Improving the Usability of an Integrated Decision Support System for Design Decision Making

Jiping Xu, 26.02.2018, Kick-off Presentation

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Outline

Motivation

Research questions

Approach

Current state

Timeline

© sebis

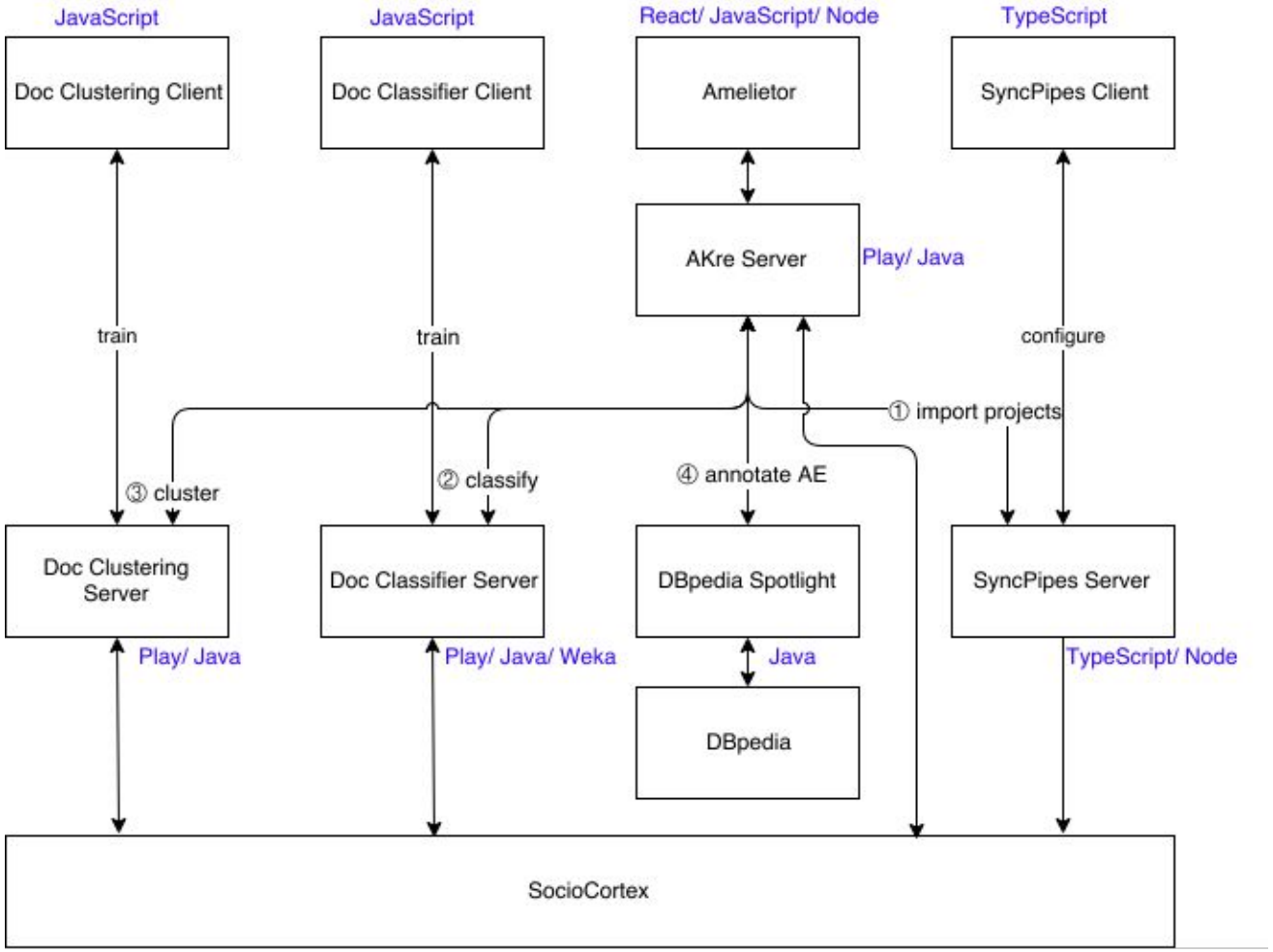
Motivation

- Software architecture is the result of a set of architectural design decisions [1].
- These design decisions involve important choices on core components to satisfy and balance stakeholders' concerns [2].
- Benefits of capturing design decisions: lower costs of change, reusability [3], …
- Difficulties of making design decisions: not rational, biased [4], …

Design decisions are important to the success of the software, but it is difficult to make good decisions. So there is a need for a system to support architects and developers in the process of design decision making - Architecture Management Enabler for Leading Industrial software (AMELIE).

