Editorial: “Complexity of Systems Evolution: Requirements Engineering Perspective”

MATTHIAS JARKE, RWTH Aachen University and Fraunhofer FIT
KALLE LYYTINEN, Case Western Reserve University

Walking on water, and programming according to specifications is easy—as long as both of them are frozen.
—Robert Glass

This introduction discusses the changing nature of complexity associated with requirements engineering (RE) tasks and how it has shifted from managing internal complexity to adapting and leveraging upon external and dynamic complexity. We note several significant drivers in the requirements knowledge that have resulted in this change and discuss in light of complexity theory how the RE research community can respond to this. We observe several research challenges associated with “new complexity” and highlight how the articles included in the special issue advance the field by defining complexity more accurately, observing more vigilantly new sources of complexity, and suggesting new ways to manage complexity in terms of economic assessments, knowledge flows, and modeling for adaptability.

Categories and Subject Descriptors: D.2.1 [Requirements/Specifications]: Methodologies

General Terms: Design, Management

Additional Key Words and Phrases: System evolution, complexity, requirements knowledge, requirements engineering, agility, edge computing

ACM Reference Format:

Matthias Jarke and Kalle Lyytinen. 2015. Editorial: “Complexity of systems evolution: Requirements engineering perspective”. ACM Trans. Manag. Inform. Syst. 5, 3, Article 11 (January 2015), 7 pages.
DOI: <http://dx.doi.org/10.1145/2629597>

1. REQUIREMENTS ENGINEERING AND COMPLEXITY

The genesis of requirements engineering (RE) research around the mid 1970s was motivated by the practitioner’s need for more disciplined software development, as software had grown large and unmanageable [Brooks 1995; Ross 1977; Ross and Schoman Jr. 1977]. In this regard, from its inception, RE has been an offspring of the need to mitigate and manage complexity associated with software. Consequently, much of the RE research since then has focused on ways of bringing about intellectual discipline in identifying, reducing, and mitigating the inner complexity of software by capturing, sharing, representing, analyzing, negotiating, and prioritizing requirements as a basis for effective design decisions and interventions. This drive is evidenced by the significant volume and impact of requirements-related papers published in top software and IS conferences and journals (for a survey, see Cheng et al. [2009]). Many results of

This work was supported by the Deutsche Forschungsgemeinschaft under Excellence Cluster 89 UMIC.

Authors’ addresses: M. Jarke, Information Systems, RWTH Aachen University, Ahornstr. 55, 52074 Aachen, Germany; email: jarke@dbis.rwth-aachen.de; K. Lyytinen, 10900 Euclid Ave., Cleveland, OH 44106, USA; email: kjl113@case.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credits permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121–0701 USA, fax +1 (212) 869–0481, or permissions@acm.org.

© 2015 ACM 2158-656X/2015/01-ART11 \$15.00

DOI: <http://dx.doi.org/10.1145/2629597>

this research stream, such as the use of business and system modeling (e.g., entity-relationship and UML diagrams, use cases [Jacobsen et al. 1997; Jarke et al. 1998]), risk-driven methodologies [Boehm 1988], structured requirements documents, and requirements tracing [Gotel and Finkelstein 1994; Ramesh and Jarke 2001], have found their way into design practice [Lyytinen et al. 2009].

2. A NEW TYPE OF REQUIREMENTS ENGINEERING COMPLEXITY

Externally, the design complexity in RE deals with how software and its components interact with and relate to increasingly heterogeneous and large numbers of technical and social components in the environment, including components that are contributed by the “edge” of organizations (freelancers, end users, small innovative companies) exploiting modern platform capabilities. The design task is more about adjusting multiple interconnected software systems and components and improving their environmental “fit” by adapting them into a growing number of technical, social, and organizational subsystems [Hanseth and Lyytinen 2010]. Another challenge is the growing size of the software initiatives. The number of included components and their interactions has grown exponentially. The initiatives span several years and often cost billions or hundreds of millions of dollars; moreover, these initiatives encounter constant changes in their requirements, designs, and underlying technologies [Wu et al. 2011]. Inadequate means to manage this complexity often have resulted in their dismal performance.

