

The Role of Computer Science and Software Technology in Organizing Universities for Industry 4.0 and Beyond

Mehmet Akşit
Software Technology
Formal Methods and Tools Group
University of Twente, Enschede,
The Netherlands
Email: m.aksit@utwente.nl

Abstract—This paper analyzes the recent developments around Industry 4.0 and beyond, identifies the necessary organizational structures of universities to assist companies in their transition processes, defines the relevant sub-disciplines in computer science and finally describes the software engineering and technology challenges in designing and implementing economical and robust high-quality Industry 4.0 systems.

I. INTRODUCTION

INDUSTRY 4.0 aims at increasing the efficiency and effectiveness of manufacturing processes with the help of large-scale computerization. It is stated that Industry 4.0 is the 4th industrial revolution in the history of manufacturing. Publications on Industry 4.0 generally try to set-up a conceptual framework to explain what the Industrial 4.0 phenomena is. There are also some ongoing efforts on structuring universities and knowledge institutes so that their research and education activities can seamlessly support this upcoming transition towards Industry 4.0. We consider Industry 4.0 as a natural development in continuous transition from traditional to modern manufacturing processes. Since transitions occur with the help of technology, organizational structure of universities are crucial in accomplishing the Industry 4.0 objectives. The role of computer science and software technology is undisputable, since the success of Industry 4.0 largely depends on effectiveness and efficiency of computing systems. Based on our experience with four large high-technology companies, this paper introduces the attributes of the phenomena Industry 4.0 and beyond, depicts the required organizational structures of universities to assist companies in their transition processes, identifies the relevant sub-disciplines in computer science and finally describes the software engineering and technology challenges in designing and implementing economical and robust high-quality Industry 4.0 systems.

II. INDUSTRY 4.0 AND BEYOND

The term Industry 4.0 refers to computerization of manufacturing processes. There are four principle scenarios in Industry 4.0 [1]:

- **Interoperability** meaning that sensors, devices, machines, and people can connect and exchange information with each other.

- **Information transparency** meaning that a rich set of data can be gathered from various sources.
- **Technical assistance** meaning that machines, systems, processes, human beings, etc. can be intelligently and effectively assisted to monitor, control and optimize the overall manufacturing process.
- **Decentralized decisions** meaning that subsystems can autonomously take decisions where possible.

It is claimed that Industry 4.0 is the 4th industrial revolution in the history of manufacturing [2]. With Industry 4.0, it is expected that machines and systems will become more self-aware and self-learning so that their effectiveness and maintenance can be improved. In addition, due to networked data gathering and intelligent and autonomous process control, the manufacturing processes will be much more efficient and effective than traditional manufacturing processes.

One criticism to these claims is that industrial innovation is continuous and as such one cannot talk about a revolution [3]. Moreover, although there are some attempts to define the technical implications of Industry 4.0, it seems that Industry 4.0 touches to a large set of disciplines from sensors, industrial manufacturing to computer science and software technology (CS-ST). In particular, almost all disciplines of CS-ST are relevant for Industry 4.0.

We think that the concepts relevant to Industry 4.0 must be defined and understood in the process of on-going transition from traditional to modern manufacturing processes. It is important to stress that such transitions are not abrupt in nature but gradual, depending on the characteristics of manufacturing, technological and societal progresses. In the following we will make an attempt to compare traditional and modern manufacturing processes with each other.

Traditionally, the term industrial manufacturing referred to labor-intensive factories with specialized product portfolio. Production processes and products to be manufactured had to be predefined precisely. Process and product control and optimization activities were carried out in each phase of production separately. Due to advent of new technologies and changes in social structures, however, there have been continuous changes to the ways how industrial manufacturing is realized. For example, during the last two to three decades, the following transitions have been observed in highly industrialized countries:

