Survey: Classification Scheme for Software Engineering (SE) Research

Participant – General Data

Working package ID:	WP-ID-1
What is your **current position**?	Research Scientist (PhD Candidate)
Researcher year?	3
What is your **research topic**? **keywords** of participant	model-based quality analysis, self-adaptive systems, design-time analysis, performability evaluation
What is your **research topic**? **ACM** Computing Classification System from: https://dl.acm.org/ccs	model-driven software engineering, software performance, model-development and analysis
Short description of proposal working package: (Note: Mark items that are unclear or not defined yet with „unclear“)	WP 'Performability evaluation for self-adaptive systems' .This workpackage shall develop an approach to determine the quality property 'performability' of an SAS at system design-time. Expected outcome: formal language to describe adaptation rules, approach to specify failure model, prototypical tool implementation.
Phase of working package: (Select: in planning phase not finished yet, planned, planned and in conduction, conducted)	partly planned and in conduction; this WP's has already some prototypical work but is not finally planned in a clear formal way.

Phase of Working Package (WP)	
In every „Planning Phase“: * How was the procedure conducted? * What is/was challenging in planning phase of the proposal working package? - Challenges in methodology - Technical challenges - Organizational challenges (i.e., management of research work) * What are still open questions?	conduct planning phase: planning phase is from my experience no linear process that can be done upfront. instead I have started with an initial picture in mind about the content of this WP; then I started with prototypical work in order to undertand the addressed problem better. Meanwhile I am in the refinement phase, where I can now start with the actual WP planning which means that now the required outcome must be defined both in terms of definint the approach that need s to be developed and also the way how to evaluate it. Open questions: clear definition of required approach and output artefacts is missing; how everyting is evaluated is also missing.
* How you try to solve them? (e.g., get a hint from superviser for metrics etc.)	/
In „Conducting Phase“: * What was challenging in each conducting phase of the proposal working package? - Challenges in methodology - Technical challenges	As I'm not yet in the actual conducting phase, (I would not count prototypical work as part of the actual conducting phase) I would leave this part empty. But as a general comment that I already see as one of the challenging technical problems is the integration of your own solution/work in the existing ecosystem and code base due to poor existing code quality. Another challenge will be if the choosen case study will be sufficient to show/apply the approach. also designing and conducting the experiment will be challenging. the experiment
* How you try to solve them?	As usual, I will address the implementation step-by-step in an agile way, which means that after each partial implementation I will review and adapt accordingly. Also I will implement feedback mechanisms to check as early as possible if everything will go in the right direction.

Classification Scheme for Software Engineering Research

The following multi-dimensional classification scheme intent to classify SE research artefacts.
The proposed solution tries to classify SE research according different kind of statements according to their research object, their kind (e.g., relevance), and their evidence.
A statement is defined as „a fact, a hypothesis or conclusions out of data results of a research object with a given evidence. The statement can either be about an intrinsic property of the research object, a relationship between the object and its context or about the relevance of the object.“ [1].
The following description and example is adapted from [1].

For the **research object (dimension 1)**, we identified five classes:

(1.1) Problem:
The object of research can be a problem, even if no solution is presented. However, the investigation and understanding of a problem is the base for a future solution. Most of these papers take place on social or social-technical context settings.

(1.2) Method:
This object of research refers to any human-driven approach for fulfilling SE tasks (e.g., code reviews, pair programming). This can include tool-supported and automated sub-tasks and processes. However, the main approach is driven by humans.

(1.3) Model:
Models are abstractions from the real world for a certain designated use for which certain properties of the modeled entity hold. In SE, models are, e.g., reference architectures for software systems, but also (prediction) models of software process or software product qualities.

(1.4) Metamodel/Language:
Metamodels serve as language specifications (like grammars do). Such languages can be results and objects of SE research. Examples range from programming languages to specification languages.

(1.5) Automation:
This subclass refers to algorithms, tools, and applications, hence fully automated tasks in SE. Of course, elements from this subclass can perform sub-tasks of human-driven approaches (see (1.2)).

Different **kind of statements (dimension 2)** can be made about a research object.
This dimension is divided into three main classes:

(2.1) Property-as-such:
Statements about intrinsic properties of the research object, which do not depend on external factors. E.g., "Compiler C is modularized into modules for lexing, parsing, abstract syntax tree attributation and code generation and a symbol table". Although this is certainly not the only way to modularize a compiler, it is a statement on the modularization of the described compiler C.

