Winter 12-2019

ArchiCap – A tool for capability-based IT architecture exploration

Julian Ereth

ArchiCap – A tool for capability-based IT architecture exploration

Prototype proposal

Julian Ereth, University of Stuttgart

Abstract

IT architects often face a high uncertainty and a lack of methodical support when it comes to architectural decisions in emerging IT environments. Here, a capability-based approach can help to ensure business-orientation and strategic alignment. However, in practice it turns out to be difficult to identify relevant capabilities and adequate architectural possibilities. The prototype presented in this proposal constitutes a web-based software tool to support the exploration of capabilities and link them to architectural decisions. For this, the software allows (i) the definition of environmental setups for certain application areas (e.g. capabilities and architectural possibilities for the internet of things) and (ii) the application to certain use cases (i.e. the analysis of the IT landscape and the exploration of relevant capabilities and architectural possibilities). The prototype is part of an overarching research and was developed and used to explore the area of distributed analytics systems.

Introduction

The goal of a well thought out IT architecture is the provision of an IT landscape that meets current business requirements and that is able to evolve according to the company's long-term vision (Cloutier et al. 2010; Muller 2008). Accordingly, architectural reasoning should follow business requirements rather than technological initiatives. To ensure such a business-orientation, IT architects draw back on Enterprise Architecture Management (EAM) frameworks, like TOGAF (Group 2018), which provide methods to link the business strategy with the technological landscape of a company. Here, TOGAF uses capabilities, as abilities that an organization, person, or system possesses, to derive requirements for the IT landscape. Figure 1 outlines an architectural decision hierarchy according to this understanding.

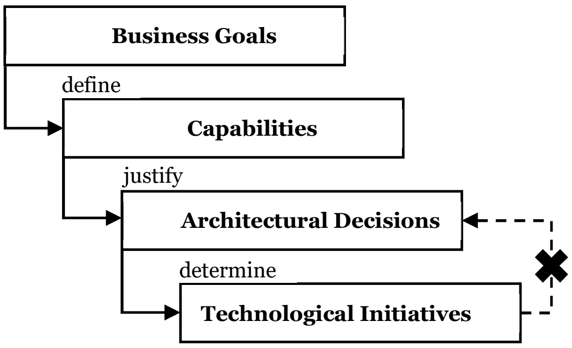

Figure 1. Architectural Decisions Hierarchy (IEEE 2011)

In practice, however, a strict compliance to such a decision hierarchy turns out the be hard, due to an increasing complexity in IT landscapes, rapidly-changing business requirements and fast-evolving technological advancements (Nakagawa et al. 2013). This is especially true for novel areas, like the Internet of Things (IoT) or cloud-native systems, where only little experience exists. Here, solutions often have a very short life-span and there is a lack of blueprints and reference architectures. Consequently, IT architects face a high uncertainty and a lack of methodical support when it comes to architectural decisions in such environments.

The goal of this research is to provide a more flexible way to explore relevant business capabilities and adequate architectural consequences. For this, this prototype serves, firstly, as a tool to gather capabilities and link them to architectural possibilities (exploration), and secondly, helps users to analyze their IT landscape and identify capabilities, architectural possibilities and pain-points in their landscapes (application). The prototype is thereby part of an overarching research and was developed to evaluate the process of architectural decision making described above, and to identify concrete capabilities and architectural possibilities in the emerging area of distributed analytics systems.

Research Context

The prototype is part of an ongoing research that explores the architectural aspects of distributed analytics systems. It was developed as a tool to analyze case studies and derive general-accepted capabilities and architectural possibilities in this area. Hence, the prototype is designed in an abstract manner in order to derive properties of certain artifacts and thereby constitutes a level 2 artifact for design science research (Bækgaard 2015). To derive this prototype, we follow the design science process suggested by Von Alan et al., as we initially used related work to explore the problem relevance and come up with a first prototype that we then iteratively evaluated and improved (Von Alan et al. 2004).

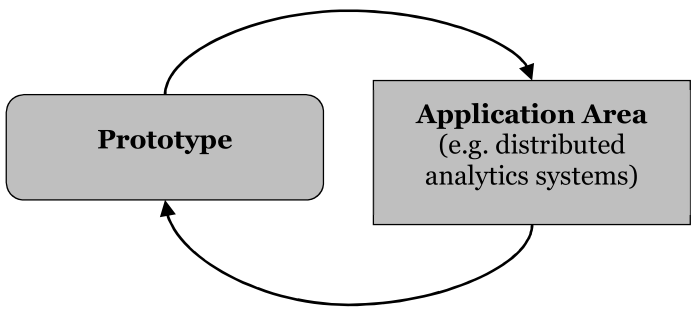

Figure 2. Role of prototype in research process

Figure 2 illustrates the role of the prototype in the research process. It was used to initially explore the application area (distributed analytics systems) by analyzing case studies. In the case studies, we interviewed IT architects and business stakeholders to derive capabilities and architectural solutions. The results were then presented and discussed by means of the prototype. Next to the evaluation of the actual insights about architectures for distributed analytics systems, we also gathered feedback for the prototype to improve the usability and logic of the software.

