2008 15th Asia-Pacific Software Engineering Conference

Software Process Simulation Modeling: Facts, Trends and Directions

He Zhang
Lero Research Center
Department of Computer Science
and Information Systems
University of Limerick, Ireland
he.zhang@lero.ie

Barbara Kitchenham
School of Computer Science
and Mathematics
Keele University, UK
barbara@cs.keele.ac.uk

Dietmar Pfahl
Simula Research Laboratory
Department of Informatics
University of Oslo, Norway
dietmarp@simula.no

Abstract

Software Process Simulation Modeling (SPSM) research has increased since the first ProSim Workshop held in 1998 and Kellner, Madachy and Raffo (KMR) discussed the "why, what and how" *of process simulation. This paper aims to assess how SPSM has evolved during the past 10 years in particular whether the reasons for SPSM, the simulation paradigms, tools, problem domains, and model scopes have changed. We performed a systematic literature review of software process simulation papers from the ProSim series*[1] *publications in the last decade. We identified 96 studies from the sources and included them in this review. The papers were categorized into four major types and data needed to address each research question was extracted. We found a need for refining the reasons and the classification scheme for SPSM introduced by KMR. More emerging SPSM paradigms and model scopes were added to enhance KMR's discussion. Trends over time showed that interest in continuous modeling was decreasing and interest in* micro-processes *was increasing. Hybrid models were based primarily on system dynamics and discrete event simulation and were all implemented by vertical integration. We recommend SPSM research concentrate more on recent software processes and on making SPSM more reusable and thus easier to build.*

1. Introduction

Software Process Simulation Modeling (SPSM) was introduced into the software engineering domain by Abdel-Hamid and others' pioneering work summarized in [1]. In the last two decades, it has been emerging as an effective tool to help evaluate and manage changes made to software projects and organizations. As a major research event, the ProSim series conference has taken place since 1998, and focuses on up-to-date theories and applications of SPSM research in addressing real-world problems.

At ProSim'98, Kellner, Madachy, and Raffo (KMR) discussed a variety of aspects of software process simulation in their widely-cited paper, "*Software process simulation modeling: Why? What? How?*" [4], such as the reasons for undertaking simulations of software process models, simulation approaches/paradigms, and modeling scopes. Recently, we undertook a systematic review to revisit and update the state-of-the-art of SPSM research, to summarize the 10-years (1998-2007) progress, to discover trends, and to suggest the possible directions of our future research activities in this domain. In [16], we reported the process and primary results of the systematic review, especially from the Category A studies (see Section 2.3). This paper instead emphasizes the concrete *facts*, the underlying *trends*, and the potential *directions* derived from the review results.

Section 2 briefly introduces systematic literature review and describes the method and studies in this review. Section 3, 4 and 5 respectively summarizes, discovers and recommends the '*facts*', '*trends*', and '*directions*' of SPSM research. We present our current conclusions in Section 6.

2. The Systematic Review

2.1. Systematic Literature Review

In 2004, Kitchenham *et al.* [8, 3] suggested software engineering researchers should adopt "Evidence-Based Software Engineering" (EBSE). EBSE aims to apply an evidence-based approach, which was initially developed in medicine and is being adopted in many domains, to software engineering research and practice. In this context, evidence is defined as a synthesis of best quality scientific

[1]Proceedings of International Workshop on Software Process Simulation Modeling (ProSim) and International Conference on Software Process (ICSP), and the special journal issues

1530-1362/08 $25.00 © 2008 IEEE
DOI 10.1109/APSEC.2008.50

IEEE computer society

studies on a specific topic or research question. The main synthesis method is Systematic Literature Review (SLR).

In contrast to an ad-hoc literature review, a systematic literature review (also known as systematic review) is a methodologically rigorous review of research results. It is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest [6]. A systematic review is a form of secondary study, the identified individual studies contributing to an SLR called primary studies. The state-of-the-art of SLR and EBSE will be reflected in [7].

