An Approach to Architecture-centric Design and Management in Large Heterogeneous Research Projects

Aneta Kabzeva [1, 2], Markus Hillenbrand [1], Nicolas Repp [2], Paul Müller [1], Ralf Steinmetz [2]

[1]AG ICSY - Integrated Communication Systems
University of Kaiserslautern, Germany
[2]KOM - Multimedia Communications Lab
Technische Universität Darmstadt, Germany
{kabzeva, hillenbr, pmueller}@informatik.uni-kl.de
{kabzeva, repp, steinmetz}@kom.tu-darmstadt.de

Abstract: Architecture-centric software development is a standard in the realisation of industrial software intensive systems. Software architecture is normally defined before the design and implementation of the concrete system components and serves as a guideline during the whole development lifecycle. However, standard architecture-centric development approaches cannot be applied in heterogeneous research projects by implication. The large number of involved partners and the innovative nature implicate much information coordination and many architecture consistency challenges. This paper addresses all challenging aspects with an iterative architecture-centric design and management approach. The approach is based on the experience collected during the architecture design and management process in the research project THESEUS TEXO which is used here as example for the practicability of the offered method.

1 Introduction

The role of the architecture as a determining quality factor in the development of large software systems has already been recognised in last decades [Gar00, IEE00]. Architecture-centric development approaches have been set up as a standard for the realisation of software-intensive products. Thereby the architecture serves as a manual during the whole development process and provides for the compliance with the functional and qualitative requirements on the system.

In industrial software-intensive projects architecture modelling is done preferably directly after the specification of the system goals, requirements, and constraints and before the design and the implementation of the concrete system components. Thus, an evaluation of the architecture permits the inspection of the selected architectural approaches and examination of their effect on the quality of the future system before its realisation. This approach saves a lot of time, correction work and costs [BCK03, Gar00].

The application of an architecture-centric development process within the scope of large heterogeneous research projects puts some challenges. The coordination of a huge number of financially and legally independent project partners from different domains, the spreading of uniform knowledge about the architecture, the consistency protection of the architecture, and the unexplored influence of novel technologies and methods on architectural quality require some modifications in the existent architecture management process.

Purpose of this work is to define an iterative architecture design and management method for large heterogeneous (research) projects that addresses all recognised challenges resulting from the context. The approach should facilitate the co-operation between project partners, grant consistency of the information spread between them, reduce the communication overhead, and ease the propagation of changes to all affected views and documents.

The remainder of the paper is structured as follows. A statement of the challenges in the architecture definition and management process in heterogeneous research projects is included in the next chapter. After a review of the related work in Section 3, Section 4 suggests an iterative approach as solution to the challenging aspects. How the proposed method can be applied to design and manage the architectural process in a heterogeneous research environment is demonstrated in Section 5. The last section summarises briefly the contributions and outlines some future research activities.

2 Fundamentals

This section gives an overview of the aspects considered by the suggested approach. Even if the critical role of software architecture in the development process of software-intensive systems is already recognised [Gar00, IEE00], there exists no uniform definition for architecture. Therefore, how architecture is considered within this work is specified before the discussion of the differences between research and industrial projects and the architecture modelling and management problems that result from these differences.

2.1 Architecture

Since the early 1990s, when software architecture gained its popularity as a software engineering discipline, many definitions were published in research works and from practitioners [SEI09]. A commonality between all these definitions is that architecture should consider the structure and the behaviour of the system it describes. Following the definition of Bass et al. [BCK03] software architecture within the scope of this work is "the structure or structures of a system", which consist of software elements, the relationships among these elements, and the externally visible properties of the elements presented from different viewpoints [IEE00] and on different abstraction levels.

An architecture should reflect all business and technical decisions in a project [BCK03]. It should fulfil the requirements of all stakeholders involved in the project, follow the rules of the development organisation and the organisational structure, and consider the technical environment in which the future system will be applied. Last but not least, the experience of the architects also influences the architectural design and management process.

Based on these influencing factors the rest of this chapter presents the differences between industrial and research projects. Consequently, the architecture design and management challenges in the research field are identified.

