Evidence in Architecture Degradation and Consistency Checking Research

Preliminary Results from a Literature Review

Sebastian Herold
Department of Mathematics and Computer Science
Karlstad University
651 88 Karlstad, Sweden
sebastian.herold@kau.se

Martin Blom
Department of Mathematics and Computer Science
Karlstad University
651 88 Karlstad, Sweden
martin.blom@kau.se

Jim Buckley
CSIS/Lero - The Irish Software Research Centre
University of Limerick
Limerick, Ireland
jim.buckley@ul.ie

ABSTRACT

Theory suggests that software development may benefit from applying architecture consistency checking and hence, academics have focused on this area. In such circumstances it is important to characterize the overall focus of the field, in order to assess the actual impact of the research work carried out to date, identifying where researchers are placing their effort and which sub-topics are being neglected. In doing so a map is provided for researchers allowing them focus on relevant research gaps and avoiding saturated sub-topics, to deliver results relevant for academia and industrial practice.

The goal of this paper is to shed some light on the current state of research in software architecture degradation and consistency checking, particularly focusing on empirical evidence. Preliminary, yet significant results from an ongoing mapping study, as a precursor to a more detailed literature review, are presented and discussed. These results show that solution-proposals constitute a large part of the body of work. In addition, they show that case studies, with potentially limited external validity, are prevalent in terms of empirical design. We argue that the proportion of empirical studies needs to be expanded and that existing case studies should be complemented by experiments and surveys, assessing its impact in practice.

CCS Concepts

• Software and its engineering → Software architectures; Empirical software validation; Software evolution;

Keywords

Software architecture degradation; software architecture consistency; systematic mapping study; literature review; empirical evidence

1. MOTIVATION

Software architecture degradation is the process of continuous divergence between the implementation of a software system and the intended software architecture [17, 20]. Software systems that are affected are less likely to be aligned with the design decisions and the desired quality attributes manifested in the intended software architecture [1]. Architecture consistency checking approaches are one strategy to address software architecture degradation [7]. They partially automate the detection of inconsistencies between the intended software architecture and the implementation, supporting software architects and developers in their efforts to keep software architecture and code consistent.

Theory and logical argumentation suggest that software development can benefit from applying architecture consistency checking. However, the actual impact and prevalence of software architecture degradation, the benefits and drawbacks of existing architecture checking techniques, and the gain and efforts associated with applying them need to be empirically investigated. This is required to efficiently deliver research results with impact on industrial practices and to focus future research efforts in the field.

The research community has developed a significant body of work regarding the related concepts of “erosion”, “drift”, “degradation”, and “decay” [17, 9, 12, 19, 21]. A few surveys were published which give general, technically-focused overviews and taxonomies of the field [7, 11, 8, 2]. However, a summary of the state of empirical evidence is missing. Without such a summary, the degree to which these important questions have been addressed is questionable.

This paper presents and discusses initial results from a literature review addressing this gap. Specifically, this initial analysis focuses on the following research questions:

RQ1: What are the prevalently investigated aspects of software architecture degradation and consistency checking research?

RQ2: What kind of empirical evidence exists regarding software architecture degradation and consistency checking research, and what can be said about the validity and reliability of the evidence?

The actual literature review is a two-step process in which we first perform a systematic mapping study of the field to identify promising research issues, as a precursor to a more detailed, systematic literature review, during which we will...
refine the research questions and assess them with respect to the identified studies [18, 13]. The results discussed in this paper are based on intermediate findings from the mapping study and, as such, are not final. But already themes are apparent that may stimulate a discussion of potential research directions.

The remaining article is structured as follows. The following section describes the study design. Section 3 presents the results which are discussed in Section 4. Related studies are discussed in Section 5. The article is concluded in Section 6.

2. STUDY DESIGN

The design of the mapping study follows the process suggested by Petersen et al. [18] as shown in Fig. 1. They propose the phases of (a) defining the research questions for the study (presented in Sec. 1), (b) conducting the search, (c) screening the papers, (d) keywording of abstracts, and (e) data extraction and mapping. Each of the following subsections covers one of the steps (b)-(e).