The environment in which RE is practiced has changed dramatically from the software landscape of 40 years ago. Whereas most of the interest in the past focused on understanding and managing the *inner* and *static* complexity of the design task by using abstraction, modularization, and related principles, today’s complexity is of a different ilk. It is also *external* and *dynamic*. Accordingly, the sources, nature, and dynamics of complexity in RE have changed dramatically.

Partly, this externalization of complexity is due to advances in hardware and telecommunications technologies that have lowered computing cost and extended software functionality and flexibility, making software ubiquitous. Partly, this change is due to changes in task and organizational environments where software is either produced or deployed. Organizations now operate globally, they need to be nimble, and they need to innovate at a constant pace with digitally created options [Sambamurthy and Zmud 2003]. This has widened the boundaries and functions of information systems into interorganizational or industry-wide systems creating new types of interdependencies and new forms of complexity. At the same time, software functionality has become “liquefied” as time to market is more important than product quality, whereby software is run in a perpetual state of beta versions [O’Reilly 2005]. As a result, the field’s focus and scope has shifted from engineering of individual systems and components toward the generation, adaptation, and maintenance of software-intensive *ecosystems* consisting of software, hardware, human and organizational agents, business processes, regulators, and more.

We illustrate the changing nature and scope of requirements and associated complexity between the traditional and new RE in Table I by enlisting their differences in six “V’s of RE”: volume, veracity, volatility, vagueness, variance, and velocity. Those features that are unique and novel to the new environment are expressed in a grey shade. In particular, we can note that the nature of RE activity today is significantly different in terms of the requirements of volatility, vagueness, variance, and velocity. All of these are manifestations of heightened levels of external complexity, which is likely to result in different dynamics of RE task.

This shift in the RE knowledge, and how it is managed, has thus changed the complexity into a highly dynamic one. Traditional RE approaches tended to focus on static, internal complexity and primarily dealt with volume and veracity. Therefore,

Table I. Six V's of Requirements in Traditional and New Requirements Engineering

Requirement Feature	Definition	Old Complex Requirements	New Complex Requirements
Volume	The size of the requirements pool influencing the scope of the work	Major focus of RE as it influences effort estimation (medium to large)	Significant during RE as it influences effort estimation [Hansen et al. 2009] (large to ultralarge)
Veracity	To what extent requirements express the needs of the stakeholders and are consistent	Emphasized as the key feature of RE task; works well if requirements can be frozen	Important as an ideal, but not key, feature of most RE efforts [Ross 1977; Jarke et al. 2011]
Volatility	The rate at which the requirements change over a given period of time	Recognized as a key reason for the failure of waterfall (e.g., Brooks [1975])	Constant feature of software development for most environments [Hansen et al. 2009]
Vagueness	To what extent designers and other stakeholders understand the content and consequences of the requirement	Not recognized as an important element other than to be avoided during RE task	Inherent feature of many RE initiatives due to initial lack of user learning or understanding of the dynamism introduced by the software in the environment
Variance	The variation in the design scope and consequences of the requirement pool and the heterogeneity of design components involved	Not recognized as an important element in RE activity	Significant element influencing RE dynamics and complexity [Jarke et al. 2011; Hansen et al. 2009]
Velocity	The rate at which requirements are changing over time	Not important and recognized	Significant contributor to specific context of RE, especially in software platforms [Fisher et al. 2013]

indeed, they appear no longer to be able to “scale to manage these increasingly complex, globally distributed systems at reasonable cost or project risk” [British Computer Society, 2004, p. 1].

