solutions in quality and/or performances. 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 preparation process. We just want to point to the step in the typical process for the targeted thesis type, where our methodology may help both the candidate and the mentor. Obviously, the point in this process where our classes of methods for generating ideas for innovations can serve as a road sign for PhD candidate is phase 3.

The 10 classes presented in this paper 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 unique first letter, 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) example(s) that illustrate(s) 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 and computer science and well described both in formal literature and informal sources such as the Wikipedia web site [http://www.wikipedia.org/].

Mendeleyevization (M)

Description. If one of the classification classes in an existing taxonomy of 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 can 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, researcher sometimes is not aware of the classification, or the classification does not exist in the moment when the new approach is invented. We consider Mendeleyevization as a method of generating a research idea only in the case when 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.

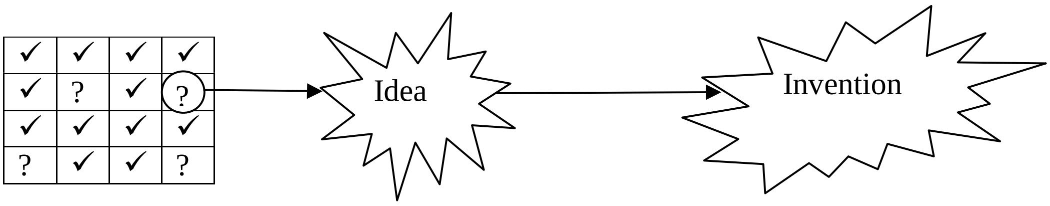

Figure 1. Existing classification is represented by the table were some cells contain existing solutions (✓), while others are empty (?); empty-cell analysis may lead to an idea for innovation.

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 can assume that Flynn's taxonomy influenced the inventions.

Generalization (G).

Description. Frequently, there are many versatile concrete solutions of a problem, but there is no common model that can encompass all of the existing solutions. Somebody may catch the important common properties of the existing solutions and can 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.

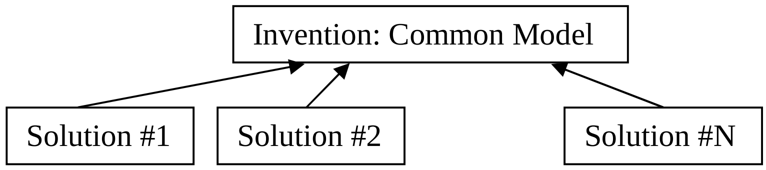

Figure 2. Generalization of particular concrete solutions to a common model.

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.

Specialization (S).
Description. Starting from well established general approach, someone can 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.

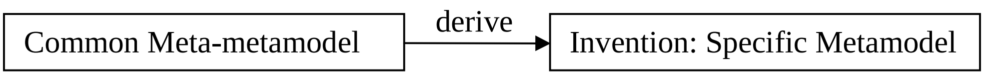

Figure 3. Deriving specific metamodel from common meta-metamodel.

Examples. Starting from common meta-metamodel, someone can 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).

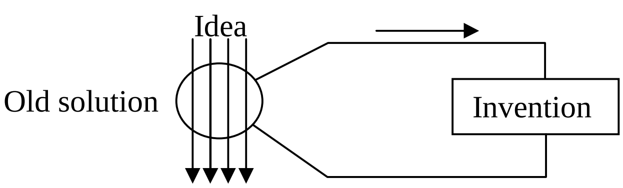

Figure 4. Dead theoretical invention (resistor) becomes alive when new idea (variable magnetic field) uses new technology (inductor).

Example. Computational model of artificial neural networks (ANN) was invented in early 1940s [McCulloch1943], but it revitalized and research in the field exploded in early 1980s [Hopfield1982], when 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). Degree of the solution modification during crossdisciplinarization may vary. On the first end of scale, the solution may be ported directly, and that 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 from different field, so the new solution has almost nothing to do with the initial solution.

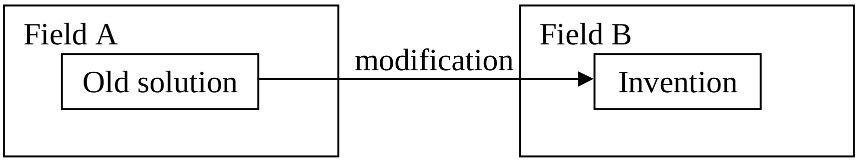

Figure 5. Existing solution of a problem in the field A inspires the idea on invention in the field B. Modification is possible but not necessary.

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. New solution is invented by implanting a resource into an existing solution. Characteristics of new solution overcome simple sum of characteristics of old solution and implanted resources, it brings new quality or significant performance gain.

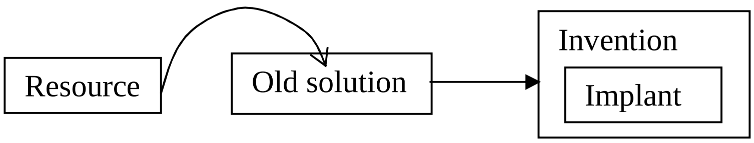

Figure 6. A resource implanted into an existing solution creates a new solution.

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 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.

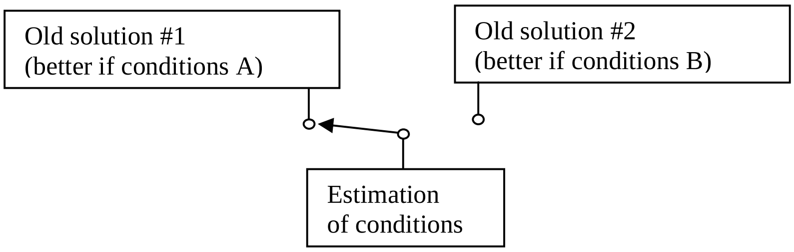

Figure 7. Selecting among existing solutions to adapt new solution on the conditions.