2.1 Conducting the Search

We identified the primary studies by querying four large databases of scientific publications, namely Scopus, ACM Digital Library, SpringerLink, and IEEEExplore. The concrete search strings used for the databases varied slightly, based on the query format required and the number of operators allowed by the particular database. However, all search strings expressed a search for entries containing the term “software architecture” and synonyms of “degradation”, “consistency”, or “inconsistency”, such as “erosion” or “conformance”. Searches were restricted to the software engineering discipline if possible or to computer science in cases where a more specialized subdiscipline was not offered by the particular database. Conducting the search resulted in 2,434 papers after elimination of duplicates.

2.2 Screening of Papers

The papers retrieved from the databases in the first step were screened for relevance based on several inclusion and exclusion criteria. Only papers published in journals or conference proceedings, technical reports or books were included, whereas slide presentations, keynote abstracts and forewords were excluded. The main content-related inclusion criterion was that the title, the abstract, or the list of keywords attached to a paper must mention software architecture degradation or architecture consistency checking (or one of their synonyms) in a way that the screening researcher could conclude that the focus of the paper is in one of these areas. This particularly excludes papers mentioning these topics in general, introductory sentences only. We also decided to exclude papers on architecture recovery as the focus of this mapping study is architecture (in-)consistency between existing, intended architectures and code bases, not reverse engineering of architectures. Moreover papers were excluded if they mentioned “architecture” or “consistency” in unrelated contexts, such as in the description of a system’s architecture alone or in the consistency between two other types of artefacts.

All papers were screened by two of the authors and classified as “relevant”, “irrelevant”, or “unclear”. In cases in which both agreed on “relevant” or “irrelevant”, the paper was accepted or rejected, respectively (129 papers were rated “relevant” by both authors). 114 papers were categorized inconsistently by the two authors concerned and the third author had the casting vote: if the third vote led to a majority of “relevant” it was accepted. 45 more papers were considered relevant and added to the pool of papers going forward. There were 34 remaining papers, for which no consensus could be reached through rating, and these were also included for the subsequent steps to err on the side of safety.

2.3 Keywording and Classification

The wording of the abstracts and, in many cases, of parts of the introduction, evaluation, and conclusion sections of papers was to be performed incrementally in four iterations. After each iteration, the classification scheme is revised based on any newly identified keywords, which may lead to refinement of categories, splitting of categories or merging of categories. In the first iteration of keywording, a subset of the papers was reviewed by two of the authors to assess potential misunderstanding in the keywording required to characterize a study; since no significant misinterpretation occurred during this process, papers in iterations two and three were only single-reviewed to reduce the workload; papers in iteration four will be double-reviewed to avoid laxity potentially caused by interpreter drift.

The preliminary results presented here are based on the keywording and the classification scheme after iteration three. So far 129 papers have been classified, 10 of which were excluded, based on deeper insights from reading beyond their abstracts.

2.4 Mapping

For this article, three subcategories of the classification scheme were picked as facets for an initial map of the research field. Although reflecting only a subset of the overall set of relevant articles, these facets already show significant trends. These three facets are:

- Study type facet: What kind of study was performed and, in the case of empirical studies, which strategy was applied, i.e. create & design research, case study, experiment, survey? This facet addresses RQ2.
- Evaluation focus facet: If the paper contains an evaluation, what was the focus of the evaluation? Does it assess the technical quality of the tool for example, or the erosion in a subject software system. This facet addresses RQ1 and RQ2 indirectly.
- Focus of the Approach: Which subtask of tackling architecture erosion and checking erosion is addressed or covered by the paper’s contribution? This facet also addresses RQ1 and RQ2 indirectly.

The categories of these facets will be explained in more detail in the following section.
3. PRELIMINARY RESULTS

The results of mapping the first 119 papers of the study according to the three facets are depicted in Fig. 2.

