

Systematic Literature Review of Architecture Recovery through Dynamic Analysis

Leonardo Humberto Silva

May 2014

Agenda

- Introduction
- Research question
- The need for a review
- Developing the SLR protocol
- Conducting
 - Literature Search Process
 - Data Extraction and Synthesis
- Ongoing work / next steps
- References

Introduction – Motivation

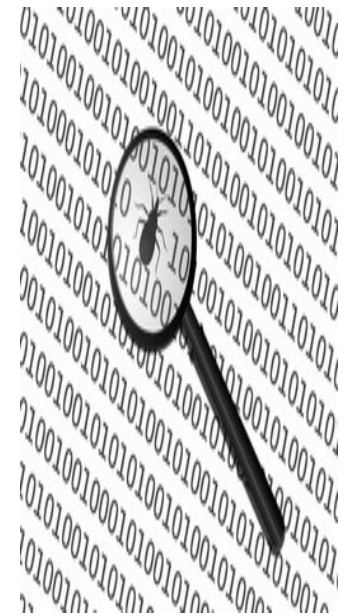
- **Architecture Erosion**
 - Software architecture erosion designates the progressive **gap** normally observed **between** the **planned and the actual architecture** of a software system as implemented by its source code

**Stop the Software
Architecture Erosion**



Introduction – Software Analysis

- The use of **more than one view** allows to **analyze separately different features** that interest to those involved with the quality of a system
- **Dynamic analysis**, or the analysis of data gathered from a running program, has the potential to provide an accurate view of a software system because it **exposes the system's actual behavior**



Introduction – Technical Challenges

- Any approach that supports dynamic discovery of architectures must address three challenges
 - **Monitoring:** observing a system's runtime behavior
 - **Mapping:** interpreting that runtime behavior in terms of architecturally meaningful events
 - **Architecture building:** representing the resulting architecture

Research Question

- The **goal** is to analyze specifically the use of dynamic analysis to generate new architectural views of systems
- Main question: can **dynamic analysis** provide new architectural information that “escapes” from **static analysis**?