2.2 Influencing factors in industry projects

All four influencing factors named in the previous subsection are usually clearly defined in an industrial software-intensive project.

Assessment of system requirements

A customer orders some software by a certain software development organisation. The customer defines the desired functional and qualitative requirements on the system. Customer, developer and user revisit these requirements, discuss what is possible from the developer's point of view and settle together the characteristics of the future system. Thus, the requirements on a system can be examined from the viewpoint of all involved stakeholders.

Organisational structure

Normally, an industrial project is handled by a single development organisation. Such an organisation has a clear organisational structure with a concrete employees' configuration. The employees have a specific knowledge and based on this knowledge the organisation has a preferential development culture. This organisational culture alleviates the choice of methods, techniques, and approaches to be applied in the architecture design and management process. If more than one development organisation is involved in a project, one of them always has a leading function and determines the organisational structure.

Technical environment

The technical environment of a future system in industrial projects is predefined by the customer. Thus, during the design process all technical influences on the architecture and consequently on the future system can be considered.

Architectural experience

Every industrial software development organisation considering software architecture modelling as part of its development lifecycle has an architect responsible for the management of this activity. An architect brings architectural knowledge based on rich experience. Thus, an expert estimation of the best models, methods, and techniques, which correspond to the specified system requirements, the organisational structure, and the available technical environment, can be met in industrial projects.

2.3 Influencing factors in research projects

Contrary to industrial projects, which pursuit the development of software for customers, research projects aim at gaining new knowledge in up-to-date research areas using innovative and up-to-date technologies and methods. This different goal reflects in the architectural influencing factors.

Assessment of system requirements

A software-intensive system which results as an output of a research project is specified by a group of researchers. All stakeholders involved in such a project are representatives of the developers group. The lack of customers and users results in the specification of the system requirements only from developer's point of view[1]. To consider also the other perspectives on the system the researchers have to put themselves in the customer's or user's role. Additionally to the lacking stakeholders, another difference between industry and research is the heterogeneous knowledge of the involved developers. Large research projects often engage experts from different domains. On the one side they bring much valuable knowledge into the project. On the other side the discussion, combination, and coordination of all requirements coming from the different domains makes the assessment process a complex procedure. Furthermore, the innovative nature of research projects allows no complete definition of all requirements before the architecture modelling phase. All innovative approaches and techniques studied in such a project lead to constant addition, deletion, and adjustment of the requirements.

Organisational structure

Large research projects always have a stringent decentralised structure. The research partners come from different domains in which they are experts, have different employees' organisations, use different terminology, and apply different development approaches. As already mentioned, these differences bring a lot of knowledge and agility in the project, but the filtration of the best techniques, methods and approaches for the specific research problem is a cumbersome process. The communication between all partners, the coordination of the information, and the preservation of consistency are much more complex compared to industrial projects. Since all partners in a research project have equal rights and are legally independent, it is important in such an environment to find the best composition of modelling methods and techniques. Each partner should preserve the freedom to apply his own approaches in the own work. Interfaces with other partners should be defined in a consistent way, approved by everyone involved.

Technical environment

Industry projects apply up to date standard technologies and tools to develop a system. The goal of research projects is innovation. Combination of existing standards and implementation of new technologies are the main objective in research environment. This is the main reason why the technology influence on the architecture cannot be estimated a priori in research projects.

[1] Project reviewers can be seen as potential customers in research projects. However, they do not participate actively in the requirements specification process, but only evaluate the researchers' proposals.

Architectural experience

Generally in a research project there are seldom architects who manage the architecture modelling process. Architectural knowledge comes mostly from the different project partners who probably have heterogeneous experience. The selection of an architecture design and management process that combines the best approaches of all partners is a great challenge. This challenge is amplified by the innovation character of research projects. The introduction, usage, and analysis of new approaches, methods, and technologies are the main goal of each research project. The consideration of all these innovative aspects in the architecture and the estimation of their effect on the quality of the future system is an essential difference between research and industrial projects. No architectural knowledge considering the novel aspects is existent.

