THE CONTRIBUTION OF ARCHITECTURE DESCRIPTION LANGUAGES TO THE EVALUATION OF SOFTWARE ARCHITECTURES

R Bashroush, I Spence, P Kilpatrick, TJ Brown
School of Computer Science, The Queen's University Belfast, Belfast

Key words to describe the work: Software Architecture, Architecture Evaluation, Tools, Architecture Description Languages (ADL)

Key Results: The design (and currently implementation) of the *ADLARS Development Studio* a tool suite that contributes and aids the evaluation process. A list of characteristics an ADL should posses in order to better support evaluation. The restructuring and fine tuning of ADLARS (an Architecture Description Language designed within our research group) to better capture architectures and support evaluation

How does the work advance the state-of-the-art?: The overall outcome of the work (a tool suite, an ADL, and an ADL characteristics to support evaluation) contributes towards a more formal architecture evaluation process benefiting from a formal ADL with the proper tool support. Also our work draws a first attempt in identifying the characteristics an ADL should posses to support architecture evaluation.

Motivation (problems addressed): The architecture evaluation methods are usually informal and manual processes that require advanced skills from both architects and evaluators. The use of a formal ADL (Like ADLARS) for architecture description, with the proper tool suite (Like the *ADLARS Development Studio*) for information extraction and presentation, helps to streamline and partially automate the architecture assessment process.

Introduction

Identifying limitations and mistakes within software architectures at the design stage is often cost-efficient and reduces the overall system's development and marketing time. A number of techniques have emerged over recent years, for assessing both single-systems, and product-line architectures. These techniques do not assume any particular format or language for the description of the architecture. Often however, they do require the ability to extract a range of information from the architecture description. In this research, we looked at the relationships between the features that might be provided by a formal architecture description language (ADL), and the information required for architecture assessment purposes. We also designed a set of visual tools for use within the architecture development and assessment process in order to alleviate and aid the human part of the process.

Research Progress and Contribution

Architecture Trade-off Analysis Method ATAM [1] was the evaluation method used in our experiments, a well-established process that was developed at CMU (and the only available formal and general purpose method at the moment). During the early stages of the research, ATAM was analysed and modelled as a flow of information. Then, potential bottle-necks were identified within the process where an ADL with the appropriate tool support could contribute. For example, one of the potential bottle-necks identified was in the data flow from the architecture team to the evaluation team. This flow is generally not governed by any standard protocol or data format. This could cause a problem if the first team (architects) does not pass all the required information in the appropriate format to the second team (evaluators) as the evaluation team might not always contain domain experts. This is stated in [1, pp. 105] confirming the importance of the clarity and completeness of the documentation of the architecture to the evaluation process.

Possible ways in which a formal language, with an appropriate set of tools, could contribute to the above process, especially with reference to alleviating the bottle-necks were researched.

First, we arrived to a list of characteristics an ADL should posses to be capable of capturing all the necessary information about a candidate architecture. Figure 1 below shows the brief list of the ADL requirements. For a detailed information about this list, please refer to [2].

Second, a tool suite was designed for ADLARS [6] (an ADL developed within our research group and

1. The ability to contain all the necessary information that would enable the user to extract different architectural views of a candidate system (Functional view, Concurrent view, Code view, etc.) i.e. to be capable of distinguishing among the different views.
2. A facility for including textual descriptions within more formal definitions of components and connectors within the architecture. This kind of facility can be helpful for documentation purposes. An example might be information that has to do with non-functional requirements. This kind of textual information embedded within the architecture definition, can be useful at the evaluation stage as it can be extracted and composed to provide ready-made documentation of the architecture.
3. The ability to capture information about the architectural styles being used, and the ability to correlate styles with required quality attributes. Important recent work in this area, has led to the emergence of the concept of Attribute-Based Architectural Styles ABAS [3] and more recently Tactics [4].
4. The application of Use Cases and Scenarios to architecture descriptions is a central feature of ATAM and other evaluation methods. Support for the definition of scenarios and their application to architecture descriptions is therefore an important capability. The Use Case Maps (UCM) notation [5], has attracted much research interest as a way of modelling scenarios.