Prototype Description

The prototype is built as interactive web software using modern web technology (i.e. HTML, CSS and JavaScript). It can be accessed with a usual web browser and is publically available on the internet. The source code was published in a public repository on GitHub™[1]. The prototype comes with some predefined environment setups that can be loaded on demand. However, during the use of the prototype each user has the possibility to customize the environment and define his own setup.

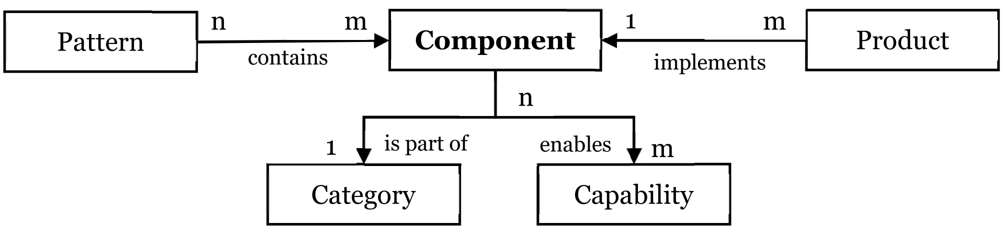

Figure 3. Prototype data model

On a logical level the prototype draws on the data model illustrated in Figure 3. The central element are *components* that represent abstract types software systems (e.g. relational databases or CRM systems). These components are then linked to an arbitrary number of *products*, which constitutes concrete implementations for a component (e.g. MySQL as a relational database). Moreover, a component can be part of an architectural *pattern*, which is a combination of components to provide a certain functionality (e.g. an event hub as a combination of edge processing, message queuing and stream processing). Moreover, it is possible to define *categories* to structure the defined components. Categories can either

[1] https://github.com/JEreth/archiCap

be used to bundle certain types of components, e.g. databases for different types of databases, or to introduce layers in the architecture, e.g. data management or data presentation. Lastly, capabilities are the linking pin to the business. Capabilites are defined as free text elements that are linked to a set of components that are required to enable this capability (e.g. the capability *identify patterns in financial stock data in real time* requires a *complex event processing* component).

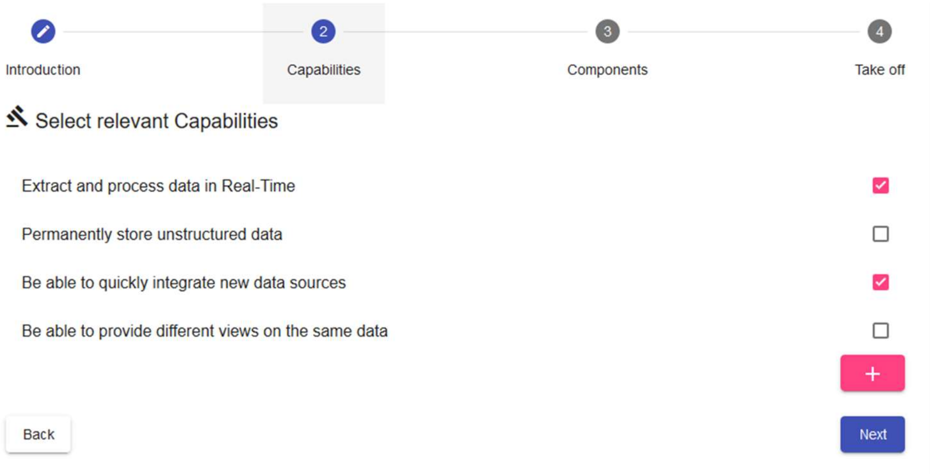

Figure 4. Wizard for easy setup

For a usability perspective the prototype can be divided into two modes. First, there is the application mode, where it is assumed that the environment is set up and a user can simply select and explore relevant capabilities and architectural possibilities that are relevant for the current use case. Here, the users are supported with a wizard (c.f. Figure 4) that helps them to select relevant capabilities and components. In the next step the user selection is visualized on a stage, where all relevant elements are presented in a logical order and the user is able to highlight relations and explore his possibilities (c.f. Figure 5). Moreover, the software also provides some automatic analysis function like the automatic identification of implemented patterns in an IT landscape, i.e. which components are available of certain patterns (c.f. Figure 6).