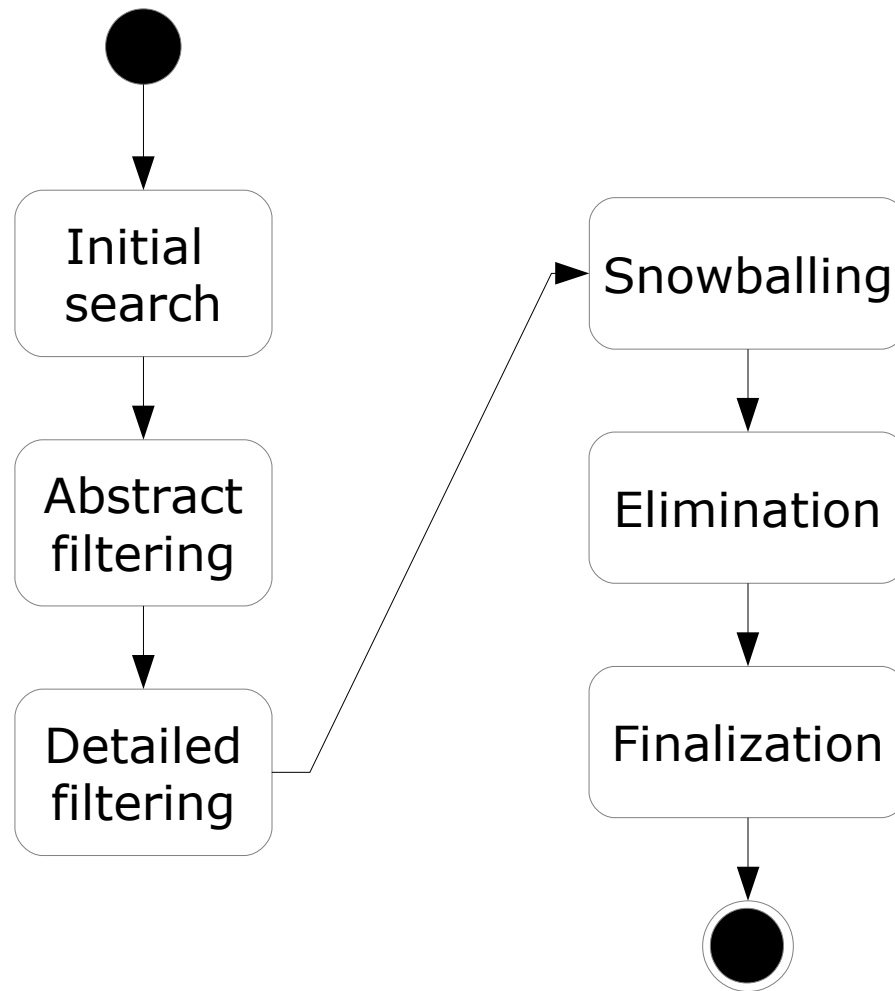
Identification of the need for a review

- Main related work identified
 - **Cornelissen et al.** (2009) reported a SLR aimed at the identification and structuring of research on program comprehension through dynamic analysis. They selected 172 articles between July 1999 and June 2008.
 - **Kitchenham et al.** (2010) provided an annotated catalog of SLRs available to software engineering researchers and practitioners. They do not mention any specific SLR related to architectural recovery through dynamic analysis.

Developing the SLR protocol

- The protocol is important to structure the work and to avoid bias
- The protocol should be peer-reviewed
- Search strategy
 - Search strategies in SLRs often use keywords search in papers abstracts
 - Brereton et al. (2007) pointed out that, in comparison to other disciplines, **the standard of abstracts** in software engineering publications **is poor** (“*snowballing*”)

The Literature Search Process



The literature search process proposed by Szvetits and Zdum (2013)

Snowballing Process

- The bibliographies of every selected publications are checked for **useful articles** that are missed in the initial search
- **Useful articles** = articles from well-known conferences and journals and with a considerable number of citations
- Snowballing is performed until a convergence is reached and no more new relevant articles are found



Literature Selection Criteria

- Inclusion criteria
 - Articles found in the ACM Digital Library or IEEE Xplore Digital Library
 - Articles found through snowballing
- Exclusion criteria
 - Doctoral dissertations and Master theses
 - Books, magazines, web sites and technical reports
 - Surveys and Systematic Literature Reviews
 - Publications without abstracts
 - Publications that contain models not related to software development (e.g. mathematical models)

Search Phase (search strings)

- **P** = {dynamic analysis, dynamic architecture, event based architecture, event-based architecture}
- **Q** = {dynamic, dynamically, runtime, run-time, run time}
- **R** = {static analysis, static view, static based, static-based}
- **S** = {architecture recovery, architecture discovery}

$$\mathbf{Result} = \mathbf{P} \vee (\mathbf{Q} \wedge \mathbf{R} \wedge \mathbf{S})$$