2.4 Architecture design and management challenges in the research environment

This subsection summarises all challenges identified during the analysis of the differences between industrial and research projects.

Management of the modelling process

The predominant lack of professional architects and requirements engineers on the one side, and the assessment of all partners' interests and knowledge demand a corresponding structure of the architectural team. This need is also amplified by the lack of clear hierarchy and the consequential communication overheads in a research environment.

Complete specification of system requirements

The main obstacle for the definition of requirements, which are needed as input for the architecture design, is the innovative character of research projects. The analysis of new methods and technologies leads to new findings during the whole system lifecycle, which influences the requirements of the system. Consequently, an architecture design and management approach for research projects should offer adjustment of the requirements and respectively of the architecture during the entire lifecycle. The lack of customers in a research environment hampers also the specification of the requirements under consideration of all viewpoints.

Estimation of technology influence on the architecture

The greatest challenge in a research environment for the architecture design and management process is the lack of experience with the innovative technologies examined within the project. All architectural approaches consider the impact of technology on the quality of the future system during architecture design. This is nearly impossible in research projects. Therefore, an architecture design and management approach for research projects should allow a stepwise analysis of the new technologies and their meaning for the quality of the software.

Choice of appropriate architecture-design and management approach supporting parallel operation of all project partners

The heterogeneous structure of research projects and the parallel work of all project partners require an approach that considers the individual interests of each partner, i.e. is not an obstacle for his research process. The coordination of the communication between the partners, comprehensibility of the information for all involved researchers, and consistency of the architectural views should also be granted. The larger the number of involved partners is, the more difficult to identify methods, technologies, and notations satisfying all needs.

3 Related work

The heterogeneous structure of research projects and the challenges specified in the fundamentals section forebode an iterative and incremental development approach for the environment. For this purpose this chapter does not discuss sequential development models [Roy87].

The spiral lifecycle model of Boehm (1988) [Boe88] is the first research work on iterative incremental development. The spiral model envisions the instability of system requirements and the changeable risks, which should be repeatedly analysed. Every iteration of the model specifies the goals, possible solution alternatives, and accomplishes an evaluation of these alternatives under consideration of risk minimisation. Thus a cyclic improvement of the product is achieved. However, even if the model allows a stepwise improvement of requirements and architecture, it does not support the parallel mode of operation in a research project. Detailed design and coding still remain late activities in the development process.

In the late 1990s, Extreme Programming (XP) was introduced by Beck [Bec99] as a software engineering methodology. XP regards the partial change of system requirements and strives for iterative exploring of implementation technologies early in the development cycle. Nevertheless XP focuses on the code production at the expense of requirements and architecture specification.

The Twin Peaks model of Nuseibeh (2001) [Nus01] is a finer-grain spiral model concentrating on the iterative development of requirements and architecture during the software development process. Thus, it is a simplified version of Boehm's spiral model that allows swift reaction on rapid changes. The concurrent evolution of requirements and architecture during the implementation process complements the XP model and addresses the challenges resulting from the heterogeneous nature of large research projects.

The next section presents an approach based on the Twin Peaks model concept. This new approach specifies a concrete requirements and architecture definition procedure aligned with the particularities of a research project's structure.

4 Architecture-centric design and management approach in heterogeneous research projects

An approach for architecture-centric development in research environments has to allow stepwise evolution and refinement of both the requirements and the architecture of the target system to support an incremental analysis of novel methods and technologies, which are the main goal of scientific projects. Expanding the spiral Twin Peaks concept [Nus01] the approach presented in the following defines requirements and architecture specification procedures considering the knowledge existing in a research project. Before presenting the specifications this section proposes a structure of the architecture-design and management team which will have to control the whole modelling process and grant for consistency.

4.1 Structure of the architectural team

An architect defines the system structure. He acts as mediator between all stakeholders. The product of his work, the architecture, provides an explanation of the system's goals for developers and a presentation of the solution structure for customers. There is no such role classification in research environments. As stated in the fundamentals section, a research project seldom involves a professional architect or requirements engineer. Even if this is the case, the architecture modelling team should consider the interests and knowledge of all involved partners coming from heterogeneous domains. The role of an architect in such an environment is not the one of intermediary between customer and developer, but a mediator between the separate project partners and domains.

