

Dan Tofan, Matthias Galster, Paris Avgeriou
 Software Engineering and Architecture Group
 Department of Mathematics and Computing Science
<http://www.cs.rug.nl/search/>

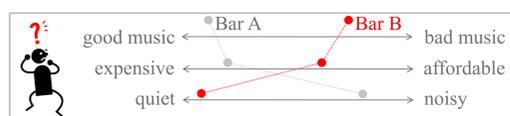
Introduction

Problem: The software architecture is the result of a set of architectural design decisions [1]. Such decisions and their rationale are an important part of the architectural knowledge (AK) of a system [2]. Such knowledge is often lost, resulting in problems like increased maintenance costs and design erosion [7].

Claim: The architecture community may benefit from utilizing knowledge capturing approaches from other disciplines. Thus, we propose to use an inter-disciplinary technique for capturing tacit architectural knowledge.

Repertory Grid Technique (RGT)

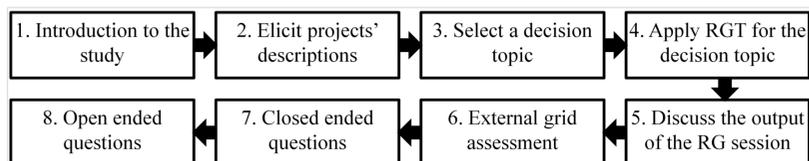
According to the personal construct psychology, people create representations of their experiences in their minds [5]. For example, a person who needs to choose a bar, may use contrasting “poles” (e.g., quiet vs. noisy) to characterize elements (bar A and B):



Empirical Study Design

RQ: What are the advantages and disadvantages of the Repertory Grid Technique for capturing architectural knowledge?

We used seven subjects. All were undergraduate or graduate students at the University of Groningen. We performed these steps with each subject:



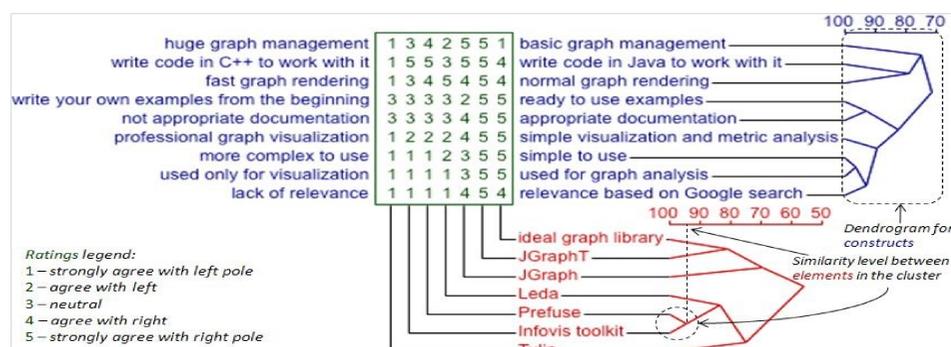
Decision Topics Explored with RGT

The outcome of the first three steps of the study is summarized below. For each of the seven subjects, the table contains their degree program, a short description of the project and its type. The last column has the decision topic we explored with RGT.

ID	Study	Project description	Project type	Decision Topic
1	Ph.D.	Network metrics for power grids	Academic	Graph theory library
2	B.Sc.	Visualization of architectural decisions	Academic	Data visualization tech.
3	Ph.D.	Support medical disease treating	Academic	Features for the first release
4	B.Sc.	Same as ID 2	Academic	Database selection
5	M.Sc.	Smart Grid application	Course	Main hardware
6	M.Sc.	Same as ID 5	Course	Main hardware
7	Ph.D.	E-banking reporting system	Industrial	Reporting engine

Grid Analysis

The RGT sessions (step 4) took on average 57 minutes. As an example, the figure below shows the output for the session of ID1, as analyzed by WebGrid [6].



ID1 perceives Prefuse and Infovis as around 95% similar, with regard to the ratings of the criteria (i.e., constructs) used to evaluate each option. If one regards two elements as very similar, then one may select any of the two, when considering a decision topic.

Advantages and Disadvantages of RGT

The main **advantage** of RGT is *reasoning support* for architects. RGT was characterized as *systematic* (quoting a subject: ‘very professional’), *offering new insights*, *reflective* and providing *decision support*. Another advantage is *readability*, by offering a concise (‘much information with less text’) and a graphical output.

Regarding its **application context**, RGT seems better suited for *big projects* instead of small ones. RGT is useful at the *beginning of the project*, to help in the decision making. RGT output is useful to software architects, but less to testers.

The main **disadvantage** concerns *effort*. As a new approach, RGT has a *learning curve*. It may be *straining*. RGT lacks user-friendly *tool support*, and it is *time consuming*.

Conclusion

We propose to capture architectural knowledge, by applying an inter-disciplinary approach. We identified advantages and disadvantages of the technique. RGT offers good reasoning and decision support to architects. As future work, we will apply RGT in the industry.

References

- [1] Bosch, J. Software architecture: The next step. In *Proceedings of First European Workshop on Software Architecture (EWSA 2004)*, (2004), 194–199.
- [2] Babar, M.A., Dingsoyr, T., Lago, P., and Van Vliet, H. *Software Architecture Knowledge Management: Theory and Practice*. Springer-Verlag New York Inc, 2009.
- [3] Clements, P., Garlan, D., Bass, L., et al. *Documenting software architectures: views and beyond*. Pearson Education, 2002.
- [4] Fransella, F., Bell, R., and Bannister, D. *A Manual for Repertory Grid Technique*. Wiley, 2004.
- [5] <http://gigi.epsc.ucalgary.ca:2000/>, WebGrid 5, accessed on Nov. 2010.
- [6] Jansen, A., Bosch, J. and Avgeriou, P., Documenting after the fact: Recovering architectural design decisions, *Journal of Systems and Software*, Elsevier, Volume 81, Issue 4, pp.536–557, 2008.

Acknowledgements

This research has been partially sponsored by NWO SaS-LeG, contract no. 638.000.000.07N07. We thank Tim Menzies and the study participants for their help.