Conceptual Framework for the Use of the Service-oriented Architecture-Approach in the Digital Preservation

Christian Saul, Fanny Klett

Fraunhofer Institute Digital Media Technology
Business Area Data Representation & Interfaces
Ehrenbergstrasse 31
98693 Ilmenau, Germany
{Christian.Saul; Fanny.Klett}@idmt.fraunhofer.de

Abstract

This paper presents a conceptual framework for the use of the SOA[1]-approach in the digital preservation. The focus of this work reflects the service composition part within the SOA service concept. Previously released approaches have been separately using process-oriented models to describe the behaviour of services, and structure composition models to represent service interactions. In this paper, the authors attempt to combine the even mentioned disjunctive models to obtain a comprehensive model, which represents both, the structure and the behaviour of the services. For this purpose, the novel SCA[2]-BPEL[3] serves as basis for the implementation in a future-oriented SOA-compliant digital preservation software system. The SCA-model specifies the architecture of the intended system, while the BPEL-model indicates the behavior of each service, which is defined in the SCA-model. We can conclude that the SCA-BPEL approach is well-suited for building a scalable, adaptable and service-oriented software system. The knowledge gained from the conceptual framework will serve as a basis for future digital preservation developments.

Keywords

Service oriented Architectures, Digital Preservation, Service Component Architecture, Business Process Execution Language

Introduction

Digital preservation refers to the management of digital information over time. It is defined as a long-term, error-free storage of digital information, in terms of retrieval and interpretation, of the entire time span the information is required for. Long-term is defined as "long enough to be concerned with the impacts of changing technologies, including support for new media and data formats, or with a changing user community. [1]

Due to the rapid grow of digital information the data transfer volume to digital repositories rises continuously. This makes it necessary to find new levels of automation in digital archiving and preservation solutions. The increasing diversity in size and complexity of new digital resources implies that the repository systems must become highly automated and adaptable to various types of input, storage, access, and simultaneously to the users. The level of automation and technology support in current digital preservation solutions is low, and involves several manual stages. The scalability of existing preservation solutions has been poorly demonstrated by now, and solutions often have not been properly tested against diverse digital resources, or in heterogeneous environments.

Research in digital preservation domain has moved away from trying to find one ideal solution to the digital preservation problem towards focusing on the definition of practical solutions for preservation situations. This approach has to utilize the experts' know-how in memory institutions, to implement industry standards, and moreover, to involve solutions that are scalable and adaptable to heterogeneous environments.

Related Work

SOA is an emerging approach [2] that describes flexible software architectures, which can offer proper solutions to the above-mentioned problems of digital preservation systems.

The functionality of those architectures is provided as loosely coupled services over standardized interfaces. The aim is to map the business processes through a suitable composition of various services, in order to achieve a high flexibility related to process variability.

[3] associates four key concepts (see figure 1) with a service-oriented architecture namely application front-end, service, service repository, and service bus. The focus of the SOA-approach is the service concept itself.

Services are packaged software resources, which are well-defined, self-contained modules that provide standard business functionality and are independent of the state or

[1] Service-Oriented Architecture (SOA)
[2] Service Component Architecture (SCA)
[3] Business Process Execution Language (BPEL)

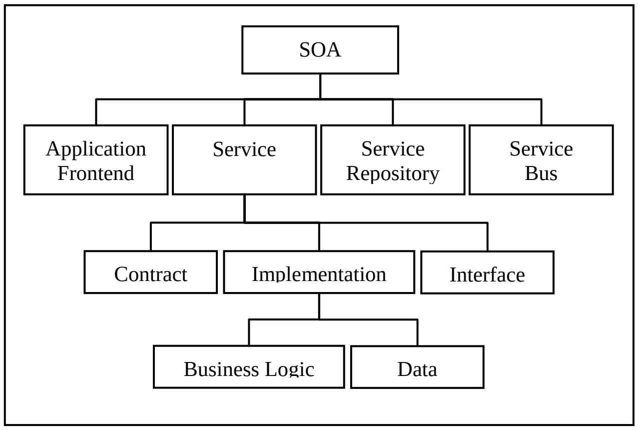

Figure 1: Elements of SOA [3]

context of other services. They are described in a standard language, have a published interface, and communicate with each other by requesting application of their operations, in order to collectively support a common business task or process. [4]

[5] states the service concept in one concise sentence that summarizes the important facts. Services are autonomous, platform-independent entities that can be described, published, discovered, and loosely coupled in novel ways.