In this new situation, IS scholars need to heed to and deal with a new sort of dynamic complexity, filled with new degrees of the requirements of volatility, vagueness, variance, and velocity. This invites the exploration of new intellectual strategies to deal with such change. Most importantly, scholars must ask how we can reconceptualize RE tasks and conditions for their effective execution in light of this dynamic complexity. Taking such complexity seriously requires scholars to acknowledge that this sort of complexity—as a phenomenon—is no longer a marginal feature of RE. Consequently, they need to expand and deepen the range of theoretical frameworks that help conceptualize new RE complexity and generate strategies to mitigate its effects.

To address these issues, scholars need to first obtain a firmer intellectual grasp of what this sort of complexity is and ground it on quite extensive theoretical work on complexity and systems theory (see the work of King and Simon in this issue). In this stream, complexity generally refers to an emergent property of a system composed of large numbers of self-organizing agents that interact in a dynamic and nonlinear fashion and share a path-dependent history [Cilliers 1998]. In complex systems, tight

couplings and complex interactions are likely to amplify positive feedback such as drastic changes in velocity. They may also lead to catastrophic events such as those resulting from significant variance in the requirements knowledge (and related challenges to integrate and make sense of it). Furthermore, the new complexity operates between order and disorder of (social) systems at the “edge of chaos,” whereby software systems gain generative properties (increasing vagueness), shift unexpectedly to new developmental trajectories (again increasing volatility, velocity, or variance), and consequently cannot be managed solely by classical RE methods that mitigate against internal complexity

3. ADDRESSING NEW REQUIREMENTS ENGINEERING COMPLEXITY: GOALS OF THIS SPECIAL ISSUE

The realization of the presence of a new type of complexity has significant repercussions on how scholars and practitioners alike approach requirements management, requirements evolution, RE tasks, or software economics. To address some of these issues, the guest editors of this special issue have been involved and organized a series of scholarly workshops that invited participants both from academia and industry. These workshops were organized both in the United States (see Lyytinen et al. [2010]) and Europe (Dagstuhl workshops in 2009 [Jarke and Lyytinen 2010] and 2012 (this issue)), and they deeply examined and debated these topics. This exploration has also produced an RE manifesto [Jarke et al. 2011] that outlines some unique research challenges in RE, as well as some solution strategies associated with this new form of complexity.

This special issue advances this line of inquiry. It is an outgrowth of debates held in the third workshop organized in Dagstuhl in October 2012, which specifically focused on new types of requirements complexity and its sources [Huang et al. 2012]. It addressed the following challenges: (1) to deal better with context changes and business goal management to reduce the “black swan” rate of badly failed large projects, (2) to exploit recent theories of technological and institutional evolution to understand better how to control complexity and leverage it for software-based innovation, and (3) the demand for runtime reorganization of existing large-scale systems with respect to new operational and RE goals.

When we proposed this special issue, we had several goals in mind. First, we wanted this special issue to be part of an ongoing effort to frame, influence, and direct the discourse around the new “brave” world of RE to become more relevant to the increasingly complex world of software systems. Second, we wanted to shift the focus to examining RE as constant shaping of “sociotechnical systems” in the form of design requirements that are attentive to and reflect the emerging interactions among people, software, and technologies. These all need to be recognized while managing complex system requirements, and they require simultaneous and dynamic understanding of social and technical elements, as well as their complex interactions. Third, we emphasized—in the spirit of new complexity—RE as an evolutionary approach that constitutes an ongoing, nonlinear process of adoption, mutual adaptation, and co-evolution of mappings between problem spaces to solution spaces across multifaceted sociotechnical systems. Fourth, we wanted to embrace issues of RE scalability in the context of multidimensional change. There is some proof that digital infrastructures now enable a different scale of RE—for instance, open source software development or app ecosystems. There is, as yet, little systematic investigation of how scalability and dynamism interact and how such interaction influences RE.

To this end, this special issue focuses on a series of interrelated questions:

- (1) How are different forms of complexity within RE theorized and studied?
- (2) What are the new perspectives on complexity, such as biological systems and complexity evolutionary economics?