To address this management challenge this approach proposes an architectural team that includes a representative from each domain involved in the project. Each of these representatives is responsible for spreading the specific domain knowledge between all team members and advancing the view of all partners in that domain on the system. To be able to answer questions on the methodologies and techniques used within his domain, a representative has to have an overview of the most important aspects within the domain, or at least the assignment of the activities to the domain personnel. Additionally, every domain representative is in charge of spreading all decisions of the architectural team between his domain associates and collecting feedback from them.

This team arrangement grants for consideration of all domain specific features and usage of the best existent knowledge during the modelling process. Information spreading is managed in both directions - from the architectural team to all partners and vice versa, without a great overhead for the single person.

4.2 Requirements specification

Detailed and understandable specification of system requirements is an important input for the architecture. The implementation of a system meeting all user requirements necessitates a clear definition of all functional and qualitative properties. As noticed in the fundamentals section the innovative character of research projects and the dynamic development during the whole lifecycle do not allow complete specification of the system requirements before beginning with the architecture modelling phase. To address this challenge this approach inherits the iterative incremental parallel improvement of requirements and architecture from the Twin Peaks model. However, a start set of requirements has to be comprised before modelling the first version of an architecture. In research projects the project proposal is the first specification of the system's requirements. Due to the lack of customers and consumers, the project proposal is also the single requirements source for the researchers. A reasonable result from the analysis of this proposal is a set of scenarios presenting the notion of the researchers on the future system. In a brainstorming session the heterogeneous project partners can introduce their innovative ideas and become acquainted with the ideas of others. The number of involved partners in heterogeneous research projects is large. Even if each partner represents a single scenario it will be a very time consuming and complex process to examine all of them during the architecture modelling phase. For this reason this approach proposes as next step the prioritisation of all collected scenarios based on their comprehensibility and completeness. The best scenario has to be consequently adjusted to the mutually agreed core functionalities. The resulting integrated scenario (Figure 1) is subsequently modelled. This model comprises all initial expectations on the future system which has to be realised with the innovative ideas of all project partners and presents them in a way understandable for all researchers, independent from their domain knowledge.

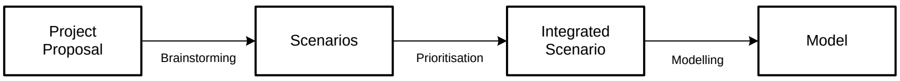

Figure 1: Requirements specification

4.3 Architecture specification

The work planned in a research project proposal is normally assigned to different work packages that together achieve the specified goal of the project. Hence every work package is responsible for a small part of the planned solution. At the end these partial solutions have to be integrated to form the "big picture". Considering this specific structure of research projects, it is impossible to establish and evaluate the architecture of the planned system before the design and implementation of the separate components planned in the different packages. Under these circumstances the architectural team has to consider the package results during the architecture modelling.

Thus, a "meet-in-the-middle" [TF08] approach is most suitable in a research environment, i.e. a parallel top-down and bottom-up definition of the system structure. The architectural team models thereby a top-down view on the architecture based on the integrated scenario. This model should be an abstract high-level view on the system. At the same time every work package has to model a package architecture. These bottom-up package specific views should comprise all components planned for development within the package and the communication between them. Thus, project partners are also given the possibility to present existing own products that should be used within the project and show how they should interact in the new environment. Further on, a package specific view should indicate possible or needed connections to other packages. Hence, the package architectures define requirements on the overall architecture. The overall architecture establishes some restrictions for the package architectures in return.

