A Systematic Literature Review of Requirements Modeling and Analysis for Self-adaptive Systems

Zhuoqun Yang[1], Zhi Li[2,3], Zhi Jin[3], and Yunchuan Chen[1]

[1] Institute of Mathematics, Academy of Maths and Syst. Sci., Chinese Academy of Sciences, Haidian Dstr, Beijing 100190, P.R. China
zhuoqun.y@gmail.com, yunchuan001@163.com
[2] Software Engineering Dept., College of Computer Science and Information Technology Guangxi Normal University, Guilin, Guangxi 541004, P.R. China
zhili@gxnu.edu.cn
[3] Key Laboratory of High Confidence Software Technologies (MoE), Peking University Ministry of Education, Beijing 100871, P.R. China
zhijin@sei.pku.edu.cn

Abstract. [Context and motivation] Over the last decade, researchers and engineers have developed a vast body of methodologies and technologies in requirements engineering for self-adaptive systems. Although existing studies have explored various aspects of this topic, few of them have categorized and evaluated these areas of research in requirements modeling and analysis. **[Question/Problem]** This review aims to investigate what modeling methods, RE activities, requirements quality attributes, application domains and research topics have been studied and how well these studies have been conveyed. **[Principal ideas/results]** We conduct a systematic literature review to answer the research questions by searching relevant studies, appraising the quality of these studies and extracting available data. The results are derived by synthesizing the extracted data with statistical methods. **[Contributions]** This paper provides an updated review of the research literature, enabling researchers and practitioners to better understand the research trends in these areas and identify research gaps which need to be further studied.

Keywords: systematic literature review, self-adaptive systems, modeling method, RE activity, requirements quality attribute

1 Introduction

Self-adaptive systems (SASs) are able to adjust their behaviors in response to the dynamic changes in the environment and themselves. Due to the inherent volatility of the deployed environment and frequent interactions between software systems and the environment, SASs are faced with the challenges of meeting demands on some quality attributes, such as fault-tolerance, replaceability, etc. To maintain these attributes, we need to build adaptation mechanisms in SASs for endowing them with the capability of self-reconfiguring, self-healing, self-protecting and self-optimizing, which are

C. Salinesi and I. van de Weerd (Eds.): REFSQ 2014, LNCS 8396, pp. 55–71, 2014.

known as self-* properties [1]. Therefore, when developing SASs, engineers should take both domain logic and adaptation logic into account.

Requirements engineering (RE) is known as the first stage in the lifecycle of software development, aiming at defining domain logic, identifying stakeholders' needs and documenting information for subsequent analysis and implementation [2]. Different from traditional RE, RE for SASs focuses more on defining adaptation logic, since SASs need adaptation mechanisms. Thus, during RE for SASs, engineers must address what changes in the environment and the system themselves to be monitored, what to adapt, when to adapt and how to adapt. Requirements modeling is a fundamental activity in RE. Various kinds of artifacts produced during the modeling process are involved in the latter analysis, such as specifying requirements, diagnosing requirements, verifying requirements, etc.

Over the last decade, researchers and engineers have developed a vast body of work on requirements modeling and analysis for SASs. Existing studies [3-7] have summarized some of the achievements, provided insight in this field and outlined challenges in each direction. However, to the best of our knowledge, no systematic study has been performed on categorizing and evaluating these emerged modeling methods and corresponding RE activities. Thus, there is no clear view on where the researches are conducted and where the results are published, to what extend each kind of modeling method or RE activity is studied, how the method is evaluated, how the quality of studies varies against each method and what the most active topics are.

The objective of this paper is to systematically investigate the research literature of requirements modeling and analysis for self-adaptive systems, summarize the state-of-the-art research trends, categorize the used modeling methods and relevant RE activities, classify the quality attributes and application domains, assess the quality of current studies and generate the most active research topics. To conduct the investigation and report analysis results, we adopt the research methodology of systematic literature review [8] [9] in the evidence-based software engineering paradigm [10].

The rest of the paper is structured as follows. Section 2 briefly describes the systematic review and the protocol underpinning this study, followed by the presentation of the analysis results in Section 3. Section 4 discusses the results and threats to validity, followed by the conclusions and discussions on future work in Section 5.

2 Research Method

Evidence-based software engineering (EBSE) aims to improve decision making related to software development and maintenance by integrating current best evidence from research with practical experience and human values [11]. The core tool of the evidence-based paradigm is the Systematic Literature Review (SLR), which is a systematic methodology of defining answerable research questions, searching the literature for the best available evidence, appraising the quality of the evidence, collecting and aggregating available data for answering the identified questions. The whole process of SLR is presented in Figure 1. To complete SLR, three phases are needed: planning, conducting and reporting. During the planning phase, a protocol is produced for defining basic review procedures, on which the conducting phase should depend.

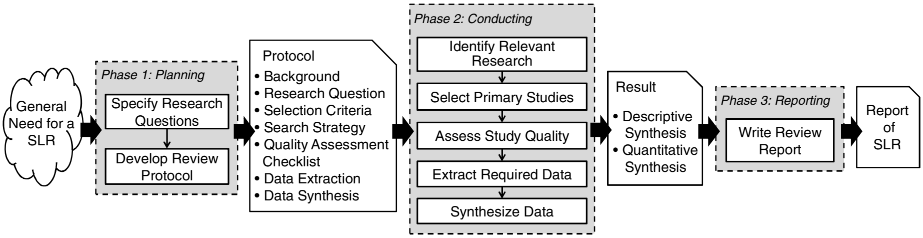

Fig. 1. Process of Systematic Literature Review

Due to the limitation of space, a detailed account of our SLR protocol is beyond the scope of this paper, but can be found in [12], which is available online.

