HLRS

Institut für
Höchstleistungsrechnen

FORSCHUNGS – UND ENTWICKLUNGSBERICHTE

INCREASED FLEXIBILITY AND DYNAMICS IN DISTRIBUTED APPLICATIONS AND PROCESSES THROUGH RESOURCE DECOUPLING

Alexander Kipp

ISSN 0941 - 4665 Oktober 2014 HLRS-12

Höchstleistungsrechenzentrum
Universität Stuttgart
Prof. Dr.-Ing. Dr. h.c. Dr. h.c. M. Resch
Nobelstrasse 19 - 70569 Stuttgart
Institut für Höchstleistungsrechnen

INCREASED FLEXIBILITY AND DYNAMICS IN DISTRIBUTED APPLICATIONS AND PROCESSES THROUGH RESOURCE DECOUPLING

von der Fakultät Energie-, Verfahrens- und Biotechnik
der Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

Alexander Kipp
aus Stuttgart

Hauptberichter:	Prof. Dr.- Ing. Dr. h.c. Dr. h.c. Michael Resch
Mitberichter:	Prof. Dr.-Ing. Stefan Wesner
Tag der Einreichung:	23. Januar 2013
Tag der mündlichen Prüfung:	16. Oktober 2014

D93

ISSN 0941 - 4665 Oktober 2014 HLRS-12

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die wesentlich zum Gelingen dieser Arbeit beigetragen und mich in all der Zeit unterstützt haben.

Mein erster Dank gilt Prof. Dr. Michael Resch für die Möglichkeit, meinem Promotionsvorhaben am HLRS zu folgen, sowie für die Erstellung des Erstgutachtens.

Prof. Dr. Stefan Wesner möchte ich für die laufende Unterstützung, vor allem während der finalen Phase der Ausarbeitung, sowie für die Erstellung des Zweitgutachtens danken. Insbesondere die zielführenden Diskussionen haben sehr zum Gelingen dieser Arbeit beigetragen.

Prof. Dr. Michael Schmidt danke ich für die Übernahme der Leitung des Prüfungsausschusses.

Bei den Kollegen des HLRS möchte ich mich für die angenehme Zusammenarbeit, die Hilfsbereitschaft sowie die Kultur des offenen Austauschs und das gute Betriebsklima bedanken. Insbesondere bedanke ich mich bei Eugen Volk, Tao Jiang, Jia Liu und Ralf Schneider für die äußerst gute sowie konstruktive Zusammenarbeit.

Freddie Klank danke ich für die Unterstützung beim Druck dieser Arbeit.

Mein besonderer Dank gilt Lutz Schubert. Vielen Dank für die tolle Zeit in unserer Arbeitsgruppe, die vielen inspirierenden Diskussionen und die exzellente Zusammenarbeit, die kurzweilige Zeit im Büro, sowie das große Interesse an dieser Arbeit und die fortlaufende Unterstützung.

Mein größter Dank gilt meiner Familie. Meinen Eltern danke ich, da sie in jeglicher Hinsicht die Grundsteine für meinen Weg gelegt haben. Meinen Schwiegereltern möchte ich für die kontinuierliche Unterstützung vor Ort danken. Meinen Kindern Leonie, Lukas und Julian danke ich für ihr Verständnis für die unzähligen Arbeitsabende und -wochenenden, sowie für die schöne Ablenkung in den Arbeitspausen. Meiner Frau Heike danke ich für die liebevolle, dauerhafte Unterstützung und Motivation, das häufige Rückenfreihalten, sowie für all das, was man nicht mit Worten ausdrücken kann.

Freiberg a.N., im Oktober 2014 Alexander Kipp

Abstract

Continuously increasing complexity of products and services requires more and more specialised expertise as well as a relevant support by specialised IT tools and services. However, these services require expert knowledge as well, particularly in order to apply and use these services and tools in an efficient and optimal way.

To this end, two applications fields have to be distinguished: There is the need to integrate human experts within the respective product engineering process, but also to support these experts with complex services for their specific expertise domain.

Domain experts are characterised by having an explicit, profound knowledge within their domain. However, these domain experts typically do not have the required knowledge to apply and execute specialised services and IT environments in a target-oriented and efficient way. For example, a doctor is not expected to have profound knowledge in the operation of high performance computing environments as well as the applied simulation environments. Expert doctors have a profound knowledge in a specific field of expertise (domain), e.g. in cancellous bone implants, which they like to provide in a computer-supported way for the benefit of their patients. In this case a doctor could simulate the expected outcome of an operation beforehand, without forcing the patient undergoing a medical intrusion. To this end, the optimal proceeding for the operation and the treatment can be determined in advanced. Obviously a doctor does not have the required IT knowledge to develop, execute and analyse the technical outcome of a simulation. I.e. an infrastructure is required, allowing for the consumption of such services for users not having the required technological knowledge to use these services in an optimal and efficient way.

Faster and more dynamic markets require an infrastructure allowing adapting to flexible and continuously changing circumstances. I.e. an infrastructure is required supporting domain experts as well as the service infrastructure to dynamically adapt to changing requirements.

In this context Gartner coined the term Service Oriented Architecture (SOA) in 1996[1]. The introduced concept follows the proven concept from the electrical engineering domain, which abstracts the service realisation and the service functionality via a technological independent interface. Basically, this proceeding correlates with the paradigm ”Programming in the large“ and ”Programming in the small“, introduced by DeRemer and Kron[2], decoupling conceptually the service realisation and the respective technology independent service orchestration with high-level process modelling languages.

The web service concept took up the cause of the operational realisation of these concepts. However, the intended target to provide a technical platform for the presented SOA paradigm was not achieved due to several reasons.

First of all, web services focus on the development and the integration of services, but not of human experts. In particular this framework does not provide a

---

[1] Research Note SPA-401-068, 12 April 1996, "'Service Oriented' Architectures, Part 1"

[2] DeRemer F, Kron H. Programming-in-the-Large Versus Programming-in-the-Small. *IEEE Transactions on Software Engineering*. 1976;SE-2(2):80-86

collaborative support for this purpose.

In addition, the required abstraction from technological details of the offered services is not given. The usage of web services requires at least a basic knowledge of the respective communication- and interaction protocols, as well as of the used technologies for the service realisation. This results in a significant reduction in flexibility, since service consumers are not able to easily change service providers, not to mention adapting their internal processes dynamically to changed circumstances. The latter one applies for example to changed corporate objectives. Furthermore, service providers have to announce changes in their service infrastructure to their customers, and these have to adapt their services to the new structure. Additionally, service providers are not able to integrate third-party suppliers within their services in a transparent fashion. Finally, the given approaches do not allow for a domain-specific encapsulation of both, expertise and services, i.e. allowing domain experts to apply their expertise to the respective application fields, and provide these expertise to end users via standardised interfaces abstracting from the underlying technologies.

To this end, the starting point of this thesis was the derivation of generalised key requirements from collaboration scenarios, depicting the essential requirements to be addressed in order to enable such a dynamic collaboration environment.

Based on these key requirements existing approaches have been analysed and evaluated in how far these approaches face these key requirements. This analysis showed significant gaps in current approaches.

The developed and realised concept in this thesis tackled the identified gaps of existing approaches by

- Enhancing the concept of web services with a service virtualisation layer, allowing for the transparent usage, adaptation and orchestration of services
- Enhancing the developed concept towards a “Dynamic Session Management” environment, enabling the transparent and role-based integration of human experts following the SOA paradigm
- Developing a collaboration schema, allowing for setting up and steering synchronous collaboration sessions between human experts. This enhancement also considers the respective user context and provides the best suitable IT based tooling support

The developed concept has been applied to scientific and economic application fields with a reference realisation. In particular, the following scenarios have been realised and evaluated in the context of several European research projects:

- Enhanced collaboration support during the construction of an aircraft
- Service- and Resource orchestration and optimization during the operation of an airport
- Integration of medical simulations in standards process description languages
- Quantitative and qualitative evaluations of the developed gateway concept and the respective reference realisation

The evaluation showed that the developed gateway concept addressed the identified gaps of existing approaches.

Zusammenfassung

Kontinuierlich komplexere Produkte und Dienstleistungen erfordern mehr und mehr spezialisiertes Wissen von Personen sowie eine entsprechende IT Unterstützung durch Bereitstellung entsprechender, spezialisierter Dienste. Solche spezialisierten Dienste setzen allerdings ebenfalls Expertenwissen voraus, um effizient und optimiert eingesetzt und bedient zu werden.

Es werden somit zwei Anwendungsfelder unterschieden: Zum einen bedarf es der Integration von Experten in entsprechende Produktionsprozesse, weiterhin auch der Unterstützung dieser Experten durch spezialisierte Anwendungen.

Domänen-Experten zeichnen sich durch explizites, tiefgehendes Wissen innerhalb der entsprechenden Fachgebeit aus. Allerdings haben diese Domänen-Experten typischerweise nicht das nötige Wissen, die entsprechenden Dienste zielgerichtet und effizient einzusetzen. Exemplarisch ist es nicht zu erwarten, dass ein Mediziner umfangreiche Kenntnisse mit dem Einsatz von Höchstleistungsrechnern sowie der entsprechenden Simulationsumgebungen besitzt. Spezialisierte Mediziner haben Kenntnisse in bestimmten Fachgebieten (Domänen) und möchten diese Rechner-unterstützt zum Wohle Ihrer Patienten einbringen. So wäre es beispielsweise möglich, dass ein Mediziner bei einer Operation deren erwartetes Ergebnis mithilfe einer Simulation im Vorfeld überprüfen kann, ohne dass der Patient einem Eingriff unterzogen wird, und somit für den Patienten die optimale Vorgehensweise und Behandlung bestimmt werden kann. Offensichtlich bringt der Mediziner nicht die nötigen IT Fachkenntnisse mit, diese Simulation zu entwickeln, auszuführen und die technischen Ergebnisse auszuwerten. D.h. es bedarf einer Infrastruktur, welche die Nutzung solch komplexer, spezialisierter Dienste auch für Anwender ermöglicht, welche nicht über das nötige technologische Fachwissen verfügen, diese optimal einzusetzen.

Schnellere dynamische Märkte erfordern eine Infrastruktur, welche es ermöglicht, flexibel auf sich kontinuierlich ändernde Anforderungen zu reagieren. D.h. es wird eine Infrastruktur benötigt, welche es ermöglicht, Domänen-Experten sowie die entsprechende Service-Infrastruktur dynamisch an diese Anforderungen anzupassen.

In diesem Kontext wurde 1996 erstmals von Gartner der Begriff der serviceorientierten Architektur (SOA) geprägt[3]. Dieser folgt dem bewährten Prinzip aus der Elektrotechnik, welches vorsieht, dass von der konkreten Dienstrealisierung abstrahiert wird und dessen Funktionalität über eine standardisierte, technologisch unabhängige Schnittstelle verwendet werden kann. Im Wesentlichen entspricht diese Sichtweise dem von DeRemer und Kron geprägten Paradigma “Programming in the large” und “Programming in the small”[4], welches konzeptionell die Entkopplung der Dienstrealisierung von der entsprechenden technologie-unabhängigen Dienst-Orchestrierung durch entsprechende Prozessbeschreibungssprachen vorsieht.

Die operative Umsetzung dieses Konzepts wurde mit der Einführung von Web

---

[3] Research Note SPA-401-068, 12 April 1996, "'Service Oriented' Architectures, Part 1"

[4] DeRemer F, Kron H. Programming-in-the-Large Versus Programming-in-the-Small. *IEEE Transactions on Software Engineering*. 1976;SE-2(2):80-86

Services eingeläutet. Das Ziel einer technologischen Plattform für das vorgestellte SOA Paradigma wurde jedoch aus mehreren Gründen nicht erreicht.

Zum Einen fokusiert das Web Service Konzept auf die Entwicklung und Integration von Diensten, aber nicht auf die Einbindung von menschlichen Experten und deren Wissen. Im Speziellen wird in diesem Rahmenwerk keine Kollaborationsunterstützung zur Verfügung gestellt.

Weiterhin ist die vom SOA Paradigma geforderte Abstraktion von den zugrundeliegenden Technologien der angebotenen Dienste nicht gegeben. Die Nutzung von Web Service basierten Diensten erfordert zumindest ein grundlegendes Verständnis der eingesetzten Kommunikations- und Interaktionsprotokolle, sowie der zum Einsatz kommenden Technologien. Dies resultiert in einer signifikanten Reduktion der ebenfalls essentiell wichtigen Flexibilität, da Dienst-Endkunden beispielsweise nicht einfach zwischen Dienstanbietern wechseln können, geschweige denn ihre internen Prozesse dynamisch an neue Rahmenbedingungen anpassen können. Letzteres tritt beispielsweise im Rahmen von geänderten Unternehmenszielen auf. Weiterhin müssen Dienstanbieter Änderungen Ihrer Service-Infrastruktur an Ihre Kunden postulieren, und diese ggf. deren Dienste auf die neue Struktur anpassen. Weiterhin ist es nicht möglich, Dienste von Drittanbietern transparent in das Dienste-Portfolio einzubinden. Schließlich ist es mit den gegebenen Ansätzen nicht möglich, Fachgebiet-spezifische Kapselungen von Wissen und Diensten vorzunehmen, d.h. dass Domänen-Experten Ihre Expertise in spezifischen Feldern einsetzen, und transparent Endanwendern über standardisierte, von der entsprechenden Technologie abstrahierten, Schnittstellen zur Verfügung stellen können.

Basierend hierauf wurden in dieser Arbeit anhand konkreter Kollaborationsszenarien allgemeine Anforderungen abgeleitet, welche zur Erreichung eines solchen dynamischen Kollaborationsumfelds erfüllt werden müssen.

Anhand dieser Schlüsselanforderungen wurden bestehende Ansätze bzgl. der Adressierung der identifizierten Schlüsselanforderungen untersucht und ausgewertet. Im Rahmen dieser Untersuchung und Lückenanalyse hat sich gezeigt, dass bestehende Ansätze für die Umsetzung eines den Szenarien entsprechenden, dynamischen Umfelds nicht gerecht werden.

Das zur Schließung der identfizierten Lücken erarbeitete Konzept dieser Arbeit adressiert diese durch:

- Die Erweiterung des Web Service Konzepts mit Virtualisierungsmöglichkeiten, welche es ermöglichen, transparent Dienste zu verwenden und diese zur Laufzeit transparent anzupassen und zu orchestrieren
- Die Erweiterung des entwickelten Konzepts zu einem “Dynamic Session Management”, welches die transparente und rollenbasierte Integration von menschlichen Experten entsprechend des SOA Paradigmas ermöglicht
- Die Entwicklung eines Kollaborationsschemas, welches es ermöglicht, synchrone Kollaborationen zwischen menschlichen Experten aufzusetzen und zu steuern. Dies geschieht unter Berücksichtigung des jeweiligen

Anwenderkontexts sowie der bestmöglichen Unterstützung durch IT basierte Dienste.