Building a software system typically requires combining multiple existing services. These composite services can be recursively composed with other services into higher level solutions. According to [6], the two models of service composition in SOA's are both, the process-oriented- and the structural composition model described below.

Process-Oriented Model

The process-oriented composition combines services by using a workflow model to define a new service component. BPEL [7] is the most applied specification for this composition model.

Business Process Execution Language (BPEL). BPEL defines a model and a grammar for describing the behavior of a business process based on interactions between the process and its partners. The interaction with each partner occurs through Web Service interfaces, and the structure of the relationship at the interface level is encapsulated in a partnerLink. The BPEL process defines how multiple service interactions with these partners are coordinated to achieve a business goal, as well as the state and the logic that are necessary for this coordination. BPEL also introduces systematic mechanisms for dealing with business exceptions and processing faults. Moreover, BPEL introduces a mechanism to define how individual or composite activities within a unit of work are to be compensated in cases where exceptions occur or a partner requests reversal. [7]

Summarizing it can be stated that BPEL is concerned with business logic and the sequence of operations, which are performed to execute an individual business process.

Structural Composition Model

In contrast to the process-oriented composition, structural composition focuses on identifying the participating components, and the component connections that represent component interaction. The SCA [8] is the specification of a structural composition model for SOA.

Service Component Architecture (SCA). SCA represents a flexible SOA architecture standard for building composite applications using reusable services and extends, and complements prior approaches to implementing services. The SCA builds on open standards such as Web services. The SCA is based on the idea that business function is provided as a series of services, which are assembled together to create solutions that serve a particular business need. [8]

The SCA is concerned with what components exist in a business application, what services those components offer, what services reference those components, how the components are connected together, what endpoint addresses and communication methods are used for the connection, what policies are applied to components and to the connections between them.

Service Component Architecture Assembly Model. The SCA Assembly Model (see figure 2) [9] consists of a series of artefacts, which define the configuration of an SCA domain in terms of composites, which contain assemblies of service components, the connections and related artefacts, which describe how they are linked together.

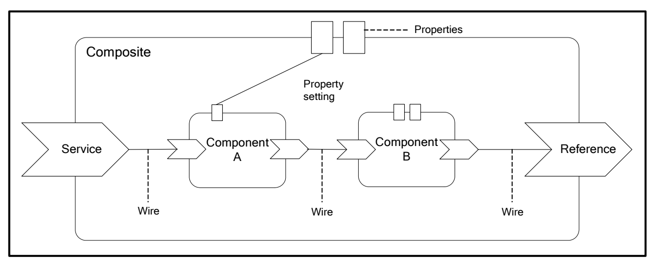

Figure 2: Service Component Architecture Assembly Model

One basic artefact of SCA is the component, which represents the unit for the construction of the SCA. A component consists of a configured instance of an implementation, which provides business functions. The business function is offered to be used by other components as a service. Implementations may depend on services provided by other components. These dependencies are called references. Implementations can provide properties, which are data values, which influence the operation of the business function. The component configures the implementation by providing values for the properties and by wiring the references to services provided by other components.

The SCA describes the content and linkage of an application in assemblies called composites. Composites can contain components, services, references, property declarations, and wires, which describe the connections between these elements. Composites can group and link components built from different implementation technologies by allowing appropriate technologies to be used for each business task. In turn, composites can be used as complete component implementations: providing services, depending on references, and with settable property values. Such composite implementations can be used in components within other composites by allowing for a hierarchical construction of business solutions, where high-level services are implemented internally by a set of lower-level services. The content of composites can also be used as a group of elements, which can contribute to build higher-level compositions.

Combination of Process-Oriented and Structural Composition Model

[10] argues that the implementation of the components as BPEL processes within an overall SCA assembly represents a good combination. Our work focuses on the application of the novel SCA-BPEL service composition approach to build a SOA-compliant digital preservation system. We combine the above explained service composition models to get a combined view of the structure and sequence: The SCA shows the structure of our composite service application while BPEL processes determine the flow sequence for each operation.

SOA-Approach in the Digital Preservation

The underlying Digital Preservation Workflow

The aim of our project is to develop a SOA-compliant digital preservation system by including a methodology that facilitates preservation work based on Web-Services. The intended digital preservation system refers to a preservation workflow that starts with a Pre-Ingest Phase over a Transfer Phase to an Ingest Phase.

Pre-Ingest is the preparatory phase for transfer of records from producer to the repository. During this phase the producer describes and normalises the content to comply with the requirements of the repository.