We selected two different sources for the categories of the study type facet using several empirical research strategies by Oates as our guideline: design & creation research, case studies, surveys, and experiments [16]. Further research categories, as suggested by Wieringa et al., were adopted, as long as the categories were not already covered by the existing empirical research strategies defined [22]. This led to the additional categories of solution proposals, opinion papers, experience reports, and philosophical papers of which only the first is included in the discussion here for the sake of brevity and because of the other categories’ limited informative value regarding the motivating research questions. Table 1 lists the categories and provides a short description.

46.2% of the 119 investigated studies are categorized as solution proposals, and a further 36.1% are design & creation research. These latter studies typically contained an evaluation of the approach suggested and, in these, the prevalent design employed was case studies; 39 of these studies follow this evaluation strategy, and three followed an experimental strategy. Surveys and experiments are described in 2.5% and 5.0% of the reviewed papers, respectively.

The categories of the evaluation focus facet were derived during the keywording of the papers and are described in Table 2. Based on all paper reviewed so far, 65.5% of studies primarily focus on technical aspects, mostly illustrating general applicability or evaluating performance of proposed techniques. Non-technical aspects were evaluated in 10.1% of the papers. About 15.1% of the papers had a rather system-oriented focus, either primarily assessing erosion (11.8%) or its impact (3.3%).

Table 3 shows and describes the categories of the approach focus facet which stretches over different operational aspects of acting against architecture degradation. According to this categorization, a majority of papers address the detection of erosion and inconsistencies (65.5%) while the other aspects, analysis, fixing, and enabling prevention—are subject to a similar but lower number of studies (13.4%, 14.3%, and 16.8%, respectively).

![Figure 2: Systematic map of the reviewed studies, showing absolute and relative numbers of papers per category and per combination of study type and evaluation focus and study type and approach focus, respectively. Multiple mentions allowed.](image)

Table 1: Categories of the study type facet.

<table>
<thead>
<tr>
<th>Solution proposal</th>
<th>Novel or significant extension of a technique, applicability and benefits shown by small example or good line of argumentation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design &amp; creation</td>
<td>Focus on developing new tools, models or methods. Followed by empirical evaluation (case study, survey, experiment).</td>
</tr>
<tr>
<td>Case study</td>
<td>Focus on single or small number of instances of the phenomenon of interest, its boundaries to its context are often blurred/unclear.</td>
</tr>
<tr>
<td>Survey</td>
<td>Focus on obtaining same kind of data from a large group of people/events/systems in a systematic way.</td>
</tr>
<tr>
<td>Experiment</td>
<td>Focus on cause and effect relationships carefully excluding undesired factors.</td>
</tr>
</tbody>
</table>

\[1\] Please note that the studies in each empirical-evaluation category are not exact subsets of those in the creation & design category and that some creation & design papers evaluated their approach with more than one strategy.
The results show that, independent of evaluation focus and approach focus, design & creation research evaluated through case studies and solution proposal validated through argumentation or small examples are the prevalent study types. Experiments and surveys are comparably scarce across all evaluation focus and approach focus categories. Technical aspects are significantly more often investigated than other aspects and are almost the only evaluation focus category for which experiments are reported (with the exception of a single study on non-technical aspects). The same prevalence is true for “Detection” in the approach focus facet whereas experiments are reported for all four categories.

| Technical | Evaluation focuses on technical aspects of contribution, such as accuracy, performance, or general applicability in terms of correctness. |
| Non-technical | Evaluation focuses on non-technical aspects of contribution, such as usability or user perception. |
| Erosion assessment | Evaluation focuses on assessing the degree of erosion or the state of consistency/inconsistencies in the studied system(s). |
| Erosion impact | Evaluation focuses on investigating the impact of erosion / the benefit of consistency on/for the studied system(s)/organization. |

Table 2: Categories of the evaluation focus facet.

| Detection | Study investigates/addresses the detection of architecture degradation or architectural (in)consistency. |
| Analysis | Study investigates/addresses the analysis of architecture degradation, such as how to measure degradation, understanding causes of degradation, etc. |
| Fixing | Study investigates/addresses how to fix architecture degradation/inconsistency and re-establish consistency. |
| Enabling prevention | Study focuses on tasks that support and enable degradation prevention, such as traceability or constraint enforcing in the architecture-to-code context. |

Table 3: Categories of the approach focus facet.