(2.2) Property-in-relation:
Statements about the research object which are related to its context.
This can be either statements on (i) the existence of influence factors. E.g., "Compiler C's performance mainly depends on the size of the input file and the performance of the file system." or (ii) it can be statements on the influence itself: "Compiler C's performance depends polynomially on the given input file size and linearly on the performance of the file system."

(2.3) Relevance/Novelty:
Statements about the importance of a research object, necessarily relating to a given context. Of course, the term relevance itself includes many properties such as the broadness of applicability, the size of the impact of the application. From a scientific point of view, the depth of thought and the originality can be also seen as aspects of relevance. An example is "Object-oriented modeling can be applied beneficially to all application domains of software except driver programming." However, this is not a statement about its applicability (how difficult the application is) but about the potential areas of a positive application.

The aforementioned classes of statements are underlined with different **kind of evidences (dimension 3)**, i.e., empirical or non-empirical:

(3.1) Argumentation:
This kind of statement has no significant empirical or non-empirical evidence (i.e., formal proof). Statements here could imply the description of solution approaches (e.g., algorithms in pseudo-code or text-based descriptions).

(3.2) Motivating example:
A first exemplary demonstration of a research object with an example made by the authors or taken from the community.

(3.3) Case study:
Properties of the research object are validated in a case study. Further specific sub-classes are possible (e.g., community case studies).

(3.4) Survey through interview or questionnaire:
Empirical data on a property of a research object taken from surveys (either qualitatively or quantitatively evaluated).
Sub-classes in terms of unstructured, (semi-) structured interviews can be defined.

(3.5) Experiment:
Data on properties of a research object gained through a controlled experiment, usually with humans or automated as benchmark.

(3.6) Verification:
Mathematical proofs relating to a set of axioms and proof-rules. This can be, e.g., analyses of properties of algorithms and so on.

(3.7) Mining software repositories:
This class represents software repositories such as version control systems, requirements/bug-tracking systems, and mail archives that record software developers' activities.

The concept can be display with three dimensional axes. Each coordinate should further links to evaluation / research questions that should contain action items and methodological recommendations to conduct the research (cf. Appendix). In this way, SE research papers could be easier to plan, write in a structured way, assess and are more comparable. This classification could also be used for information retrieval in a knowledge management system (KMS).

[1] Kaplan, A. et al. 2021. A Classification for Managing Software Engineering Knowledge. In Proceedings of the Evaluation and Assessment in Software Engineering 2021; accepted, to appear.

Example Application

In the following, an example application on existing research work for illustration purpose is presented. The example approach in [2] extends the work of [3] and uses an architectural model together with a context model to analyze the dataflow regarding confidentiality violations in dynamic Industrial IoT scenarios. The illustration paper [1] consists of 3 contributions, an extension to the overall modeling and analysis process, a metamodel to describe context properties, and an extension to the analysis part (realized by a model to model transformation).

For the **research objects** the classes (1.2), (1.4), and (1.5) are selected. The first class (1.2) because the extended process describes a human-driven approach to model confidentiality. This description matches the definition of a human-driven SE task. Class (1.4) because of the approach contains a metamodel for modeling context attributes. The analysis extension is realized by an automated model to model (m2m) transformation and therefore matches the description of class (1.5) with fully automated tasks.

The **statement and evidence** assignments for each contribution are:
1. process extension: The paper describes only the need for these kinds of analysis processes and describes the general applicability. Therefore we select (2.3) relevance and as evidence (3.1) as well as (3.2), since we do not give empirical evidence of the general applicability but argue it and provide a motivating example.
2. metamodel: The metamodel is not explicitly evaluated but rather whether it is suitable as an input for the analysis. Therefore, we select (2.2) as it is a related property. The evidence is given by the motivating example (3.2) and the executed scenarios based on different case studies (3.3). Additionally, by using case studies from industrial partners to model the scenario, we investigated the relevance (2.3).
3. analysis: The analysis is firstly evaluated by analyzing different scenarios and comparing the output to the expected output. We evaluated there only the functional properties. Therefore we would choose (2.1) as property-as-such since these mainly check whether the behavior is as expected. The type of evidence is (3.2) and (3.3) since we use different case studies and a motivating example. Additionally, we investigated the scalability of the approach. This is a (2.2) property-in-relation statement since we investigated the execution time compared to different input parameter sizes. As evidence class, we select (3.5) since we scaled the selected input parameter and then observed the system's behavior.