Motivation - AMELIE

- A system of systems
- Developed by different people at different time



© sebis

Research questions

RQ1: What are the requirements of a decision support system (DSS) for making design decisions?

- A rather new field, various requirements from researchers
- Industry partner - Siemens AG: real-world use cases

RQ2: How to improve the *usability* of an existing DSS that supports design decision making?

- Usability (end-users)
- Maintainability (developers)
- Understandability (end-users, developers)
- Learnability (end-users, developers)

© sebis

Approach

Phase 1

- Conduct literature review
- Familiarise with AMELIE, set up all relevant sub-systems
- Artifact: a list of requirements

Phase 2

- Perform usability test to get end-user feedback about existing AMELIE UI
- Implement UI changes
- Integrate systems

Phase 3 (see next slide)

- Create end-user manual
- Create developer documentation
- Create API reference

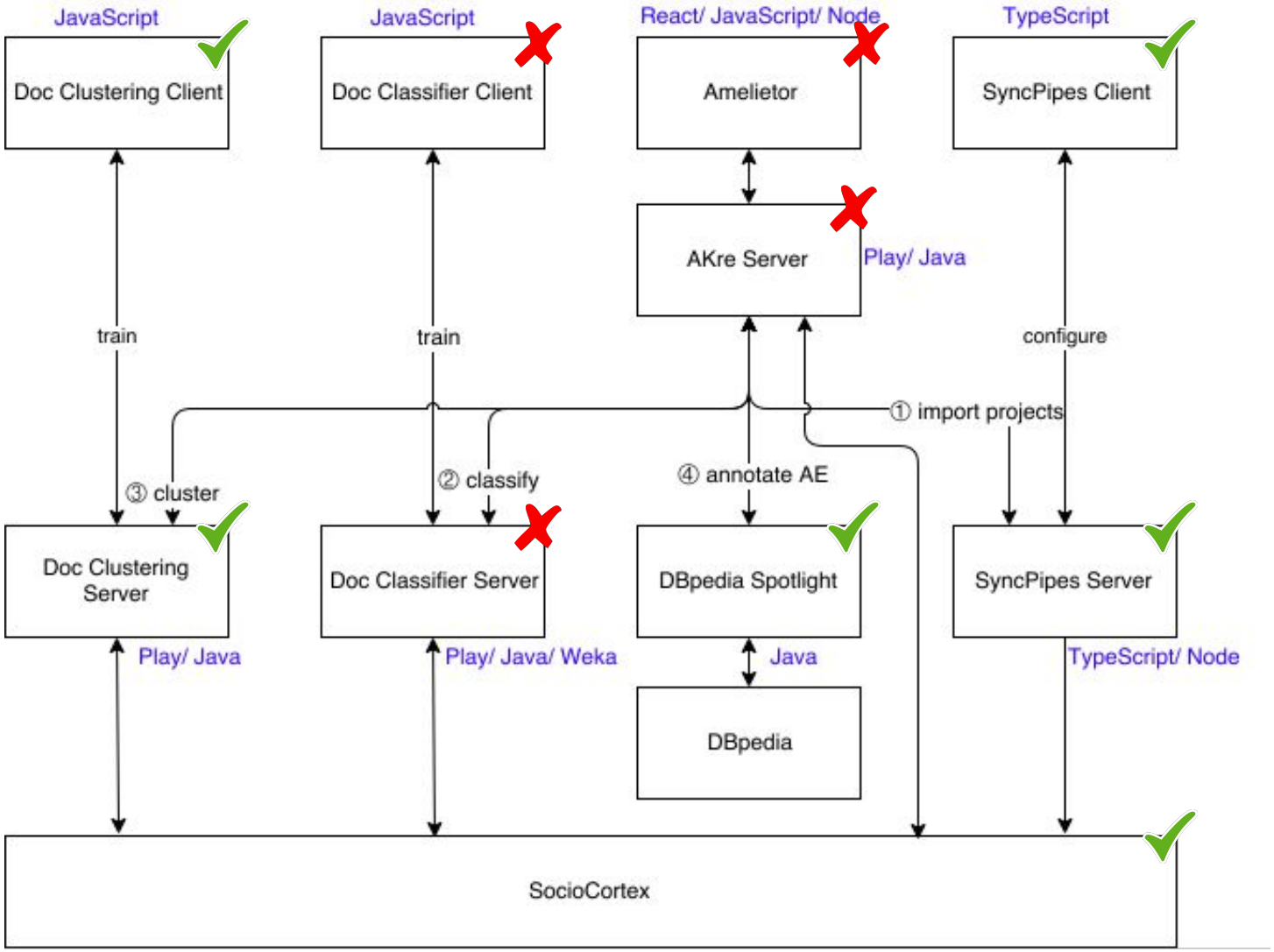
© sebis

Approach - cont'd

✓: with documentation

✗: without documentation

Phase 3