1. Knowledge intensive manufacturing **instead of** labor/resource-intensive manufacturing. There is a greater dependence on intellectual capabilities than on physical inputs or natural resources [4].
2. Focus on owning and managing knowledge and skills and intellectual property rights **instead of** focusing on labor/resource intensive manufacturing processes [5]. When planned carefully and in certain circumstances, it may be much more profitable to outsource some activities and manufacturing processes [6].
3. Dynamically managed and optimized, multi-asset portfolio **instead of** fixed/ad-hoc, single-asset portfolio. The advantages are risks reduction, controlled risk taking, capital preservation and enhanced returns [7][8].
4. Mass customization **instead of** mass production. Mass customization is the automated manufacturing of tailored products. It has the combined advantages of the low unit costs of mass production with the flexibility of building products for more customer satisfaction [9].
5. Proactive self-organizing companies **instead of** inflexible hierarchically-organized companies. Such new organizational structures aim at effectively responding to changing markets and business contexts [10].
6. End-to-end alignment and optimization of (manufacturing) processes **instead of** focusing only on the improvement of individual phases. This improves companies due to enhancement of the whole supply-chain [11].
7. Multi-disciplinary usage of teamed personnel **instead of** working with solely operating individuals. Teaming helps organizations in continuous improvement, understanding complex systems, and in successful innovation [12].
8. Organizing businesses/enterprises globally through networks **instead of** isolated and/or localized organizations. This is an increasing necessity for any entrepreneurship and value creation [13]-[15].
9. Improved time-to-market **instead of** long sequences of research, design, manufacturing and marketing phases [16].
10. Intensive use of state-of-art CS-ST as the “main enabler” of modern businesses **instead of** considering CS-ST just like any other technical skill. CS-ST is essential in fulfilling the requirements of modern businesses, such as described in [9][11][13]-[16].
11. Strong cooperation with universities for the purpose of innovation **instead of** considering universities mainly as theoretical institutions that educate people [17]-[19].

It is clear from this list that Industry 4.0 refers to (or a new name of) a part of a natural transition in manufacturing processes which has been taking place since several

decades. We therefore term this list as “a list of attributes of manufacturing processes for Industry 4.0 and beyond”.

III. 2. ORGANIZING UNIVERSITIES FOR INDUSTRY 4.0 AND BEYOND

Traditional universities have contributed to industries by educating engineers, applied mathematicians, administrative personnel, managers, etc. Within this context, two main categories of activities have been considered essential:

- **Research**, where academic personnel of the university are expected to be expert in certain fields. The selection of the topic of a field is not necessarily derived from industrial and societal needs; it can be ad hoc. The expertise is quite specific and theoretical. The excellence is measured according to number of publications in certain pre-classified journals.
- **Education**, where academic personnel of the university are expected to give lectures in their fields of expertise and examine the students by oral and written tests. In addition, students are expected to be supervised in writing their theses.

We believe that traditional universities with these characteristics cannot fulfil the requirements as demanded by modern industrial manufacturing processes and businesses as formulated by Industry 4.0. Moreover, classical education methods, like long lectures followed by classical oral and written examinations cannot give the necessary education baggage to the students as desired. As such, we think that the following strategic and tactical changes are required:

1. Excellence in knowledge and in-depth specialization for academic personnel are still required. However, **the specializations must be derived from the needs of the targeted society and industry** instead of ad-hoc selection of topics; along this line, the topics must be synthesized through the scope of industrial and societal mid-term and long-term objectives. Otherwise, universities cannot be equipped with the necessary expertise in supporting companies in their innovative processes.
2. **The academic personnel must learn to work together in multi-disciplinary teams.** For example, theory-oriented persons must be able to work with practically-oriented-persons, and vice versa, different experts in the same discipline or among different disciplines, must cooperate together in university-industry joint projects. Otherwise, the complex problems of industry and society cannot be addressed effectively.
3. **The academic personnel must be proactive in forming networks** to cooperate with national and international institutions and colleagues not only from his/her own discipline but also from other disciplines. This is necessary to share expertise, to jointly define

the desired research agenda, and to find solutions to complex industrial and societal problems.