An SLR involves several discrete activities, which can be grouped into three main phases: *1*) planning the review, *2*) conducting the review, and *3*) reporting the review. A pilot review is recommended for the reviews including multiple research questions or a large set of primary studies.

2.2. Questions and Method

This research follows Kitchenham's methodological guidelines for systematic reviews [5, 6], as adapted for PhD candidates. Three researchers were involved in this work, including one principal reviewer, one secondary reviewer, plus another researcher acting as the expert panel.

The systematic review is intended to answer the following predefined research questions:

Q1. What are the purposes or motivations for SPSM in the last decade's practice?

Q2. Which simulation paradigms have been applied in the last decade, and how popular are they in SPSM?

Q3. Which simulation tools are available for SPSM and have been in use in the last decade?

Q4. On model level, what are problem domains and model scopes focused on by process simulation models?

Q5. On parameter level, what are the output variables considered when developing process simulation models?

Q6. Which simulation paradigm is the most appropriate for a specific SPSM purpose and scope?

The time frame of sources for this study is constrained to the period from 01/01/1998 to 31/12/2007. Because the ProSim workshop series (which continued as a special track of ICSP since 2007) are regarded as the most important forum of SPSM, the sources related to ProSim (Table 1) are the primary data sources for this research at current stage.

Considering there are a number of research questions and a large set of potential primary studies (over 200 candidate papers), a pilot review was carried out after the planning phase "*to assess and refine review protocol, and further to secure the quality of the systematic review*" [6]. Further details regarding the inclusion and exclusion criteria can be found in [16].

2.3. Studies and Categorization

Based on the results of the pilot review, we refined SPSM study classification into four categories as follows:

A. Software process simulation models or simulators;

B. Process simulation modeling paradigms, methodologies, and environments;

C. Applications, guidelines, and solutions for adopting process simulation in software engineering practice;

D. Experience reports of SPSM research and practice.

Note that this categorization was not a mutually exclusive one, i.e. it is possible that a specific study falls into more than one category.

These four types of studies focus on different aspects of software process simulation research, and may give answers to the research questions and from different points of view. Figure 1 shows the relationships between these study categories and their roles in process modeling and simulation adoption. Category B studies introduce and provide effective paradigms, methods and tools for constructing process simulation models or simulators (Category A studies). These simulation models can be further adopted in software industry by following the practical solutions or guidelines (Category C studies) in given organization's context. The experience (Category D studies) collected from modeling and adoption can be used as feedback to continuously improve SPSM research.

In total, 209 papers have been published in the ProSim sources (Table 1). They form a comprehensive body of knowledge of software process simulation and modeling. Unfortunately, since the electronic proceedings were not available for ProSim'98 - '00, nine papers could not be evaluated in current review. Nevertheless, we believe that the low proportion (4.3%) of these studies published in the early years will not significantly influence the review results, especially for recent trend analysis. By carefully reviewing their titles, abstracts, keywords, conclusions and references, 96 articles were selected and identified as the primary studies. The number of papers per year and source are summarized in Table 2. The full list of the primary studies are available in [14].

We extracted 14 major attributes from these primary studies in terms of their categories. The quality of a primary study is assessed according to a question checklist, which specifies 20 questions to study categories. The detail of data extraction and quality assessment criteria can be found in [14, 16].

Table 1. Selected sources of the systematic review

Source	Acronym	Period	Search method
Proceedings of ProSim Workshop	ProSim	1998 - 2006	Manual
Proceedings of ICSP Conference	ICSP	2007	Manual
Journal of Systems & Software	JSS	1999 - 2001	Manual
Journal of Software Process: Improvement & Practice	SPIP	2000 - 2008	Manual

Table 2. Sources identified as primary studies

	1998	1999	2000	2002	2003	2004	2005	2006	2007	**Total**
Proc.	15	13	21	0	32	27	24	8	8	**148**
missing	2	1	6	0	0	0	0	0	0	**9**
JSS	11	0	12	0	0	0	0	0	0	**23**
SPIP	0	10	0	7	5	7	7	2	2	**38**
selected	13	9	14	7	16	10	13	6	8	**96**

3. Facts

A fact is "*a piece of information presented as having objective reality*" (Merriam-Webster dictionary). This section reflects some of the basic '*objective realities*' from several modeling aspects as a snapshot of SPSM in the last decade.