Transfer is the phase where the storage of records is transferred from the producer to the repository and between the repositories. It involves the transfer agreement, an optional test transfer, the actual transfer of records and their metadata, validation of the records, and acceptance from the repository.

Ingest is the phase where the repository is checking the transferred records, normalizes the transferred records and prepares them for long-term preservation in its storage, and for metadata management.

Here, we would like to point out that the OAIS workflow [1] starts with an Ingest Phase. In other words, we add a number of stages prior to the general workflow to extend the functionality of the indented system, and consequently to enhance the system support in a more extensive manner. Figure 3 illustrates this fact.

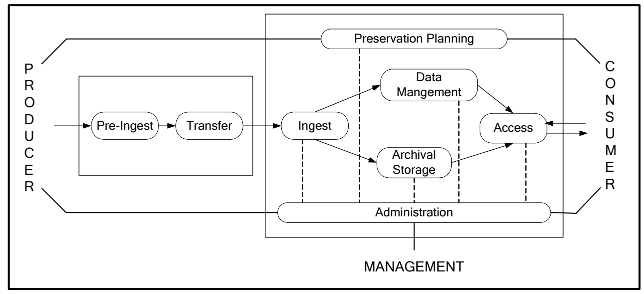

Figure 3: Extended OAIS Workflow Model

In the following the Pre-Ingest Phase is pointed out as an example to illustrate the whole process.

Building Components

To develop an innovative product with no obvious precedent, an understanding of the users and their capabilities, their current tasks and goals, the context of use of the product, and the constraints on the products performance is required. In order to communicate the user needs, requirements, objectives and expectations have to be discussed, refined, specified, and probably re-scoped.

Therefore, a variety of data gathering methods to collect sufficient, relevant and appropriate data is needed so that a set of stable requirements can be produced.

The most important needs arise from the data gathering methods that are focusing on the Pre-Ingest Phase:

- Creation of records management classification scheme
- Automate the process of appraisal
- Routine technical transactions (conversion into archival file formats, etc.)
- Compare documents and access restrictions against requirements from archival institution
- Analyze records (metadata, duplicates, classification)

Upon gathering the user needs the next step is to assemble appropriate components, which address the adequate needs. Each component has a dedicated task to fulfill functional requirements stated by user needs.

The following components for the Pre-Ingest Phase result:

- Technical Identification Component
- Digital Repository Requirements Component
- Metadata Improvement Component
- Migration Service Component

- SIP[1] Generating Component

The task of the Technical Identification Component is to identify the technical characteristics of the digital records like file formats, or the accompanying metadata formats. The Digital Repository Requirements Component analyses the digital repositories in terms of the respective requirements for long-term preservation. This could be mandatory file formats, or specific metadata elements. In order to conform to the requirements, some file formats changes are necessary. Hence, the Digital Repository Requirements Component additionally delivers a list, or proposes the tools to transform the digital records into a digital repository compliant format. The Metadata Improvement Component uses the provided information of the Digital Repository Requirements Component, and the technical characteristics of the Technical Identification Component to improve the metadata related to the records. In the same way, the Migration Service Component uses the output of the Digital Repository Requirements Component and the Digital Repository Requirements Component to migrate the digital records according to the requirements of the intended digital repository. The SIP Generating Component prepares the digital records and their metadata according to the SIP configuration accepted by the digital repository.

Modeling the Software Architecture

Based on the even identified services the architecture of the intended system can be constructed. The next step is to build the structure of the system through composing the services among each other. The result of this step is an architecture, which can supply information about their components: what services they offer, what services they use, how they are linked together, etc.

To model the structure of the SOA-compliant digital preservation system we have chosen for the new SCA-approach, because the SCA extends and complements prior approaches to implementing services, and it builds on open standard such as Web services. The SCA provides a model, both for the composition of services, and for the creation of service components by including the reuse of existing application function within the SCA composites.

As mentioned above, in the following we picked out the Pre-Ingest Phase in our digital preservation workflow to reveal the SOA-approach. The other phases can be implemented in the same manner.

The following figure 4 illustrates our Pre-Ingest SCA composite assembled from a series of components. The Pre-Ingest composite consists of five components as defined above, one offered service and three references to external services. The five components offer both references and services, and they are connected by wires, which describe the connections between those elements.