2.1 Research Questions

The high-level goal of this literature research is to review the existing research work in the literature of requirements modeling and analysis for self-adaptive systems. To achieve this goal, we refine it into seven answerable questions in Table 1. These questions can be categorized into four types: a) publication type is related to the questions that are related to publication information, such as published time, venues and authors; b) content type contains the questions that should be answered by extracting the corresponding data from the texts of papers; c) quality type consists of the questions which are answered by assessing the quality of papers; d) topic type includes the questions which are related to the topics of relevant studies.

Table 1. Research Questions and Corresponding Types

Research question	Type
RQ1: What is the time/venue/research group/region distribution of the publications?	Publication
RQ2: What modeling methods and RE activities are studied? RQ3: What requirements quality attributes and application domains are involved?	Content
RQ4: Which methods are better applied and have more rigorous evaluation? RQ5: Which RE activities are presented and discussed more detailedly?	Quality
RQ6: What topics can we generalize based on the content of selected studies? RQ7: What is the relationship between topics and modeling methods?	Topic

2.2 Search Process

Figure 2 presents the mechanism underpinning the search process. The objective of the search process is to identify relevant studies based on search strategies. Defining search strategies includes defining search sources and defining search strings.

Search sources consist of some search engines, e.g. IEEE Xplore, and publication venues. Search engines are chosen for conducting automated search, which means researchers use these online databases to thoroughly retrieve relevant studies with some search strings adapted to the given search syntax and rules. Publication venues are chosen for conducting manual search, in which researchers manually scan conference proceedings or journals for relevant studies.

To improve the reliability and the repeatability of our study, we adopt the quasi-gold standard (QGS) [13] method, which is a set of known studies established by manual search within certain venues and time span, to objectively define search strings and evaluate the performance of search strings. The retrieved results from automated search complement manual search by expanding the coverage of the relevant studies. Moreover, we conduct the "snowball" search, which means investigators scan the references in each paper derived by manual search and automated search and pick out the most relevant ones. Therefore, the final set of relevant studies consists of search results from manual search, automated search and "snowball" search.

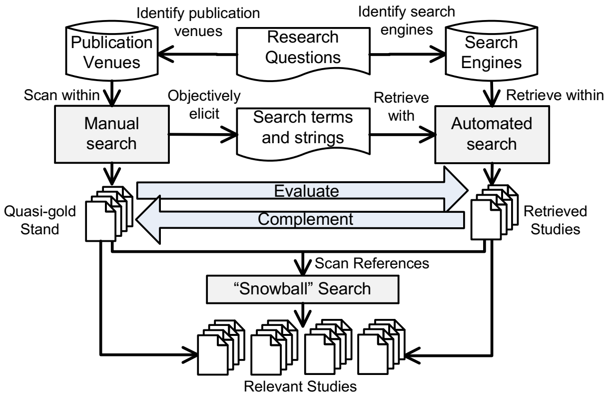

Fig. 2. Mechanism Underpinning the Search Process

Defining Selection Criteria. Inclusion criteria and exclusion criteria are defined for selecting relevant studies. Retrieved papers are firstly checked with exclusion criteria. If one paper meets any one of the exclusion criteria, i.e. C5 **OR** C6 **OR** C7 **OR** C8, it will be excluded. The remaining papers are checked with inclusion criteria. If one paper meets all the inclusion criteria, i.e. C1 **AND** C2 **AND** C3 **AND** C4, then it will be included.

Table 2. Inclusion and Exclusion Criteria

Inclusion criteria	Exclusion criteria
C1: Published time between 2003.1-2013.9	C5: In the form of books
C2: Focus on requirement modeling and analysis for self-adaptive systems.	C6: In the form of editorial, abstract, keynote, poster or a short paper (less than 6 pages)
C3: Related to concrete RE activity	C7: Opinion pieces or Position papers
C4: Involve concrete modeling methods and evaluation to the methods	C8: Focus on summarizes the existing research work, e.g. roadmap or survey

Defining Selection Procedure. We use the above criteria for establishing QGS from the manual search and deriving relevant studies from the automated search and the "snowball" search. The selection procedure consists of three rounds:

- Round 1: We first scan each paper by title, aiming to eliminate any irrelevant papers. Any paper that any researcher thinks should be included or is unsure about should be kept in the set of candidate papers for Round 2.
- Round 2: Scan the abstracts of candidate papers from Round 1 and appraise each paper with selection criteria. Any paper that any researcher considers should be included or is unsure about should be kept in the set of candidate papers for Round 3.
- Round 3: Look through the full texts of the candidate papers from Round 2 and assess each paper with the selection criteria. Any paper on which researchers cannot reach agreement should be resolved by a joint meeting.

During the selection procedure, we also consider duplicate papers and repeat studies. A duplicated paper refers to the same paper that can be retrieved from more than one search engine. In this situation, we retain only one of the duplicates in the final set of relevant studies and remove all the duplication. A repeated study means the same study published in more than one venue with the same authors' order or different authors' order. In this situation, we remove the repeated studies and retain the most comprehensive or the most recent version, except for answering RQ1.

Defining Search Sources. Search engines function as the databases for the automated search and the digital library where publication venues are provided. To ensure thorough retrieval, we choose six search engines that cover the RE literature: ACM Digital Library, IEEE Xplore, Science Direct, Springer, EI Compendex and Web of Knowledge.

Publication venues consist of a collection of proceedings and journals where the community tend to publish their research results. To ensure the quality of this study, we choose the qualified conferences and journals (Table 3) according to the Australian ERA (Excellence in Research for Australia) Outlet Ranking [14].

