CS590L **Paper Review**

Dhananjay Indurkar **di5x7@umkc.edu**

Evaluating Software Architecture for Real-Time Systems

R. Kazman, M. Klein, P. Clements - Software Engineering Institute, Carnegie Mellon University

Summary of the Paper

The paper introduces the Architecture Tradeoff Analysis Method (ATAM) that can be applied towards analysis and evaluation of software systems. ATAM is inspired from the software Architecture Analysis Method (SAAM) as well as peer-communities on quality attributes and the concept of architectural styles. With ATAM, the goal is to analyze a system vis-à-vis its quality attributes; establishing the functional correctness not being the objective here.

The paper is grouped into three sections – architecture evaluation concept, technical concepts of ATAM and the formal steps of ATAM. In the first section, the authors emphasize the need for architecture evaluation, the people/group(s) involved, namely the project decision makers, the evaluation team and any or all the stakeholders in the subject system. The evaluation should answer the questions on the suitability of the architecture for the given system and an opinion about which is a better architecture from the set of competing architecture styles. The cost-benefit analysis from a architecture evaluation includes – better communication amongst stakeholders, explicit mention of quality objectives, prioritization of quality goals that are in conflict, a clear and lucid explanation of the system architecture, opportunities for software component reuse and an overall improvement within the architecture itself and its documentation.

Architecture evaluation is based on three types of scenarios – use cases, growth scenarios and exploratory scenarios. Taxonomies are used to establish the requirement attributes. Use of screening questions to guide the evaluation process is a good way to keep the evaluation process focused. The architectural information is represented using different “views” that are derived from the 4+1 View model by Kruchten. Key decisions are identified as sensitivity points and tradeoff points. Application of ATAM helps identify the software quality milestones, key design decisions, any changes to which could affect the quality of the completed system and any tradeoff decision points in the architecture of the system.

The Attribute-based architecture styles (ABAS) approach is a useful methodology for architecture evaluation, if the software system architecture doesn’t show characteristics that could suggest use of a different evaluation technique. Qualitative analysis heuristics are derived from the ABAS to reach a coarse grained result that can be derived at an early stage.

The final section of the paper describes the ATAM steps. A full execution cycle typically requires three days and the activities carried out include scenario and architecture elicitation, mapping of the scenarios onto the system architecture and analysis. The author has given break-down steps and activities that are carried out each day. Also the relative emphasis of each activity is mentioned. The elicitation and mapping steps help delineate the ATAM activities and obtain the outputs that are used in the analysis phase. The dependencies amongst the ATAM outputs are shown in a graphical model. An activity schedule for each day of ATAM evaluation is charted out. Finally, an ATAM example evaluation is carried out.

Strengths

- Good breakdown of the ATAM evaluation activities.
- The BCS example system for ATAM evaluation helps understand the concept and shows how well structured the analysis process becomes.
- The ATAM method helps evaluate large and complex real-world software system architectures and gives valuable insight that helps make the system even better and truly functional.
- Building a repository of scenarios and other key parameters is also helping to standardize the process and make it a routine practice that does not consume great resources but gives very much valuable output.
- ATAM evaluation gives idea about the suitability of the system architecture on multiple dimensions of performance, reliability and availability, security and more.

Weakness

- Incomplete/partial details for the ATAM evaluation example.
- Relationship between ATAM and ABAS models not fully described.
- Though the paper outlines the ATAM steps, it is neither a sufficient nor a complete resource on ATAM evaluation.
- The idea is about evaluating a given software architecture, however a better approach would be to have a methodology that helps build these quality attributes as an integral part of the system architecture and have verification checks at each design milestone.

Interesting Points

- With ATAM, we not only find out how well the system meets the quality goals but also what is the mode of interaction between these quality keystones and how they balance or trade-off against each other.

- Use of Attribute-based architecture styles (ABAS) helps guide the architecture pattern towards specific design objectives based on quality attributes. It makes the architectural design robust, standardized and predictable.

Critical Questions

1. How to incorporate the ATAM process for a system that is expected to be largely assembled from different third party components – do we apply the evaluation process to the main design only?

2. Does the process factor in the tradeoffs when the software architecture uses a not so optimal design in favor of the ease of implementation?

3. How well would the evaluation process work with “agile” software development process where the focus is on adaptability to change?