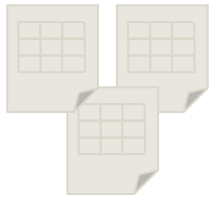


© sebis

Current State

Phase 1 completed

- Select literatures
- Collect requirements
- Remove duplicates
- Refine requirement descriptions
- Share with project manager/architect/researcher independently
- Consolidate results from them



© sebis

Current State - cont'd

- Artifact: a list of requirements
  - 31 requirements related to architectural knowledge management
  - 15 requirements focused on architectural design decisions

Example:

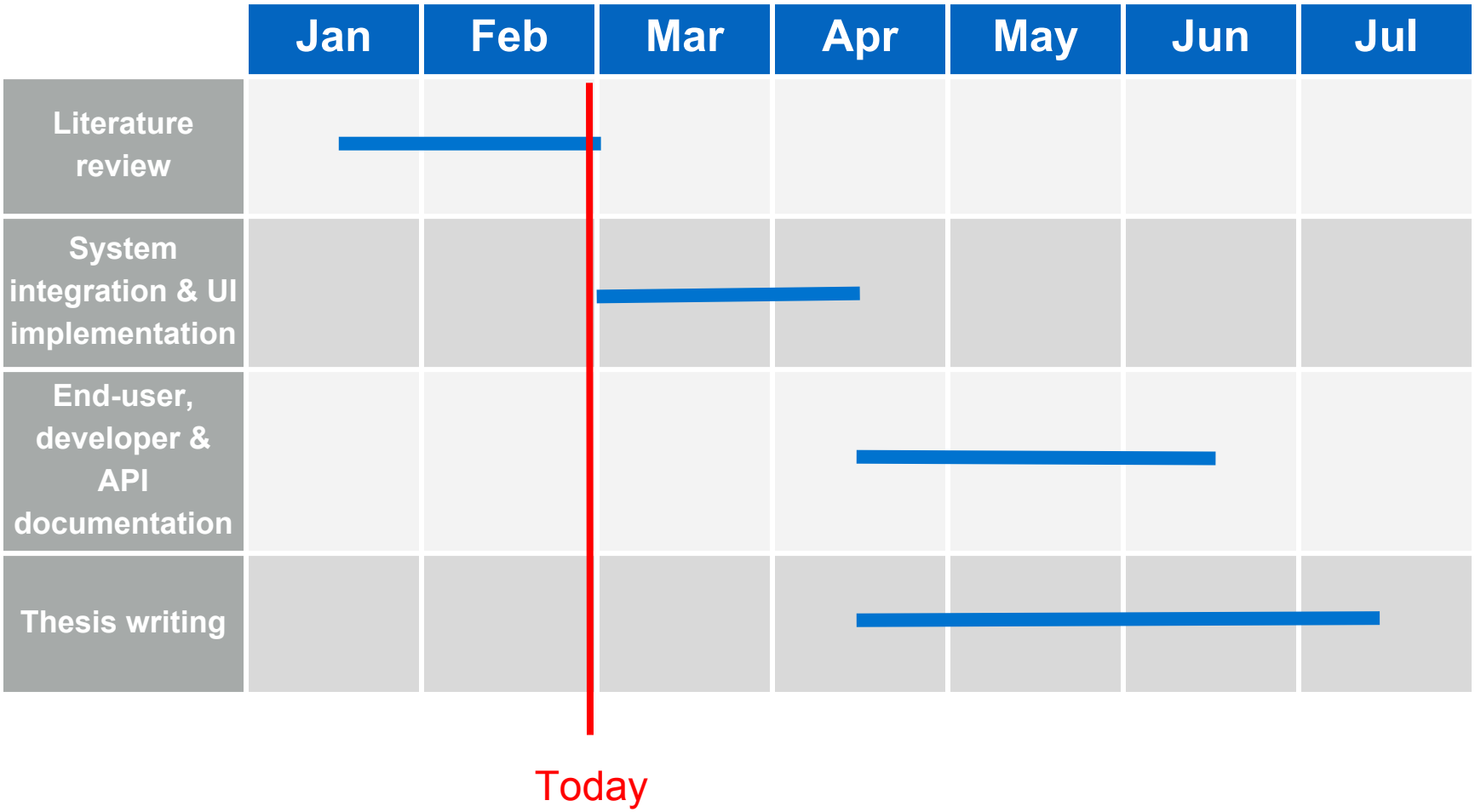
No.	Description	Siemens*	AMELIE**
1	Integration with other tools [6, 7, 10]: automatic or semi-automatic integration with others tools during different phases of software lifecycle.	Y	Y
2	Reasoning capability [7]: basic inference mechanism that can automatically detect inconsistencies, state new questions or perform some fuzzy reasoning.	Y	-
3	Architectural decision consequence analysis [8, 9, 11]: the main consequences of a decision are the changes in the model when a decision is executed. In addition, new decision topics can be introduced.	Y	P