Application Use Cases for a KMS
Potential use in a KMS: For instance, with the help of the classification we can search for all papers which describe a confidentiality process. We would then get our approach back and could see easily, that the approach uses additionally a metamodel and an automation. We could see, what kind of statements are evaluated and which evidence is given. This could then be used for further searches for example regarding metamodels and confidentiality. Another possible use case could be, that we have developed a metamodel and want to investigate what kind of evaluation is commonly used. Here we would again insert the research object and then have a look at the given evidence to get insight, how other researchers have evaluated their approaches to compare own results with state-of-the-art.

[2] N. Boltz et al. 2020. Context-Based Confidentiality Analysis for Industrial IoT. In SEAA 2020. IEEE. https://sdqweb.ipd.kit.edu/publications/pdfs/boltz2020.pdf
[3] Stephan Seifermann et al. 2019. Data-Driven Software Architecture for Analyzing Confidentiality. In ICSA 2019. IEEE, 1–10
https://sdqweb.ipd.kit.edu/publications/pdfs/seifermann2019b.pdf

Classification Scheme – Construction Method

How long did it take to read and to understand the main concept of the classification scheme (duration in minutes)?	~40 min (trying to understand the different meaning of each property took the most time)
Rate the following aspects from 1 (weak) to 5 (strong) and argue why in a comment	Rate: 1-5, comment (argumentation)
Understandability	rate: 3, especially understanding the meaning of the various properties of each dimension; some are easy to grasp, but especially the statement dimension is very complex and complicated; the examples fit for a general explanation but transfering them to my own research is challenging
Orthogonality of dimensions	rate: 5
Orthogonality of classes in each dimension - in example application - in proposal working package	example: 3, argue: depends on how familiar the research topic is for yourself, could be difficult to follow the proposed classification if you are not too familiar with the topic; wp: 4, argue: depending on granularity of wp it was easy to classify approach, if you could not categorize yourself this is a clear hint that the wp needs further refinement
Usefulness for your purpose (proposal working package)	rate: 5 applying the method brought up more questions than trying to help reducing; I would see the benefit in helpting to identify gaps; maybe because my WP is still not well-structured, i.e. covers too many aspects and should be split up in smaller parts so that maybe only one property per dimension is covered; currently it feels like all attributes are covered in the WP i.e. scope is to broad;
Adequate visualization technique?	rate: 3 in general yes, but not for my case; I would have to place several voxels in the 3-d space because as previously said, the WP is currently too general and covers several attributes;

Classification Scheme - Classification of Proposal Working Packages

Task: Classify your proposal working package according to the aforementioned classification scheme and give feedback

Task: Classify your working package according the aforementioned three dimensions of the classification and argue why (cf. example application).

Template:
* Research object
* Kind of statement
* Evidence

If available: Map explicit evaluation / research questions and method design attributes to the template.