Die Evaluierung des Konzepts erfolgte anhand konkreter wissenschaftlicher und wirtschaftlicher Anwendungsfälle durch eine Referenzrealisierung. Im Speziellen wurde folgende Szenarien realisiert und im Rahmen mehrerer Europäischer Forschungsprojekte evaluiert:

- Kollaborationssupport bei der Entwicklung eines Flugzeugs
- Dienst- und Resourcenoptimierung beim Betrieb eines Flughafens
- Integration medizinischer Simulationen in Standard Prozess-Beschreibungen
- Quantitative und qualitative Evaluierung des Konzepts und der Referenzrealisierung

Im Rahmen dieser Evaluierungen wurde aufgezeigt, dass das erarbeitete Konzept die Indentifizierten Defizite bestehender Ansätze adressiert.

Table of Contents

List of figures

List of tables

Abbreviations

AI	Artifical Intelligence
AOP	Aspect-Oriented Programming
B2B	Business to Business
BP	Business Process
BPEL	Business Process Execution Language
BPMN	Business Process Model and Notation
BPSS	Business Process Specification Schema
BSCW	Basic Support for Collaborative Work
CAD	Computer Aided Design
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
CRM	Customer Relationship Management
CT	Computed Tomography
cXML	Commerce XML
DMU	Digital Mock-Up
DNS	Domain Name System
DSM	Dynamic Session Management
EAI	Enterprise Application Integration
ebXML	Electronic Business using eXtensible Markup Language
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
GHz	Giga Hertz
HID	Human Interface Device
HLRS	High Performance Computing Centre Stuttgart
HPC	High Performance Computing
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
IDP	Identity Provider
IIS	Internet Information Service
IoT	Internet of Things
IP	Integrated Project
IT	Information Technology

ITIL	Information Technology Infrastructure Library
MEP	Message Exchange Pattern
MOM	Message Oriented Middleware
OOP	Object Oriented Programming
PaaS	Platform as a Service
PDP	Policy Decision Point
PEP	Policy Enforcement Point
QoS	Quality of Service
RPC	Remote Procedure Call
RVE	Representative Volume Element
SaaS	Software as a Service
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SIR	Service Instance Registry
SLA	Service Level Agreement
SME	Small and Medium-sized Enterprises
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SotA	State of the Art
SSO	Single Sign On
STS	Security Token Service
UDDI	Universal Description, Discovery and Integration
URL	Uniform Resource Locator
VO	Virtual Organisation
WCF	Windows Communication Foundation
WF	Workflow
WfMS	Workflow Management System
WS	Web Service
WS-CDL	Web Service Choreography Description Language
WSDL	Web Service Description Language
WSMS	Web Service Management System
xCBL	XML Common Business Library
XML	eXtended Markup Language

1. Introduction

The Internet has become an integral part of our everyday life. Beside the opportunity to use the Internet as a source for information is has become common to use the "Web" also as interaction platform for collaboration and the provisioning of services [155]. Furthermore, following the established trend towards cloud computing the Internet has gained additional attraction towards a generic communication environment allowing for the provisioning of almost any kind of resource by building upon the corresponding communication standards.

Taking into account this kind of infrastructure allowing for the integration of a broad range of distributed services and resources, the concept of so called Virtual Organisations (VO) came up at the beginning of the 21th century [56] [187]. The related VO models, as defined in [156] and [45], foresee to manage virtual resources in order to provide an aggregated, complex product consisting of different, orchestrated services and resources. In particular the enabling of such VOs towards a dynamic environment, allowing for autonomic and intelligent handling of unforeseen events, like the underperformance or the failure of a subcontracting service provider, is of significant importance in order to ensure the intended flexibility of the respective infrastructure. However, the management of such dynamic environments, in particular considering the heterogeneity of available services and resources is a challenging and still unsolved issue. Especially the matter of integrating human resources in such virtual collaboration environments, whereby taking into account various different services required to support these humans experts to achieve a specific goal, is yet unresolved. This also applies to the internal management of resources within an organization. Enterprise Resource Planning (ERP) [62] and Enterprise Application Integration (EAI) [143][73] are getting more and more important for the efficient usage of the internally available resources, whereby facing similar challenges when operating a VO consisting of different resources and services.

The conceptual solution for these issues has been faced with the Service Oriented Architecture (SOA) paradigm [50], mapping the established and proven idea to provide standardised interfaces abstracting from the underlying realisation of the respective functionality from electric engineering to the software engineering world. To allow for this approach, web service technologies have been seen as the enabler for a SOA based environment in a distributed, cloud-based infrastructure. However, this approach has only been successful in a limited way, since, due to a rank growth of a lot of web service standards, the original goal to decouple the interface definition from the respective service implementation has not been achieved. So far, only technically versed people are enabled to realise processes based on web services or consume the corresponding services. In particular the differing IT experience of users shows, that the current available environments are not satisfying the concrete needs of the corresponding users.

1.1 Motivation

Fast changing customer preferences, continuously increasing business dynamics and disruptive technological shifts require two kinds of flexibility. On one hand, the opportunity of cross-organisational correlations to support changes in offering characteristics (offering flexibility), and on the other hand the ability to alter connections to partner with different supply chain players (partnering flexibility) have to be supported [60]. From a technical and methodological perspective this requires a new kind of infrastructure allowing for the integration of resources and services in an abstract way.

Service provisioning over the internet using web service specifications becomes more and more difficult due to the vast number of different, quickly changing technologies and standards. One of the most important aspects relates to dynamic service provisioning: the straight forward web service usage aims at exposing individual resources according to a fixed description, whereas organizations offering services want to expose flexible descriptions of their complex aggregated products, allowing for the encapsulation of the underlying service infrastructure by hiding the respective complexity [66]. Additionally, due to globalization and environmental challenges, there is a growing interest in adopting collaboration technologies so as to support distributed enterprises to work together as virtual teams, thus reducing time and costs. This thesis presents an approach to reduce the technological overhead in virtual resource and service provisioning over the Internet, implicitly allowing for more flexibility. To this end, a dynamic gateway structure is introduced which is acting as virtual endpoint to message transactions, enabling to encapsulate complex business processes on behalf of service providers.

In addition, collaboration between entities is not limited to the provisioning and consumption of services, but it also provides expert knowledge of human resources, e.g. the provisioning of consulting services or expert analysis. The latter one is reflected within the design and development of complex engineering products, e.g. airplanes, requiring the consultation of external, domain specific experts. In this area, in this thesis an enhancement of the mentioned virtualisation gateway towards a “Dynamic Session Management System” is proposed, which dynamically integrates decision taking and communication tools within an entire collaboration framework. This allows for the role-based integration of humans in eBusiness collaboration environments, in particular allowing for synchronous collaboration between humans within an eBusiness process. Therefore, the introduced infrastructure enables communication channels based on decision making and communication tools, allowing the involved persons at a specific state of the process to interact with each other. The mentioned infrastructure also ensures that the corresponding applications are set up as intended. It also outlines on how data management and sharing within this dynamic infrastructure is handled while addressing the security concerns of certain individual companies.

To this end, the approach developed in the context of this thesis allows describing and executing such collaborations from the perspective of sharing data and consuming shared resources including human and computational resources in an abstract way. In particular, the introduced approach allows for an orchestration of resources with a high-level workflow language, like BPEL [194] without the need to consider the technical details about how the resource can be consumed, enabling also typical managing personnel without a profound expertise in technical aspects, to describe and setup respective processes.

In summary, the introduced gateway approach tackles the major requirement for dynamic and flexible service-oriented environments, namely to decouple interfaces from the respective resource framework, whereby taking into account human as well as computational resources.

1.2 Objectives

The SOA paradigm is well established in the design and development of complex IT services and infrastructures. However, the SOA paradigm itself has to be seen as a conceptual approach to handle the complexity of the respective systems. In order to make a SOA concept alive, a technological environment is required meeting the corresponding principles and standards. The importance and relevance of this paradigm is also reflected in the expected revenue of SOA based environments, which will likely increase from 34.595M€ in 2012 to 90,553M€ in 2020 [142].

Complex collaborations in B2B environments rely on flexible infrastructures, being able to adapt quickly to changing conditions, partners and targets. In particular in a global market, companies have to adapt to fast changing trends in order to stay in business.

In addition, B2B collaborations involve not only services, but also human resources to bring in specific expertise when developing complex products and constructions. Synchronous collaboration between humans is very important, in particular in dynamic environments, enforcing quick reactions on specific events and issues, which might require the support of a specialist promptly.

Web service technologies have been seen as the technical environment for SOA based solutions and concepts; however, as it is shown in this thesis, current approaches do not tackle the essential requirement to decouple services and the respective interfaces in an adequate way. In particular the integration of humans in these environments is not addressed in an adequate way.

The objective of this thesis is developing a concept to overcome these identified issues, in particular to enable the integration of human and IT resources in a SOA like manner. To this end, the developed concept tackles the increasing complexity of dynamic changing resource infrastructures by allowing for the definition of abstract, cross-organisational collaborations. In particular, in this thesis

1. An analysis is conducted in the first instance taking into account existing approaches based on web service technologies as technical realisation of the SOA paradigm as well as the integration of human resources in B2B collaborations;
2. An enhancement of the web service layer is introduced in order to allow the transparent usage of complex IT service environments by decoupling the respective realisations and interfaces;

3. The introduced service abstraction concept is enhanced with a respective collaboration layer in order to enhance the web service layer to integrate human resources in a SOA like manner.

1.3 Chosen Approach

The basis of this thesis is the consideration of typical B2B collaborations, involving different and potentially changing types of collaboration partners, as depicted in Figure 1. These collaborations are described within a workflow or process, depicting the single steps required to achieve a common goal. In particular, in B2B collaborations the consumption and integration of services and resources from collaboration partners is a common practise.

Therefore, the SOA paradigm enables orchestrating and modelling such collaborations in a target-oriented way by abstracting from the underlying, technical details of the respective services.

In addition such collaborations require, in particular when producing complex products and constructions, not only the integration of IT services, but also of human experts, which have to collaborate synchronously with each other in order to achieve a common goal.

Taking these aspects as starting point, in *chapter 2* common requirements for such a holistic environment are derived from selected scenarios. Additionally, the corresponding state of the art is identified and analysed in *chapter 3*.

Based on the derived, common requirements from chapter 2, the gaps of existing approaches are identified following two major lines. Namely, gaps in B2B collaborations with respect to coupling and using of IT services whereby considering different user types, and also with respect to integrating human resources and enabling them to synchronously collaborate with each other, are considered.

Therefore, a concept for service virtualisation is developed in *chapter 4* taking into account the identified gaps with respect to the provisioning and coupling of services whilst considering different user types.

This service virtualisation environment is enhanced in order to enable an integration of human resources for synchronous collaboration sessions in a SOA-like, role-based fashion. This enhanced concept and the respective realisation details are depicted in *section 4.4* .

In *chapter 5*, the developed conceptual approach is evaluated by applying a respective prototype implementation to different application scenarios. Additionally, the concept and the realisation are evaluated by taking into account quantitative as well as qualitative aspects.

Finally, this thesis is concluded and an outlook is given in *chapter 6*, highlighting further enhancement of the introduced virtualisation environment.

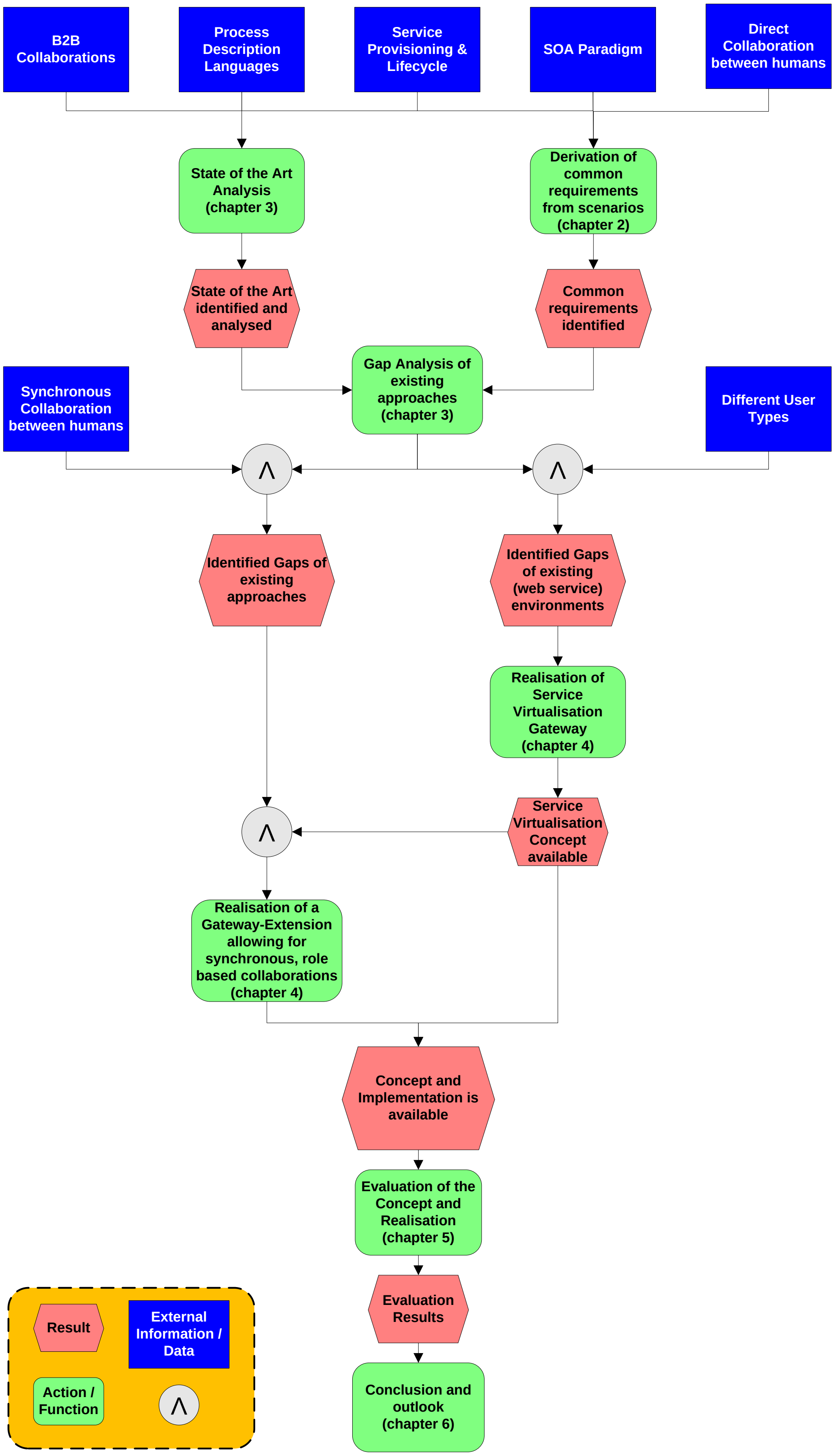

Figure 1 - Approach in Event Driven Process Chain notation as defined in [82]

1.4 Research Contribution

The research contribution of this thesis is a new virtualisation concept for SOA based environments. The essence of this thesis is a new approach allowing to couple and integrate complex services and IT environments, whereby enriching this approach towards a role-based integration model for human resources. The introduced approach allows for the orchestrating of human and IT based resources in a transparent fashion, thus allowing for the development of abstract processes bringing together human experts and complex IT service environments. The introduced approach has been presented and established by the author and others in different scientific publications ([93] [94] [95] [96] [98] [99] [81] [153] [154] [155] [157] [54]). The trend of energy-efficient usage of resources in a transparent fashion has been manifested by the author and others as well by applying the introduced approach to this domain ([87] [88] [89] [90] [91] [92] [97] [100] [102]).