The knowledge collected from the overall view and the package views has to be consolidated as a next step (Figure 2). Based on the package views the architectural team can assign the functionalities of the system to the separate components and consequently to the corresponding work packages. Functionalities provided in the overall architecture but not considered in any of the package architectures are written down by the architecture team. Double modelled functionalities and components, i.e. considered in two different package architectures, are also detected in this step. Functionalities one package needs from another, but not found in the architecture of the identified provider package, or such functionalities a package plans to offer to another one, but the receiver package has not modelled them as needed in the own architecture, are noted by the architectural team as well. These findings should be discussed with all project partners. After a successful assignment of the missing functionalities to the appropriate components and packages respectively and the deletion of the duplicates, a detailed architectural view should be provided. This view comprises all package specific components and the communication between them. This step allows the architectural team to understand where the separate packages see their responsibilities in the future system. It enables also the identification of all interfaces between the different packages and research partners.

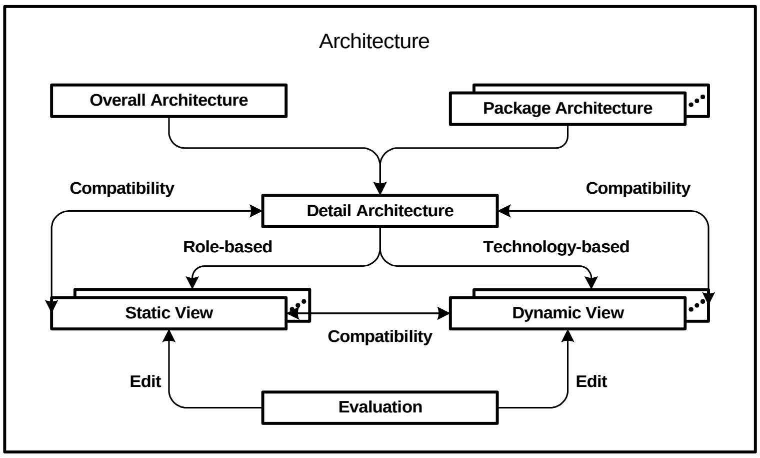

Figure 2: Architecture specification

Most likely the detailed view on the architecture in a large heterogeneous research project will be of high complexity. Aiming at reduction of this complexity and enhancement of communication between the project partners and their comprehensibility on the architecture [BCK03] the approach proposes a decomposition of the detail architecture in role-based static and technology-based dynamic views [IEE00]. For every role that can be assigned to a user during his interaction with the system, there should be a static view on the system showing only the components needed for the role to perform its role-specific functionalities. For every complex process initiated by a role interaction with the system a dynamic view should be modelled showing how the components involved in the execution interact with each other to perform the functionality.

The decomposition of the detail architecture in smaller views improves additionally the identification of new requirements resulting from the architecture and the applied technologies. It helps to spot missing components and to discover possible connections between existing ones. This fragmentation of the architecture supports also the adaptation to dynamic requirements. The connection between role and technology-based views (every process is associated with a role) allows an easy identification of all change relevant candidates, both in the views and in the detail architecture. Thus the consistency of all views can be improved. In an adjustment iteration step every change is propagated to the affected static and dynamic views and to the detailed architecture.

At the end of an adjustment iteration step an evaluation of the architecture can take place. The goal of an architecture evaluation is the identification of risks and the verification of the system requirements in the existent design [DN02]. However, considering the dynamic nature of requirements and the application of novel technologies an evaluation in a research environment will be probably more appropriate at the end of the project.

4.4 Collaboration assistance

A large number of architectural components is normal in the structure of large heterogeneous projects. Every component belongs on the one hand to a concrete work package. On the other hand it fits in a distinct functionality group (e.g. interface, backend component, or data unit). As collaboration assistance the proposed approach offers two colour coding systems for the architectural components.

The detail architecture should be presented in a work package colour coding. A different colour is assigned to every work package in the project and all components developed within a package have to be dyed in the corresponding colour. This colour coding system eases the collaboration between work packages. Every package can easily identify which are the packages he has to collaborate with.

In the static and dynamic architectural views a group based colour coding is proposed. Every component is dyed in the colour defined by its functionality. This colour coding decreases the views complexity. No explicit notification of the component's type is needed. This information is coded in the colour of the component. Thus the views are not overloaded with data, but clearly arranged without any loss of important information.