3.1. Purposes

In the first ProSim Workshop (1998), Kellner, Madachy, and Raffo presented a wide variety of reasons for undertaking simulations of software process models [4]. During our review process, however, it was difficult to handle the purpose identification clearly according to the definitions addressed by KMR. Three of their major shortcomings are: *1*) Ambiguity exists among their 6 purposes; *2*) Since the examples given in KMR's paper were mainly derived from the publications in ProSim'98 and the work prior to that, the scope of their purposes was limited; *3*) the research and arguments were not based on a systematic and rigorous review methodology, like an SLR.

In an attempt to more precisely reflect the purposes of SPSM research identified from the primary studies, we reframed them into three levels: **Cognitive level**, **Tactical level**, and **Strategic level**. They can be further detailed as 10 purposes. The ***cognitive*** level contains the purposes of *1*) understanding, *2*) communication, *3*) process investigation, *4*) training and learning. On the ***tactical*** and ***strategic*** levels purposes are similar. They are *5*) prediction and planning, *6*) control and operational management, *7*) risk management, *8*) process improvement, *9*) technology adoption, *10*) tradeoff analysis and optimizing. They differ in scope and impact between these two levels.

3.2. Domains

Problem domain identifies the problem(s) in software engineering that the simulation model investigates. Our systematic review extracted 19 problem domains from Category A studies (shown in the leftmost column of Table 3). Overall, *generic development*, modeling the normal development process of software project, has become the most common domain modeled.

3.3. Scopes

Model scope specifies the boundary of a simulation model in two dimensions: time span and organizational breadth. To more properly review and classify the model scopes of the published simulation models, we extended the 5 scopes defined by KMR to 7, i.e. *single-phase*, *multi-phase*, *project*, *multi-project (program)*, *product*, *product evolution*, and *long-term organization*. Among them, *project* is the most frequently modeled study scope.

Table 3 shows the relations between modeled domains and scopes extracted from Category A studies. Note that the number in the 'subtotal' row is not always the exact sum of the column as some studies modeled multiple domains.

3.4. Paradigms

The diversity and complexity of software processes and the richness of research questions (purposes) determine the different capabilities of simulation paradigms needed.

Overall, 10 simulation modeling paradigms were found in this review. System dynamics (SD, 49%) and Discrete-event simulation (DES, 31%) are the most widely used

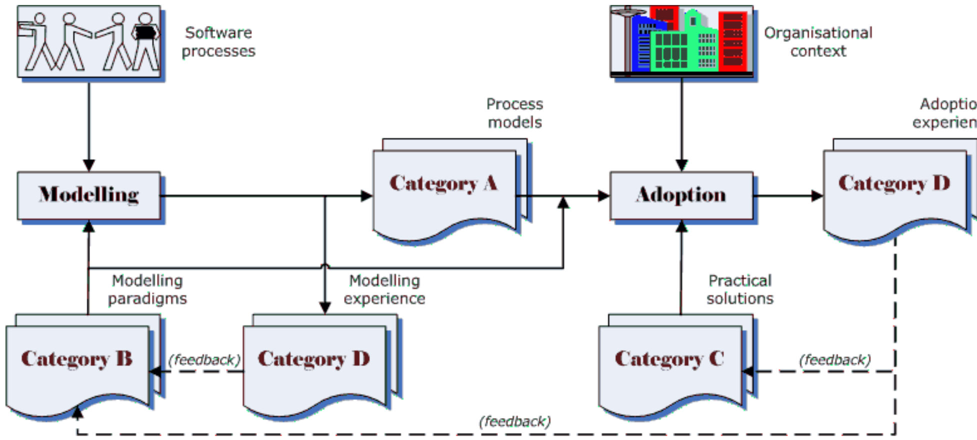

Figure 1. Relationships between study categories

techniques in SPSM. Other paradigms include state-based simulation (SBS), qualitative(semi-quantitative) simulation (QSIM), knowledge(rule)-based simulation (KBS), role-playing game (RPG), agent-based simulation (ABS), discrete-time simulation (DTS), stigmergy theory (Stig.), and emergent/unbound systems (Emrg.). However, only SD, DES, SBS and KBS were discussed by KMR.