The major research contribution of this thesis is the enhancement of the web service concept towards a virtual collaboration environment. In addition, the enrichment of this approach towards a role-based integration of human resources in a SOA-like fashion reflects a major improvement of existing approaches and concepts.

In summary, the major research contributions of this thesis are

- The enhancement of the web service concept by introducing a service virtualisation layer allowing for the encapsulation of complex service environments in a transparent, SOA-like fashion;
- The enhancement of the introduced approach towards a synchronous, role-based collaboration environment for human resources;
- The analysis and development of a respective schema to setup and steer synchronous collaboration session between humans with extended IT support.

To evaluate the relevance of the developed approach the concept has been realised and evaluated in various European research projects by applying the developed approach to concrete, scientific and business relevant application fields.

Finally, this thesis provides a detailed analysis of existing work, identifying gaps with respect to dynamic B2B collaborations and the integration of human resources.

1.5 Background

The author analysed different research projects by applying and evaluating the developed concept in respective, scientific and business relevant application fields. In particular, within the research projects *Access to Knowledge through the Grid in a mobile world (AkoGrimo)* [3] and *Trust and Contract Management Framework (TrustCoM)* [176] the author investigated the concept of VO-based environments.

Within the research project *Business objective driven Reliable and Intelligent grid for real business (BREIN)* [24], with a focus on merging agents, semantics and grids for "real" usable Grids, the author established the introduced gateway of this thesis as service virtualisation environment.

In the context of the research project *Innovative Collaborative Work Environments* *for Design and Engineering (CoSpaces)* [35], the author conceptualised and build-up an environment for synchronous collaboration environments. In addition, the prototype realisation of the dynamic session management extension of the introduced virtualisation gateway in this thesis as well as the respective collaboration session setup and steering schema have been developed, applied and evaluated in the context of this project.

Finally, within the project *Green Active Management of Energy in IT Service Centres (GAMES)* [102], focussing on the analysis and establishment of an environment allowing for the design- and runtime adaptation of hardware- and software environments towards energy efficient usage of the corresponding resources, the author investigated the analysis and determination of energy-consumption related metrics and respective monitoring solutions, in order to allow the optimisation of the entire energy consumption footprint of the executed services. Coupling this approach with the introduced service virtualisation environment enables service provider to offer services following specific energy-efficiency levels.

2. Requirement determination from application scenarios

In order to determine the requirements for flexible, dynamic SOA based environments, relevant application scenarios are analysed in this chapter. These scenarios have been applied within different European research projects, namely the BREIN, CoSpaces, and GAMES projects, reflecting the impact and relevance of the realised approach in this domain. Details about the application and evaluation of the following scenarios are given in section 5.1.

In the first instance, the respective user perspectives and the application scenarios are depicted, in order to identify common requirements.

After that, three application scenarios are described reflecting different aspects and viewpoints on complex IT infrastructures, namely:

- A scientific simulation scenario, highlighting the requirement of integration and simplified usage of complex IT resources as well as considering different user types and their needs;
- A scenario focussing on coupling and communication of autonomous self-organising software agents based on complex IT infrastructures;
- An environment allowing for the synchronous collaboration between experts based on complex IT infrastructures.

Finally, an overview of the identified key requirements is given in section 2.5.

2.1 Customer and Service Provider perspective

In the following section, these two roles are briefly summarised by analysing the specific requirements and targets.

2.1.1 Service provider

In a B2B scenario a service provider offers a service intended to be consumed by customers via a well-defined interface, addressing the customer's specific requirements. To this end, service providers encapsulate a specialised knowledge in the corresponding service. E.g. by providing a HPC simulation service for cancellous bones, the service provider implicitly encapsulates knowledge about running and evaluating a respective simulation, as well as operating or at least using a HPC environment in an efficient way. In particular, in this context, not only specific IT infrastructures are sold, but also specialised services for specific tasks, including a corresponding knowledge that might be needed by customers.

Therefore, a service provider intends to hide the service specific environments to avoid potential competitors to copy their infrastructure and environment in order to provide a similar service.

In addition, service providers try to avoid providing monitoring information about their infrastructures to external customers. Although this information is of great interest for service customers, e.g., allowing them to determine if the agreed service quality is achieved by the service provider, service provider try to avoid providing this information due to several reasons. For example in case of a problem with the service infrastructure, a service provider might re-direct service invocations to external service providers as well, without informing the customer and thus masquerading that everything is working fine. In some cases an external service provider is involved from the very beginning, providing some specific functionality which is necessary to orchestrate a complex IT service, and the service provider does not want to provide this information to his customers. In some cases, the service provider is even not allowed to do so due to some legal restrictions. In addition, by providing detailed monitoring information a service provider would allow external customers to gain information about the used infrastructure and potential settings of the corresponding runtime environment.

Finally, a service provider tries to bind his customers closely to his environment. This is achieved by providing services via specific, non-interoperable or standardised interfaces, enforcing potential customers to bind their services to these interfaces and thus complicating the exchange of a service provider by another one. This process enforces to take into account technical specialities of the service and IT infrastructure of the service provider, in particular if the customer intends to achieve a good performance. If these technical specialities are not taken into account, customers run a risk to waste resources, and thus money. However, as soon as the customer’s application is bound to the service provider interface, additional efforts are required from the customer in case he wants to use another service provider. In this context, service customers tend to continue consuming the services of the original service provider, in order to avoid these additional costs.

In summary, for the service provider the optimal proceeding foresees to sell specific services, hide as much of the service and IT infrastructure as possible, and finally deliver the requested results.

2.1.2 Service customer

Depending on the customer’s needs, detailed monitoring and infrastructure information is required. For example in the HPC domain a profound knowledge of the underlying service and software environment is of significant importance in order to achieve adequate performance of the corresponding service or application. Often it is also required to fulfil security constraints, e.g., avoiding specific data sets being processed in specific countries, or to prevent potential competitors to determine when jobs are executed or services are invoked. However, users do not want to be aware of the technical details of the underlying IT infrastructure, e.g., if experts from a specific domain want to execute a specific simulation service, they are just interested in the simulation results, but not in the concrete technical execution steps of the simulation. The latter one applies in particular to SOA based environment, enabling a service consumer or orchestrator to concentrate on the functionality of *his* application without being forced to take into account technical specialities of the IT environments of the service provider.

In addition, service customers want to keep their application and services as flexible as possible. This includes the easy replacement of service providers, e.g. if the service provider is no longer fulfilling the agreed QoS or if another service provider is providing a similar functionality with better conditions. In these cases, the service customer tries to stay as independent as possible by avoiding binding his own applications and services too closely to a specific service provider, like it is the case if the service provider enforces the service customer to use a non-interoperable, non-standardised service interface.

So, in summary, depending on the expertise of the customer it is sensible to hide technical details, but some IT experts might want to know technical specialties of the underlying IT infrastructure, in particular allowing them to optimize the application or service execution. However, most users prefer to consume a service without being forced to take into account the technical details of the underlying service infrastructure, in particular with respect to assembling new, complex services by just orchestrating existing ones. This requires an environment ensuring the easy, non-technical integration and potential replacement of service providers on a conceptual level.

2.1.3 Identified conflict

The discussed profiles are clearly conflicting. On one side, a service provider tries to bind his customers as close as possible to his infrastructure and services, whereas a service customer tries to avoid this. In addition, a service provider avoids providing detailed information about the behaviour of his infrastructure to his customers, but in particular customers are interested in this kind of information, since this allows them to determine potential contract violations and performance related analyses.

2.2 Offering and consumption of complex IT services

The integration of complex IT services by encapsulating its complexity in an abstraction layer is a common issue to be addressed in SOA based environments. The intention of this section is to identify specific requirements for this application scenario, derived from corresponding use cases. In this context it is assumed that a user with only limited IT knowledge wants to use a complex IT service. The consideration of a simulation service offered by a HPC service provider within a well-defined user process requires the integration of complex IT infrastructures. Additionally, complex HPC applications can also be distributed geographically [83] [109]. Within the engineering domain there is a trend towards simulating complex constructions, rather than building expensive physical prototypes, which have to be reconstructed every time an optimisation has been applied. Costs for virtual prototypes are much lower. However, constructing complex physical constructions engineers must have a profound knowledge in the relevant constructing or engineering domain, but not in IT. Following this trend towards the close coupling and integration of simulations in this construction process, more and more profound IT knowledge is required in order to realise effective simulations and reliable results. This scenario depicts a “Simulation as a Service” scenario within which users with only limited IT knowledge are enabled to consume complex, scientific IT HPC services in an easy-to-use and abstract fashion.

2.2.1 Scenario description

As depicted in section 2.1 the integration of complex IT services has to consider two viewpoints: form a customer as well as from a service provider perspective. A service provider intends not to burden his customers with technical details in order to enable them to consume the respective service, but also to hide the underlying IT and service infrastructure to keep his competitive advantage. This ensures on one side customers' satisfaction, and enables the service provider additionally to handle potential changes within the respective infrastructure in a transparent fashion. On the other hand, a customer is interested in consuming a specific product and the expected results in a reasonable time, but not in the technical details of the service.

In this scenario it is assumed that a doctor is preparing a hip surgery of one of his patients. In order to do so, a well-defined process has to be followed, including formal and administrative issues, and also concrete preparation and analysis steps. The latter one includes the creation of a CT and the medical analysis in order to determine the optimal position of the implant. In order to keep the impact on the patient as little as possible, this analysis has to be supported by a simulation, allowing to determine the optimal position of the implant without hurting the patient and being enabled to analyse how the implant is going to behave in case of specific movements of the patient. In particular this proceeding allows for determining different optimised implant positions, depending on the way the patient typically behaves in his everyday life.

Therefore, the doctor's clinic administration contracted with an HPC service provider, specialised in simulation services for cancellous bones. As soon as the patients CT data is available, following the well-defined process for hip surgeries, the service of the service provider is invoked and the simulation results are provided to the doctor for a detailed medical analysis and preparation of the surgery.

On the other side, the service provider enables customers to easily consume these complex services by offering an abstract service interface. This process ensures that

a) The service provider does not have to burden customers with technical details;
b) The service provider can provide the best suitable service for his customers In particular, the respective configuration and optimization can be done by internal experts (from a methodological and technical viewpoint) in order to ensure that the service execution does not cause additional costs and provides the best suitable results for the customer;
c) The service provider can handle internal errors and platform replacements or updates in a transparent fashion.

2.2.2 Scenario context

The Computer-Aided simulation of complex real-world problems provides a significant improvement in the development of new methods and new technologies, whilst minimizing the related costs and efforts. This is in particular the case with respect to the simulation of processes

- that are unsolvable by traditional and experimental approaches, like the prediction of future climate changes or the fate of underground contaminants;

- which are too hazardous to study in laboratories, like the characterization of the chemistry of radionuclides, other toxic chemicals or the distribution and effect of medicine within a human body;
- that are time consuming or expensive to solve by traditional means, e.g. car crashes, development of new materials or aerodynamic optimisations (requiring a physical model for each potential change).

In summary, such simulations provide a significant impact in the engineering, environmental and medical sector allowing for a detailed analysis of the relevant method or prototype whilst not affecting the environment at all.

Many applications in the HPC domain intend to tackle optimisation problems, being realised as parameter studies in order to archive the best suitable solution for a given problem. Therefore, specific characteristics of a surface have to be simulated by varying input parameters following the trial and error principle, since for a lot of real-world problems no formal procedure exists in order to achieve specific goals. So, to find the optimal solution for a given problem, different variants have to be tested.

Although the general procedure within parameter studies for different domains is similar, these complex procedures enforce a lot of detailed knowledge of both the simulation itself as well as the appropriate IT infrastructures and the corresponding software environments. Current environments enforce engineers and scientists to have this knowledge, beside the knowledge of their relevant domain as well, which is a big problem. Following the idea of SOA it is advantageous to have experts for each of the mentioned domains allowing for the encapsulation of specialised knowledge and thus optimised operation of these environments.

As an example for such a complex simulation process the micro-mechanical simulation of cancellous bone tissue targeting on determining its macroscopic elastic material properties on the continuum scale, is presented. The following overview gives a rough overview of the complexity of the simulation process. A detailed overview is given in [93].

Nowadays implant and prosthesis development is mostly done on experimental basis. So the choice of type, size and positioning of the implant is based mainly on the experience of the surgeon, which is a source of potential errors. This is also reflected by the fact that most postoperative mechanical complications are not explainable respectively and cannot be quantified, e.g.

- the Z-Effect of intramedullary implants with 2 proximal fixation devices
- Cut-out of gliding screw
- Loosening (~12% Revisions of placed implant are required per year in Germany) [113]

In order to minimise, or ideally even eliminate, these drawbacks, an extended support is needed to

- Chose the correct type of implant
- Support the correct positioning of the implant
- Predict the healing process

- Predict and quantify possible mechanical complications

That continuum mechanical simulation of bone-implant-systems has the potential to provide this support, which is shown by the work of numerous research groups during the past years. For static Finite Element simulations of bone-implant-systems, in general three prerequisites are essential: (1) the bone geometry, (2) the forces acting on the bone structures including body- as well as muscle forces and (3) a detailed modelling of the inhomogeneous, anisotropic material distribution of the bony structures.

