Protocol for Mapping Study

Current Version: 1.3
Date: 01/04/2021

Involved Researchers: Ricardo Caldas[1], Razan Ghzouli[1], Patrizio Pelliccione[2], Danny Weyns[2], Thorsten Berger[2]

1 Motivation

Self-adaptive systems are designed to make decisions on their own and adapt to seek their goals continuously in face of uncertain operating conditions. However, accounting for all unforeseen disturbances that may lead the system into failure is unfeasible. The openness to which self-adaptive systems are prone poses unique challenges for ensuring that the system will behave within a desired envelope. With the aim to increase the trustworthiness of such systems, various types of formal methods have been employed that are applied in various stages of life cycle of self-adaptive systems [1]. With a formal method we refer to a composition of one or more formally specified properties (e.g., the probability that the system reaches a particular state should be below a certain value), a formally specified model (e.g., a reliability model of the system specified as a Markov model), and a tool that uses both for some purpose (e.g. a model checker that verifies that the property holds for the model). While some efforts have been devoted to characterize the formal techniques used in self-adaptive systems, these efforts are either outdated [2], or they have a specific focus, e.g., autonomous robotics systems [3].

Therefore, we are keen to build upon these works and provide an up-to-date map of literature on the characteristics and usage of formal properties and models in self-adaptive systems. Such a map will provide researchers with the state of the art overview and has the potential to identify open challenges. To that end, we will perform a mapping study. This document describes the protocol for this study.

2 Research Method

To reach our objective, we will conduct a mapping study on the published knowledge in the area of software engineering for self-adaptive systems. A systematic mapping study consists of a process with 5 steps [4] as shown in Figure 1: Research Questions Definition, Conduct Search, Screening of Papers, Design of Classification Scheme, and finally Data Extraction and Mapping.

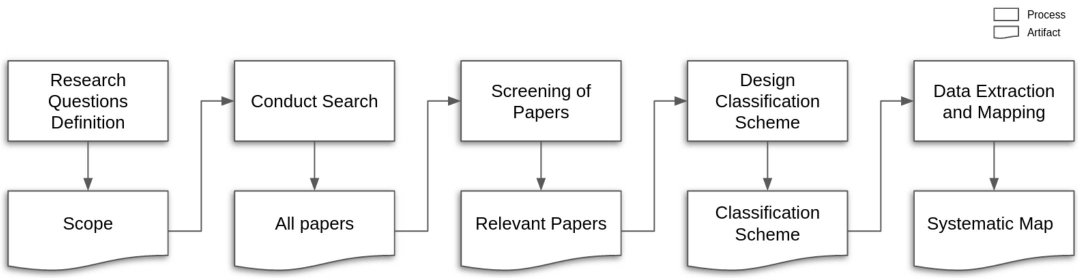

Figure 1: Systematic mapping process, based on [4]

In the remainder of this document we elaborate on each of these steps and explain how the study is organized accordingly. With this document in hands, the reader should be able to execute or replicate the mapping study.

3 Research Questions Definition

The research questions drive the mapping study and all the following steps are based on them. The process of eliciting and specifying the research questions involves all researchers that participate in the mapping study. We define the overall goal of the mapping study using the Goal-Question-Metric (GQM) approach [5], see Table 1.

With RQ1 we aim to get an overview of the formal properties that have been specified and used in self-adaptive systems. When we use the term “formal” we mean properties (and models) based on a mathematical framework. Our interests include the languages used for specifying properties, and specific characteristics of property specifications, such as an explicit notion of time. While characterizing formal properties in self-adaptive systems is the main focus of the work, formal properties are inherently linked with formal models. With RQ2, we want to get a view on the models associated with the properties derived from answering RQ1, following a similar investigation process. With R3, we aim to get an overview of the tools that are used to apply properties and models. Tooling plays an important role in

Table 1: Goal-based research questions.

Goal	
Purpose	Characterize
Issue	the specification and usage of
Object	formal properties in self-adaptive systems
Viewpoint	from a researchers perspective.
RQ1	What formal properties are specified in self-adaptive systems?
RQ2	What formal models are used in combination with these properties?
RQ3	What tools are used to apply the properties with the models?
RQ4	Why formal properties are used in self-adaptive systems?

the provision of guarantees for properties over models, and ease-of-use can lead to preferences for certain types of property/model combinations [3]. Finally with RQ4, we are keen to characterize why formal properties, models and tools are used in self-adaptive systems, in particular with respect to use case, the stage in the lifetime of the system, and the application domain.

4 Conduct Search

Ricardo ►*needs refactoring to reflect our new intentions*◄ The set of all papers is defined with respect to two dimensions, time and space. Whereas we quantify the time in years descending from 2020 and the space is defined by the venues that might contain relevant papers for our study.