3.5. Outputs

The systematic review identified 12 model variables as simulation outputs from the Category A studies. They are shown in Table 4, in which the third column indicates the number of studies including the leftmost output variable. It is evident that *time*, *effort*, *quality*, and *size* are most common drivers for simulation study of software process.

The more details of the *facts* or the statistical results of this review can be found in [14, 16].

4. Trends

Trend means "*the general movement over time of a statistically detectable change*" (Merriam-Webster dictionary). This section attempts to detect the important '*movement*' or '*change*' derived from the review results over the decade.

4.1. Paradigms

In 1998, there were only three simulation paradigms employed by the Category A studies (models) published in the first ProSim workshop (Table 5). They were system dynamics (SD), discrete-event simulation (DES), and knowledge-base simulation (KBS). KMR discussed four types of simulation in their semial paper [4]. However, our SLR identified ten simulation paradigms from the primary studies.

Table 5. Paradigms applied in simulation models over years

	1998	1999	2000	2002	2003	2004	2005	2006	2007	**total**
SD	4	5	7	1	4	3	5	2	1	**32**
DES	1	2	3	1	2	2	4	1	2	**18**
SBS				1	1				1	**3**
KBS	1		1	1						**3**
QSIM				1				1		**2**
RPG						1	1			**2**
ABS							1	2		**3**
DTS						1				**1**
Stig.							1			**1**
Hybrid	*1*	*1*	*2*		*1*	*1*	*3*	*1*		***10***

Trend 1 System dynamics and discrete-event simulation form the main stream of SPSM paradigms.

Table 5 shows the number of Category A studies (simulation models) per paradigm and per year. Note that some studies employed more than one paradigm in their simulation research. The bottom line indicates the number of hybrid models each year, which integrate two or more techniques above it. The majority of process models for simulation each year were built using SD or DES (the top two rows), including their combination.

Trend 2 New simulation paradigms continue to be introduced into SPSM research between 2000 & 2005.

Table 6 presents a more comprehensive view of all primary studies, including Category A, B, C, and D studies.

Table 3. Modeling problem domains vs. model scopes

Domain \ Scope	single phase	multi-phase	project	multi-project	product	evolution	long-term org.	unknown or n/a	**Total**
generic development			9					1	**10**
software evolution			1			7			**8**
software process improvement	1	1					1	3	**6**
requirement	2		1			1		1	**5**
incremental development	1	2	1					1	**5**
open-source development	1			1		2			**4**
global development			1			3			**4**
software economics	1		1		1				**3**
software product-line					1		1		**2**
agile process			1					1	**2**
quality assurance		1						1	**2**
acquisition/outsourcing		1						1	**2**
software engineering education			2						**2**
software test	1								**1**
software design	1								**1**
software services					1				**1**
productivity analysis			1						**1**
risk management			1						**1**
software reliability								1	**1**
subtotals	7	4	19	1	2	8	2	9	

Unlike Table 5, the study numbers are represented with the solid dots, which make it clear which paradigms were applied each year. The four rows at the top of Table 6 are the paradigms discussed in KMR's paper in 1998, which are separated from others by a solid line. The emerging paradigms were introduced to SPSM research since 2000, and their application became dominant at 2004/2005.

4.2. Granularities

As two complementary types of software process, *macro-process* and *micro-process* research (identified by Osterweil [9]), are applicable to SPSM research as well. However, the systematic review reflects that there exist three different granularity levels that process simulation models can focus on: namely ***system***, ***process***, and ***entity*** levels, supported by different simulation paradigms.

At ***system level***, a software process is modeled as an overall system. The behavior of the process modeled is described by a set of external parameters that continuously vary over time. This granularity corresponds to the *macro-process* research that is supported by the conventional *continuous* simulation.