4. **The academic personnel must be flexible enough to adapt themselves** in changing demands from industry and society. Otherwise, in due time, the expertise of academic personnel can be outdated or become less relevant.
5. The **education process must be tailored** to answer the mid-term and long-term needs of industry and society:
 - It must focus on the core concepts **instead of** hypotheses.
 - It must focus on gaining analytical skills, critical thinking and reasoning **instead of** memorizing what are in the books.
 - It must aim at teaching problem solving/synthesis **instead of** gaining knowledge which cannot be utilized for solving actual problems.
 - It must emphasize working in multi-disciplinary projects **instead of** only focusing on mono-disciplinary exercises.
 - It must enhance communication skills, such as oral and written presentation and argumentation skills **instead of** educating students with non-communicative and introvert attitude.
 - It must aim at increasing consciousness of students in ethical concerns **instead of** educating students with irresponsible and/or indifferent attitude.
6. The university must create **suitable organizational structures** to enable the academic personnel efficiently and effectively fulfil the objectives listed above. These include:
 - **Proactive and self-adaptive organization** to support the objectives of the university in dynamically changing contexts.
 - Organization to set-up and carry-out **multi-disciplinary projects** for industry and society.
 - Organization with an **award system to motivate** the academic personnel and students along the objectives of the university.
 - Organization which **emphasizes CS-ST** since it is the “main enabler” of all disciplines at the university.

To derive the required specializations within universities the following activities can be carried out:

1. **Understanding the context** of the university.
2. **Defining the strategic needs** of companies and society.
3. **Analyzing the current structure** of the university.

4. **Identifying the strong and weak points** of the university.
5. Based on the observations of the future trends as stated in the paper, **formulate a transition plan**.

These steps look quite obvious but due to involvement of many stakeholders, they are harder to implement than one may expect. While realizing the transition, the following quality attributes can be considered:

- **Relevancy:** The university must be highly relevant in addressing technical and social needs. To this aim, research, education and organization activities must be defined in close cooperation with the relevant companies and societal organizations.
- **Alignment with the current state-of-the-art research:** The research and education activities to be carried out must advance the state-of-the-art so that the companies and businesses can be matured to be the leaders in their context.
- **Cross-fertilization:** Different university research and education activities can benefit from each other. To maximize the benefit, it is important to strongly coordinate the related activities with each other.
- **Industry-as-laboratory:** To identify the relevant problems and to test the proposed solutions, it is important that the principle investigators and the affiliated (Ph.D. and/or M.Sc., etc.) students visit the companies regularly and carry out experiments within industrial and societal context. To this aim, companies must provide personal assistance and industrial case studies.
- **Academic research steering committee:** To coordinate the activities effectively and efficiently, it is important to monitor the progress of research and education activities and evaluate them with respect to the desired objectives. To this aim, academic research steering committees can be established where all the relevant stakeholders participate.

Inspired from the developments around Industry 4.0, recently, there have been some attempts to classify universities accordingly. To this aim, the concept of University 4.0 has been introduced for meeting the demands of Industry 4.0. In [20], University 4.0 is defined as “an university which is outward looking, deeply connected to industry and the communities around it, and committed to serving the needs of its students”. Although this definition is largely consistent with the definitions given in this paper, we consider evolution of industries and as well as universities as continuous processes which influence each other. Our focus in this paper is more from technological perspective, which needs to be supported by dedicated methods. Although we consider discussions around University 4.0 is useful, the conceptualization of this terminology is still in a premature state.

IV. THE DEVELOPMENTS IN COMPUTER SCIENCE AND SOFTWARE TECHNOLOGY

CS-ST is the main force in almost all industries; it creates added value for products and businesses. There is almost no product in the market which does not contain software or is not produced by a process controlled by software. To accomplish the objectives of Industry 4.0 and beyond, advanced CS-ST is needed.

There are all kinds of hypes over CS-ST in the popular media. Nevertheless, the recent developments in CS-ST are more or less shaped around the following topics:

1. Large infrastructures, service-oriented architectures, cloud computing, systems-of-systems [21]-[23].
2. Sensors, Internet of Things (IOT), and pervasive computing [24]-[26].
3. Big data and big data analytics [27]-[29].
4. Security and cybersecurity [30][31].
5. Cyber-physical systems [32].
6. Artificial intelligence and related topics including computational intelligence, machine learning and multi-agent systems [33]-[36].
7. Graphical processing, and visualization including virtual reality [37][38].
8. High performance, and/or multi-core/parallel architectures including parallel programming [39].
9. Theoretical and practical work on algorithms and/or constraint-based “solvers” [40][41] to address a large category of mathematical problems. In general algorithms/solvers are applied to every category of computer science specializations listed in this section.
10. Software (engineering) methods and techniques [42] to fulfil the functional and qualitative requirements of software systems. The concepts of software engineering can be applied to every computer science specialization listed in this section.