Establishing QGS. The manual search is conducted by two researchers individually and should be terminated when the Kappa value depicts a good or very good agreement. We scan all papers in the chosen venues by title, abstract and full texts with the selection criteria. The Kappa value is above 0.8, which indicates good agreement [15] and disagreement is eliminated by discussion with other investigators. Finally, the QGS is established by aggregating the selected results of two researchers. Table 3 provides the frequency and percentage of the 61 papers that compose QGS.

Defining Search Strings. Search terms are derived by using text mining. A frequency analysis of information of papers in QGS is undertaken followed by a statistical analysis of most frequently occurring words or phrases by using QDA Miner and WordStat [16]. We import the title-abstract-keyword segment of each paper in to QDA Miner and derive search strings (Table 4). The use of the search strings can be combined with Boolean operator as: S1 **AND** (S2 **OR** S3 **OR** S4 **OR** S5 **OR** S6 **OR** S7 **OR** S8 **OR** S9 **OR** S10).

Table 3. Publication Venues and Paper Frequency

Conference	Frequency	%	ERA	Journal	Frequency	%	ERA
SEAMS	12	27%	N/A	REJ	4	25%	B
RE	6	14%	A	JSS	3	19%	A
RE@runtime	5	10%	N/A	SESAS	2	19%	N/A
REFSQ	5	10%	B	ASEJ	1	6%	A
ICSE	4	8%	A	IST	1	6%	B
MODELS	4	8%	B	SoSyM	1	6%	B
ASE	3	6%	A	TAAS	1	6%	B
ICAC	2	6%	B	ToSEM	1	6%	A
CAiSE	2	4%	B	TSE	0	0%	A
FSE	2	4%	A	ESE	0	0%	A
SASO	1	2%	N/A	—	—	—	
Total	46	100%		Total	15	100%	

Table 4. Derived Search Strings

Item	Search string
S1	(“self-adaptive systems” **OR** “dynamically adaptive systems” **OR** “adaptive system” **OR** “Adaptive software” **OR** “self-adaptive software” **OR** “adaptive service” **OR** “web systems” **OR** “socio-technical system” **OR** “self-adjusting systems” **OR** “autonomic computing” **OR** “self-adapting software”)
S2	“model requirements” **OR** “modeling requirements” **OR** “Requirements modeling”
S3	“specify requirements” **OR** ”specifying requirements” **OR** “requirements specifying” **OR** “requirements specification”
S4	“monitor requirements” **OR** “monitoring requirements” **OR** “requirements monitoring”
S5	“aware requirements” **OR** “requirements-aware” **OR** “requirements awareness” **OR** “requirements-awareness”
S6	“diagnose requirements” **OR** “diagnosing requirements” **OR** “requirements diagnosing” **OR** “requirements diagnosis”
S7	“detect requirements” **OR** “detecting requirements” **OR** “requirements detection”
S8	“verify requirements” **OR** “verifying requirements” OR “requirements verifying” **OR** “requirements verification”
S9	“requirements” **AND** (“self-adaptation” **OR** “self-reconfiguration” **OR** “self-repair” **OR** “self-healing” **OR** “self-tuning” **OR** adaptation **OR** configuration **OR** reconfiguration **OR** “decision making” **OR** “decision-making” **OR** “adaptation behavior” **OR** “behavior”)
S10	“evolution requirements” **OR** “requirements evolution”

Automated Search and Evaluating Search Strings. We conduct automated search within each search engine by splitting and inputting the strings according to the search syntax demanded. After eliminating disagreement, we finally record 79 papers and 47 of them can be found in QGS (Figure 3).

Quasi-sensitivity is an important criterion for evaluating the quality and efficiency of search strategies [13]. It refers to the proportion of relevant studies covered by the QGS. Thus, the value of our quasi-sensitivity is 77.04% (47/61), which is between 72%~80%. It means that the search strategies are acceptable according to [13].

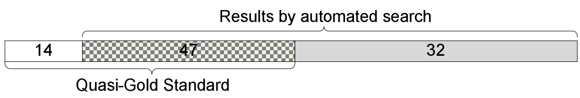

Fig. 3. Relationship between QGS and Automated Search Results

2.3 Quality Assessment Checklist

To answer RQ4 and RQ5, a quality assessment checklist (Table 5) is defined based on the assessment items introduced in [9] and [17]. We use the checklist to evaluate whether a method or an activity is maturely or rigorously conveyed in the literature.

Table 4. Quality Assessment Checklist

Assessment question	Optional answer and score
A1: How clearly is the problem of study described?	Explicitly=1/Vaguely=0.5/No description=0
A2: How clearly is the research context stated?	With references =1/Generally=0.67/ Vaguely=0.33/No statement=0
A3: How detailedly is the modeling method conveyed?	Step by step=1/Relatively detail=0.67/ Generally=0.33/Vaguely conveyed=0
A4: How detailedly is the RE activity elaborated?	Explicitly=1/General steps=0.67/ Vaguely=0.33/Disorderly=0
A5: How rigorously is the method evaluated?	Simulation=1/Detailed case study=0.67/ General case study=0.33/No evaluation=0
A6: How explicitly are the contributions presented?	Explicitly=1/Generally=0.5/No presentation=0
A7: How explicitly are the limitations discussed?	Explicitly=1/Generally=0.5/No discussion=0
A8: How explicitly are the insights and issues for future work stated?	With recommendations=1/Generally=0.5/ No statement=0

2.4 Data Extraction

To answer RQ1, corresponding information can be extracted directly from the papers. To answer RQ2, we extend the modeling method category presented in [2], investigate RE activities at requirements time, design time and runtime. To answer RQ3, we classify requirements quality attributes based on ISO 9126 Software Quality Characteristics [18]. Application domains can be elicited from the motivating example of each paper. To answer RQ4 and RQ5, we read full texts and appraise each paper according to the quality assessment checklist. To answer RQ6 and RQ7, we extract text segments, coding texts, and translate codes into topics or themes [19, 20].