5 Case study THESEUS TEXO

To show the practical applicability of the proposed architecture-centric design and management approach, this section describes how it was used in the THESEUS TEXO [TEX09, JRS08] application scenario. THESEUS is a large research program that has as goal the development of new Internet-based infrastructure for better usage of the knowledge available on the Internet. The purpose of TEXO as part of THESEUS is to provide a service-oriented architecture-centric [Erl05] platform which allows services on the Internet to be traded and composed in value-added services.

5.1 Architectural team in TEXO

TEXO involves more than 50 researchers coming from 14 universities and companies. They are organised in twelve work packages. Every work package contains researchers from one or more of the partner organisations according to the domain in which they are in action within TEXO. One of the packages is responsible for the management of the project. However all packages work in parallel.

The architectural team in TEXO comprises two researchers from the management work package and one representative from every other package. All these researchers form the so called Architecture Task Force (ATF). Every ATF member is responsible for presenting the ideas of his work package and explaining the methods and technologies used within his domain to the other ATF members. Consequently, all knowledge gathered in the ATF is spread in the own package and feedback is collected. Thus all ATF decisions consider the interests of all researchers and all conclusions made within the ATF are reported to all researchers.

To ease the communication process TEXO uses also a semantic Wiki. All phone calls and meetings of the ATF with the belonging agenda and results are documented there. All relevant documents are administrated on a Web-based cooperation platform.

5.2 Requirements specification in TEXO

The specification process for the initial TEXO requirements followed the steps defined in the proposed approach. Based on the project proposal a brainstorming session with all involved partners took place. Every partner presented scenarios on his picture of the platform - what should be offered by TEXO (e.g. search, publication, composition, purchase of services), in what situations in real life should it be applicable (e.g. to satisfy needs of every person, or to support small and medium-sized enterprises), and how should it be presented to the user (e.g. a Web-based interface, integration in the user's environment, or stand alone application). To avoid passive partners every one of the institutions participating in the project was obliged to present a scenario. Consequently an integrated scenario, comprising the best ideas collected during the brainstorming session was defined. It was modelled as a video sequence showing how a user interacts with the TEXO platform. This model was used as first requirements set in the architecture modelling process. To assure consistent terminology understanding within the partners a TEXO glossary was defined which contains the definitions of the most important terms.

5.3 Architecture of the TEXO platform

The ATF has modelled an overall architecture (based on the integrated scenario model) in the first iteration of the design process. Simultaneously every one of the work packages (except for the management package) has modelled a package architecture. Additionally, the ATF defined a component description template which the researchers had to use to describe the planned components and their behaviour in a uniform way. The ATF has analysed all architectural models and integrated them in a detail architecture. During the integration phase missing or double existing functionalities and such components were identified. After a presentation of these results and the elimination of the discovered weaknesses an architecture with more than 60 components was presented to the researchers. Every one of these components was coloured according to its package affiliation. Thus, the detailed architecture helped the researchers to identify with which other packages they have to collaborate, who wants to use their own components and how.