V. THE ROLE OF SOFTWARE ENGINEERING METHODS AND TECHNIQUES

Economical, sustainable and robust software systems which fulfil functional and qualitative requirements are essential for all software systems. To accomplish the requirements of Industry 4.0 and beyond, software engineering methods and techniques are crucial. **No matter how intelligent a software solution is, if it cannot be realized with the desired quality attributes, one cannot expect an economical value out of it.** As such software engineering methods and techniques can be defined as crosscutting (meta-level) concerns that relate to all developments within CS-ST.

The trends in software engineering methods and techniques cannot be considered separately from the recent developments in computer science. After 4 years of intensive cooperation with high-tech industry, for example, we have identified the following trends [19]:

1. Product-line **instead of** product design. Most products are developed and manufactured by specialized

companies, which market families of products. It is not economical to develop each product from scratch [43][44].

2. Systems of systems **instead of** systems perspective. Software systems for Industry 4.0 are generally adopted in large distributed settings. Scale-ability and interoperability of systems are essential. Systems of systems architectures, are therefore the natural candidates of the platforms of Industry 4.0 architectures [23].
3. Ecosystem design **instead of** platform design. Software ecosystems are an effective and economical way to construct large software systems for Industry 4.0 on top of a software platform by adding up software modules developed by different actors. In ecosystem design software engineering is spread outside the traditional borders of software companies to a group of companies and private persons [45].
4. Auto-adaptive control architectures **instead of** architectures without any control mechanisms. To realize the monitoring and controlling activities in Industry 4.0 and to cope with the changing requirements and context, software systems are expected to be more reactive and self-adaptive. This generally requires built-in feedback control mechanisms in software. Self-adaptation can be realized at system level, sub-system level and/or at component-level. In addition, different styles can be adopted, such as single, master-slave, hierarchical and/or peer-to-peer control architectures [46][47].
5. Distributed problem solving including distributed algorithms, coordinating systems and multi-agent architectures **instead of** centralized problem solving with monolithic and/or localized architectures. Since computer systems for Industry 4.0 are distributed, to reduce complexity and enhance reliability/availability, algorithms and intelligence in systems must be distributed as well. Accordingly, programming languages and techniques must adequately support distributed programming efforts by offering expressive and flexible abstractions [48]-[52].
6. Model-based development **instead of** straight-forward programming. Since more and more companies are specialized in certain product categories and in manufacturing processes, deriving software architecture from relevant domain models can help in reducing complexity, enhancing reuse and testability/verifiability of software systems. Model-based development has been adopted in various approaches such as product-line engineering (SPLE), model-driven engineering (MDE), domain specific architectures (DSA) and domain-specific programming languages (DSL), model-based verification (MBV), etc. [53]-[56].
7. Multi-objective optimization **instead of** ad-hoc hand-crafted and/or single objective optimization. Along the line of model-based development, various algo-

rhythmic techniques and search-based methods have been introduced to compute the “optimal” architectural decomposition with respect to certain quality attributes. In addition, various run-time optimization techniques can be adopted in computing optimal control strategies and scheduling processes [57]-[59].

8. Modularization of semantic concerns **instead of** traditional abstraction mechanisms based on implementation concerns such as data or function. As a consequence of model-based development, software abstractions more and more correspond to the concerns of models. The concerns of a model are naturally based on the semantics of the model, and these cannot always be effectively represented as a data or function. Moreover, concerns in Industry 4.0 systems can be emerging meaning that they may appear or disappear dynamically. As such, programming languages and techniques must adequately support programming efforts by offering expressive and flexible abstractions for emergent semantic concerns [60][61].
9. A rich set of composition mechanisms **instead of** a fixed set of language constructs for hierarchical organization of programs (such as class-inheritance). To support flexibility in control strategies and to cope with various evolution schemes, languages must offer generic and/or domain specific composition mechanisms to express, for example, object, aspect and event compositions and transformational techniques in a uniform manner. The languages must maintain their closure property in compositions so that scalability of systems can be provided [62]-[67].
10. Uniform integration of verification techniques **instead of** independent tool- and technique-specific verification approaches. There are various model-based verification and testing approaches available. Examples are model-checking, static and dynamic analysis, run-time verification, model-based testing, adopting model-specific verification (simultaneously) based on continuous and/or discrete models, etc. Most of these techniques are complementary and as such combined usage of these may help in finding faults with less false-positive and false-negative cases [62][67]-[69].