The process of micromechanical simulations of cancellous bone tissue aims at the third of the above mentioned prerequisites and becomes necessary due to the lack of resolution of modern clinical imaging techniques. To separate the micro structures of cancellous bones, which have a diameter of ≈0.1mm, from volumetric imaging data, an isotropic resolution of ≈ 0.02mm would be necessary. Since this resolution is approximately 20 times higher than the one achievable with modern, clinical, dual source computer tomography (CT) an additional modelling step has to be introduced to enable continuum mechanical simulations with anisotropic material data on the resolution scale of clinical CT-data.

The general process of this additional modelling step for one element is shown in Figure 2.

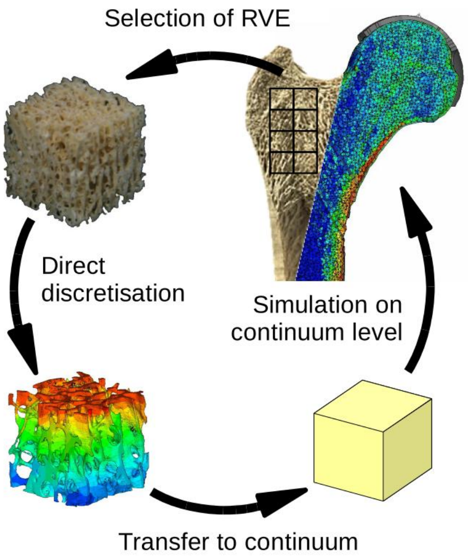

Figure 2 – Direct mechanical simulation proceeding of a cancellous bone [93]

In detail, the following steps are processed:

1. In the first step, a representative volume element (RVE) is selected and loaded from the initial data set.
2. Now this RVE is discretised and boundary conditions are applied in order to proceed with static simulations. In this step, a domain decomposition has to be processed by conducting a 2D-3D algorithm splitting. Therefore, the domain is decomposed with cutting planes in a normalised way to the global coordinate axis. Then, the resulting sub-domain faces are analysed for iso-lines with 2D marching cubes, in order to allow for the meshing of the iso-lines with 2D

advancing fronts. These iso-lines are attached to each sub-domain connected to the corresponding face. For each of these sub-domains closed seed fronts are constructed. Finally, this described 3D advancing fronts algorithm is executed in each sub-domain on the closed front. Afterwards, the geometry setup is processed as depicted in Figure 3.

3. After the direct discretisation the RVE is simulated in 6 different load cases to gain the local microscopic strain and stress distributions on the microscopic level. The procedure is depicted in Figure 4.

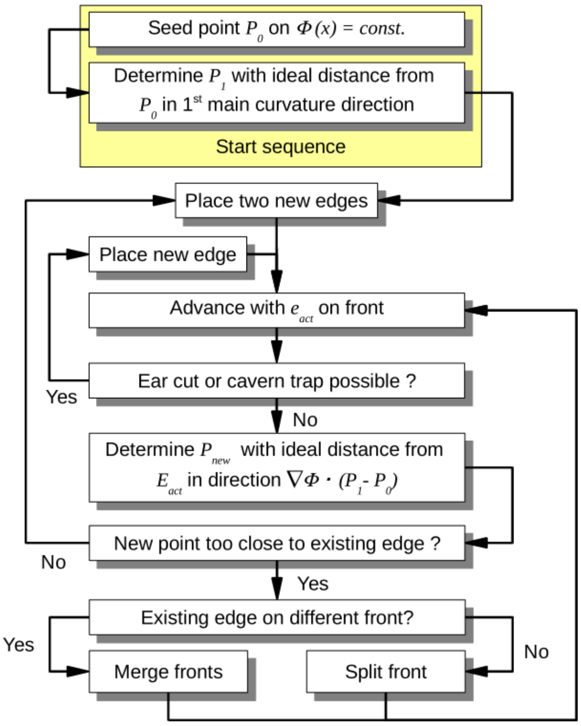

Figure 3- Geometry Setup [93]

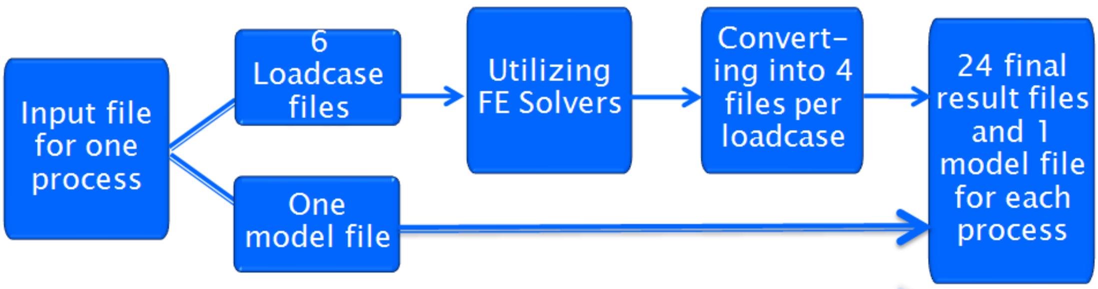

Figure 4 – Static Simulation Processing

4. The simulation results are transferred back to the continuum level by calculating the effective stiffness tensor for the RVE. In detail four connections between local microscopic and averaged macroscopic elastic properties have to be considered to determine the effective stiffness tensor of the RVE.

The average strain over a RVE $\bar{\varepsilon}_{ij}$ is computed as the integral average of the local strain field ε_{ij} by

$$\bar{\varepsilon}_{ij} = \frac{1}{|V_{RVE}|} \int_{V_{RVE}} \varepsilon_{ij} dV_{RVE} \quad (1)$$

The connection between the local microscopic strains and the average strain is further given by the linear transformation

$$\varepsilon_{ij} = M_{ijkl} \bar{\varepsilon}^{kl} \ (2)$$

with M_{ijkl} being the so called local structure tensor [69].

Now, starting from the generalized Hooke's law on microscopic level

$$\sigma_{ij} = C_{ijkl} \, \varepsilon^{kl} \ (3)$$

with σ_{ij} the local strain filed and C_{ijkl} the local stiffness tensor, it can be derived [69] that the so called effective stiffness tensor of the RVE $\bar{C}_{ijkl}$ is given by

$$\bar{C}_{ijkl} = \frac{1}{|V_{RVE}|} \int_{V_{RVE}} C_{ijmn} M_{mnkl} \, dV_{RVE} \quad (4)$$

Executing the procedure over an entire bone, (4) becomes

$$\bar{C}_{ijkl} = \bar{C}_{ijkl} \, (x, y, z) \ (5)$$

allowing for a correlation to measures derived from clinical-CT data

$$\bar{C}_{ijkl} \, (x, y, z) \sim CT(x, y, z) \ (6)$$

The effective stiffness is represented as a tensor of rank four $\bar{\boldsymbol{C}} = \bar{C}_{ijkl} \, e^i \otimes e^j \otimes e^k \otimes e^l$, depending on four bases (coordinate systems) $e^i, e^j, e^k and \ e^l$. All these four bases are equal to the global coordinate system. Due to symmetries the 81 coordinates of the effective stiffness tensor can be reorganized in a symmetric 6x6 matrix. So, for each of the 6 simulated load-cases one column of the matrix can be calculated, resulting in a dense matrix in general. This proceeding is depicted in Figure 5.

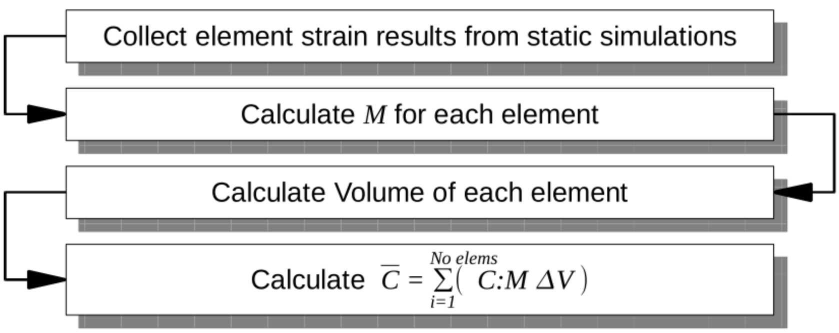

Figure 5 – Calculation of the effective stiffness matrices [93]

Finally, the computed results have to be correlated to the relevant clinical-CT data in order to enable a detailed analysis of the cancellous bone behaviour for the processed forces and implant positions on the continuum level. This correlation takes place with multivariate statistics and is done via a cluster analysis and linear discriminant analysis.

This rough description of the simulation process shows the complexity of the entire processing of such kind of simulations, often overburden experts with a detailed medical knowledge background.

2.2.3 Identified key requirements

In order to enable the depicted scenario in the previous section, it has to be ensured that *different types of users* with varying IT knowledge *are supported*. This service provisioning environment should allow *orchestrating* provided *services in a transparent fashion*, facilitating users with a specific field of expertise to use and integrate services in order to solve related issues. The user should be able to use the corresponding services based on the *provided functionality*, and not based on the used technology. To this end, the service provisioning environment should allow *hiding technical details* of the underlying IT infrastructure. This proceeding ensures that the users can concentrate on the problems and issues to be solved, and not being forced to tackle technical issues with the underlying IT infrastructure.

To allow such an environment within a B2B context, *accounting mechanisms* have to be *transparently integrated* in order to ensure that users can still concentrate on the solution of the problem. In particular in larger companies it is common practise that the central procurement department negotiates with several service providers and finally provides a pool of services to be used by all employees. In this context the transparent integration of accounting mechanisms permits employees to use the provided services from a functional viewpoint, and do not have to take care of service specific accounting issues. In particular when replacing a service provider this does not enforce an adaptation of applications relying on these services.

From the service provider perspective the environment should allow to provide *abstract* and *stable service endpoints* to the users, whereas the internal infrastructure of the service provider may change without affecting the relevant users. In particular, the development of services should be independent from the technological environment of the potential users. In summary, the environment should support the *"Programming in the small"* and the *"Programming in the large paradigm"* [44]. Therefore, a clear distinction between the service development and the service integration has to take place, in particular enforcing an enhanced support for *service lifecycle management*.

Finally, the service environment for this scenario should also allow for *dynamic process evolution* by allowing the static description of well-defined process flows, whereas enabling to dynamically replace linked services.

2.3 Inter-Organisational Collaboration in a Multi-Agent environment

Fast changing customer preferences, continuously increasing business dynamics and disruptive technological shifts require on one hand the ability of cross-organisational correlations to support changes in offering characteristics (offering flexibility), and on the other hand the ability to alter connections to partner with different supply chain players (partnering flexibility) [60]. From an organisational perspective this requires a method ensuring that the description of the respective processes can be abstracted from concrete entities, but concentrates on the functional orchestration of the steps required to produce a specific product. From a technical perspective this proceeding requires an infrastructures allowing for transparent service interception as well as communication content transformation. This technical requirement relies on the fact that changing a service provider results in the necessity to adjust runtime environments to new service interfaces. However, in order to ensure transparent changes in the service environment the respective interactions have to be adapted.

A common requirement in such environments is encapsulating functionalities provided by the involved collaboration partners via standardised interfaces. The respective technological basis for this kind of SOA based environments are web services and integration technologies, which are depicted in detail in section 3.4.

The collaboration between different organisational partners requires flexible and dynamically adjustable environments, taking into account dynamically changing requirements in B2B environments. Collaborations within a B2B environment come into existence because all involved parties benefit from each other, e.g. by buying a specific service which cannot be provided by a company on a pay-per-use manner or by taking advantage of external expertise in a specific field. This enables business entities to act more flexibly on changing requirements, adapt to new trends, and finally save costs by referring to external services and expertise only in case it is required. On the other hand, service providers are enabled to specialise on specific topics, thus provide best suitable services and support for this topic. This enables customers to easily take advantage of the provided services.

In particular, a customer wants to be able to choose and easily replace a service provider for a specific purpose if he does no longer meet the expected requirements, or if the customer changes his business goals.

On the other side, service providers want to bind customers closely to their infrastructure, whereby ensuring not to give too many insights of their own environment. This is due to the following aspects:

a) if a service provider offers a service in a better way than competitors, e.g. by offering the service cheaper or with better QoS for the same price, this is a advantage of this service provider compared with his competitors;
b) If a service provider contracts third party services to serve the users requests.

In addition, service providers also want to avoid that customers get informed about problems within their own environment, in particular during service execution.

Furthermore, there is also a trend towards intelligent and self-managed resources. These resources are getting more intelligent and self-organised. These intelligent environments are realised based on software agent technologies. These IT resources follow a specific goal, in which they are, at least to some degree, independent in the way to achieve this goal. These resources are able to learn from previous actions if they have been suitable to achieve the given goal. This might include the adaptation of the strategy, the modification of specific data, but also in the change of consumed and integrated services and service providers. In particular this latter trend enforces more flexibility to enable these resources to interact with different services to achieve a specific goal.

2.3.1 *Scenario description*

In this scenario it is assumed that a company provides complex logistic services. In this context, some services are provided directly by the company, whereas others are provided and integrated by third-party service providers, in order to enable the company to react quickly on changing requests from the customers, e.g. the transportation of specific goods and timing restrictions.

In addition, some of the services are goal-oriented and self-managed as so-called software agents.

The logistic plans are prepared and organised in advance, mainly based on experience made in the past, e.g. taking into account peak loads in specific time frames. However, the provided services have to be adapted quickly in case of unforeseen events, like an unexpected delay, the loss or underperformance of a involved service or partner, or surprisingly arising payload peaks or failures.

In order to cope with such unforeseeable issues, some of the most critical services are realised as self-organising ones, being realised as so-called software agents. These software modules are able to adapt themselves to unforeseen events by learning from the handling of previous events. This enables the respective services to adjust to changing requirements and environmental conditions in order to achieve the intended goal following some given side conditions, like latest date for delivery or QoS restrictions. These software agents are also able to adapt their internal processing strategy, including in particular the replacement and the adaptation of consumed services.

In summary, the exemplary company provides a holistic logistic service approach to external customers, whereby transparently hiding the self-organised infrastructure, including also third-party services. In addition, the company takes advantage of intelligent, self-organised software modules, enabling it to react quickly on changing requirements and unforeseen events. In this scenario, it is expected, that an unforeseen delay occurs and the service infrastructure adapts itself to the changing requirements by taking advantage of the respective, flexible environment as depicted in detail in section 5.1.2.

2.3.2 *Scenario context*

Logistics in airports require a very strict planning and flexibility due to often and quickly changing conditions.