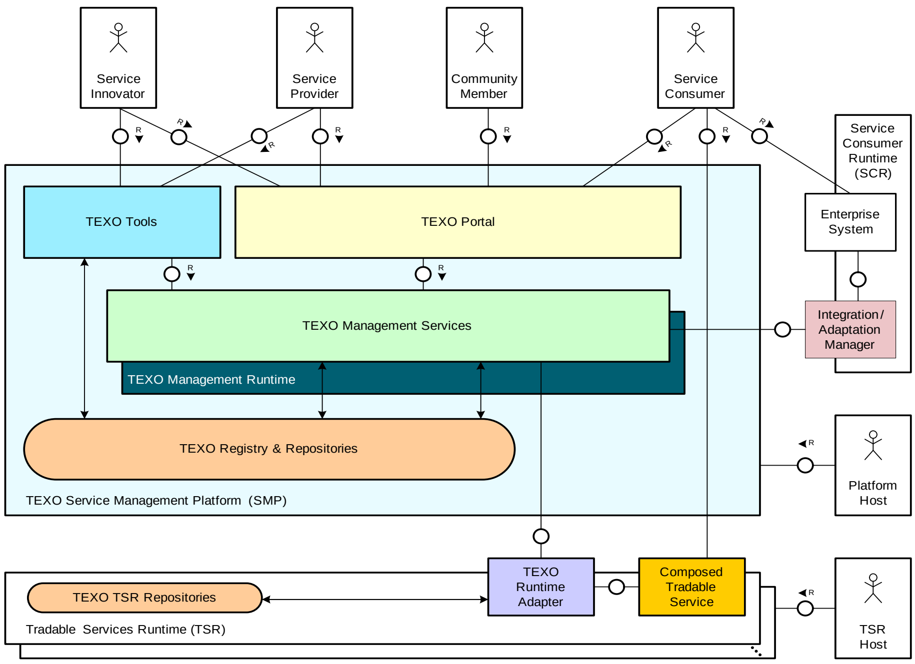

Figure 3: TEXO high-level architecture [KHMS09]

Before defining the role-based and technology-based views in TEXO a simplification process of the detailed architecture was performed. For this purpose, all components from the detail architecture were divided into large blocks according to their functionality and the role they play on the platform (e.g. management services, interfaces, adapters and so on). The resulting model presented a high-level view on the architecture (Figure 3)[KHMS09]. The identified large blocks were coloured differently. Those colours were used later in the static and dynamic views to specify to which block a component belongs.

At the present state the TEXO architecture comprises six role-based static views, one for each of the roles shown in Figure 4. These views show only via which interfaces a specific role interacts with the TEXO platform and which backend components implement their functionality. How a concrete functionality is implemented is presented in the corresponding technology-based views.

Role-based views abstract even from connections between components on the same level (i.e. components from the same block) to keep the views clear and of low complexity. Figure 4 shows an example of a role-based view.

Together with the six static views the TEXO ATF manages seven technology-based views at the moment. These views show how the complex processes by a role interaction with the platform are executed, i.e. which components interact with each other, in which order, and what information they exchange to perform a given complex functionality. E.g. for the Innovator's role from Figure 4, there is a dynamic view showing how an innovation idea is published in TEXO, evaluated by experts, and realised as a service.

The usage of the proposed approach in the THESEUS TEXO project has approved its applicability. The settled ATF as a mediator between the work packages, i.e. the different domains, manages the modelling process in TEXO. Since including a representative of every domain, every design decision met within the ATF considers the interests and knowledge of all partners. The parallel improvement of both requirements and architecture allows a stepwise enhancement of the requirements and analysis of the novel technologies applied in the project. The low complexity of the partial views improves the comprehensibility of the platform by each TEXO partner. The role-based and technology-based distinction makes the communication and change propagation in the project much easier, both on a structural and a behavioural level.

6 Conclusion and Future Work

This paper proposed an approach for architecture-centric development in heterogeneous (research) environments. The challenges resulting from the innovative nature of research projects for existing methods were identified. An architecture-centric development during parallel development in different domains demands a special management team. The specification of system requirements and an architectural design has to be performed under consideration of innovation technologies evolving during the entire research process. Furthermore, the heterogeneous interests and knowledge of all involved parties have to be considered to choose an acceptable development strategy leading to the target system. A