VI. CONCLUSIONS

This paper claims that Industry 4.0 refers to a continuous evolution process in manufacturing processes enabled by the advancements in CS-ST. Within this context, trends in university organizations and theoretical and practical developments in CS-ST must be monitored carefully. This paper has presented a set of attributes in carrying out actions in strategical planning of universities and CS-ST departments.

REFERENCES

- [1] M. Hermann et al. “Design Scenarios for Industrie 4.0 Scenarios,” in *Proc. 49th Hawaii International Conference on System Sciences*, 2016, pp. 3928-3937.
- [2] S. Vaidya et al. “Industry 4.0 – A Glimpse”. in *Proc. of International Conference on Materials Manufacturing and Design Engineering*, 2018, pp. 233-238.
- [3] E. Garbee. “This Is Not the Fourth Industrial Revolution”, –via *Slate* at http://www.slate.com/articles/technology/future_tense/2016/01/the_world_economic_forum_is_wrong_this_isn_t_the_fourth_industrial_revolution.html.
- [4] W. Powell and K. Snellman, “The Knowledge Economy”, *Annu. Rev. Sociol.* 2004. doi: 10.1146/annurev.soc.29.010202.100037, 2004, pp: 199–220.
- [5] D. Modic and N. Damij, *Towards Intellectual Property Rights Management: Back-office and Front-office Perspectives*, Springer International Publishing AG, 2018.
- [6] S. Cullen and M. Lacity, *Outsourcing- All You Need To Know*, White Plume Publishing, 2014.
- [7] P. Sironi, *Modern Portfolio Management: From Markowitz to Probabilistic Scenario Optimisation*, Risk Books, 2015.
- [8] Y. Lustig, *Multi-Asset Investing: A practical Guide to Modern Portfolio Management*, Harriman House Ltd., 2013.
- [9] H. Kull, *Mass Customization: Opportunities, Methods, and Challenges for Manufacturers*, Springer Science Business Media, 2015.
- [10] B. Robertson, *Holacracy: The Revolutionary Management System that Abolishes Hierarchy*, Henry Holt and Company LCC, 2016.
- [11] Oracle, *Why End-to-end Visibility is Key to a Modern manufacturing Process*.
- [12] A. C. Edmondson, *Teaming: How Organizations Learn, Innovate, and Compete in the Knowledge Economy*, Wiley, 2014.
- [13] D. Sherman, *Maximum Success with LinkedIn: Dominate Your Market, Build a Global Brand, and Create the Career of Your Dreams*, McGraw-Hill Education Books, 2014.
- [14] S. Thomas, *Instant Networking: The Simple Way to Build Your Business Network and See Results in Just 6 Months*, John Wiley and Sons Ltd., 2016.
- [15] E. Kaynak, R. Ajami, and M. Bear (Eds.), *The Global Enterprise: Entrepreneurship and Value Creation*, International Business Press, 2012.
- [16] P. Smith, “Accelerated Product Development: Techniques and Traps”, in the *PMDA Handbook of New Product Development. Second Edition*, K. Kahn (Ed.), John Wiley and Sons Inc., 2004, ch. 12.
- [17] C. Mascarenhas, J. Ferreira, and C. Marques, *University–industry Cooperation: A Systematic Literature Review and Research Agenda*, Science and Public Policy, Oxford Academic, 2018, pp. 1–11.
- [18] M. Akşit, B. Tekinerdogan, H. Sözer, H. F. Safi and M. Ayas, “The DESARC Method: An Effective Approach for University-Industry Cooperation”, in *Proc. of the International Conference on Advances in Computing, Control and Networking, ACCN 2015*, 2015, pp. 51-53.
- [19] M. Akşit, B. Tekinerdogan, H. Sözer, H. F. Safi and M. Ayas, “Identifying the Research Needs of Four Large High-Technology Companies”, in *Proc. ACCN 2016*, 2016, pp. 21-24.
- [20] J. Dewar, “University 4.0: Redefining the Role of Universities in the Modern Era”, in *Higher Education Review Magazine*, August 2017.
- [21] K. M. Dhara, M. Dharmala, and C. K. Sharma, *A Survey Paper on Service Oriented Architecture Approach and Modern Web Services*, All Capstone Projects, <http://opus.govst.edu/capstones/157>, 2015.
- [22] D. C. Marinescu, *Cloud Computing: Theory and Practice*, Morgan Kaufmann, 2017.
- [23] DoD, *Systems Engineering Guide for Systems of Systems*, version 1.0, 2008.
- [24] G. Ferrari (Ed.), *Sensor Networks Where Theory Meets Practice*, Springer, 2010.
- [25] R. Buyya, and A. V. Dastjerdi (Eds.), *Internet of Things Principles and Paradigms*, Elsevier, 2016.
- [26] N. Silvis-Cividjian, *Pervasive Computing Engineering Smart Systems*, Springer, 2017.
- [27] H. Mohanty et al., *Big Data A Primer*. Springer, 2015.
- [28] S. Pyne et al. (Eds.), *Big Data Analytics Methods and Applications*, Springer, 2016.