4. DISCUSSION OF RESULTS

In this section, we discuss the results in the light of the motivating research questions as formulated in Sec. 1.

4.1 RQ1: Prevalently investigated aspects

The prevalence of the detection category of the approach focus facet was rather expected. Degradation or inconsistency detection is a necessary precursor to analysing and fixing degradation, such that approaches categorized as “Analysis” or “Fixing” often build upon existing and published “Detection” work (e.g., [10] builds on [5]). Additionally, even in the later years, novel techniques and tools focused on degradation detection still contribute strongly to the overall body of knowledge. Consequently this category still scores highly in comparison to the other approach focus categories.

It can also be stated that research focuses on the evaluation of tools and techniques related to architecture degradation and consistency rather than on the non-technical assessment of these phenomena and their impact in software systems. For example, few studies report on effects related to human aspects, such as the non-resolution of identified architectural inconsistencies, due to behavioural aspects [4]. Although mitigating against architecture degradation touches human-related or cognitive aspects, such as visualization of systems, organizational issues, etc., these aspects seem not to be intensively researched.

Overall, this means that findings on causes of degradation in practice, on its prevalence, and on costs and benefits of consistency checking are under-reported. Although technical evaluations are prevalent, it must be stated that 34 of 78 articles with technical evaluations are solution proposals which, especially when validating through logical argumentation only, provide very limited evaluation regarding the practicality of the applications. Hence, regardless of the apparent focus of the research so far, there seems to be potential for further improvement, even with regard to technical evaluation of software architecture consistency approaches.

4.2 RQ2: Empirical Evidence

The state of empirical evidence in the research field must be seen as critical. Almost 50% of studies are solution proposals that either argue logically for the usefulness, applicability, or the like, of the presented contribution, or provide small synthetic examples for illustration and/or validation purposes. The generalisability of the validations must be judged on a case-by-case basis. However, the average external validity of these types of studies is lower than for other forms of evidence-based research, especially since size and complexity of systems are considered to play a crucial role in the context of software architecture degradation. The external validity of smaller example systems must be critically investigated, as well as potential bias towards the contribution to be validated. Looking at the most recently published studies between 2011–2016, a slight shift to empirical studies can be noted: we categorized 24 of the papers published before this period as solution proposal whereas 21 were empirical. For papers published in and after 2011, this changed to 30 solution proposal papers and 35 empirical evaluations.

Although case studies usually provide detailed insights into the studied subjects, their almost “monocultural” usage in the field of architecture degradation and consistency checking bears challenges. First, case studies are considered problematic w.r.t. external validity because of their deep-but-narrow look at the case under study and the potentially blurred boundaries between the studied phenomenon and its context[16]. This means that we must have a critical look at the generalizability of results in this paper’s area of interest. Second, the large majority of case studies are performed as evaluations of design & creation research with a technical focus, i.e. they provide evidence that a developed technique or tool is applicable/correct/well-performing in a single or a small number of contexts. In these cases, but also in others, the system which the technique or tool is applied to is part of the context and often not easily available for replication studies, especially in case studies involving commercial, closed source software systems. These lessens the ability of the researcher to report openly on detailed findings (e.g.,...
examples of source code contributing to degradation) and does not facilitate other researchers testing alternative approaches in the same setting. Furthermore, few case studies are longitudinal studies and, given the temporal aspects of architecture degradation, longitudinal studies are essential.

There is a large potential for surveys to strengthen external validity of results and experiments to clearly identify cause-effect relationships in architecture degradation and consistency checking research. Particularly the under-researched aspects of the impact of degradation/consistency checking and degradation assessment would benefit from these studies in order to obtain a “big picture” of degradation prevalence in practice. One reason for the lack of these studies, and a challenge to perform them in the future, might be related to the temporal aspects of degradation as relevant and significant effects and impacts might require long-running experiments which are costly and time-consuming to execute.