Example. Adaptive switching mechanism in computer networking [Intel1997] changes port running mode from cut-through switching mode, which is normal switching mode that give better performances 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 complete solutions can 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 set of existing solutions in a way to overcome performance of each of the existing solutions.

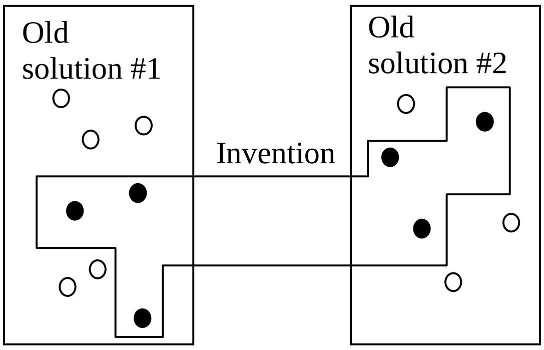

Figure 8. New hybrid solution combines parts of the existing solutions.

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

Transgranularization (T).
Description. Sometimes a similar algorithm or mechanism may be applied with different level of granularity, solving a problem never solved so far. Direction of transformation

may be to coarser or to finer granularity from the existing solution. Such an approach we will refer to as transgranularization.

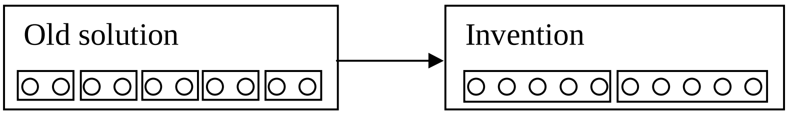

Figure 9. Elements of an old solutions are fine-grained, and invention is achieved by changing granulation of the solution elements.

Examples. One example is applying well known virtual memory [Fotheringham1961] principles to lower level of memory hierarchy - processor cache [Wilkes1965]. Similar principles may be applied to 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 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 of sophisticated solution of the problem. Of course, it is possible to start form a more complex model and to downgrade it to a simpler model by ignoring some of the negligible (in some conditions) parameters, but introducing extra parameters to the model is a more frequent requirement for new model inventions.

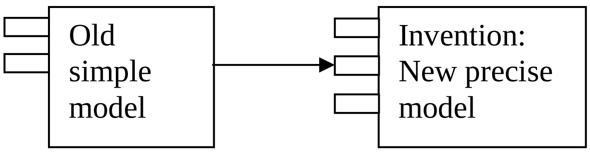

Figure 10. Adding an extra parameter leads an existing simple model to more complex, but more precise or more efficient one.

Examples. Introduction of extra parameters can lead to a refined model. Contrary to intuitive expectations, new parameters and variables can make it easier for programming implementation, which enables experiments with broader range of basic parameters. The experiments can result in optimal solution for reconfiguration of the system. An example can be found in 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 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 can be found. On the other side, introduction of the new parameters on the conceptual level can 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 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 paper and classifies them into the 10 idea generation methods introduced in this paper.

The PhD thesis of Vladimir Blagojević under the title "Analysis of anharmonicity of ZnSFe monocrystal reflectivity spectra" is 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 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, 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 introduces an innovation predominantly along the lines of two methods: M and G.

The PhD thesis of Dragan Bojić is 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 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 answer given by a student with the correct solution. This research introduced an innovation predominantly along the lines of the method M in case of conceptual database models, and the method C in case of logical database models. Results were published in [Cvetanovic2011].

In name-space architectures, which are the subject of the PhD thesis 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, and 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 are 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 of Slavko Gajin focuses 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 research of Veljko Milutinović covered the field of suboptimal detection of data signals, and introduced a method which 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 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 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 M. A referent simulator implementation created according to the methodology was performed. Results were published in [Radivojevic2011].

The PhD thesis 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 an 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 a survey paper [Protic1996], and detailed analysis in [Protic2000].

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 one proposed in a static (compile-time) cache coherence scheme. First results were published in [Tartalja1992] and later in [Tartalja1996].

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 more alike to the *H* method. The contemporary solutions at that time followed the principle of full block invalidation which can 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 which 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	**Examples**
1	Mendeleyevization	[Blagojevic], [Bojovic], [Djurdjevic], [Milutinovic], [Radivojevic], [Tartalja]
2	Generalization	[Blagojevic], [Gajin]
3	Specialization	[Djordjevic]
4	Revitalization	[Djurdjevic]
5	Crossdisciplinarization	[Radivojevic], [Tartalja]
6	Implantation	[Bojovic], [Milutinovic]
7	Adaptation	[Vuletic]
8	Hybridization	[Gajin], [Protic], [Tartalja], [Tomasevic]
9	Transgranularization	[Bojic], [Nikolic]
10	Extraparameterization	[Protic], [Vuletic]

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

Conclusions

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

The presented methodology implies that the 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 examples, which opens doors for *Mendeleyevization*. In this process, one can also catch the important common properties of the existing solutions and make an abstraction, which leads to *Generalization*. On the other side, if a well-established common meta-metamodel is identified, it can 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, can 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 of the problem, so one can perform *Extraparameterization*. Finally, 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. Also, as the experiences are gained in 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 talents using the idea generation methodologies advocated in this paper.

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.

[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.

[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.

[Linn87] 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. and 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.

[Omerovic2010] Omerovic, S., Tomazic, S., Milutinovic, M., and 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.

[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/.

Common examples

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

[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/`.

[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.

[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..

[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.

[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.

[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.

[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.

[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.

[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.