[1] Submission Information Package (SIP)

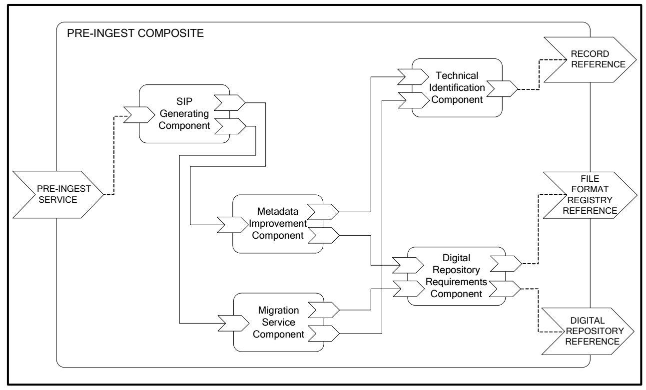

Figure 4: Pre-Ingest SCA Composite

The architecture of the intended digital preservation system only represents the fixed structure of the system without any information about the implementation of the components. BPEL aims at addressing this problem in particular, and its role is reflected in the following section.

Implementing the Software Architecture

The BPEL is a language for specifying business process behaviour based on Web Services. The processes in the BPEL export and import functionality by using Web Service interfaces exclusively, and determine the flow sequences for individual operations.

In the following the Digital Repository Requirements Component is picked out to highlight the possible business logic implementation with BPEL. The Digital Repository Requirements Component is aimed at gathering requirements of digital repositories in terms of file formats, mandatory metadata elements, standards, etc., and at informing about appropriate transform tools.

Figure 5 visualizes the implementation of the Digital Repository Requirements Component with the BPEL.

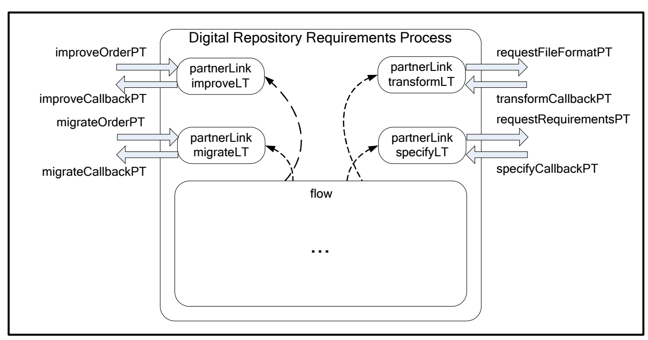

Figure 5: Digital Repository Requirements Component BPEL Implementation

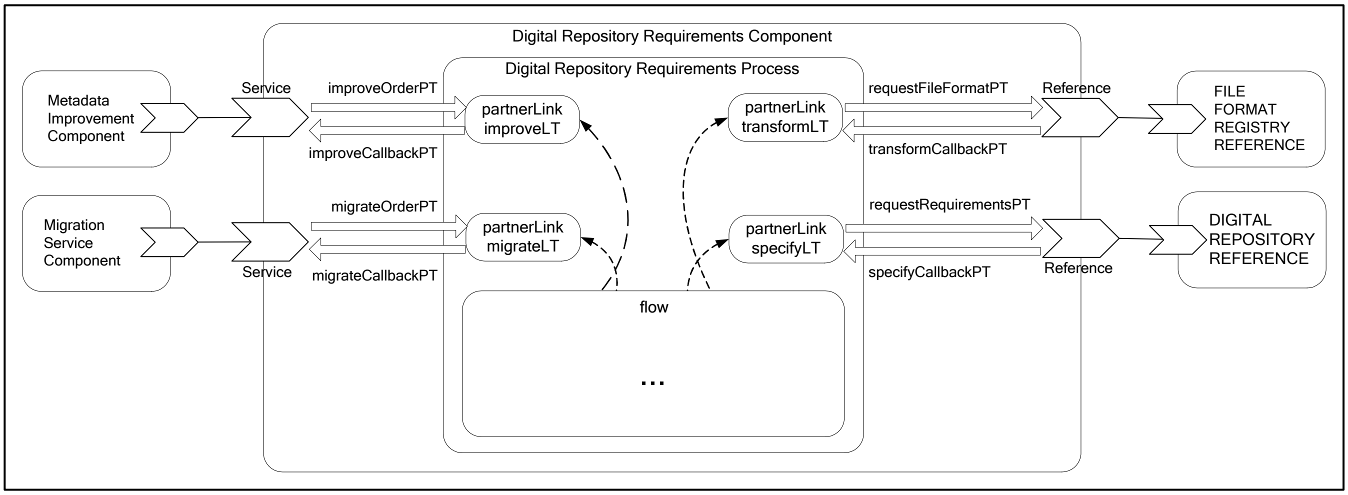

Figure 6: Pre-Ingest SCA Composite with a BPEL Implementation

The BPEL process invokes two services, one to request tools for the transformation of file formats, and another one to request the requirements of the indented digital repository for long-term preservation. In addition, the process offers two services one to provide recommendations for improvement, and the other to provide recommendations for migration. These relationships are captured in four partnerLinks.