More details on how the data is extracted and synthesized can be found in [21] and the theory underpinning the extraction process is elaborated in our protocol [12].

3 Results and Discussion

***RQ1**: What is the time/venue/research group/region distribution of the publications?*

After the search process, we select a total of 101 relevant papers, in which 11 of them are identified as repeated studies. The time distribution of the studies is provided in Figure 4. Publication venue distribution can be found in Table 3.

To derive the research group and region distribution, we investigate the authors' affiliations. The results depict that the selected papers are from 29 research groups in 13 regions and the researchers are from 43 groups in 17 regions. Most of these papers are from European countries (58/101), followed by American countries (25/101) and Asian countries (18/101). Figure 5 and Figure 6 present the top 10 research groups and regions with the frequency of published papers and corresponding researchers.

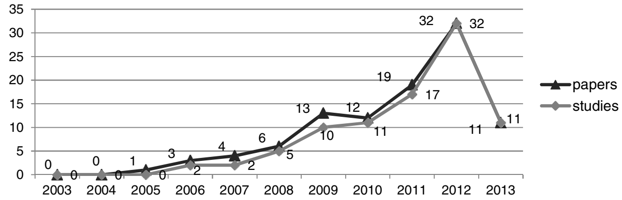

Fig. 4. Time Distribution of Selected Papers and Studies

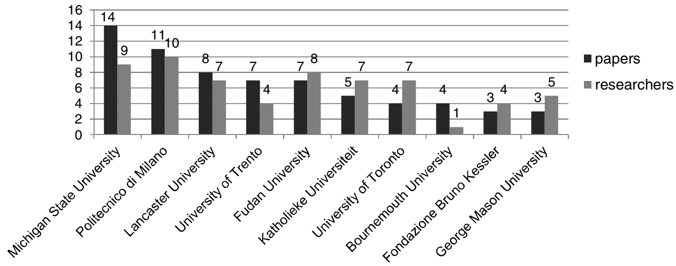

Fig. 5. Top 10 Research Groups and Researchers

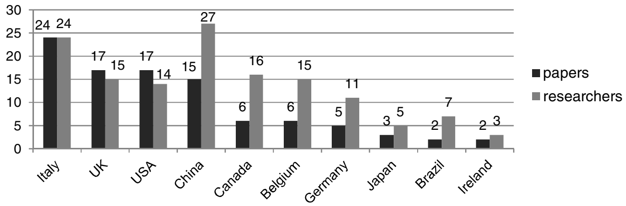

Fig. 6. Top 10 Region and Researchers

RQ2*: What modeling methods and RE activities are studied?*

Figure 7 presents the modeling methods and the corresponding frequency of studies. These modeling methods are categorized and synthesized according to the objective of modeling activities, including requirements, context and system. Goal-oriented methodologies, including KAOS [25], i* [26] and Tropos [27], are the most popular requirements modeling methods in the literature. They can clearly describe stakeholders' intension and systems' requirements. Temporal logic, including LTL [28], CTL [29] and FBTL [30] are always used as specification languages. They are utilized to specify the properties that should be held by the system. When modeling context, context models [31] are always built to capture the environmental properties. Z notation [32] is used to specify systems' behavior. Transition systems including Markov Chain [33], Petri Net [34] and DDN [35] are adopted to describe systems' states and state transitions. In addition, UML models [36] are also used to model systems' behavior. Problem frame, feature model and feedback control mechanism are more close to design level. Business process model and domain-specific model focus more on business logic and domain logic, respectively.

Figure 8 presents the categories of RE activities and the corresponding frequency of studies. RE activities are classified into activities at requirements time, activities at design time and activities at runtime. Activities at requirements time focus on modeling and specifying requirements [30], modeling adaptation mechanism [37] and verification [38]. Activities at design time mainly aim to map requirements model to architecture model [39] or derive design decisions based on requirements [40]. Activities at runtime include achieving adaptation through MAPE loop [41], runtime verification [33], runtime reconfiguration [42] and runtime evolution [43].

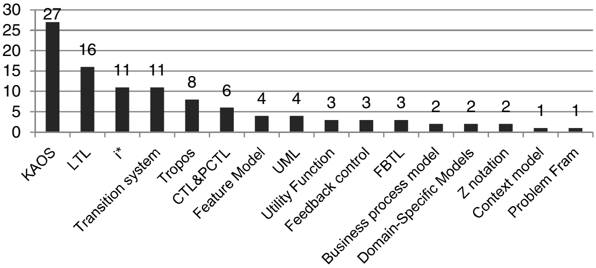

Fig. 7. Modeling Methods and Corresponding Frequency of Studies

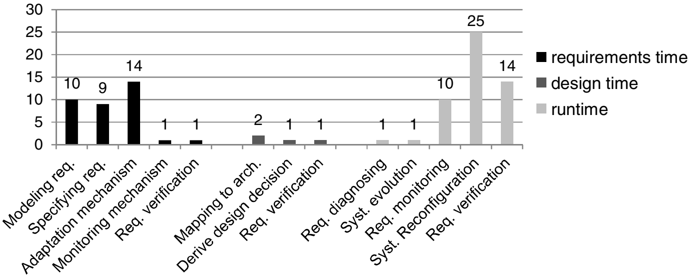

Fig. 8. RE activities and Corresponding Frequency of Studies

Challenges in Modeling Methods and RE Activities. Figure 7 depict that context modeling in RE for SASs still lacks study. The modeling method proposed in [31] may inspire us to work out other innovative context models. Promising research topics related to context may include: model and specify context uncertainty, reasoning with context uncertainty and requirements-driven adaptation with context uncertainty. Figure 8 depicts that there are research gaps in mapping requirements to architectures. Promising research topics may include: requirements-driven architecture adaptation and requirements-driven evolution.