Figure 1. ADL properties to support evaluation [2]

was used in our experiments). *ADLARS Development Studio* [2], the tool suite, was designed to help in alleviating the aforementioned bottle-necks, a summary of the tools can be found in figure 2 below.

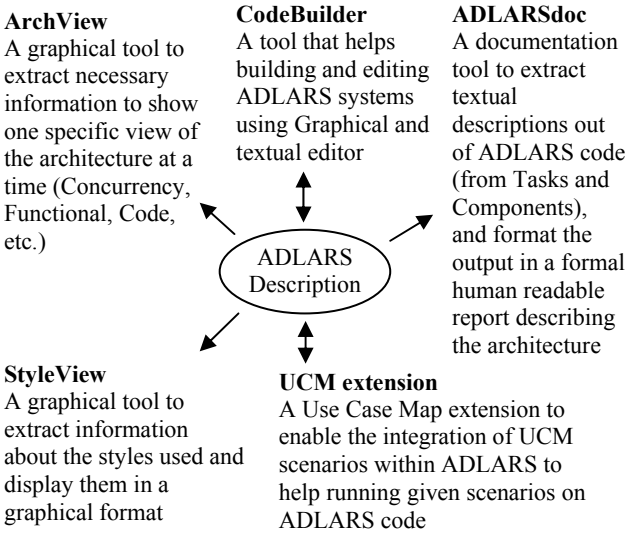

Figure 2. ADLARS Development Studio

The possibility of fully automating the evaluation process was considered, but rejected for two reasons: first, the difficulty involved, and second, the fact that those with experience in architecture evaluation using ATAM stress the benefits of bringing all stakeholders together, i.e. the human component.

Evaluating and Validating Research Outcomes

The validation and evaluation of the research outcomes is an important stage of any research practice. The software architectures for three case studies were designed and descriptions were constructed using ADLARS. This by itself helped in refining and spotting limitations within the language itself. The three case studies are: Floating Weather Station, Network Emulator [7], Load Balancing over Mirrored Web Servers. These case studies are to serve as a test-bed for applying ATAM utilizing the tool suite as soon as it is ready.

Conclusion

My research was concerned with studying the different architecture evaluation techniques to see how these techniques could benefit from the use of a formal language for architecture description, and to identify the features such a language should possess in order to be capture all necessary information. The aim was also to design a set of tools for extracting and presenting this information in an appropriate form. Taken together, the use of the ADL for architecture description, and the tool suite for information extraction and presentation, help to streamline and partially automate the architecture assessment process.

References.

[1] P. Clements, R. Kazman, M. Klein. Evaluating Software Architecture: Methods and Case Studies. SEI series in software engineering. Addison-Wesley 2002.

[2] R. Bashroush, I. Spence, P. Kilpatrick, and TJ Brown. Towards an Automated Evaluation Process for Software Architectures. *Proc. of the IAESTED international conference on Software Engineering SE 2004*, Innsbruck, Austria, 2004.

[3] M. Klein, R. Kazman, L. Bass, J. Carriere, M. Barbacci, H. Lipson. Attribute-Based Architecture Styles. *Proc. of the First Working IFIP Conference on Software Architecture (WICSA1),* San Antonio, TX, 1999.

[4] F. Bechmann, L. Bass, M. Klein, Deriving Architectural Tactics: A step Toward Methodical Architectural Design (CMU/SEI-2003-TR-004).

[5] R.J.A. Buhr, R.S. Casselman. Use Case Maps for Object-Oriented Systems. Prentice Hall, 1996.

[6] T.J. Brown, I. Spence, P. Kilpatrick. ADLARS: A Relational Architecture Description Language for Software Families. *Proc. of the 5th International Workshop on Product Family Engineering*, Siena, Italy, 2003

[7] R. Bashroush, I. Spence, P. Kilpatrick and T.J. Brown. A Real-time Network Emulator: ADLARS Case Study. *Proc. of the 3rd Asia Pacific International Symposium on Information Technology*, Istanbul, Turkey, 2004.