In addition, the organisational structure of airports changes towards a networking organization with different service providers. For example, as depicted in [81], ground services are also offered by some airlines, so the airport is providing in such a case just basic infrastructural services, such as buildings, networks, flight related information, electricity, and so on.

In particular, in airports quick reactions on changing conditions and unforeseen events have to be ensured. Due to sudden changes, either on the resource provisioning side (e.g. by sudden failures of critical infrastructures) or on the scheduling (e.g. due to delayed flights, suddenly changing weather conditions, …) there is a great need in being able to quickly adjust and reorganise the available resources to ensure a smooth operation of the flight handling. In particular the latter one enforces a strict following of respective rules and guidelines, in order to ensure a secure operation as well.

In order to cope with this issue, the available resources have to be enabled to interact with each other as well, which causes, in particular in case of involved third party services, a significant interoperability and interaction issue.

In addition, in case of such a spontaneous enforced adjustment and rescheduling of resources, every resource has to solve a local optimization problem, since the original mission the respective resource is intended for, still remains unaccomplished. For example it has to be decided if it is better to delay processing of one flight, which will cause a respective delay and maybe anger of the corresponding passengers, or to take over an additional mission to cope with another flight and thus ensure a better workload balance of the resource.

To ensure a sufficient quality of service, service providers have specific SLAs with the respective service consumers. These conditions, and maybe restrictions, have to be fulfilled and to be transferred to the goals of the corresponding services to ensure a smooth operation. Therefore, in this scenario, the airlines act as customers and varying ground handling service providers act as respective service providers [81].

2.3.3 Identified key requirements

In order to enable the described scenario of this section, the following key requirements have been identified, in particular with respect to dynamic adaption and technical realisation of such a service infrastructure.

The service provisioning environment should allow *integrating internal* as well as *external services in a transparent fashion*. The user should be able to select required services based on their *functionality*, and not be forced to take into account technical details and the location of the service. Additionally, during *runtime* the *transparent replacement of equivalent services* should be supported. This proceeding facilitates the development of concrete process flows, whereas the underlying service infrastructure can be adapted dynamically during runtime. This ensures that the *well-defined processes* are flexible enough to *cope also with unforeseen events and conditions*. In particular the environment should cope with errors in a transparent fashion by not affecting the involved users.

To ensure this flexibility on a technical level, the environment should support *various existing standards*, including different and potential incompatible versions of these standards. This ensures that changes on technical level can be tackled in a transparent fashion by the service environment. In particular in an eBusiness related context *security* is an important issue to be addressed in a *transparent way*, since dynamic changes in the service landscape enforce a respective adaptation of the security infrastructure as well. This should be done without affecting the user as well.

Finally, it is of essential importance to ensure *effective access to the service environment*. In particular in complex environments this can be seen as the most important aspect.

2.4 Distributed workspaces – Integration of Human resources

The construction of complex products, like airplanes or complex buildings, enforces a lot of different domain experts to work together. To this end, this requires an enhanced support for distributed teams working in virtual enterprises that need to come together frequently to assess the project from different engineering points of view and to ensure that the product they are producing is meeting the specifications and the set quality standards. Such teams have both planned and ad-hoc meetings to resolve conflicts and to finalize the design. Such meetings are usually organized as traditional face-to-face meetings involving large organizational efforts and travel costs with direct impact on the environment and the quality of life of the team members, affecting also their social and family life. Furthermore, such face-to-face meetings are difficult to organize since the diaries of individuals need to be consulted to find a suitable time, consequently delaying the decision and the time for finalizing the product.

There is a clear trend towards simulating complex environments in advance before building a concrete physical prototype (cf. section 2.2). Although the simulation of such complex environments allows determining potential conflicts in advance before constructing a physical prototype, the simulation results have to interpreted and analysed by relevant experts in order to minimise the possibility of errors occurring during the operation of the physical construction.

As highlighted before, the construction of complex products requires the integration of complex, specialised IT services and the involvement of different domain experts, who have to work together in order to finalise the respective construction in time and within the intended budget. To this end, although specific knowledge is also integrated in included services, there is a specific need to also allow human experts to interfere in these complex processes and to enable them to collaboratively work on respective solutions.

The ongoing development of internet technologies is opening up new possibilities for people to collaborate across organizational and geographical boundaries. From the user's point of view, this means access to remote experts or colleagues, data and services on demand at any time and from any place. However, such on demand access needs to be managed within a given context to ensure that the privacy of others is neither invaded nor violated. For example, before contacting an expert the user needs to know the context of the expert. Is the expert currently available? Is he/she working or on holiday? Is he/she busy? What is the best way to contact her/him (video link, audio, chat)? This would require tracking the user context and making such information available to the rest of the team. Another context dependent issue is to understand the devices, interfaces, and services that are available to support the collaboration [148] [186]. By publishing such information, a collaborative working session could be initialized to suit the context with appropriate hardware and software settings.

Classically, a collaborative session is organized in advance by informing the relevant people and agreeing on a common date and medium. Additionally, the collaborative environment for supporting the meeting needs to be installed at each user site before the meeting. If such collaboration is a frequent activity, the collaborative technology infrastructure needs to be permanently installed at each partner site. However, current collaborative technologies only support access to a limited number of application services and therefore the type of discussions that could take place effectively is small. Due to the lack of integration between the applications and the collaborative environments, the users are not in a position to execute various business applications on demand for presenting their results to others and contributing to the discussion. Furthermore, current distributed environments provide little support for adaptation according to the available hardware configurations in order to provide an optimized service to the users.

Therefore, an enhanced collaboration environment is required, enabling also geographically distributed experts to get in touch synchronously in order to discuss and resolve issues being raised during the virtual construction. The following scenario reflects this issue. The respective implementation of this scenario is given in section 5.1.3.

2.4.1 Scenario description

Bob is a manager of a large European airplane manufacturer and has designed a process describing the steps for the construction of a specific part of an airplane. Since design as well as construction of the airplane is distributed all over Europe, management and process planning is of utmost importance. Additionally, the construction of an airplane involves a couple of external service providers which also have to be integrated in the execution of the process. For the sake of simplification, just a small extract from the workflow is considered, namely the distributed development of an aircraft fuselage. Figure 6 shows this simplified process.

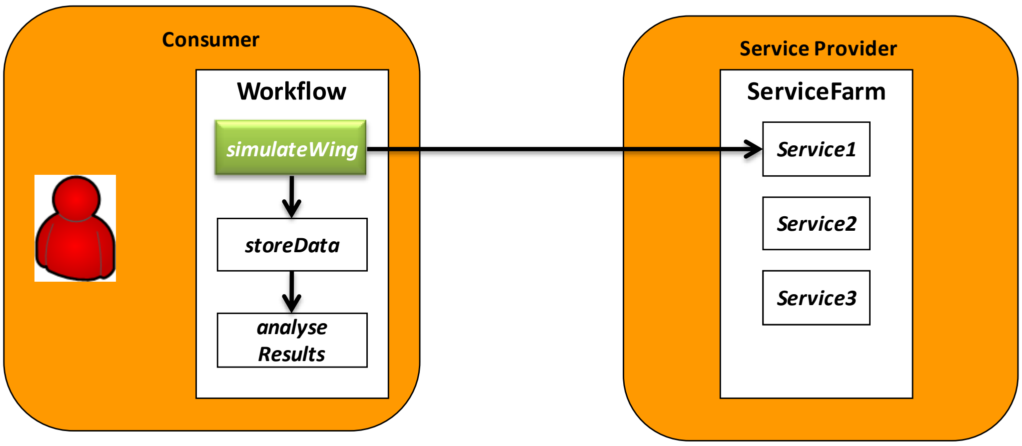

Figure 6 - Example Workflow

In this example, Bob designed a workflow representing the construction of an airplane and is responsible for its execution. To this end, Bob requires strong computational resources to execute the actual simulation, and therefore negotiates with different HPC providers to acquire their services. It may be safely assumed that the airplane manufacturer company, for which Bob works, already has a series of existing contracts with different computing providers. In order to enable the communication, Bob had to contact his IT department a couple of times in order to set up a corresponding IT-Infrastructure locally as well as for the integration of the target platforms of the corresponding service providers.

Therefore, Bob will want to run multiple parallel tests to identify the optimal configuration parameters such as lifting force, noise, material stress etc. Since optimisation involves multiple decision makers under different circumstances depending on the simulation results, Bob will devise a second process for improving the wing design. This process involves sending the design and simulation results to various experts, who may update the design and / or the simulation parameters, potentially causing the simulation range to spread out, to be altered etc.

Obviously, any of these iterations cause massive overhead in configuring the communication protocols, arranging for common discussion dates etc. – the more people involved in the discussion group, the more complex the setup. Even though the group may consist of full-time contracted participants, a more dynamic group that involves relevant experts according to the specific simulation results would speed up progress and improve the quality of the results significantly. What is more, since each iteration can be seen as a stand-alone process, there is no need to involve the same experts every time, but they may be replaced by experts from identical fields.

In summary, it should be possible to describe and execute such collaborations from the perspective of sharing data and consuming shared resources including computational and human resources. This would allow for an orchestration of resources with a high-level workflow language, like BPEL [194] without the need to consider the "technical" details about how the resource can be consumed, enabling also "typical" managing personnel without a profound expertise in technical aspects, to describe and setup such kind of processes.

2.4.2 Scenario context

As highlighted in section 2.2 and 2.3, the construction of complex products requires the involvement of varying domain experts, who have to work together in order to finalise the construction in time and within budget.

Therefore, an enhanced collaboration environment is required, enabling these experts to get in touch synchronously in order to discuss and resolve issues being raised during the virtual construction.

In this scenario, a European airplane manufacturer develops a new airplane. The respective engineers of the airplane manufacturer are distributed over different countries within Europe, whereas each country is constructing some specific parts of the airplane. Before the relevant parts are physically constructed, a virtual design phase takes place. In this design phase, 3D CAD models are produced for each of the parts. Finally, before the corresponding parts are constructed physically, these parts have to be coupled together in a digital mock-up. Figure 7 shows the virtual integration of different parts being constructed virtually at different places.

However, during this integration process different problems can occur, e.g. non-fitting parts or non-compatible connections. This is often due to the fact of the distributed design of the respective parts. To tackle these issues, the involved engineers have to get in touch with each other, resolve the issue and protocol the obtained solution.

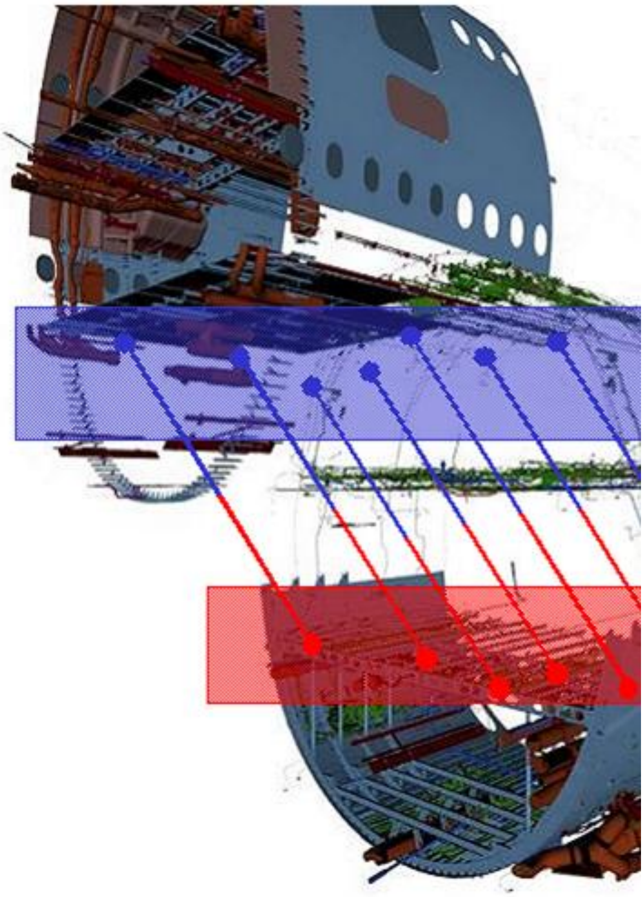

Figure 7 - The DMU integration process [36]

2.4.3 Identified key requirements

The facilitation of this scenario requires an enhanced support for the integration of human resources in well-defined process descriptions. In particular, in the engineering domain a synchronous exchange between different experts is of essential importance. In this context, a flexible and dynamic IT landscape is required as well, coping with changing users and thus collaboration contexts, like dynamic adaptation of access rights.

Even when teams can meet locally, significant administrative overhead is involved in gathering for common sessions, such as setting up the resources as needed and distributing the relevant data. These administrative efforts are even more demanding for globally dispersed teams. In truth, the setup of such collaborative working sessions requires several complex, and mostly manually executed, steps: required applications (such as a shared whiteboard) are installed and configured offline, with the respective configuration / login information being distributed by phone or mail. The problem becomes even more complicated, if more complex applications than a shared whiteboard are required, e.g. to discuss a 3D CAD model etc.

To ensure correct functioning, additional user context information is required, in particular:

- The current status of a specific user, such as availability
- Device status, such as resource type, installed applications/software etc.

This contextual information is necessary for initialising a collaborative working session, so as to determine if all the relevant team members, as well as the required applications, are available. Respecting the fact that in these collaborations the expertise rather than a specific expert is needed and assuming an enhanced collaborative environment allowing for the smooth integration of various collaborative engineering tools, enabling the geographical location of experts is becoming a minor issue. This proceeding enables companies to use their available resources, e.g. manpower and application services, more efficiently, as well as users to save time and effort in setting up an effective collaboration.

To this end, the environment should enable experts to synchronously collaborate with each other in a transparent fashion by providing an infrastructure dynamically adjusting the required services and access rights, whereas the control of the respective data sets remains at the data owner. The environment should enable to share relevant information in an easy and for the end user transparent way. Therefore, a corresponding role model should be supported to describe different roles in domains, and define the necessary access rights based on these roles. This proceeding ensures that companies can assign roles to different persons (like fuselage expert) and set up an environment with all required access rights for this role. If the person holding this role is not available, a representative can be chosen, and due the role-based access model this person can access the required data sets for a specific collaboration context.