The BPEL process definition appears as the implementation of the prior defined Digital Repository Requirements Component. The process definition is the foundation for a deployment by a BPEL engine. A BPEL engine interprets and executes business processes described in the BPEL.

Against this background, we have two independent models to describe a SOA-compliant digital preservation software system. The SCA-model reflects the architecture of the intended system. It specifies the interaction of services, and assembles them together to form a composite application. In contrast to the SCA-model, the BPEL-model specifies the behaviour of each service, which is prior defined in the SCA-model. It is obvious that a combination of the explained models is sufficient.

Modeling and Implementing the Software Architecture

The SCA and the BPEL are complementary technologies. The BPEL is an execution language while a SCA captures only the dependencies. But a BPEL process can be an implementation type of a service within the SCA.

BPEL captures relationship between the process and an interactive web service as a partnerLink with different roles linked to port types. The SCA maps the partnerLink with a single role (port type) to a reference.

Figure 6 shows the SCA+BPEL Pre-Ingest Composite consisting of the components, references and services which are linked together. In our case the partnerLinks define two roles, one for the BPEL process, and one for the partner. Depending on the message flow direction, one of them becomes a reference, and the other one becomes a service. In our solution, two BPEL process interfaces are exposed as a service entry point of the composite while two other partnerLinks are mapped as references. The BPEL process does not know the implementation, the references, and their binding. This loose coupling and flexibility is a power of the SCA architecture.

Conclusion

This paper has presented a novel SCA-BPEL service composition approach applied to build a conceptual framework towards a SOA-compliant scalable and adaptable digital preservation system.

The example showed the Pre-Ingest phase of the Digital Preservation workflow according to the OAIS model. Each stage in our workflow has been modeled according to the SCA-BPEL approach. The SCA describes the structure of a workflow component (i.e. Pre-Ingest Phase, Transfer Phase, Ingest Phase, etc.), and the connections between them. The sequences, in which the particular services are involved, are determined by the BPEL. It implements the business logic of the digital preservation system.

Directions for Future Work

The knowledge gained from the conceptual framework will serve as a basis for future digital preservation developments after completing a proper formative and summative evaluation in several iterative stages of the system design.

Acknowledgements

This work is partly supported by the Grant EU project Nº 216746 PReservation Organizations using Tools in AGent Environments (PROTAGE), FP7 and the Fraunhofer Institute for Digital Media Technology, Ilmenau.

References

[1] Consultative Committee for Space Data Systems: *"Reference Model for an Open Archival Information System (OAIS)"*, Washington, DC, CCSDS Secretariat, 2002.

[2] Mike P. Papazoglou; Willem-Jan von den Heuvel: *"Service oriented architectures: aproaches, technologies and research Issues"*, The VLDB journal, volume 16, 2007.

[3] Dirk Krafzig, Karl Banke, Dirk Slama: "*Enterprise SOA: Service Oriented Architecture Best Practices"*, Prentice Hall International, 2005.

[4] Fremantle, P., Weerawarana, S., Khalaf, R.: "*Enterprise Service"*, Commun. ACM 45(10), 2002.

[5] Papazoglou, M.P., Traverso, P., Dustdar, S. Leymann, F.: "*Service-oriented Computing: State of the Art and Research Challenges"*, in Computer – Innovative Technology for Computer Professionals - IEEE Computer Society, 11/2007.

[6] Mike Edwards: "*Relationship between SCA and BPEL"*, Open SOA Collaboration, 2007.

[7] OASIS: *"Web Services Business Process Execution Language Version 2.0"*, http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html, 2007.

[8] OASIS Open SOA: *"Service Component Architecture (SCA)"*, http://oasis-opencsa.org/sca, 2007.

[9] OASIS Open SOA: "*Service Component Architecture Assembly Model Version 1.00"*, http://www.osoa.org/download/attachments/35/SCA_AssemblyModel_V100.pdf, 2007.

[10] Francisco Curbera: *"Component Contracts in Service-oriented architectures"*, in Computer – Innovative Technology for Computer Professionals - IEEE Computer Society, 11/2007.