- [29] P. Pääkkönen, and D. Pakkala, "Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems", *Big Data Research*, Elsevier, vol. 2, no. 4, December 2015, pp. 166-186.
- [30] L. Thames and D. Schaefer, *Cybersecurity for Industry 4.0*, Springer, 2017.
- [31] M. Lehto and P. Neittaanmäki (Eds.), *Cyber Security: Analytics, Technology and Automation*, Springer, 2015.
- [32] S. C. Suh, et al. *Applied Cyber-Physical Systems*, Springer, 2014.
- [33] Y. Laalaoui, and N. Bouguila (Eds.), *Artificial Intelligence Applications in Information and Communication Technologies*. Springer, (2015).
- [34] R. Kruse et al. *Computational Intelligence A methodological Introduction*, Springer, 2016.
- [35] I. Goodfellow et al., *Deep Learning*, MIT Press, 2016.
- [36] M. Hadzic et al., *Ontology-Based Multi-Agent Systems*, Springer, 2009.
- [37] C. Ware, *Information Visualization Perception for Design*, Elsevier ScienceDirect, 2012.
- [38] F. R. Leta (Ed.), *Visual Computing*, Springer, 2014.
- [39] T. Rauber and G. Rüniger, *Parallel Programming for Multicore and Cluster Systems*, Springer, 2013.
- [40] B. Vöcking et al., *Algorithms Unplugged*, Springer, 2011.
- [41] F. Rossi et al., "Handbook of Constraint Programming", in *Foundations of Artificial Intelligence*, vol. 2, Elsevier, 2006.
- [42] I. Sommerville, *Software Engineering*, 10th Edition, Pearson, 2016.
- [43] F. van der Linden et al. *Software Product Lines in Action*, Springer, 2007.
- [44] G. Orhan, M. Akşit and A. Rensink, "A Formal Product-Line Engineering Approach for Schedulers", in *Proc. 22nd International Conference on Emerging Trends and Technologies in Convergence Solutions*, L. Jololian, D. E. Robbins and S. L. Fernandes (Eds.), Nov 2017, pp. 15-30.
- [45] K. Manikas and K. M. Hansen, "Software Ecosystems – A Systematic Literature Review", *The Journal of Systems and Software*, 86, 2013, pp. 1294–1306.
- [46] S. Kounev et al., *Self-Aware Computing Systems*, Springer, 2017.
- [47] M. Akşit and Z. Choukair, "Dynamic Adaptive and Reconfigurable Systems Overview and Prospective Vision", in *Proc. Workshop on Distributed Auto-adaptive Reconfigurable Systems (DARES)* - International Conference on Distributed Computing Systems (ICDCS), May 2003, Rhode Island, Providence, USA, pp. 84-89.
- [48] M. Raynal, *Distributed Algorithms for Message-Passing Systems*, Springer, 2013.
- [49] M. Akşit and L. Bergmans, "Guidelines for Identifying Obstacles When Composing Distributed Systems from Components", in *Software Architectures and Component Technology*, The Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, Dordrecht, 2002, pp. 29-56.
- [50] M. Akşit, K. Wakita, J. Bosch, L. Bergmans and A. Yonezawa, "Abstracting Object Interactions Using Composition Filters", in *Object-Based Distributed Processing*, Lecture Notes in Computer Science 791, Springer Verlag, 1993, pp. 152-184.
- [51] D. Shuai and X. Feng, "Distributed Problem Solving in Multiagent Systems: a Spring Net Approach", *IEEE Intelligent Systems*, vol. 20, no. 4, 2005, pp. 66 – 74.
- [52] M. Hadzic et al., *Ontology-Based Multi-Agent Systems*, Springer, 2009.