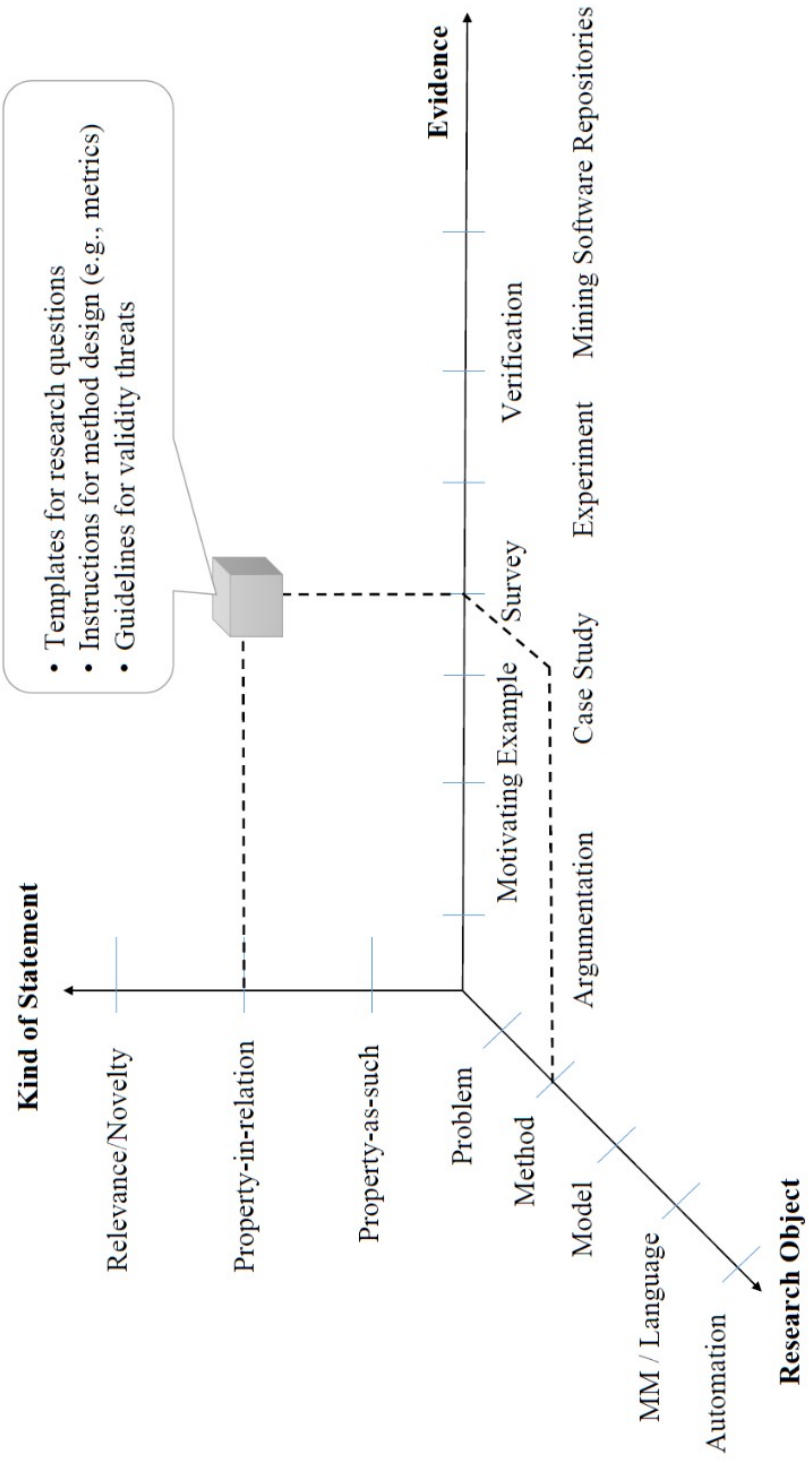
(1) Research object: performability evaluation of self-adaptive systems (SAS):
method -> mathematical approach how to analyze SAS;
meta-model -> language to describe e.g. dependability model of SAS + rules for adaptation strategy;
automation -> tool that automatically conduct the performability evaluation

(2) kind of statement:
relevance -> argued by a motivating example why such an analysis would be helpful at system design-time (e.g. support architectural design decisions);

(3) evidence: motivating example (standard example: loadbalancer) + case study from MOSAIC project (mini exemplar)

Describe strongest aspect of classification process: How did the classification scheme support you?	research object and evidence dimension can be identified easily compared to statement dimension;
Describe weakest aspect of classification process:	a research object usually has multiple attributes, i.e. depending on what kind of facet I currently work on I would categorize it into different aspects. Thus categorization heavyl depends on my own argumentation and not so much on the properties that must be fulfilled in order to group it into a distinct property; also difference between case study and experiment was not clear to me, is a case study not already some kind of experience?
Further comments:	the method requires a very good understanding about the different meanings of the properties and dimensions; The main challenge that I see for applying this categorization is, that generally a paper will cover multiple properties of this classification because the scope of a research work covers different aspects of a problem; each problem again can be categorized by using this classification; but how do you aggregate an overall rating of all problems? I think e.g. the result for a paper would be a 'Punktewolke' instead of a single voxel and the challenge would then be to trace the data points back to the arguments given in the paper.

Appendix


Kind of Statement
Relevance/Novelty
Property-in-relation
Property-as-such
Problem
Method
Model
MM / Language
Automation
Research Object
Evidence
Motivating Example
Survey
Verification
Argumentation
Case Study
Experiment
Mining Software Repositories
• Templates for research questions
• Instructions for method design (e.g., metrics)
• Guidelines for validity threats