4.3 Recommendations for Future Research

Based on the preliminary findings of this study, we recommend addressing the following four areas of research that are currently lacking attention:

- More empirical studies: Almost 50% of the papers are solution proposals, meaning that only slightly more than 50% of the papers in this field provide any substantial empirical evidence on the efficacy of the approaches proposed.

- More empirical studies beyond technical assessments are required: Technical research has been, and still is, a dominant part of published articles whereas non-technical aspects, such as usability and user perception, analysing and assessment of architecture degradation and its impact in practice are under-represented.

- More empirical studies beyond case studies are required: The strong focus on case studies may be a threat to the generality of the results in this research field. More experiments and surveys could improve validity and provide more evidence regarding cause-effect relationships related to architecture degradation.

- More longitudinal studies are desirable: Most case studies performed to evaluate techniques and tools are snapshot studies at a single point in time. As degradation is a potentially long-running process, historical and longitudinal studies are desirable.

- Design of studies for replicability and comparability: in order to compare approaches, improve them and focus future efforts to the most relevant aspects, more studies need to be designed for easier replication and easier comparison of approaches. Particularly a repository of “ground-truth architectures” of open source systems seems desirable, allowing the community to compare research results and to report openly on findings.

4.4 Validity and Reliability

Validity refers to the extent to which a claim or conclusion is justified [15]. Reliability is a related concept and refers to the data gathering and analysis being consistent, meaning that measuring the same thing twice will return the same results [15]. This section assesses potential validity and reliability issues of the study and describes the steps taken to address them.

We follow the validity categorization scheme described by Brewer [3]. Construct validity refers to the degree to which the measurements performed relate to the studied phenomenon [15]. In this study, the studied phenomenon is focus of, and empirical evidence in, published research articles on software architecture degradation and consistency checking. The measurement is the sorting of articles into the systematic map as explained in Section 2 and Section 3, based on a keywording of abstracts. We often noticed, in line with existing studies, that abstracts often lacked information regarding our categorization scheme and were misleading at times [6]. Mendes, for example, noted a high number of papers designated incorrectly, when using terms incorrectly (for example suggesting a specific research strategy, such as “experiment” [14]). However, we were aware of this potential issue and adapted the reading depth of papers if the content of a paper’s abstract did not provide sufficient information for categorization, as suggested by Petersen et al. [18].

External validity refers to the degree to which the results of a study can be generalized to a wider population [15]. In this study, this refers to how far the focuses and gaps of the research articulated in the reviewed articles are representative for the overall body of published papers. Two different threats to the external validity in this case exist. First, the set of papers identified for classification might not be representative because irrelevant papers might have been included or relevant papers might not. By retrieving papers from different literature databases and carefully forming the query strings, we tried mitigating against this risk. However, literature not contained in these databases, e.g. grey literature, was not considered. This risk could be further minimized by applying snowballing techniques [24] which were considered too time-consuming for this mapping study. We will, however, use snowballing to extend the scope of the intended refined literature review based on the papers identified in this study. It can be assumed though that the most significant results and empirically valid studies have been published through peer-reviewed, academic channels, as reflected in the data sources we used. Second, the mapping study is ongoing such that the results presented and conclusions made are preliminary and hence potentially not valid for the full set of papers ranked as relevant. We will mitigate against this threat by completing and refining the mapping study in the near future and expect the main preliminary results—dominance of the technical evaluations, solution proposing research and case studies—to be confirmed and refined.

Internal validity refers to the degree to which a study establishes that a factor causes a certain effect and that it is only that factor that causes the effect [15]. As this study does not investigate any cause-effect relationships, we do not consider threats to internal validity.

As mentioned before, reliability refers to consistency in data gathering and analysis. In order to improve reliability in data gathering, we queried multiple databases of academic publications. Furthermore, the initial screening was independently undertaken by at least two of the authors, coming to an identical judgement for 2,320 of 2,434 papers (95.3%), increasing to 2,400 papers with a decision after involving an independent third vote (98.6%). Only for 34 papers (1.4%) could no consensus be reached. To ensure reliability during
the keywording and mapping process, the initial papers were reviewed by two researchers in the first iteration to ensure a common understanding of the predefined categories. The categorisation scheme was refined and discussed after each iteration, to avoid misunderstandings of the categories.