***RQ3**: What requirements quality attributes and application domains are involved?*

We investigate the requirements quality attributes (Figure 9) related to SASs according to ISO 9126. We do not intend to elaborate the definitions of these quality attributes, but reveal the relations implied behind. According to [18], adaptability and replaceability belong to portability. These two quality attributes are involved in the studies of building adaptation mechanism or runtime adaptation. Analyzability is considered in the studies of monitoring or diagnosing requirements. Time behavior and resource behavior are always concerned in the evaluation of the adaptation process. Reliability is studied in the work on the topic of verification. Fault tolerance is always derived by relaxing the requirements. Security is discussed in security requirements engineering. Understandability is involved in the study of producing more understandable requirements model.

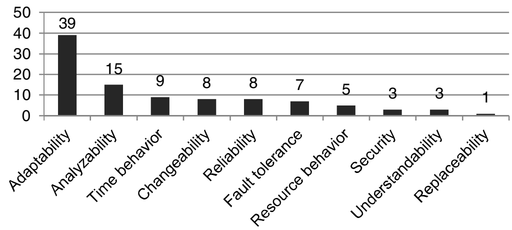

Fig. 9. Requirements Quality Attributes and Frequency of Studies

The application domains are presented in Figure 10. The top 5 most widely cited application domains depict that the community concentrate on investigating online applications, web services, mobile computing systems, social-technical systems and smart living systems. We find the common characteristic of this application domain is that they all need to interact with other software, systems or the human. These results will benefit researchers and practitioners to choose the most appropriate demonstrations and design the most reasonable experiments for their research work.

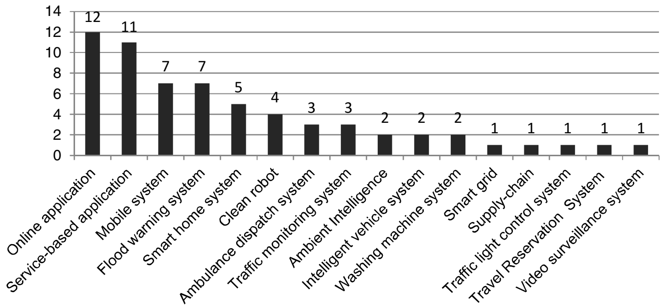

Fig. 10. Application Domains and Frequency of Studies

Challenges in Requirements Quality Attributes and Application Domains. Figure 9 shows the gaps in research on security requirements engineering, adaptation mechanisms that can provide explanations to the human and compositional adaptation. Researchers can also consider other quality attributes, e.g. recoverability. Besides, the application domain should be chosen to underpin the quality attributes.

RQ4*: Which methods are better applied and more rigorously evaluated?*

Relevant studies are appraised according to the quality assessment checklist (Table 5). Figure 11 depicts that KAOS and i* both have relatively low score, because some of modeling methods and adaptation mechanisms proposed based on KAOS or i* lack rigorous evaluation. The highest scored logic is CTL&PCTL, for they are widely used

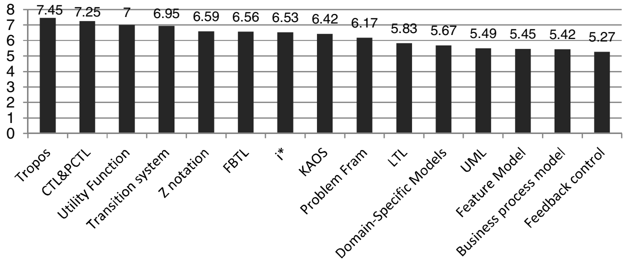

Fig. 11. Quality Score of Modeling Methods

to model system properties for model checking, which is inherently a rigorous approach. Domain-specific models, UML models, feature model, business process model and feedback control need more exploration and more rigorous evaluation.

RQ5*: Which RE activities are presented and discussed more detailedly?*

Figure 8 depicts the first three activities at requirements time and the last three activities at runtime are almost explored in more than 10 studies. Therefore, the scores of these activities are more convincing than others'. In these six activities, requirements verification at runtime has the highest score, for the verification process is always based on rigorous reasoning or mathematical methods. System reconfiguration at runtime comes after runtime verification, because this process is related to decision making and the evaluation process is always elaborately designed. The next is runtime monitoring, which also includes rigorous analysis processes. The three activities at requirements time are lower scored because they are always involved in qualitative studies and most of the evaluations are based on qualitative demonstrations.