2.5 Summary of identified key requirements

The following table summarises the identified key requirements from the described scenarios. As it has been shown, there are key requirements to be addressed in the fields of

- Simplicity (Ease of use)
- (Service) Provisioning
- (Service) Interoperability
- Time Constraints
- Reliability and Robustness
- Collaboration Support

Table 1 – Identified key requirements from scenarios

ID	Definition	Description
1	**Simplicity (Ease of use)**	
1.1	Support of different user types	Enable users with different levels of IT knowledge to use and integrate services in their own applications and services. In particular differing needs of users with different IT knowledge have to be supported. Exemplary users with profound IT knowledge want to be able to cope with technical details in order to optimise the service behaviour with respect to the objective to be tackled, whereas users with only limited IT knowledge prefer to "just" use the selected service without studying in depth the technical configuration opportunities.
1.2	Dynamic service coupling and orchestration	In order to allow dynamic service coupling and orchestration in a SOA-based environment, the technical complexity of the respective IT infrastructure has to be encapsulated from the users. In particular the integration, consumption and on-demand replacement of internal and external services should be doable in a transparent fashion.
1.3	Support the integration of heterogeneous application, service and IT environments	Allow the integration of heterogeneous environments in a transparent fashion.
2	**Provisioning**	
2.1	Enable the "Programming in the small" and "Programming in the large" paradigm	The framework should allow separating the development of services and components and the orchestration of these services and components to new services. In particular, this proceeding enables developers on the one hand to concentrate on the technical and functional aspects of the respective services. On the other hand, domain experts are not forced to take into account the corresponding technical details, but concentrate on their process knowledge and assemble the respective services and components respectively.
2.2	Service Lifecycle Management support	The environment should allow supporting the entire service lifecycle, namely the design,

ID	Definition	Description
		development, operation and transition of services.
2.3	Transparent and integrated accounting support	The service developer should not be forced to integrate certain accounting functionality within his service. The corresponding modelling and runtime environment should allow defining and applying the respective functionality in a transparent fashion by automatically considering the corresponding user context (e.g. differing QoS depending on varying service contract).
3	**Interoperability**	
3.1	Application and support common standards to ensure interoperability and flexibility	In order to ensure interoperable interactions between different development and runtime environments, common standards have to be applied by all vendors, which is the key for enterprise service integration. Additionally, it has to be ensured that changes in the environment and standards (also from different vendors, e.g. supporting various security standards for secure information exchange in a transparent fashion) do not affect the corresponding, agreed standards and formats.
4	**Time Constraints**	
4.1	Effective Access to service environment	It is of essential importance to ensure effective access to the service environment. In particular in complex environments an effective access can be seen as the most important aspect to be considered.
5	**Reliability and Robustness**	
5.1	Dynamic Process evolution	The environment should allow for the evolution of processes, describing required steps to be taken in order to achieve a specific goal. This is not limited to the structure of the process itself, but should also enable for the modification of services / service providers during runtime. The environment should allow the transparent replacement of services and the corresponding interfaces, without affecting the client application / customer at all.

ID	Definition	Description
6	**Collaboration Support**	
6.1	Synchronous collaboration support	Support synchronous collaboration between humans, in particular enable humans to collaborate synchronously (remotely) with each other, e.g. enable a scientists running a cancellous bone simulation to easily consult a doctor in case of unforeseeable situations, or enable an engineer determining a problem in the CAD model of an airplane to contact another expert directly (including the sharing of the corresponding data / information) in an easy way.
6.2	Role-based collaboration support	Ensure that only foreseen roles are enabled to consume specific resources, e.g. ensuring that the external design engineer is only enabled to access a specific CAD model not showing all technical details. However, with the reduced data set the design engineer is enabled to do his work, whilst meeting privacy constraints.

3. State of the Art

In chapter 2, common requirements for dynamic, flexible and heterogeneous B2B environments have been identified. In this chapter, the relevant existing concepts and approaches are analysed in how far they can cope with these identified requirements. This chapter follows a top-down approach, starting by analysing common solutions and approaches from an abstract viewpoint and ending up with the analysis of respective technological realisations, trying to face to the identified requirements.

In particular, with respect to the identified key requirements, the following field are analysed in detail:

In the first instance, an overview about *different user types* is given in section 3.1 in order to reflect the varying requirements from different user groups, which have to be reflected in dynamic B2B environments. This is of essential importance in order to ensure to satisfy the needs of a broad range of users groups.

In the following, in order to allow users to consume provided and for their purposes relevant services, common coupling mechanisms are depicted in section 3.2, namely by analysing common B2B environments, the so-called *Enterprise Service Bus (ESB)* environments.

Afterwards, the required conceptual approach to describe such kind of environments is analysed in section 3.3, namely the concept of *Service Oriented Architecture (SOA)*. This concept foresees to describe services in such a way to enable the integration in previously depicted environments. To this end, this approach is analysed in order to determine in how far the concept meets the identified requirements.

Following the analysed conceptual approaches, the relevant realisations are analysed. Therefore, in section 3.4 *Web Services* as intended technical realisation of the SOA paradigm are examined and analysed.

In section 3.5 common process description languages and runtime environments are analysed, in particular reflecting in how far these approaches allow for the description of abstract processes, whereas ensuring the required dynamicity and flexibility.

Additionally, in section 3.6 the lifecycle management of services is reflected by analysing the common ITIL approach.

In section 3.7 the integration capacities of human resources with common approaches are reflected. As identified in section 2.4, there is a need in integrating humans as well in complex engineering and construction processes, so this section will show in how far common approaches meet this identified requirement.

Finally, in section 3.8 a holistic overview about the coverage of the identified key requirements in chapter 2 by the analysed common, state of the art concepts and approaches is given.

3.1 User Profiles

When referring to resource decoupling within distributed applications this relates mainly to the “programming in the large vs. programming in the small paradigm” [44]. In particular, when referring to the “programming in the large” paradigm which signifies that services are consumable in an easy way as well as easy to integrate by ensuring interoperability between different services, the person assembling these services has to be considered. This person is a domain expert assembling available services, each providing a specific functionality, to a new, global application solving a specific issue.

To this end, three different types of users have been identified [122]:

1. The *casual user*: This user consumes services not very often, but is mainly focusing on default functionality. This user only has limited IT knowledge. He just wants to use the service without concrete understanding of the underlying processing and is interested to gain the expected service outcome following some specific agreed QoS criteria.
2. The *experienced user*: This user consumes services quite often and is experienced in using default functionality. However, this user type also has just limited IT knowledge and is not able to do specific adaptations by himself.
3. The *expert user*: This user has a specialized and profound IT-knowledge. He uses services very often and is able to do specific adaptations / programming actions. In particular, this user is also able to optimize the service by taking into account the respective software and hardware environment, e.g. by optimising an HPC application for specific runtime environments.

3.2 B2B Collaboration – The Enterprise Service Bus

In order to allow such a dynamic SOA based environment as depicted in section 2.2 and 2.3, in particular with respect to the goal of service providers to hide their service infrastructure from their customers, the essential backbone of such an environment is the communication infrastructure, namely the so-called Enterprise Service Bus (ESB). This communication infrastructure should conceptually allow for both, the interconnection of local services within a domain, as well as the orchestration and integration of services from external service providers [78] in a transparent fashion. The conceptual approach of an ESB is given in Figure 8. As depicted in this figure, the ESB is an intermediate layer between different service instances acting as a messaging bus. This messaging bus allows different services to plug-in to this bus in order to receive and send messages from and to different services. From a conceptual perspective the ESB should enable this kind of service interaction in a transparent fashion, however, since these interactions have to be modelled in such a way that the respective services are able to process the received messages, from a technical perspective this concept cannot be addressed with current environments, in particular since current realisations burden service consumers to consider the relevant service provider’s runtime environment and used technologies. As depicted in section 2.5, a service consumer just wants to use the respective service and get back the corresponding results. However, current environments based on web service technologies only provide a limited support for such kind of environments [98], making their usage a limiting factor in such dynamic environments. A detailed overview of established ESB realisations is given in [140].

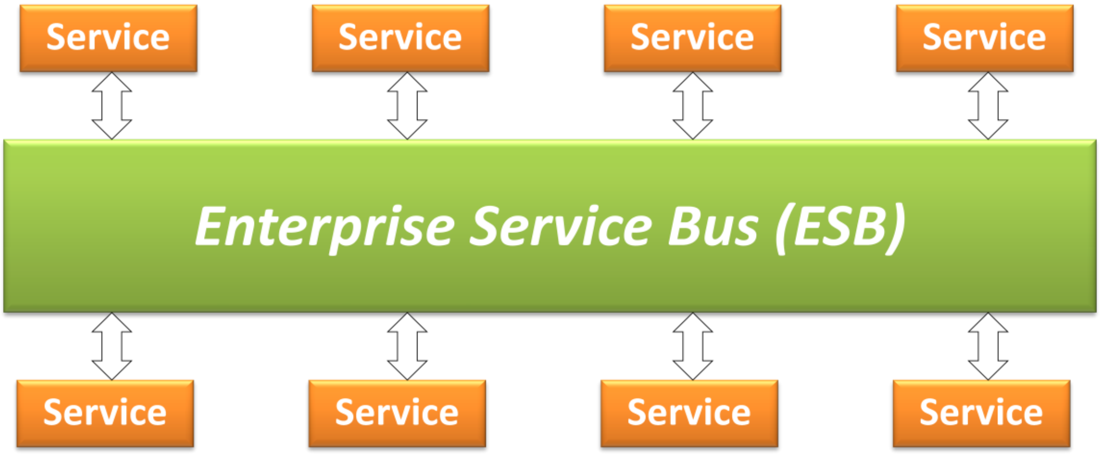

Figure 8 – Conceptual view Enterprise Service Bus (ESB)

Additionally, common ESB Systems have problems in recovering from unforeseen errors. This is mainly due to the missing support for evolution during runtime [12]. Furthermore, most ESBs are designed to allow for an orchestration and integration of services within a specific domain. Typically, the ESB ”ends“ at the boundaries of an organisation[61], not allowing to integrate services from external service providers as well.

The ESB within a conceptual model should be able to act as a mediator service, beside others, allowing for the virtualization of service endpoints, the transformation of messages as well as the loose coupling of services. Therefore, the ESB acts as interconnect between services [152]. In order to allow such an environment, service metadata is seen as one of the essential criteria for dynamic service mapping. In particular, the corresponding service provider has to announce an interface and the relevant invocation modalities to the ESB. This announcement of interfaces enforces the service consumer to have a concrete knowledge about the services he intends to invoke, namely the service location as well as the syntactical interface.

In order to tackle this issue of this close binding between services, in [108] a mapping of BPEL elements to pub/sub semantics is introduced, allowing the realisation of a defined BPEL process within a distributed agent environment. Therefore, every agent takes over a specific task within the workflow. The message routing is realized with a hierarchical message broker infrastructure. However, this approach does not allow for the provision of complex, aggregated IT services. Furthermore, the hiding of the underlying infrastructure by the published service endpoint of a service provider is not considered as well.

Other approaches facing this challenging issue of dynamically mapping service invocations to potentially changing service providers [127], proposing a new approach for monitoring and dynamic service selection for BPEL processes. Therefore, an Aspect-Oriented Mapping approach is introduced allowing for the dynamic mapping of messages, depending on previously defined rules. Although this approach allows for a dynamic adaptation of service invocations, this message interception approach is limited to BPEL processes and does not support other runtime environments. The major gap in this approach is that it does not foresee an interception or virtualisation on the service provider side, which is a limiting aspect with respect to dynamic environments.

Furthermore, an ESB has to provide an extensive support for [123]

- Security issues, in order to ensure that only foreseen users are allowed to access and consume the relevant services, as well as to ensure data privacy and protection;
- Adapters, in order to allow for the integration of legacy applications with no web service interconnection, as well as for the transformation of messages and data;
- Management and monitoring opportunities, in order to allow adjusting the environment during runtime, whilst also ensuring to provide a detailed overview of the current runtime state. The latter one can also be taken into account in order to adjust the ESB setting respectively, e.g. in order to invoke services with a better energy-efficiency behaviour for specific tasks. This could be determined by taking into account the corresponding monitoring information;
- Process orchestration, in order to allow the combination of existing internal and external services;
- Complex event processing, in order to allow an intelligent steering of events and the corresponding actions, in particular in order to support the processing, the correlation as well as pattern matching for the raised events;
- Integration tooling. Ideally, this tooling should enable user to define their processes and flows within a graphical user interface. Furthermore, an extensive support should be given for deploying and testing the respective services.

The ESB can be seen as the conceptual backbone for a SOA-based environment, enabling loose coupling of services by providing a corresponding messaging infrastructure [179]. This conceptual approach simplifies the usage of internal as well as external provided services significantly, e.g. compared to CORBA.

These proprietary approaches, like CORBA and EAI, did not address the identified requirements as well, in particular with respect to loosely coupled, distributed SOA environment, since these approaches rely on closely coupled interfaces [28]. Due to the proprietary nature of these approaches, the respective EAI [43] and ESB [28] solutions do not allow for dynamic B2B environments [39].

There are also similarities to the concept of message oriented middleware environments (MOM) [116]. As depicted before, an ESB should enable to provide virtual communication and interaction channels to services, including also service outside of the organisational boundaries. Taking this into account, the amount of required connections between the services can be reduced. This is due to the fact that all services have to connect to the ESB only. Otherwise, if all components have to connect to each other, at most $\frac{n*(n-1)}{2}$ connections are needed, by applying an ESB just n.

Therefore approaches, like depicted in [86] and [133] conceptually introduce a central messaging interception. However, these approaches do not allow to couple also cross-organisational services and resources, since every service provider runs his own respective environment.

Additionally, a specific support should be given to web service based environments, which are seen as "technical" realisation of the SOA approach [98]. In particular, the integration and usage of abstract business services and entities has to be supported in order to allow for the required dynamicity. To this respect, an intelligent routing has to be supported, in particular with respect to message routing based on the corresponding content, context and / or business rules. Furthermore, in order to allow the dynamic replacement of services within a SOA based environment, a respective transformation support has to be given. This allows for the transparent replacement of services in a running environment without affecting the invoking applications / services at all, which are not addressed by current approaches [192].

3.2.1 Summary

To summarise, an ESB should address concrete IT and Business needs, namely the need to support loosely coupled and distributed business units and business partners [28].

To this end, dynamic SOA-based environments have to support abstract service endpoints in order to hide the technical details of applications / services [96], in particular when taking into account the integration of domain internal services as well as of services from external service providers. In particular the latter one allows the provision of complex IT products by

- Orchestrating existing internal and external services
- Allowing for the provision of this aggregated, complex IT service to external customers in a transparent way
- Allowing for the integration and provisioning of explicit domain knowledge (process knowledge) in an encapsulated, transparent fashion

However, these essential requirements are not met by current solutions, since these enforce a strong collaboration between technical and domain experts and thus do not allow for the required, transparent dynamicity [192].