This work, is inspired by a literature review on the use of formal techniques in self-adaptive systems that was conducted in 2012 [2]. That review covered the period from 2004 until 2012. Our intention is to update their findings and provide new insights regarding the state-of-the-art, therefore, we decided to restrict our time-span to January of 2012 until December of 2020.

From the space dimension, our choice of venue was mainly focused on publications from Software Engineering for Adaptive and Self-Managing Systems (SEAMS) conference. Based on knowledge and experience, we know that SEAMS provides a representative sample of research on software engineering of self-adaptive systems and contains the most relevant papers targeted by this study. Other work presented at software engineering venues also focused on one specific venue, examples are ICSE [6] and WICSA [7]. In addition, our decision is backed up by the current version of ACM SIGSOFT Empritical Standards[1]. So, with the aim of ensuring quality work and less noise in data [8, 9] and of having good sample with respect to SAS formal methods, we decided to only target SEAMS publications. We used a database that keeps track of SEAMS proceedings since 2006.

Finally, the set of all publications are papers from 2012 until 2020 in SEAMS proceedings. Totaling 196 papers.

5 Screening Papers

Screening papers consists of passing the complete set of potentially useful papers through a filter. The outcome is the set of relevant papers. Thus, in this section we define the filter, by means of inclusion and exclusion criteria, and then the procedure to be followed when discarding papers.

From Sect. 4, one can notice that formal properties and models descriptions might appear in the abstract or even title of the publications. However, it is our intention to include papers that provide formal specification (IC_3) of properties and models. This demands an in-depth scanning of the papers. For this reason, automatic search engines cannot be used in this phase and manual work must be employed.

The individual work of screening consists of checking if the paper is in compliance with the criteria presented in Table 2. It is done by parsing the paper in search for self-adaptation, which is usually spotted in the abstract and introduction. Then, looking for formalized properties, that are usually in the

[1] `https://arxiv.org/abs/2010.03525`

Table 2: Inclusion Criterion (IC) and Exclusion Criterion (EC)

ID	Description	Reasoning
IC_1	Self-adaptation as a first-class concern	Self-adaptation is fundamental for the scope.
IC_2	Explicit properties formalization	Papers with non- or semi-formal definitions may hinder ambiguous interpretation.
IC_3	Papers that specify and use formal properties	We want to focus on the usage of such properties. Not in the proposal of new methods.
EC_1	Tutorial, artifact, short paper, keynote, secondary studies, roadmaps	Such papers do not provide enough contextual information.
EC_2	No separation of concerns between target system and adaptation mechanism	We are interested in characterizing models/properties for separate components.
EC_3	Model or property are elicited for the evaluation section of the paper	Such models/properties are not related to the system but to the work itself.
EC_4	Papers using control theory design	Razan ►*update the reason*◄ .

form of equations. Once a property is found, the surrounding text is read to confirm if the equation is a property. Reading the methodology section of the paper helps to make an assessment of IC_4.

In this phase, two researchers ('A' and 'B') work in parallel on a complementary set of papers, compare and merge the results into two sets of papers. Papers that both agree that they are sure about the inclusion and the ones that are unsure. One or more researchers apart from 'A' and 'B', namely 'C', participate from the process in a validation role. 'C' receives the papers that were agreed to be unsure and a random small sample from the initial set of papers. With this procedure, we can improve efficiency, agility and minimize bias in the set of selected papers. See Fig. 2 for a representation of the process, where 'A', 'B', and 'C' are subscriptions to the processes.

This validation process may be an overkill in terms of validation and might slow the process. Therefore, we change x (percentage of papers) accordingly to how much we have advanced in the study. In the first rounds, x is the full set of papers and it reduces until each of the researchers get 60% of the papers, which gives a good efficiency with still small cross-validation.