* Is this a real-world use case from our industrial partner?

** Is this already supported by AMELIE?

© sebis

Timeline



© sebis

B.Sc.

Jiping Xu

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for
Business Information Systems

Boltzmannstraße 3
85748 Garching bei München

Tel +49.89.289.17132
Fax +49.89.289.17136

matthes@in.tum.de
wwwmatthes.in.tum.de

References

[1] Jansen, A., & Bosch, J. (2005). Software architecture as a set of architectural design decisions. In Software Architecture, 2005. WICSA 2005. 5th Working IEEE/IFIP Conference on (pp. 109-120). IEEE.

[2] Zimmermann, O. (2011). Architectural decisions as reusable design assets. IEEE software, 28(1), 64-69.

[3] Tofan, D., Galster, M., Avgeriou, P., & Schuitema, W. (2014). Past and future of software architectural decisions–A systematic mapping study. Information and Software Technology, 56(8), 850-872.

[4] van Vliet, H., & Tang, A. (2016). Decision making in software architecture. Journal of Systems and Software, 117, 638-644.

[5] Bhat, M., Shumaiev, K., & Matthes, F. (2017, September). Towards a framework for managing architectural design decisions. In Proceedings of the 11th European Conference on Software Architecture: Companion Proceedings (pp. 48-51). ACM.

[6] Tang, A., Avgeriou, P., Jansen, A., Capilla, R., & Babar, M. A. (2010). A comparative study of architecture knowledge management tools. Journal of Systems and Software, 83(3), 352-370.

[7] Capilla, R., Jansen, A., Tang, A., Avgeriou, P., & Babar, M. A. (2016). 10 years of software architecture knowledge management: Practice and future. Journal of Systems and Software, 116, 191-205.

[8] Van Der Ven, J. S., Jansen, A., Avgeriou, P., & Hammer, D. K. (2006, July). Using architectural decisions. In Second International Conference on the Quality of Software Architecture (Qosa 2006).

[9] Jansen, A., Van Der Ven, J., Avgeriou, P., & Hammer, D. K. (2007, January). Tool support for architectural decisions. In Software Architecture, 2007. WICSA'07. The Working IEEE/IFIP Conference on (pp. 4-4). Ieee.

[10] Regli, W. C., Hu, X., Atwood, M., & Sun, W. (2000). A survey of design rationale systems: approaches, representation, capture and retrieval. Engineering with computers, 16(3-4), 209-235.

[11] Tang, A., Jin, Y., & Han, J. (2007). A rationale-based architecture model for design traceability and reasoning. Journal of Systems and Software, 80(6), 918-934.