The ESB is the central integration component for a SOA-based environment by providing a messaging channel allowing for the delivery and on demand transformation of messages, whereas additionally methods for authentication and authorization should be provided and supported in a transparent fashion as well. However, as soon as the organisation boundaries are left, the integration and coupling of external services and resources require manual integration efforts, and thus hinder the dynamic, transparent management and operation of these environments [98].

3.2.2 Addressed key requirements

As depicted in chapter 2, simplicity of service usage as well as service provisioning is of significant importance in order to ensure a dynamic, flexible SOA-based environment. Hiding the technical complexity of the underlying IT infrastructure is an essential issue to be addressed to meet these key requirements. To this end, there are several proprietary approaches available trying to address this issue in a restricted way. In particular, existing solutions focus on the integration via non-standardised wrapper techniques, not allowing supporting third-party services as well. This becomes an additional limiting factor when referring to cross-organisational collaboration scenarios, which are supported only when the respective collaboration partner is using the same environment as well. This proceeding enforces a strong collaboration between domain and technical experts as well, dissenting from the intended SOA paradigm. The integration of external services requires a manual integration effort, hindering significantly the intended flexibility and interoperability. In particular the latter one deals with the dynamicity required in dynamic B2B environments, especially with respect to changing service providers during application execution. The integration of non-ESB compatible services enforces a manual integration effort, and thus does not allow a continuous operation, resulting in decreased flexibility. Especially in case of occurring errors during application execution most proprietary ESB environments allow for an internal error escalation. However, in case of errors only internally known services can be considered in order to replace a failing one, as depicted in [171]. These approaches cope with an enhanced error and recovery management, but being limited to specific environments, in the mentioned case to BPEL processes. Additionally, these approaches are all limited to internal services only. In case of integrated external services there is no possibility to cope with this situation in an adequate way.

3.3 Service Oriented Architecture (SOA)

A Service Oriented Architecture (SOA) describes an architectural pattern within IT based systems. In particular, this paradigm foresees to describe different services to be combined to new services / IT products as components in an abstract way, enabling developers to orchestrate these components based on the required functionality to new, more complex applications and services. Technically speaking, this concept has been adapted from the electrical engineering domain. In electric engineering it has been determined soon that, in order to build up complex devices and machines, this complexity can only be handled when going for a hierarchical approach by encapsulating functionality within well-defined interfaces, enabling the device layer to combine the provided functionality with other components of the device, without the explicit burden to take into account the implementation details of the integrated functionality. This proceeding enables in particular to build up new devices by also integrating components from external vendors, which are often specialised for specific functionalities and thus can provide the corresponding services with a better cost-benefit ratio. Implementing all functionality by itself would cause significant costs for the entire development process, a long duration until the product is finalised, as well as a potential high effort during product maintenance since in this context there is no access to a corresponding support by an external vendor.

In other words, the SOA paradigm describes a methodology targeting on application and service interoperability and reusability. Even the design and implementation of operating systems starts following this approach [157] [153] [154]. The major focus of these approaches is the architectural viewpoint of an environment, in particular the inter-relation and composition of services are considered in this context, but not the concrete implementation of these services. Therefore, an essential aspect is as well the context-independence between different service instances. This means no explicit dependencies are allowed between services, since this would be a limiting factor in dynamic IT environments due to the fact that changing a service instance would also require transmitting the respective service context.

To sum up, the major intended benefits of SOA environments are:

- *Service re-usability*: due to the ensured context-independence the provided services can be seen as atomic units, which can be re-used and combined with other services.
- *Efficiency*: new services and applications can be easily and efficiently developed by combining and coupling new and already existing services.
- *Loose coupling between the provided services and the underlying technology*: By separating the service interfaces from the underlying technology a decoupling of the service implementation from the service environment is achieved. In particular, since from a conceptual viewpoint just the message exchange between the involved services is considered, new services and applications can be modelled without the need to consider the underlying technology, enabling the developer to concentrate on the problem to be solved.
- *Distribution of responsibilities*: by decoupling the process description from the technology layer of service runtime environment new libraries can be developed reflecting the requirements of different users' types. For example, a process designer is not interested in the technical details of a service implementation he intends to integrate within his process. The process designer wants to concentrate on his original tasks, the development of process patterns. However, the operator of the runtime environment of such process models is interested in the characteristics of the runtime environment. By decoupling process descriptions from technologies one is enabled to develop technology-independent libraries, describing steps required to achieve a specific goal. This separation is referred to as the "programming in the large" and "programming in the small" paradigm [44]. The first one refers to the development of services and applications by combining and orchestrating existing and new services on the basis of well-defined interface descriptions. The latter one refers to the development of services within a specific runtime environment.

The essential difference to the common Object Oriented Programming (OOP) paradigm is the focus on the message exchange between services within a SOA-based environment. This abstract contemplation of the interaction between services describes a higher abstraction as being reflected by the method signature of a class definition. The latter one typically contains parameters required by the runtime environment, whereas the abstract message exchange focuses on the data to be submitted as well as the semantics of the calling context.

Another advantage resulting from a SOA-based environment is the simplified realisation of outsourcing projects [13]. This is due to the fact that, within an SOA environment, all services can be accessed via interfaces abstracting from the underlying technologies, and thus also services from external service providers can easily get integrated. This decision might be taken in case that an external service provider offers services which are more efficient compared to the own realisations, or that a service provider offers services that are not yet available or realised. A service is more efficient than another one, if it provides at least the same functionality, but causes fewer costs during execution. By taking into account QoS and SLA related aspects as well, a SOA enables to easily integrate external services.

In addition, by following the SOA paradigm, services and applications can be designed, realised and orchestrated at varying granularities. This is mainly due to the mentioned context-independence of services, enabling the simplified combination and orchestration of these services. These combined services can be offered again as new services, thus enabling to realise specific sub-tasks by dedicated services, and subsequently orchestrate these services to a new service, covering a larger scale of tasks. Following this bottom-up approach enables reducing the complexity in designing and developing complex systems, whilst increasing the maintainability.

The message exchange between services follows certain patterns, namely the message exchange patterns (MEP) [50]. These MEPs define how messages are to be exchanged between services, e.g. by defining if a synchronous interaction should take place following the MEP “request/response”, or if an asynchronous interaction should take place following the MEP “notification”.

The idea to decouple the interface and the concrete service implementation is not new. The first attempts of this approach can be found in the realisation of the Component Object Model COM [169], the Common Object Request Broker Architecture CORBA [169], the Windows Communication Foundation (WCF) [184] or the Java 2 Platform Enterprise Edition [169]. The major difference compared to the SOA paradigm within these realisations is the close coupling of the interface definition with the respective runtime environment. Within an SOA environment the service interface has to be decoupled from the corresponding runtime environment.

3.3.1 SOA stack

To describe a SOA requires certain functions and roles. Basically, the realisation of a SOA-based environment consists of the following three roles

- *Service provider*: A service provider offers a specific functionality via a well defined interface. In particular the latter should enable to consume the offered service in an interoperable fashion. This interface description, reflecting the available functionality, is also stored in a *service registry*.
- *Service registry*: The service registry allows for storing and retrieving interface descriptions of services. Furthermore, additional descriptions about the service can be provided as well.

- *Service requestor*: The service requestor inquires a service satisfying his specific needs. These requirements should be met by a service, whose description is stored within the service registry. Therefore, the service requestor queries the service registry. Subsequently, the service registry provides the relevant service information, in case a service fitting the service requestor requirements has been found. Finally, the service requestor can invoke the best fitting service via the retrieved interface.

To ease the proceeding for the service requestor, a middleware solution can transparently enable him to consume services without being forced to query a service registry in the first instance. This middleware is typically called a *service bus*. This concept is described in more detail in section 3.2.

3.3.2 Conceptual approach

Since the complexity in developing and providing IT services is increasing continuously, the concept of SOA has been adapted for IT services and applications as well. Laskey et al. define a SOA based environment as follows [106]:

"From a holistic perspective, a SOA-based system is a network of independent services, machines, the people who operate, affect, use, and govern those services as well as the suppliers of equipment and personnel to these people and services. This includes any entity, animate or inanimate, that may affect or be affected by the system".

In other words, a SOA describes a paradigm reflecting a holistic, abstract and ideally technology independent viewpoint on the complexity of IT service environments. In particular, within a SOA environment, the following perspectives have to be considered:

- the services being provided;
- the corresponding IT infrastructure;
- the people operating, affecting and using these services;
- external suppliers providing equipment and personnel support to operate these services.

The major impact of a SOA-based environment is to decouple service implementations and the corresponding interfaces, in order to enable an interoperable integration and orchestration of resources. In particular, from a technical perspective, this so-called loose coupling refers to binding services as late as possible to specific interfaces, as well as to the messaging infrastructure. The latter one is necessary since otherwise a loose coupling would not be applicable. A messaging environment, comparable to the eMail principle, allows invoking a service also in an asynchronous way, and thus enables an application to submit an invocation and let the messaging environment take care of all required adaptations to fit the invocation for a specific service implementation [122].

Based on this service oriented approach and decoupled resources, a directory service has to be provided as well, enabling service consumer to search, find and bind the best suitable service at runtime for their applications, providing the functionality required in a specific context. The latter one may also take into account specific QoS and SLA criteria that might change during the execution of an application. Typically, there are three types of directory services provided, namely

- *White pages*: This is the most basic way in finding a service from a service provider for a specific application. This kind of directory service allows browsing the overview of provided services from a specific service provider.
- *Yellow pages*: This kind of directory service groups the available services based on their provided functionality. Thus, this directory service enables to search for a specific functionality without the need to select a service provider in the first instance. In particular this enables to find the most suitable service provider for a specific functionality.
- *Green pages*: This model of a directory services contains the interfaces of the corresponding services. So this model enables to search for a relevant interface specification. This proceeding enables to determine the best suitable service for a specific issue from a syntactical perspective.

Following [122], *"A SOA describes a system architecture representing manifold, different and even eventually incompatible methods and applications as re-usable and free accessible services, thus allowing for a platform and language neutral usage and re-usage."*

In order to achieve this, the usage of commonly agreed standards is of essential importance. In addition, the usage of services in this environment should be applicable in a simple way, following the "programming in the small and programming in the large" paradigm [44].

In addition to the technical perspective of consuming, integrating and orchestrating service, the lifecycle of a service within a SOA-based environment has to be supported as well. Namely, the following phases have to be supported:

- *Design*: combination of existing modules to provide new services or define completely new services
- *Operation*: running services as intended in the SOA model
- *Evolution / Change*: long runtimes are not unusual (~30 years possible). Changing requirements / service environments enforces the adaptation of the affected application and services.
- *Service Shutdown*: controlled shutdown of specific services which are no longer used. This refers in particular to the required context-independence between involved services as stated before.

A detail overview of the service lifecycle management related standard ITIL is given in section 3.6.

3.3.3 Application to B2B environments

Today's eBusiness scenarios require a consequent realization of the SOA paradigm, providing benefits for both service provider and service consumer: service provider can easily expose their services in the sense of aggregated business processes in such a way that potential service consumers can integrate these services in their own services. This has to be done in an abstract way which means in particular that no implementation details of the underlying service implementation need to be considered. This enables service consumers to use provided services without worrying about integration and interoperability issues, providing them the required flexibility in dynamic B2B environments.

To achieve this, in particular operational integration and lifecycle issues have to be faced by the respective environments, making these aspects critical success factors for SOA [129].

The abstraction of services from their implementation is already an integral part in our everyday life, e.g. online banking services hide the complex processes to execute financial transactions behind comparatively simple interfaces. Even more commonly, DNS servers replace the complex and dynamic IP addresses in a network with (potentially meaningful) names, thus enabling the network users to connect to machines in an intuitive fashion.

In modern eBusiness scenarios it is necessary to decouple service implementation from the corresponding interfaces in order to maintain the services efficiently, without affecting users or providers.

3.3.4 Summary

In summary, the SOA paradigm provides a conceptual approach easing the combination and implementation of services. Therefore, commonly agreed standards have to be followed, because otherwise the coupling of services from different service providers enforces in-depth knowledge of the respective IT environments.

3.3.5 Addressed key requirements

However, available standards are defined in a fuzzy away, allowing for "some" interpretations with respect to the standard implementation, and thus results in different and incompatible realizations. In particular the interoperability between different platform providers is not given in this context [18].

In addition, SOA mainly focused on computers by providing mechanisms to automate the communication between different service instances [122], but not on the integration of human resources.

The SOA paradigm is intended to enable for the "programming in the small" and "programming in the large" paradigm [44]. "Programming in the small" stands for the concrete implementation of a specific service, and thus enforcing a profound technological knowledge of the respective environment. On the other hand, "programming in the large" enables user with specific process knowledge to combine and orchestrate services without the need to know in detail how the respective service is working. At this level, only an abstract view of the corresponding services is required.

Additionally, it is of essential importance in a B2B context to provide an integrated, transparent accounting mechanism at the process definition level in order to charge customers for using a specific service. This mechanism has to be decoupled from the service implementation. This enables the service developer to concentrate on the respective functionality, but not on eventually changing charging mechanisms following the company's business model.

Finally, existing environments trying to face the identified issues do not address the required technology independence and flexibility in an adequate way, due to missing support of the different perspectives required to develop and maintain SOA-based environments [84].

3.4 Web services as SOA-Enabler

As depicted in section 3.3 it is of essential importance for a SOA-based environment to decouple the service interface from the corresponding service implementation. The conceptual realisation of the SOA paradigm as described in section 3.3 is now going to be mapped to a concrete technical realisation. Namely, web services have been seen as the technical realisation of the SOA paradigm [137], so in this section a brief description of the respective software stack is given, as well as an analysis about the gaps of current web service technologies towards a SOA-based environment.

3.4.1 Technical concept

Basically speaking, web services encapsulate a piece of software, allowing for the interoperable invocation of these services by using XML artefacts, relying on messaging-based interactions [5][185][137]. As depicted in section 3.3.1, the SOA concept involves different roles which have to be mapped now to concrete technologies. The technical realisations of these roles are mainly driven by three standards, namely SOAP [65], WSDL [31] and UDDI [33]. In particular, SOAP describes the messaging format allowing for the creation and delivery of invocation messages between services. WSDL describes the corresponding interface description language, targeting on the interoperable access to the respective services by encapsulating the technical details of the service implementation. Finally, UDDI provides the required registry functionality, enabling service providers to announce their service as well as potential service consumers to query this registry in order to find a suitable service for their purposes.