- [53] M. Akşit, B. Tekinerdogan, F. Marcelloni, and L. Bergmans, "Deriving Object-Oriented Frameworks from Domain Knowledge", in *Building Application Frameworks: Object Oriented Foundations of Framework Design*, John Wiley & Sons, 1999, pp. 169-198.
- [54] J. Z. Pan et al., *Ontology-Driven Software Development*, Springer, 2013.
- [55] Reinhartz-Berger et al., *Domain Engineering Product Lines, Languages, and Conceptual Models*, Springer, 2013.
- [56] A. R. Da Silva, "Model-Driven Engineering: A Survey Supported by the Unified Conceptual Model", *Elsevier Computer Languages, Systems and Structures*, 43, 2015, pp. 139-155.
- [57] A.J. de Roo, H. Sözer, L. Bergmans, M. Akşit, "MOO: An Architectural Framework for Runtime Optimization of Multiple System Objectives in Embedded Control Software", *Journal of Systems and Software*, 86 (10), 2013, pp. 2502-2519.
- [58] H. Sözer, B. Tekinerdogan, M. Akşit, "Optimizing Decomposition of Software Architecture for Local Recovery", *Software Quality Journal*, 21 (2) 2013, pp. 203-240.
- [59] M. Harman et al., "Search-Based Software Engineering: Trends, Techniques and Applications", *ACM Computing Surveys*, vol. 45, Issue 1, Article no. 11, 2012.
- [60] S. Malakuti Khah Olun Abadi, and M. Akşit, "On Liberating Programs from the Von Neumann Architecture via Event-based Modularization". in *Companion Proc. of the 14th International Conference on Modularity*, New York: Association for Computing Machinery (ACM), 2015, pp. 31-34.
- [61] S. Malakuti Khah Olun Abadi and M. Akşit, "Emergent Gummy Modules: Modular Representation of Emergent Behavior", in *Proc. of the 2014 International Conference on Generative Programming: Concepts and Experiences (GPCE)*, 2014, pp. 15-24.
- [62] de Roo, A.J. and Sözer, H. and Akşit, M. (2014) Composing domain-specific physical models with general-purpose software modules in embedded control software. *Software and Systems Modeling*, 13 (1). pp. 55-81.
- [63] L. Bergmans, W. Havinga, and M. Akşit, "First-Class Compositions-- Defining and Composing Object and Aspect Compositions with First-Class Operators", *Transactions on Aspect-Oriented Software Development*, IX, 2012, pp. 216-267.
- [64] S. Malakuti Khah Olun Abadi and M. Akşit, "Evolution of Composition Filters to Event Composition", in *Proc. 27th ACM Symposium on Applied Computing (SAC 2012)*, 2012, pp. 26-30.
- [65] L. Bergmans and M. Akşit, "Composing Crosscutting Concerns Using Composition Filters", *Communications of the ACM*, 44 (10), 2001, pp. 51-57.
- [66] T. Elrad, M. Akşit, G. Kiczales, K. Lieberherr and H. Ossher, "Discussing Aspects of Aspect-oriented Programming", *Communications of the ACM*, 44 (10), 2001, pp. 33-38.
- [67] M. Akşit, "The 7 C's for Creating Living Software: A Research Perspective for Quality-oriented Software Engineering", *Turkish Journal of Electrical Engineering & Computer Sciences*, 12 (2), 2004, pp. 61-95.
- [68] S. Ciraci, S. Malakuti, S. Katz and M. Akşit, "Checking the Correspondence Between UML Models and Implementation", in *Proc. of the 1st International Conference on Runtime Verification*, 2010, pp. 198-213.
- [69] B. Nielsen, "Towards a Method for Combined Model-based Testing and Analysis", in *Proc. of the 2nd International Conference on Model-Driven Engineering and Software Development*, 2014, pp. 609-618.