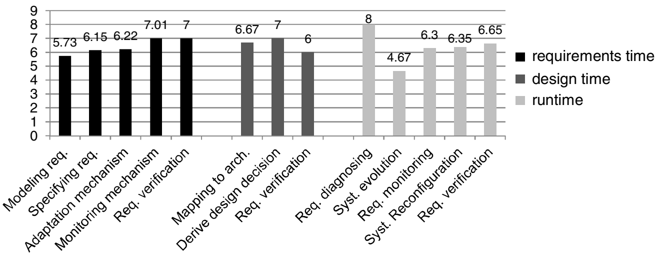

Fig. 12. Quality Score of RE activity

Challenges in Quality of Research. To achieve more precise and more effective adaptation decisions, we expect to derive quantitative models and quantitative representations during requirements modeling and analysis. Therefore, we should incorporate research results in other disciplines into RE for SASs. Fuzzy set theory, probabilistic theory and probability theory can be applied to describing uncertainties of both requirements and context. Control theory can be utilized to design adaptation mechanisms in SASs. Optimization theory, decision theory and game theory can be used to derive adaptation decisions. In this way, the quality of modeling methods and RE activities may get improved.

***RQ6**: What topics can we generalize based on the content of selected studies?*
***RQ7**: What is the relationship between topics and modeling methods?*

We code segments of relevant studies with 135 key phrases and 44 of them are kept after removing duplicate phrases. Then, these codes are categorized into 7 topics according to the content of each paper. Table 6 presents the relationship between topics and the related modeling methods. The bar in the table indicates the relative frequency of each method. One study may have more than one code and adopt more than one modeling method.

Table 6. Relationship between Topics and Modeling Methods

		KAOS	i*	Tropos	Feature Model	Context model	Problem Frame	UML	Business process model	Domain-Specific Model	Transition system	LTL	CTL&PCTL	FBTL	Z notation	Utility Function	Feedback control
Modeling requirements, contexts and systems	context modeling and analysis	1		1		1											
	defining SAS development framework			2													
	describing adaptation in feedback loop										1						1
	modeling adaptation mechanism	5	2	1			1	1	1								
	model adaptation with security requirements	2															
	modeling and resoning on NFR										1						
	modeling domain requirements for SAS									1							
	modeling requirements evolution	1															
	modeling RE activities of SAS		1														
	modeling security requirements	1															
	modeling systems behavior											1					
	modeling variant of self-adaptive systems									1							
	cope with requirements changes		2														
	customize software with preferences	1															
Specifying adaptive elements	specifying and managing self-* properties														1		
	specifying adaptation mechanism										1						
	specifying adaptation semantics	3										2					
	specifying adaptive programs											2					
	specifying adaptive requirements	1															
	specifying self-adaptive systems							1				1			1		
Dealing with uncertainty	addressing environmental uncertainty	1						1									
	decision making with uncertainty		1								2					1	
	mitigating uncertainty through adaptation	1	1											1			
	modeling sources of uncertainty		1														
	modeling uncertainty in requirements													1			
Verification and validation	QoS verification										1		1				
	requirements modeling and validation											1					
	validating the qualities of system										1	1					
	validating requirements at design time													1			
	verifying NFR at runtime										1		1				
	verifying adaptive programs											3					
	verifying requirements at runtime										3	1	3				
Monitoring and detecting	monitoring requirements	2		1								1				1	1
	detecting inconsistency within contextual req.			1													
	detecting requirements violation	1										1				1	
Adaptation and decision making	self-tunning with unanticipated changes				1												
	dealing with runtime variability reconfiguration				1												
	requirements-driven runtime reconfiguration	5	2	1	2												1
	runtime evolution by dynamic reconfiguration	1											1				
	runtime reconfiguration with model evolution								1								
	decision making to protect security req.	1															
	optimizing design decision			1													
	trade-off between FR and NFR							1									
mapping	mapping requirements model to arch. model		1									2					

Challenges in Research Topics

Researchers can refer to this table for investigating how different modeling methods are applied to a certain research topic. They can also explore how a certain modeling method can be adopted into different research topics. The blank areas in the table present research gaps in requirements modeling and analysis for self-adaptive systems. Promising topics may include: quantitative reasoning with NFR, modeling adaptation behavior with transition systems, runtime verification with context uncertainty. Indeed, to generate new topics or new motivation, a flexible way is incorporating uncertainty into the existing topics, since uncertainty has been a first-class concept in requirements engineering for self-adaptive systems.

4 Threats to Validity

Potential Bias. During conducting the review, researcher's bias may affect the analysis results. We adopt Kappa coefficient to assess the selection results and categorizing results of different researchers. When there is disagreement, we eliminate it by conducting a joint meeting and discussing with external researchers.

Internal Threats. Internal threats to validity deal with systematic errors in design and conduct of the review. To reduce this threat, we establish a rigorous protocol in advance and the protocol is reviewed by external reviewers. When conducting the review, the participants are divided into two groups. The final results are derived by integrating their individual results together.

External Threats. There may be some threats to external validity with respect to the generalization of the conclusions of this study. We note that with the increasing number of works in the literature we cannot guarantee complete capture of all the material in this area. There are still numerous unpublished papers, which cause the decrease of paper frequency in 2013 (Figure 4). We diminish this threat by taking into account all the primary venues in this area and integrating manual search, automated search and "snowball" search together to get the final set of relevant studies.

5 Related Roadmaps and Surveys

During the last decade, roadmaps and surveys of the literature have summarized the achievements and provide insight is this field. Cheng, et al. [5] and Salehie, et al. [7] are both highly qualified roadmaps in the literature. The former one presented challenges of software engineering for self-adaptive systems in four aspects: modeling dimensions, requirements, engineering and assurances. The latter one provided more details and insights in requirements engineering, design, implementation and test. More recently, Weyns, et al. [22] summarized several interesting research areas based on the research results of SEAMS from 2006 to 2011 and Dagstuhl seminar in 2008. Besides, they [23] also investigated the formal methods used in self-adaptive systems with research work between 2000 to 2011. Moreover, Patikirikorala, et al. [24] summarized various kinds of control engineering approaches used in designing self-adaptive systems with the publications between 2000 to 2010.