Software processes are modeled and observed with more details at ***process level***, where the executed tasks with the resources required and their sequential relationships (unlike causal relationships alone on *system* level) can be tracked during simulation. The modeling on this level is supported by *discrete* simulation.

Trend 3 Continuous modeling gradually lost its dominant position in SPSM research in comparison with discrete approaches during the decade.

When revisiting the top two rows of Table 5, both SD and DES supported simulation studies of software process each year. Although the total number of SD models is larger than any others', it is not difficult to observe a decreasing trend of SD application in the time frame, by absolute number or percentage. The studies using SD dominated in the early years (prior to 2000). Recently, the number of published studies using SD has decreased and approached the number of studies using DES. It implies that simulation research has become more interested in *micro-process* modeling.

The simulation models focusing on *process* level are based on the conventional life-cycle or sequential process

Table 4. Summary of simulation outputs

Output	Description	Studies	Percentage
Time	Project schedule or elapsed time	20	35.7%
Effort	Effort or cost	16	28.6%
Quality	Product quality or defect level	11	19.6%
Size	Requirement size or functionality	11	19.6%
Resource	Resource or staffing level	7	12.5%
Plan	Project or development plan (e.g. task allocation)	3	5.4%
ROI	Return on investment or cost/benefit analysis	2	3.6%
Productivity	Development productivity	1	1.8%
Market share	Product market share	1	1.8%
Index	Nominal index	1	1.8%
Behaviour	Behaviour patterns	1	1.8%
Flow	Process flow	1	1.8%

Table 6. Paradigms applied in primary studies over years

	1998	1999	2000	2002	2003	2004	2005	2006	2007
SD	●	●	●	●	●	●	●	●	●
DES	●	●	●	●	●	●	●	●	●
SBS	●		●	●	●				●
KBS	●		●	●					
QSIM				●		●		●	●
RPG					●	●	●		
ABS							●	●	
DTS						●			
Stig.							●		
Emrg.			●						
Number	**4**	**2**	**5**	**5**	**4**	**5**	**5**	**4**	**4**

modeling. However, software development, like other sectors of society, requires synergistic collaboration among many diverse contributors, such as the careful coordination among designers, programmers, testers, managers, and so on [10]. For instance, agile process models simply offer a framework and set of practices, rather than describe what to do next. The real process executed often depends on the practitioners' *on-the-spot* adjustments [11]. Moreover, as a human activity, software development's performance is influenced to a large extent by team coordination, which means "*the articulation of the individual actions accomplished by each of the agents in such as way that the whole ends up being a coherent and high-performance operation*" [13]. It is difficult to simulate the team effects by *process* level modeling through aggregating individuals. This becomes an issue when modeling a process with many participants, such as an open source developer community.

Accordingly, the modeling and simulation at ***entity level*** require quite different characteristics from the technology used on *process* level. Simulation at *entity* level consumes more resources to track all entities and their relationships individually than the approaches using aggregation alone. Thanks to the advance in computer and electronic engineering, this approach has now become possible.

Table 7 indicates for each simulation paradigm the inherently supported research granularity level(s) as identified from the review. Here, RPG is a special case, which usually need to be combined with other approach(es) to construct a simulator. Thus, its supporting granularity level often depends on its companion.

Table 7. Paradigms in support of granularity

	SD	DES	SBS	KBS	QSIM	RPG	ABS	DTS	Stig.	Emrg.
System	●				●	❍				
Process		●	●	●		❍		●		
Entity						❍	●		●	●

● - inherently supported ❍ - condition applying

Trend 4 Most of newly introduced paradigms enhanced the research capability at the *micro-process* level.

All newly introduced paradigms are listed at the right side of the vertical line (between KBS and QSIM) in Table 7 (compared with the four paradigms discussed by KMR at leftmost). If we define the scope of *micro-process* to cover the *entity* granularity level, then most of the new paradigms

Table 8. Granularity levels of studies

	1998	1999	2000	2002	2003	2004	2005	2006	2007	**total**
System	4	5	7	2	4	3	5	3	1	**34**
Process	2	2	4	3	3	3	4	1	3	**25**
Entity							2	2		**4**

are capable of the *process* or *entity* level research, which correspond to the *micro-process*.