* {P,Q} => abstract only

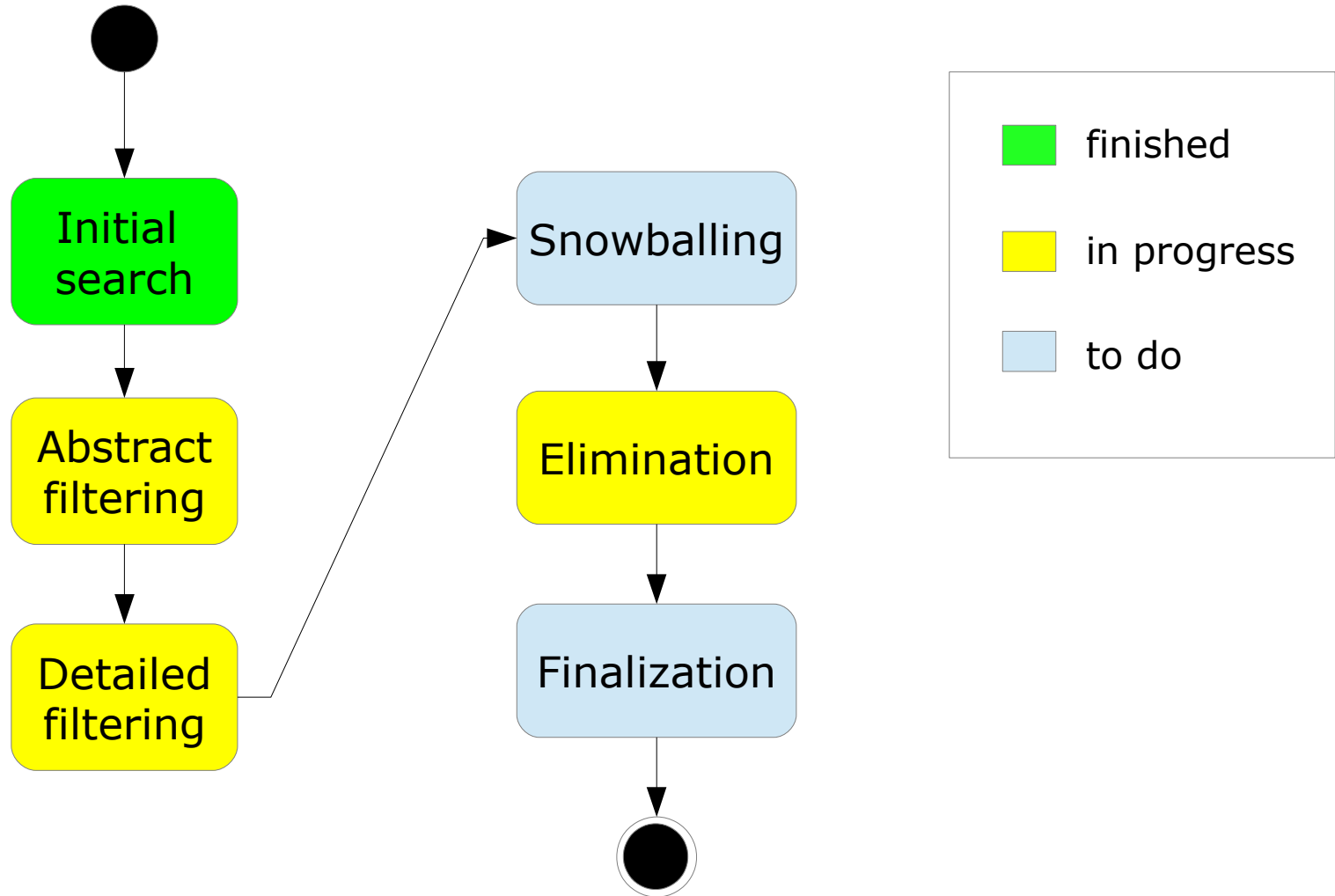
Search Phase (initial results)

String	Digital Library	Number of Results
“dynamic analysis”	ACM	289
	IEEE	1327
“dynamic architecture”	ACM	6
	IEEE	61
“event based architecture”	ACM	1
	IEEE	8
“event-based architecture”	ACM	1
	IEEE	9
Abstract:dynamic AND “static analysis” AND “architecture recovery”	ACM	24
	IEEE	26
Abstract:dynamic AND “static analysis” AND “architecture discovery”	ACM	18
	IEEE	0
⋮	⋮	⋮
⋮	⋮	⋮

Data Extraction and Synthesis

- Once the list of primary studies is decided, data can be extracted and analyzed
- Expected outcomes (*work in progress*)
 - **Quality assessment:** classification of papers based on the technical challenges (monitoring, mapping and architecture building)
 - **Data synthesis:** answers to research questions
 - **Analytical synthesis:** numerical values (# of papers, date of extraction, publication venue, etc)
 - **Lessons learned**

Ongoing work / next steps



References (cited in this presentation)

- [Brereton et al. 2007] Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., and Khalil, M. (2007). **Lessons from applying the systematic literature review process within the software engineering domain.** J. Syst. Softw., 80(4):571–583.
- [Cornelissen et al. 2009] Cornelissen, B., Zaidman, A., van Deursen, A., Moonen, L., and Koschke, R. (2009). **A systematic survey of program comprehension through dynamic analysis.** IEEE Trans. Software Eng., 35(5):684–702.
- [Kitchenham et al. 2010] Kitchenham, B., Pretorius, R., Budgen, D., Pearl Brereton, O., Turner, M., Niazi, M., and Linkman, S. (2010). **Systematic literature reviews in software engineering – a tertiary study.** Inf. Softw. Technol., 52(8):792–805.
- [Szvetits and Zdun 2013] Szvetits, Michael and Zdun, Uwe. **Systematic literature review of the objectives, techniques, kinds, and architectures of models at runtime.** Springer-Verlag Berlin

Thank You!