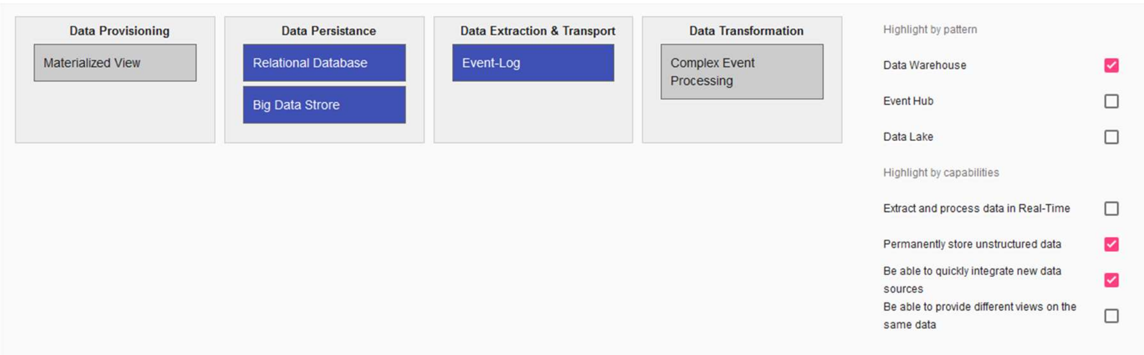

Figure 5. Stage mode to explore architectural possibilities

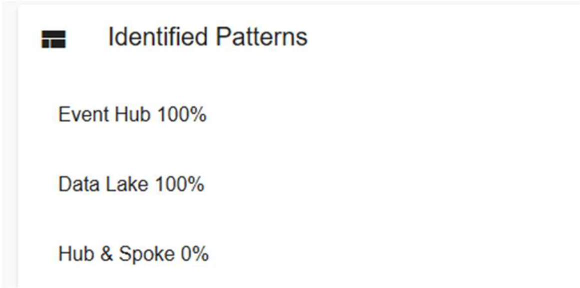

Figure 6. Automatic pattern check

The other part of the prototype focuses the definition of certain environment setups. Here, the user can define capabilities, components, products and patterns that are relevant for a certain application area. This setup is intended to contain as much elements as possibility that can then later be used by other users to analyze concrete use cases. For instance, an environment setup for the IoT might contain various capabilities (e.g. real time analytics, remote control etc.) and related architectural possibilities (e.g. event hub, edge processing, IoT gateways etc.) of which, later, only a few render relevant for a certain use case.

Conclusion and Future Work

The prototype can still be referred to as work in progress, however, it already has proved itself to be useful as it helps to gather capabilities and outlines architectural possibilities in an easy and understandable way. It must be noted that the intention of this prototype is to support an initial exploration of possibilities and it does not raise the claim to generate full-fledged architectural sketches or replace more comprehensive EAM tools.

Currently, the prototype is only used to further explore the area of distributed analytics systems. For future research it can be insightful to adapt the prototype to other emerging areas, like the IoT or Artificial Intelligence systems. For this we suggest to conduct in-depth case studies and use the prototype as a tool to structure and generalize the results. Moreover, feedback from a crowd-based evaluation (Love and Hirschheim 2017) can be helpful as the prototype is publically available and all users can give feedback via the GitHub™ platform. Next to an improvement of the software, we also encourage users to provide new environment setups to make the prototype useful to a larger audience.

References

Bækgaard, L. 2015. "Conceptual Model of Artifacts for Design Science Research," in *Proceedings of the 21st Americas Conference on Information Systems*, Fajardo, Puerto Rico.

Cloutier, R., Muller, G., Verma, D., Nilchiani, R., Hole, E., and Bone, M. 2010. "The Concept of Reference Architectures," *Systems Engineering* (13:1), pp. 14-27.

Open Group. 2018. "*Togaf Version 9.2*".

IEEE. 2011. "Iso/Iec/Ieee Systems and Software Engineering -- Architecture Description," *ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)*), pp. 1-46.

Love, J., and Hirschheim, R. 2017. "Crowdsourcing of Information Systems Research," *European Journal of Information Systems* (26:3), pp. 315-332.

Muller, G. 2008. "A Reference Architecture Primer," *Eindhoven Univ. of Techn., Eindhoven, White paper*.

Nakagawa, E. Y., Becker, M., and Maldonado, J. C. 2013. "Towards a Process to Design Product Line Architectures Based on Reference Architectures," in: *Proceedings of the 17th International Software Product Line Conference*. Tokyo, Japan: ACM, pp. 157-161.

Von Alan, R. H., March, S. T., Park, J., and Ram, S. 2004. "Design Science in Information Systems Research," *MIS quarterly* (28:1), pp. 75-105.