Trend 5 In recent years, *micro-processes* have been attracting more simulation research.

In terms of the mapping in Table 7, Table 8 records the number of published studies on each granularity level over years. Some hybrid simulation models appear at more than one level. When applying the extended concept of *micro-process*, since 2004 the number of *micro-process* simulation models has been no less than *macro-process* models. It is also an enhanced statement of Trend 3.

4.3. Integration

Like the interaction between *macro-process* and *micro-process* [9], process modeling on the three granularity levels is not mutually exclusive, and sometimes combined.

Hybrid modeling employs more than one simulation paradigms in developing a process simulation model. The systematic review concludes that hybrid process simulation models have attracted interest as a possibility to avoid the limitations of applying single modeling method, and more realistically capture complex real world software processes.

Trend 6 System dynamics and discrete-event simulation are the most common combination for constructing hybrid simulation models.

The bottom line of Table 5 shows the number of hybrid models published per year. All these 10 models were at least based on the combination of SD and DES, or even more.

Generally speaking, there exist two broad approaches to develop a hybrid process model: ***vertical integration***, which primarily implements discrete modeling at the lower *process* level, then continuously calculates the process factors and incorporates the feedback loops at *system* level; and ***horizontal integration***, in which the sub-processes or phases within a large scale and/or complex software process may be modeled using different approaches respectively and sequentially, and the data flow has to be converted at the interface between them for process phase transition.

Trend 7 Hybrid simulation modeling concentrated on *vertical integration*.

All the studies published in ProSim series conferences have tried to construct the hybrid models using *vertical integration* of continuous (SD) and discrete (DES) approaches, such as the recent work published in [2] and [12]. On the other side, there are few attempts on *horizontal integration* until the recent work reported in [15].

5. Directions

Again, according to Merriam-Webster dictionary, there are two meanings of '*direction*' relevant to what is applied in this section: "*the line or course on which something is moving or is aimed to move or along which something is pointing or facing*", as well as "*assistance in pointing out the proper route*". In this section, we cannot point out the *exact 'line or course'* that SPSM is moving towards in the future. Instead, here we try to point out what SPSM faces, and present our consideration threads of the '*proper routes*' raised from the *facts* and *trends* derived from the review.

Direction 1 More recent modern software development processes need to be further investigated in SPSM research.

Some examples in this direction include agile processes, open-source development, and global development. These modern software processes are quite distinct from the traditional software processes discussed by KMR.

Direction 2 More new simulation paradigms need to be experimented and introduced into SPSM community.

The recent changes of software processes require the simulation modeling capable of coping with higher complexity, scalability, uncertainty, and agility in the process, especially for modeling and simulating lower granularity level processes, i.e. *process* and *entity* levels.

Direction 3 More attempts are needed to effectively tackle the uncertainty of software process in practice.

The lack of complete knowledge or data of software processes is one of the most common problems obstructing the effective use and adoption of SPSM in practice. However, most purely quantitative simulation modeling approaches observed in the review count on the classical probability based methods only and require very strict assumptions.

Direction 4 Hybrid simulation models should address more than SD and DES in vertical integration.

The interest in micro-process and the increasing use of different simulation paradigms for micro-process suggest the need for hybrid models to integrate *macro-* and *micro-processes* including entity level models, to cater for new paradigms capable of modeling modern software processes. The integration is not limited by the combination of SD and DES, continuous and discrete, or *system* and *entity* levels.

Direction 5 Process simulation models should become more reusable, which makes them easier to build.

Simulation modeling is often time-consuming and requires expertise and experience. However, we found from the review that model reuse was omitted by many SPSM studies. Most Category A studies developed one-off models for simulating the specific processes.