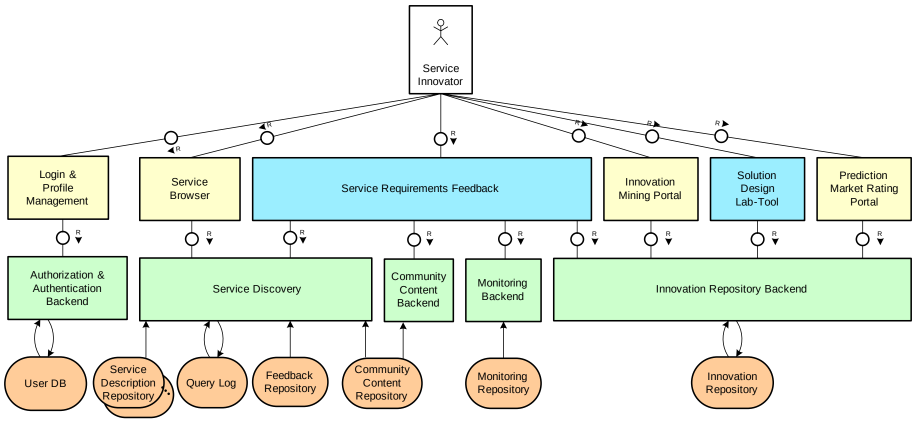

Figure 4: Service Innovator Architecture

method based on iterative incremental adjustment of both requirements and architecture during considering the novel technologies applied in a research environment was specified. Practical applicability of the approach was demonstrated with TEXO, a large research project where the proposed modelling and incremental enhancement of role-based and technology-based architectural views parallel to the component design and development process has positive influence on the communication and collaboration between the partners.

This work concentrated on the requirements and architecture specification process. Detail analysis of the evaluation process for ensuring software architecture quality and tool support for the approach will be subject of future research. As future step a survey among the TEXO project partners will be conducted to provide empirical validation of the aplicability of the proposed approach against other design approaches they are familiar with. A definition of general criteria for the usefulness of the approach (e.g. in terms of project size, projects structure etc.) and analysis of its applicability in distributed industrial environment are also planned as further activities.

Acknowledgements

The project was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference ”01MQ07012”. The authors take the responsibility for the contents.

The information in this document is proprietary to the following THESEUS consortium members funded by means of the German Federal Ministry of Economy and Technology: University of Kaiserslautern and Technische Universität Darmstadt. The information in this document is provided ”as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to

References

[BCK03] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*, volume 2nd ed. Addison Wesley, Boston, 2003.

[Bec99] K. Beck. *Extreme Programming Explained: Embracing Change*. Addison Wesley, Boston, 1999.

[Boe88] B. Boehm. A Spiral Model of Software Development and Enhancement. *Computer*, 21 (5):61–72, 1988.

[DN02] L. Dobrica and E. Niemelä. Survey on Software Architecture Analysis Methods. *IEEE Transactions on Software Engineering*, 28 (7):638–653, 2002.

[Erl05] T. Erl. *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*. Prentice Hall, 2005.

[Gar00] D. Garlan. Software Architecture: a Roadmap. In *Proceedings of the Conference on The Future of Software Engineering, Limerick, Ireland*, pages 91–101, 2000.

[IEE00] IEEE Std 1471-2000, IEEE Recommended Practice for Architectural Description of Software-Intensive Systems, 2000.

[JRS08] C. Janiesch, R. Ruggaber, and Y. Sure. Towards an Infrastructure for the Internet of Services. *HMD - Praxis der Wirtschaftsinformatik*, (261), 2008.

[KHMS09] A. Kabzeva, M. Hillenbrand, P. Mueller, and R. Steinmetz. Towards an Architecture for the Internet of Services. In *Work in Progress Proceedings of the 35th EUROMICRO Conference on Software Engineering and Advanced Applications, Patras, Greece*, 2009.

[Nus01] B. Nuseibeh. Weaving together requirements and architectures. *IEEE Computer*, 34 (2):115–117, 2001.

[Roy87] W. W. Royce. Managing the Development of Large Software Systems. In *Proceedings of the 9th international conference on Software Engineering, CA, USA*, pages 328–338, 1987.

[SEI09] Cranegie Mellon Software Engineering Institute. How Do You Define Software Architecture? http://www.sei.cmu.edu/architecture/definitions.html, Last access: May 2009.

[TEX09] THESEUS TEXO. Business Webs in the Internet of Services. http://theseus-programm.de/en-us/theseus-application-scenarios/texo/default.aspx., Last access: May 2009.

[TF08] M. B. Twidale and I. R. Floyd. Infrastructures From the Bottom-Up and the Top-Down: Can They Meet in the Middle? In *Proceedings of the Participatory Design Conference, IN, USA*, pages 238–241, 2008.