Different from these works, our review not only investigates the modeling methods and RE activities, but also explores how well these methods and activities are conveyed. In addition, we consider a wider time span and more publication venues for ensuring the coverage of existing research work. We present the state-of-the-art research trends and research gaps based on rigorously statistical results, which we hope to make this review more reliable than others. We believe that the SLR methodology we have adopted can make our review more trustworthy.

6 Conclusion and Future Work

The objective of this systematic literature review is to summarize the state-of-the-art trends of research on requirements modeling and analysis for self-adaptive systems. We found that most of these research works are from European countries and American countries, where the research groups produce more results than groups in other regions. A total of 16 modeling methods are used in 11 RE activities, and about 10 requirements quality attributes are studied, while adaptability is the most frequently concerned attribute. Online applications and service-based systems are the mostly cited application domains. It is found that some of the modeling methods need more exploration and most of the qualitative studies need more rigorous evaluation. The results of thematic synthesis (Table 6) show the gaps in using these modeling methods. In addition to these statistical results, we also analyzed the reasons implied behind the results and put forward some promising challenges implied by the results.

Our future work focuses on further investigating the relationship between requirements modeling methods and RE activities, the relationship between requirements quality attributes and modeling methods, and the relationship between requirements quality attributes and RE activities. Furthermore, we will also explore how the modeling methods and RE activities are evaluated in case studies and how the application domains are chosen for illustration. We also plan to publish all the research details and the relevant studies in the form of journal paper for helping researchers and practitioners better understand our research results and the research literature.

Acknowledgements. We thank Prof. Barbara A. Kitchenham and her team at Keele University for reviewing our protocol and all the received advice. This research is supported by the National Natural Science Foundation of China under Grant Nos 61232015 and 91318301. This research is also supported in part by the Natural Science Foundation of Guangxi Province under Grant No. 2012GXNSFCA053010.

References

1. Ganek, A.G., Corbi, T.A.: The dawning of the autonomic computing era. IBM Syst. J. 42, 5–18 (2003)
2. Nuseibeh, B., Easterbrook, S.: Requirements engineering: a roadmap. In: ICSE 2000 Proceedings of the Conference on The Future of Software Engineering, pp. 35–46 (2000)
3. Kephart, J.O., Chess, D.M.: The Vision of Autonomic Computing. Computer 36, 41–50 (2003)

4. Dobson, S., Denazis, S., Fernandez, A., Gaiti, D., Gelenbe, E., Massacci, F., Nixon, P., Saffre, F., Schmidt, N., Zambonelli, F.: A survey of autonomic communications. ACM Trans. Auton. Adapt. Syst. 1, 223–259 (2006)
5. Cheng, B.H.C., et al.: Software Engineering for Self-Adaptive Systems: A Research Roadmap. In: Cheng, B.H.C., de Lemos, R., Giese, H., Inverardi, P., Magee, J. (eds.) Software Engineering for Self-Adaptive Systems. LNCS, vol. 5525, pp. 1–26. Springer, Heidelberg (2009)
6. de Lemos, R., Giese, H., Müller, H.A., Shaw, M., Andersson, J., Litoiu, M., Schmerl, B., Tamura, G., Villegas, N.M., Vogel, T., Weyns, D., Baresi, L., Becker, B., Bencomo, N., Brun, Y., Cukic, B., Desmarais, R., Dustdar, S., Engels, G., Geihs, K., Göschka, K.M., Gorla, A., Grassi, V., Inverardi, P., Karsai, G., Kramer, J., Lopes, A., Magee, J., Malek, S., Mankovskii, S., Mirandola, R., Mylopoulos, J., Nierstrasz, O., Pezzè, M., Prehofer, C., Schäfer, W., Schlichting, R., Smith, D.B., Sousa, J.P., Tahvildari, L., Wong, K., Wuttke, J.: Software Engineering for Self-Adaptive Systems: A Second Research Roadmap. In: de Lemos, R., Giese, H., Müller, H.A., Shaw, M. (eds.) Software Engineering for Self-Adaptive Systems. LNCS, vol. 7475, pp. 1–32. Springer, Heidelberg (2013)
7. Salehie, M., Tahvildari, L.: Self-adaptive software: Landscape and research challenges. ACM Trans. Auton. Adapt. Syst. 4, 1–42 (2009)
8. Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M., Khalil, M.: Lessons from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software 80, 571–583 (2007)
9. Kitchenham, B., Charters, S.: Guidelines for performing Systematic Literature Reviews in Software Engineering. EBSE Technical Report (2007)
10. Kitchenham, B.A., Dyba, T., Jorgensen, M.: Evidence-based software engineering. In: 26th International Conference on Software Engineering, pp. 273–281 (2004)
11. Dyba, T., Kitchenham, B.A., Jorgensen, M.: Evidence-based software engineering for practitioners. IEEE Software 22, 58–65 (2005)
12. Protocol for A Systematic Literature Review of Requirements Modeling and Analysis for Self-adaptive Systems, `https://www.dropbox.com/s/t6i4ock5g11zo2x/SASProtocol.pdf`
13. Zhang, H., Babar, M.A., Tell, P.: Identifying relevant studies in software engineering. Information and Software Technology 53, 625–637 (2011)
14. ERA Outlet Rankings Access, `http://lamp.infosys.deakin.edu.au/era/?page=hmain`
15. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics 33, 159–174 (1977)
16. QDA Miner V4 and WordStat V6, `http://provalisresearch.com/products/`
17. Dybå, T., Dingsøyr, T.: Empirical studies of agile software development: A systematic review. Information and Software Technology 50, 833–859 (2008)
18. ISO 9126 software quality model definition, `http://www.sqa.net/iso9126.html`
19. Cruzes, D.S., Dyba, T.: Recommended Steps for Thematic Synthesis in Software Engineering. In: International Symposium on Empirical Software Engineering and Measurement, pp. 275–284 (2011)
20. Cruzes, D., Mendonca, M., Basili, V., Shull, F., Jino, M.: Extracting Information from Experimental Software Engineering Papers. In: 26th International Conference of the Chilean Society of Computer Science, pp. 105–114 (2007)