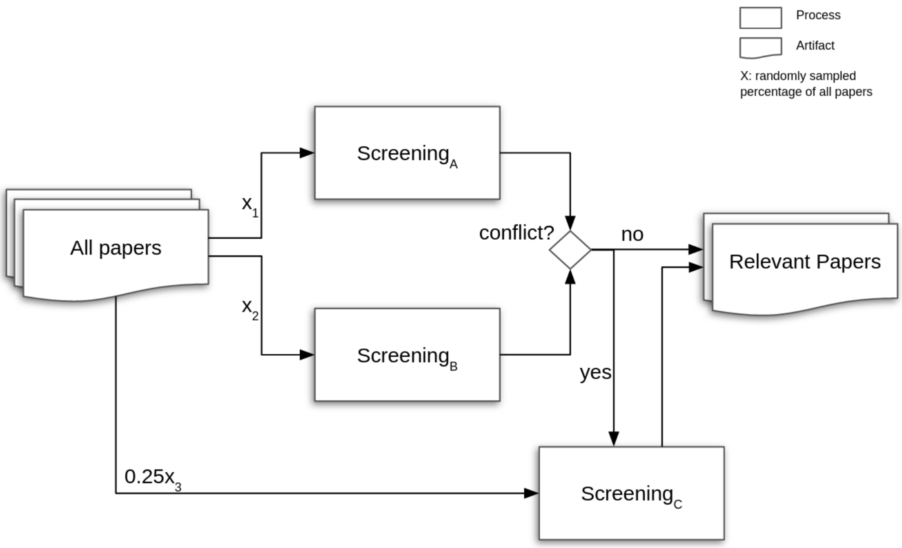

Figure 2: Screening process with cross-validation.

6 Classification Scheme

The data for mapping is extracted from the set of relevant papers. However, this process demands a scheme containing the precise information that is useful for answering the research questions. Laying on knowledge from papers that inspired this work [3, 2] we elicit and specify the data items to be extracted. Then, we provide clear-cut definitions and examples for each data item. See Table 3.

Table 3: Data items to collect.

ID	Field	Use
F1	Author(s)	Documentation
F2	Publication Year	Documentation
F3	Title	Documentation
F5	Property type	RQ1
F4	Property formalism	RQ1
F6	Property language	RQ1
F7	Explicit time	RQ1
F8	Explicit probability	RQ1
F9	Target system	RQ2
F10	Model type	RQ2
F11	Model formalism	RQ2
F12	Modeling language	RQ2
F13	Discrete	RQ2
F14	Tool	RQ3
F15	Use Case	RQ4
F16	Offline/Runtime	RQ4
F17	Application Domain	RQ4

(F1-F3) Metadata. Data about the papers collected. For documentation.

(F4) Property type In which of the common categories does the property best fits. E.g. Safety, Liveness, Reachability, Stability, Deadlock, Functional Correctness, Consistency, Completeness, Fitness, Determinism.[2]

(F5) Property formalism[2] Mathematically based technique used to define the property. E.g. Set Based, State-Transition, Logics, Temporal Logic, Dynamic Logic, Other Logics, Process Algebra, Ontology [3].

(F6) Property language[2] Language used to specify the property. E.g. Computational Tree Logic (CTL), Linear-time Logic (LTL), Probabilistic Computational Tree Logic (PCTL).

(F7) Explicit time Whether the property explicitly details time.

(F8) Explicit probability Whether the property explicitly details probability.

(F9) Target system. Whether the model used refers to the target system behavior, adaptation manager behavior, or other.

(F10) Model type Defines what is being modeled. E.g. Reliability, Performance, Functional behavior.

(F11) Model formalism[2]**.** Mathematically based technique used to define the model. E.g. Set Based, State-Transition, Logics, Temporal Logic, Dynamic Logic, Other Logics, Process Algebra, Ontology [3].

(F12) Modeling language[2]**.** Language used to specify the model. E.g. Markov Chains (MDP), Markov Decision Processes (MDP), First-order Logic (FOL).

(F13) Discrete Whether the model is discrete or continuous (mathematical definition). We can find hybrid models as well. E.g. DTMCs are discrete models and algebraic formulations are continuous.

(F14) Tool Tool used to specify the model, properties and compute them. In cases of reasoners proposed by the authors we write the given name or ad hoc.

[2]Formalism differs from language in the sense that they are mathematically based techniques for specification of models and properties. Whereas languages are derivations of formalism sets. Example: Temporal Logic is a formalism; LTL, CTL, PCTL are languages. This terminology is based on [3].

(F15) Use Case Assurance provision method. Includes frameworks or techniques to provide assurance of correctness (verification, reasoning and synthesis). E.g. Model Checking, Theorem Proving, Runtime Monitoring, Assurance case synthesis, Synthesis of correct-by-construction controllers.

(F16) Runtime Whether the approach is designed for offline or runtime. Offline techniques may be design-time or postmortem.

(F17) Application Domain Use case domain. E.g. Robotics, Self-Driving Vehicles, Internet of Things, Healthcare, Transportation, Mobile Communication, Finance, Business Analytics & eCommerce, Decision Support Systems, Forestry Farming & Urban Informatics, Manufacturing and Process Control, Logistics and Maintenance, Public Safety.

7 Data Extraction and Mapping

Data extraction and mapping is done by manual parsing, identification, classification among the data items from Sect. 6 and filling a spreadsheet[3].

By the end of this step, the researchers shall be able to answer all the research questions and provide insights delimiting the baseline for further research on formal methods for SAS.

We followed the same idea for provision of guarantee of the process as we did in Sect. 5. Please refer to that section if you have doubts with x, A, B, and C from Fig. 3.