- (3) What perspectives can inform how and why complex requirements knowledge evolves as it is generated, validated, and distributed?
- (4) How do requirements, system evolution, and environmental change interact?
- (5) How do different types of knowledge interact to shape requirements and their evolution?
- (6) What are the origins and flows of influence of requirements knowledge for complex evolving systems, and how can nonlinear influences be effectively managed in the RE evolution?

As can be seen in this list of questions, this special issue specifically seeks to integrate (sociotechnical) theories of system evolution; cognition within complex environments; models of RE and design knowledge and their economic effects; the impact of design strategy and related knowledge endowments during RE processes (e.g., explorative vs. exploitative processes of requirements discovery); and the role of vagueness, volatility, and increased variance and velocity in managing RE. Attention is placed on research approaches and methods that can be brought to bear in addressing these problems.

4. CONTENT OF THE SPECIAL ISSUE

Overall, we received 14 submissions for the special issue. After careful review, we decided to publish five articles, which amounts to a little more than a 30% acceptance rate. Overall, the accepted articles can be organized into three major themes:

- *Defining complexity* includes the article by King and Simon titled “Complications with Complexity in Requirements Engineering.” This theme addresses research question (1) with regard to the nature and challenges associated with this new complexity.
- *Sources of complexity* includes the articles “Getting to the Shalls: Facilitating Sense-Making in Requirements Engineering” by Rosenkrantz et al. and “Impediments to Regulatory Compliance of Requirements in Contractual Systems Engineering Projects: A Case Study” by Nekvi et al. This theme addresses research questions (3), (4), and (5) and focuses on the origins of new complexity in terms of heterogeneity of knowledge, system components, and their differential rates of change that drive volatility and variance up in requirements work.
- *Mitigating complexity* includes the articles “Improving Business Value Assurance in Large-Scale IT Projects—A Quantitative Method Based on Founded Requirements Assessment” by Wolf et al., “Situated Boundary Spanning: An Empirical Investigation of Requirements Engineering Practices in Product Family Development” by Ramesh et al., and “The Requirements Problem for Adaptive Systems” by Jureta et al. This theme addresses research question (6) regarding how to deal with increased complexity. The articles specifically focus on the evolution of how economic value of the large projects are assessed, how knowledge flows and knowledge representations are managed in more complex product platform environments, and how formal approaches can deal with new modeling challenges associated with system adaptability.

We want to extend our thanks to all of the reviewers who put their effort and time into conducting timely and high-quality reviews and thereby made this special issue possible: Pekka Abrahamson, Roman Beck, Nick Berente, Lam Cao, Jane Cleland-Huang, Patrick Delfmann, Jörg Dörr, Alan Fisk, Xavier Frank, Anna Hannemann, Sean Hansen, Timo Kakola, Amol Kharabe, Sengupta Kishore, Ralf Klamma, Julio Leite, Horst Lichter, Rikard Lindgren, Florian Matthes, John Mylopoulos, Selmin Nurcan, Andreas Oberweis, Barbara Paech, Barbara Pernici, Bala Ramesh, Isabelle

Reymen, Stefanie Rinderle-Ma, William (Bill) Robinson, Thomas Rose, Matti Rossi, Timo Saarinen, Zixing Shen, Alistair Sutcliffe, Richard Svensson, John Tripp, and Fan Yang-Turner. We also want to thank several colleagues who were instrumental in organizing the Dagstuhl and earlier workshops, including Peri Loucopoulos, John Mylopoulos, Bill Robinson, Sean Hansen, Nick Berente, Lin Liu, and Jane Cleland-Huang. We also thank Hsinchun Chen and Cathy Larson for their help and unwavering support while preparing this special issue.

Last but not least, this special issue appears in the very month in which Professor Reinhard Wilhelm—the founding director of Dagstuhl Castle, which internationally is probably the most important computer science meeting center—retires after more than 25 years of leadership to the computer science community. We therefore dedicate this special issue to Reinhard Wilhelm, who was always highly supportive in organizing RE-related workshops in Dagstuhl. This special issue comes thus with warm thanks on behalf of the whole RE community.