21. Extracted Data of Systematic Literature Review of Requirements Modeling and Analysis for Self-adaptive Systems, `https://www.dropbox.com/s/lksksvcyjsxhg0n/SASData.xlsx`
22. Weyns, D., Iftikhar, M.U., Malek, S., Andersson, J.: Claims and supporting evidence for self-adaptive systems: A literature study. In: ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, pp. 89–98 (2012)
23. Weyns, D., Iftikhar, M.U., de la Iglesia, D.G., Ahmad, T.: A survey of formal methods in self-adaptive systems. In: 5th International Conference on Computer Science and Software Engineering, pp. 67-79 (2012)
24. Patikirikorala, T., Colman, A., Han, J., Liuping, W.: A systematic survey on the design of self-adaptive software systems using control engineering approaches. In: ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, pp. 33–42 (2012)
25. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Sci. Comput. Program. 20, 3–50 (1993)
26. Yu, E.S.K.: Towards modelling and reasoning support for early-phase requirements engineering. In: 3rd International Symposium on Requirements Engineering, pp. 226–235 (1997)
27. Castro, J., Kolp, M., Mylopoulos, J.: Towards requirements-driven information systems engineering: the Tropos project. Information Systems 27, 365–389 (2002)
28. Zhang, J., Cheng, B.H.C.: Using temporal logic to specify adaptive program semantics. Journal of Systems and Software 79, 1361–1369 (2006)
29. Filieri, A., Ghezzi, C., Leva, A., Maggio, M.: Self-adaptive software meets control theory: A preliminary approach supporting reliability requirements. In: 26th International Conference on Automated Software Engineering (ASE), pp. 283–292 (2011)
30. Whittle, J., Sawyer, P., Bencomo, N., Cheng, B.H.C., Bruel, J.-M.: RELAX: a language to address uncertainty in self-adaptive systems requirement. Requir. Eng. 15, 177–196 (2010)
31. Ali, R., Dalpiaz, F., Giorgini, P.: A goal-based framework for contextual requirements modeling and analysis. Requir. Eng. 15, 439–458 (2010)
32. Weyns, D., Malek, S., Andersson, J.: FORMS: Unifying reference model for formal specification of distributed self-adaptive systems. ACM Trans. Auto. Adapt. Syst. 7, 1–61 (2012)
33. Filieri, A., Ghezzi, C., Tamburrelli, G.: A formal approach to adaptive software: continuous assurance of non-functional requirements. Form. Asp. Comp. 24, 163–186 (2012)
34. Zhang, J., Cheng, B.H.C.: Model-based development of dynamically adaptive software. In: The 28th International Conference on Software Engineering, pp. 371–380 (2006)
35. Bencomo, N., Belaggoun, A.: Supporting Decision-Making for Self-Adaptive Systems: From Goal Models to Dynamic Decision Networks. In: Doerr, J., Opdahl, A.L. (eds.) REFSQ 2013. LNCS, vol. 7830, pp. 221–236. Springer, Heidelberg (2013)
36. Goldsby, H.J., Cheng, B.H.: Automatically Generating Behavioral Models of Adaptive Systems to Address Uncertainty. In: Czarnecki, K., Ober, I., Bruel, J.-M., Uhl, A., Völter, M. (eds.) MODELS 2008. LNCS, vol. 5301, pp. 568–583. Springer, Heidelberg (2008)
37. Cheng, B.H., Sawyer, P., Bencomo, N., Whittle, J.: A Goal-Based Modeling Approach to Develop Requirements of an Adaptive System with Environmental Uncertainty. In: Schürr, A., Selic, B. (eds.) MODELS 2009. LNCS, vol. 5795, pp. 468–483. Springer, Heidelberg (2009)
38. Weyns, D.: Towards an integrated approach for validating qualities of self-adaptive systems. In: The 2012 Workshop on Dynamic Analysis, pp. 24–29 (2012)

39. Pimentel, J., Lucena, M., Castro, J., Silva, C., Santos, E., Alencar, F.: Deriving software architectural models from requirements models for adaptive systems: the STREAM-A approach. Requir. Eng. 17, 259–281 (2012)
40. Morandini, M., Penserini, L., Perini, A.: Towards goal-oriented development of self-adaptive systems. In: Proceedings of the 2008 International Workshop on Software Engineering for Adaptive and Self-Managing Systems, pp. 9–16 (2008)
41. Wang, Y., Mcilraith, S.A., Yu, Y., Mylopoulos, J.: Monitoring and diagnosing software requirements. Automated Software Engg. 16, 3–35 (2009)
42. Yiqiao, W., Mylopoulos, J.: Self-Repair through Reconfiguration: A Requirements Engineering Approach. In: 24th International Conference on Automated Software Engineering, pp. 257–268 (2009)
43. Inverardi, P., Mori, M.: Requirements models at run-time to support consistent system evolutions. In: 2nd International Workshop on Requirements@Run.Time, pp. 1–8 (2011)