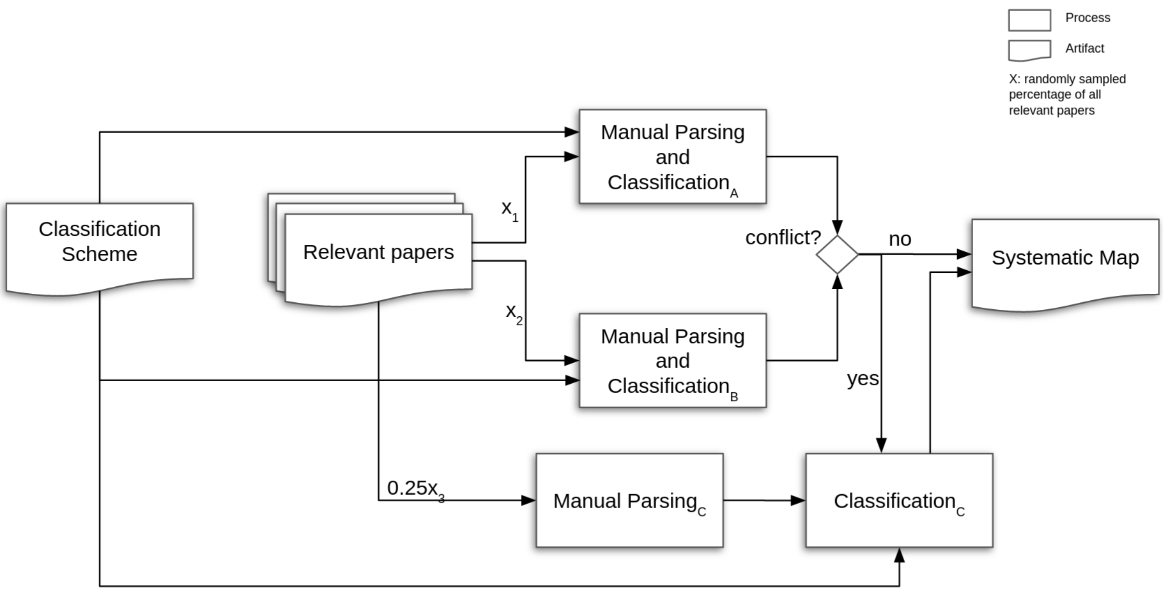

Figure 3: Data extraction and mapping process with validation.

8 Quality Assurance

To ensure the quality of the research, we follow the checklist:

- Protocol validation.
- Validation measures in conducted processes.
- All the data is publicly available.

The protocol validation consists of internal and external validation by peer-reviewing with knowledgeable researchers with different expertise. By internal validation we mean the researchers that helped to conceive the idea of the ongoing study. By external validation we mean researchers that were not actively involved in the creation of the study.

[3] https://chalmersuniversity.box.com/s/bukllq4w0tzniftnh588sykm6wrms5z0

As validation measures we include internal validation for the inclusion/exclusion of the papers, the extracted data, and the reported results.

All the data and text will be made available in the format of a replication package.

References

[1] D. Weyns, N. Bencomo, R. Calinescu, J. Cámara, C. Ghezzi, V. Grassi, L. Grunske, P. Inverardi, J. Jézéquel, S. Malek, R. Mirandola, M. Mori, and G. Tamburrelli, "Perpetual assurances for self-adaptive systems," *CoRR*, vol. abs/1903.04771, 2019. [Online]. Available: http://arxiv.org/abs/1903.04771

[2] D. Weyns, M. U. Iftikhar, D. G. De La Iglesia, and T. Ahmad, "A survey of formal methods in self-adaptive systems," in *Proceedings of the Fifth International C* Conference on Computer Science and Software Engineering*, 2012, pp. 67–79.

[3] M. Luckcuck, M. Farrell, L. A. Dennis, C. Dixon, and M. Fisher, "Formal specification and verification of autonomous robotic systems: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–41, 2019.

[4] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12*, 2008, pp. 1–10.

[5] V. R. B. G. Caldiera and H. D. Rombach, "The goal question metric approach," *Encyclopedia of software engineering*, pp. 528–532, 1994.

[6] C. Zannier, G. Melnik, and F. Maurer, "On the success of empirical studies in the international conference on software engineering," in *28th International Conference on Software Engineering*. ACM, 2006.

[7] M. Galster and D. Weyns, "Empirical research in software architecture: How far have we come?" in *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, 2016, pp. 11–20.

[8] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015.

[9] C. Wohlin, P. Runeson, P. A. d. M. S. Neto, E. Engström, I. do Carmo Machado, and E. S. De Almeida, "On the reliability of mapping studies in software engineering," *Journal of Systems and Software*, vol. 86, no. 10, pp. 2594–2610, 2013.