REFERENCES

- B. W. Boehm. 1988. A spiral model of software development and enhancement, *Computer* 21, 5, 61–72.
- British Computer Society. 2004. The Challenges of Complex IT Projects. British Computer Society, Royal Academy of Engineering. Retrieved November 15, 2005, from <http://www.bcs.org/upload/pdf/complexity.pdf>.
- F. Brooks. 1975. *The Mythical Man Month*. Addison-Wesley.
- B. H. C. Cheng and J. M. Atlee. 2009. Current and future research directions in requirements engineering. In *Design Requirements Engineering: A Ten-Year Perspective*. Lecture Notes in Business Information Processing, Vol. 14. Springer, 11–43.
- P. Cilliers. 1998. *Complexity and Postmodernism*. Routledge, London.
- J. Cleland-Huang, M. Jarke, L. Liu, and K. Lyytinen (Eds.). 2012. Requirements management: Novel perspectives and challenges, Report from Dagstuhl Seminar 12442. *Dagstuhl Reports* 2, 10.
- M. Fisher, M. Abbott, and K. Lyytinen. 2013. *The Power of Customer MisBehavior*. Palgrave & McMillan, London and New York.
- P. Frederick and F. Brooks. 1995. *The Mythical Man-Month*. Addison-Wesley.
- O. C. Z. Gotel and C. W. Finkelstein. 1994. An analysis of the requirements traceability problem. In *Proceedings of the 1st International Conference on Requirements Engineering*, 94–101.
- S. Hansen, N. Berente, and K. Lyytinen. 2009. Requirements in the 21st century: Current practice and emerging trends. In *Design Requirements Engineering: A Ten-Year Perspective*. Lecture Notes in Business Information Processing, Vol. 14. Springer, 44–87.
- O. Hanseth and K. Lyytinen. 2010. Design theory for adaptive complexity in information infrastructures. *Journal of Information Technology* 25, 1, 1–19.
- I. Jacobsen, M. Christerson, P. Jonsson, and G. Övergaard. 1997. Object-oriented software engineering: A use case-driven approach. In *Scenario-Based Design*, J. Carroll (Ed.): Wiley.
- M. Jarke, X. T. Bui, and J. M. Carroll. 1998. Scenario management: An interdisciplinary approach. *Requirements Engineering Journal* 3, 3–4, 155–173.
- M. Jarke, P. Loucopoulos, K. Lyytinen, J. Mylopoulos, and W. Robinson. 2011. The brave new world of design requirements: Four key principles. *Information Systems* 36, 6, 992–1008.
- M. Jarke and K. Lyytinen (Eds.). 2010. *Special Issue on High Impact Requirements Engineering Wirtschaftsinformatik / BISE* 53, 2.
- K. Lyytinen, P. Loucopoulos, J. Mylopoulos, and B. Robinson. 2009. *Design Requirements Engineering: A Ten-Year Perspective*. Lecture Notes in Computer Science, Vol. 14. Springer.
- T. O'Reilly. 2005. What is Web 2.0? Design patterns and business models for the next generation of software. *IEEE Software* 30.
- B. Ramesh and M. Jarke. 2001. Toward reference models for requirements traceability. *IEEE Transactions on Software Engineering* 27, 1, 58–93.
- D. T. Ross. 1977. Structured analysis (SA): A language for communicating ideas. *IEEE Transactions on Software Engineering* SE-3, 1, 16–34.

- D. T. Ross and K. E. Schoman Jr. 1977. Structured analysis for requirements definition. *IEEE Transactions on Software Engineering* SE-3, 1, 6–15.
- V. Sambamurthy and R. W. Zmud. 2003. Research commentary: The organizing logic for an enterprise’s IT activities in the digital era—a prognosis of practice and a call for research. *Information Systems Research* 11, 2, 105–114.
- W. W. Wu, G. M. Rose, and K. Lyytinen. 2011. Recognizing and managing innovation points in large information technology projects. *MIS Quarterly Executive* 10, 3, 121–132.

Received April 2014; accepted May 2014