5. RELATED WORK

A few other studies aiming at structuring or providing overviews of the field of software architecture degradation exist.

De Silva and Balasubramaniam provide a survey and categorisation of techniques and technologies employed to prevent, to detect and to repair architecture degradation [7]. They categorise approaches into three main categories of minimising, preventing and repairing architecture erosion and break them down into different strategies and discuss benefits of mixing different strategies. In contrast to their survey, our study in particular focuses on which aspects have been empirically investigated and to which degree.

Hochstein and Lindvall also provide an overview of technologies developed by researchers to combat architecture degradation [11]. They also investigate various causes of degradation and discuss how to prevent it. Even though they express studying degradation happening over time would be desirable future work, their focus is on technologies and causes, rather than the state of empirical evidence.

Ducasse and Pollet developed a taxonomy of architecture reconstruction approaches which is field closely related to architecture degradation and consistency checking [8]. Again, their overview and taxonomy is technically motivated rather than by the question about the state of evidence.

The literature review most similar to our study is a mapping study on code decay by Bandi et al. [2]. They categorise 30 primary studies regarding their detection approaches (human- or metrics-based) and further refine them regarding several attributes and features. The recommendations regarding future research that the authors give are more detailed and largely technically motivated in contrast to our focus on looking at the empirical evidence. However, our findings are backed by some of their conclusion stating that aggregating information and best practices from empirical evidence and research on relationship between code decay and maintenance effort are required. The lower number (compared to this study) of reviewed studies in [2] is due to the authors’ decision to only include studies that consider the aspect of time, i.e. that research decay in systems for at least two different points in time. The rationale is that degradation/decay is a temporal phenomenon that needs to be researched over time. Although we agree with this argumentation that historical or longitudinal studies are preferable, we find the limitation too strong for the focus of this study as it includes research on architecture inconsistencies as manifestations of degradation in a system at a single point in time.

6. CONCLUSION

This article has presented preliminary results and conclusions from an ongoing mapping study investigating the focus and the state of empirical evidence in software architecture degradation and consistency checking research. Results show that a large part of the research is still missing empirical evaluation and that the focus of empirical research is still put on design & creation research with evaluation through case studies. Given the potential lack of external validity and evidence of cause-effect relationships, there seems to be a huge potential of conducting experiments and surveys. Further recommendations expressed are to also perform evaluations beyond technical aspects. Particularly, the impact of degradation and benefits of consistency checking are not sufficiently researched and confirmed by evidence. In the near future, we will complete this mapping study, refining categories and findings. Specifically, there will be a refinement of the study focus categories and a more detailed analysis of types of systems investigated, followed by a detailed systematic literature review.

Acknowledgements

This work was supported in part with the financial support of the Science Foundation Ireland grant 13/RC/2094.

7. REFERENCES


decay: A pilot study of pattern evolution. In *First
International Symposium on Empirical Software
Engineering and Measurement (ESEM 2007)*, pages

performing systematic literature reviews in software
engineering. Report, Keele University and Durham
University, 2007.

research. In *2005 International Symposium on
Empirical Software Engineering, 2005.*, page 10 pp.,
2005.


[16] B. J. Oates. *Researching information systems and
computing*. SAGE Publications, London ; Thousand

of software architecture. *SIGSOFT Softw. Eng. Notes*,

Systematic mapping studies in software engineering,
2008.

Decay during Continuous Software Evolution and
Impact of 'Design for Change' on Software Architecture*,
pages 119–126. Springer Berlin Heidelberg, Berlin,

*Software architecture : foundations, theory, and

119, 2002.

Requirements engineering paper classification and
evaluation criteria: a proposal and a discussion. *Requir.

[23] C. Wohlin. Guidelines for snowballing in systematic
literature studies and a replication in software
engineering. In *Proceedings of the 18th International
Conference on Evaluation and Assessment in Software