6. Conclusions

We conducted the first systematic literature review of simulation modeling research in software process domain from 1998 to 2007. The process and preliminary results of the review has been published in [16]. As the continuous and enhanced work of the review, this paper presents the state-of-the-art and in-depth findings of SPSM research by not only providing the *facts* in *snap-shot* style, but also discovering the *trends* over the past decade and proposing the possible future *directions*. Especially, we gain the new insights about SPSM research on study categorization, modeling granularity and integration.

The results from this systematic review can help both insiders' and outsiders' observation and understanding of SPSM research. The study categorization can be used for future systematic review and meta-analysis with more specific focuses. Moreover, for practitioners and modelers in SPSM arena, it detects some underlying *trends* over the last 10 years, and provides them some suggestions as the potential future *directions* for realistically capturing complicated real-world software processes.

Some limitations still exist in the current study and need further improvement: *1*) the study categorization was determined by the principle reviewer's subjective judgment, which needs further examination; *2*) the missing papers may influence the integrity of this review and need to be included in the near future; *3*) the impact of study quality needs to be considered in data analysis, especially for the inclusion of low quality studies.

As part of a comprehensive systematic review, the statements drawn in this paper are based on the review results at current stage. This work will be continuously enhanced by including a more detailed analysis of the studies of Categories B, C and D, and extending our systematic review to the studies outside of the ProSim sources in the future.

7. Acknowledgements

He Zhang's research reported in this paper is also supported by National ICT Australia, which is funded by the Australian Government. Barbara Kitchenham's research is funded by the UK EPSRC project EP/E046983/1.

References

[1] T. K. Abdel-Hamid and S. E. Madnick. *Software Project Dynamics: An Integrated Approach*. Prentice Hall, Englewood Cliffs, NJ, 1991.

[2] K. Choi, D.-H. Bae, and T. Kim. An approach to a hybrid software process simulation using the devs formalism. *Software Process: Improvement and Practice*, 11(4):373–383, 2006.

[3] T. Dyba, B. Kitchenham, and M. Jorgensen. Evidence-based software engineering for practitioners. *IEEE Software*, 22(1):158–165, 2005.

[4] M. I. Kellner, R. J. Madachy, and D. M. Raffo. Software process simulation modelling: Why? what? how? *Journal of Systems and Software*, 46(2/3):91–105, 1999.

[5] B. Kitchenham. Procedures for undertaking systematic reviews. Technical report, Computer Science Department, Keele University and National ICT Australia, 2004.

[6] B. Kitchenham. Guidelines for performing systematic literature reviews in software engineering. Technical report, Keele University, 2007.

[7] B. Kitchenham, P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman. A systematic literature review of evidence-based software engineering. *submitted to Information and Software Technology*, manuscript under review.

[8] B. Kitchenham, T. Dyba, and M. Jorgensen. Evidence-based software engineering. In *26th International Conference on Software Engineering, (ICSE)*. IEEE Computer Society, 2004.

[9] L. J. Osterweil. Unifying microprocess and macroprocess research. In *Software Process Workshop*, Beijing, 2005. Springer.

[10] L. J. Osterweil. What we learn from the study of ubiquitous processes. *Software Process: Improvement and Practice*, 12(5):399–414, 2007.

[11] K. Schwaber and M. Beedle. *Agile Software Development with SCRUM*. Prentice Hall, 2001.

[12] S.-o. Setamanit, W. Wakeland, and D. Raffo. Using simulation to evaluate global software development task allocation strategies. *Software Process: Improvement and Practice*, 12(5):491–503, 2007.

[13] M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, 2002.

[14] H. Zhang. *Qualitative and Semi-quantitative Modelling and Simulation of Software Engineering Processes*. PhD thesis, University of New South Wales, 2008.

[15] H. Zhang, R. Jeffery, and L. Zhu. Hybrid modeling of test-and-fix processes in incremental development. In *International Conference on Software Process*, Leipzig, Germany, 2008. Springer.

[16] H. Zhang, B. Kitchenham, and D. Pfahl. Reflections on 10 years of software process simulation modelling: A systematic review. In *International Conference on Software Process*, Leipzig, Germany, 2008. Springer.