This announced web service stack is principally based on the W3C architecture [19], consisting of the following major building blocks [198]:

- *Communication*: In order to enable services to communicate with each other, a network infrastructure has to be provided. In addition, protocols have to be provided reflecting how messages have to be encoded and get transported to the target service, e.g. HTTP and SMTP
- *Messaging*: An essential requirement is to enable the invocation of complex operations. Therefore a document-style communication based on the communication layer has to be realised, depicting in detail how messages are assembled reflecting the operation to be invoked as well as the respective parameters. The most related web service standards in this context are
  - o WS-Addressing [21], describing the service endpoint the invocation message has to be sent to. This standard also enables to address state-full services.
  - o WS-Routing [131], providing a pre-defined route the message have to follow in order to get delivered to the service implementation.
  - o WS-Reliability[75], providing mechanisms to ensure different levels of reliability in delivering messages.

- *Description*: The description provides a functional description of the web service, reflecting the relevant functionality of the service, like WS-Policy [197] or WS-Agreement [193].
- *Discovery*: After a service has been implemented and described, it has to be published to potential service consumers.
- *Processes*: This layer enables assembling services by taking advantage of the underlying layers, and thus being enabled to orchestrate services in an interoperable and re-usable fashion.

Other approach tends towards a central instance managing all interactions via a conversation manager [16] by modelling web service conversations via a central instance, namely the service manager. However, since this approach may result in a single-point-of-failure by the service manager, it is not followed, since within dynamic B2B environments it is of essential importance to rely on a flexible, reliable and stable communication backbone.

3.4.2 Interoperability and standardisation

In order to tackle the required interoperability between services, the concept of web services is mainly driven by standards [180], ensuring the broad acceptance of web service technologies as enabler for an interoperable service platform [105]. In particular, interoperability is identified as key for enterprise integration [181].

Most web service standards are not built from scratch but on top of already existing technologies. Unfortunately, the community did not address this issue so far. Currently, there are too many, partially incompatible, overlapping and also contradicting web service standards available [122]. Altogether, there are more than 100 of competitive web service standards announced, all covering different, partially overlapping aspects. One reason for this rank growth of standards was that most web service specifications have been described in a quite rudimentary way, resulting in implementations closing this interpretation latitude and thus being incompatible with respect to technical details. To some degree this proceeding was intended by several solution providers to ensure that solutions from competitors are not compatible with the own provided solution for a specific web service standard, and thus binding customers with this proceeding to the own products and services, since integrating third party services enforces a significant integration effort. This entire standardisation process can be seen as a fragmented process, resulting in grown and partially incompatible standards [198]. Currently, there is a variety of definitions of web services available, mainly focusing on technical aspects [77]. However, the view behind the technical façade is not considered at all, thus limiting this approach to a technical one not taking into account the conceptual SOA viewpoint.

In this context there was also a paradigm shift with respect to how services have been designed and assembled. Due to the large number of web services available, the original “web” moved from a data-oriented infrastructure towards a service oriented one [139][161][71]. So, at the beginning the content of a web resource, which was typically a web site at the very beginning, was the dominating factor when designing and building resources being provided via the web. Nowadays, the web has moved to a service-oriented environment, also providing content but with a clear focus on providing services via this flexible infrastructure. Already in 2002 it was conceptually foreseeable that using this flexible, service oriented environment is going to enable moving business logic to this “web of services” as well [182]. Following this development, cooperation between and dynamic combination of services will allow for better results by providing value-added services [121].

To allow such an environment, the major enablers for this “sea of services” are the corresponding, standard-based web service technologies, as identified in [139].

Of course, in this context, interoperability is a key issue, because otherwise integrating and combining services would not be doable. Such a web service based framework should allow for a robust service composition. These compositions target on the provision of added-value services by composing available or outsourced services [162].

Assembling and combining existing services in order to provide new services is done by process modelling tools (section 3.5), which can be seen as the major tooling opportunity to compose web services [72].

As depicted before, it is of significant importance to allow not only an interoperable invocation of services, but also an understandable and interoperable description of the respective web services. Services from different service providers realising similar functionality typically only differ in the syntactical description of their interfaces. To face this syntactical issue, the application of ontologies is the enabling concept to allow for a semantically interoperability as well [52], targeting towards a common understanding across domains of people, organizations and applications [51]. In particular Yu et al. identified ontology and mediation techniques as essential key points in order to ensure interoperability within web service based system standards [198]. Enhancing web services with ontologies enables to determine suitable services not only by taking into account the syntactical interface, but also the “meaning” of the corresponding service method or the respective invocation parameter. Therefore, ontologies are acting as a metadata-schema enriching the describing capabilities of e.g. syntactical interfaces [48]. However, due to a lot of different approaches in this field, the initially intended interoperability is not given [159].

As shown, interoperability is the key requirement for a service oriented web [120]. Additionally, QoS has also to be measured to distinguish and rate web services from different providers. The management and thus fulfilment of QoS aspects is seen as the key concept of distinguishing services and service providers [34]. Interoperability can be achieved by using standardized common schemas and interfaces [15], but within dynamic B2B scenarios this is not doable, e.g. when a new service provider enters the market by offering a new service or an existing one modifies his IT infrastructure and thus has to adapt the service interfaces as well. Furthermore WS-standards advance over time, resulting in differing, non-compatible realisations [160].

Altogether, interoperability can be achieved by referring to common standards, e.g. released by a standardisation body or common “accepted” de-facto standard from the industry, ontologies and mediation techniques. The latter one applies in particular when different types of interaction patterns and protocols, like integrated RPC calls, data in non-XML format, or proprietary environments, are applied [6].

Ontologies refer to a formal and explicit specification of a shared conceptualization [46], whereas conceptualization refers to an abstraction of a domain identifying the relevant concepts. Shared means in this context that ontology captures consensual knowledge as well. The Definition of ontologies is a collaborative process involving different parties, like business experts and sociologists. Mediators as enabler for an integrated view over multiple and heterogeneous services allow the translation between different services [63].

In summary, interoperability affects two levels - the syntactical as well as the semantic level. In particular, in case of different used protocols a mediator service has to be used in order to ensure that the coupled services are able to interact with each other, without the need to adjust the respective service implementations, because this would harm the announced SOA paradigm stating that every service has to be consumable by just referring to the service interface.

A lot of efforts have been spent to define XML based interoperability frameworks [26], as well as automated schema matching mechanisms for database systems [144]. However, these frameworks concentrate on specific use cases and provide the respective service templates and descriptions, thus limiting these approaches to specific application fields only. Two famous examples are *xCBL* (XML Common Business Library), providing a definition of a set of core business documents in order to achieve a common understanding between different involved entities within a business process, and *cXML* (Commerce XML), defining an XML-based schema language for interactions as well as protocols for transactions in business interactions.

3.4.3 Security

In an environment focussing on loose coupling of services and linking these services via the internet security mechanisms have to be supported and smoothly integrated [53]. In particular when taking into account that all communication is transferred via the internet, it is of significant importance to ensure the integrity of sent messages, the encryption of the content (which might be business-critical), as well as authentication and authorisation [57]. In particular the latter one has to ensure that only foreseen customers are allowed to consume specific services, also taking into account varying QoS criteria as agreed in the respective SLA.

Furthermore, privacy is a critical aspect in distributed computing via the internet [147]. In particular, during interactions with services personal data or business secrets might be unintentionally released [1] and thus have to be protected respectively. Therefore, the required technologies for ensuring privacy in web service-based environments are digital privacy credentials, data filters and mobile privacy preserving agents [146]. In addition, [117] highlights that a flexible, declarative specification of user credentials is necessary, e.g. providing beside the standard login information also information about the location, age, context, etc.

Thus, a dynamic infrastructure, also allowing the transparent integration of security issues, is essential for the provision of dynamic SOA-based environments [47].

However, integrating all these mechanisms enforces a lot of technical work, also affecting the corresponding service implementation, thus violating the originally intended SOA paradigm.

3.4.4 Lifecycle management

As mentioned in section 3.4.2 one of the basic requirements to enable an interoperable framework is referring to standards, describing and specifying in detail how interactions, service description and localisation has to take place. This section addresses the lifecycle management aspect with respect to provided support within common web service based frameworks. In section 3.6 a detailed overview about the common ITIL framework is given.

Besides the concrete specification of webservice standards, a concrete lifecycle model has to be provided as well, describing in detail the development, deployment, publishing, discovering, execution, composing, monitoring, access, optimisation during lifetime and finally the destruction of a web service [177]. These tasks are conceptually handled by a web service management system (WSMS). In [198] a corresponding system is introduced by referring to lessons learnt in the database development of the last decade.

Another approach splitting the management of the service lifecycle is introduced in [136]. The authors provide a model relying on a manageability information model and a manageable infrastructure. This splitting has been done in order to allow an independent description of web services, whereas being able to manage and steer the infrastructure separately. In particular this manageable information of web services is described via standardized interfaces for each layer as well.

In summary, all these approaches have in common that they require a mediation service to enable the intelligent steering and transformation of invocation messages. However, such an approach is not considered in the related approaches. Due to this missing kind of service, the relevant routing logic has to become part of the service implementation, which is not compliant with the introduced SOA paradigm in section 3.3.

3.4.5 Current web service-based frameworks

Current web service frameworks support partial SOA conformant collaborative infrastructures but still lack essential dynamicity capabilities in eBusiness and collaborative working scenarios [95] [94]. An example of the latter type of scenario is addressed by the research project CoSpaces [35]. This project aimed at developing a framework enabling dynamic collaboration sessions for engineering teams that are distributed all over the world – the main issues to be addressed in this context consist in allowing the relevant people to share knowledge and necessary applications with one another. Approaches like [166] focus just on the provision of virtualized workspaces by providing the corresponding computing resources for a specific job.

Existing virtualization systems, e.g. [141] provide a first step towards a virtualization infrastructure abstracting from the technical details of the underlying service infrastructure – generally however, these approaches are limited to specific platforms and do not allow the relevant dynamicity due to limited configuration and adaptation capabilities. Other approaches tackle the virtualization aspect at a lower technical resource level, like [130] or [135]. However, these approaches do not respect the high-level requirements of eBusiness environments, which require the ability to abstract from the underlying, low-level resource infrastructures. The latter one has been addressed by [9]. However, the presented approach focuses on a quite static adaptation of web service infrastructures during runtime by supporting static mapping documents, allowing the referred mediator service to map requests between varying WSDL descriptions. In addition, by considering design and runtime environment separately, only known combination of services and exceptions can be reflected. This implies a significant decrease in dynamicity due to the fact that only well in advance known replacements can take place.

As depicted in section 2.2, an important aspect in SOA-based systems is the consideration of evolutions of the involved components. To tackle this issue, [168] therefore presents a connector component allowing mapping component invocations to different versions of the service interfaces. However, this approach lacks in facing the requirement to dynamically and transparently adapt the underlying service infrastructure. In particular, this approach does not allow replacing components without affecting the other involved components. In order to determine the current version of a service interface [79] announce an approach describing how to include also older versions of interface definitions within an interface description by enhancing UDDI. The presented approach foresees to allow service clients to get the correct interface as required by their setup. However, this approach also just considers changes within an interface of a specific component, whereas not abstracting from the syntactical interfaces of the respective services. Another approach tackling the evolution and versioning of web services is introduced by [80]. This approach tries to tackle this issue by introducing an adapter chain in order to reflect the changes of the relevant service interfaces. Therefore, for every change an adapter is provided ensuring that invocations for the out-dated interface can be matched to the current one. In order to reflect all changes of an interface, these adapters have to be organized in a chain. However, this approach does not support the virtualization of service endpoints. Additionally, this approach just supports the service provider in maintaining different versions of service interfaces. The customer side is not considered in this context.

The presented toolset in [118] enables to deploy applications on the grid by following a SOA-based approach. However, this approach does not consider abstract but static service endpoints, thus not allowing the required flexibility and dynamicity.

To this end, [66] coins the term “abstract entities” for service providers highlighting the need to hide complex, potentially distributed processes behind a straight-forward, abstract interface allowing for the integration of partners into a higher-level eBusiness process. This has to be achieved by representing partners as abstract roles. This way, collaborative processes implicitly form a hierarchy of processes, thus enabling the design of collaborative eBusiness.

3.4.6 Summary

The introduction of web services was intended to provide a technical environment for the SOA paradigm, as depicted in section 3.3. However, this approach did not succeed with this respect. Typically additional programming efforts are needed when using web services, in particular

- when data is being transferred during service invocation, especially in eBusiness environment, additional manual transformations are required and have to be applied;
- orchestrating heterogeneous services fails due to incompatible WS standard realisations. Depending on standards used and how these standards have been applied, this service orchestration requires additional integration and developing efforts, thus not addressing the required flexibility and dynamicity;
- since in cross-domain invocations additional context information has to be transmitted as well. This context information is service specific, and thus not standardized. Therefore this context transmission and interpretation has to be implemented within the respective service, including e.g. which user is allowed to access service with respective QoS, accounting information and usage conditions.

In summary to cope with these issues a profound technical understanding of the underlying service infrastructure is required, contradicting with the essential requirement for SOA-based environments to decouple service implementations from the service interfaces.

This is mainly due to the fact that the classical web has been designed and developed for information exchange, and not to act as a web for services.

To face the interoperability issues not addressed by current web service-based environments semantic enrichments have been intended to provide better interoperability between different services. However, these approaches are also limited to specific domains or runtime environments. An example of such an approach is given in [110]. The proposed solution allows for the semantically grouping of service endpoints via policies in the service registry. During every invocation this repository is checked and the corresponding method is invoked, allowing for stable business processes. However, the proposed solution is limited explicitly to BPEL, and it does not foresee message adaptations and modifications. So, there is no support in case of replaced service providers and thus the need to cope with new service interfaces.

As depicted in [119], the development of business processes should follow the evolution of the telecommunication infrastructure, which will result in a new generation of personalised, secure and highly available web services, by applying the principles of domain specificity, virtualisation, loose coupling and seamless vertical integration. The awareness and the need to address this so called “business-to-IT” problem is seen as a crucial issue [114], which is not addressed with current web service-based frameworks.

To sum up, although the web service approach has been intended to tackle the need for a technical framework for SOA-based environments, interoperability is not guaranteed in current, web service-based frameworks due to various reasons, like differences in the versions of web service standards and specifications supported, differences in error handling mechanisms, differences in protocol support etc. [104].

3.4.7 Addressed key requirements

As depicted in section 3.4.6 current web service based frameworks show some significant gaps with respect to a technical realisation of a SOA-based environment. In particular with respect to support the integration of heterogeneous applications, services and IT environments existing environments and approaches do not face the identified requirements in an adequate way. Although web service technologies in principal enable the integration of services via standardised interfaces (WSDL), the realisation of existing approaches enforces the consideration of the underlying technologies as well. This is due to the missing integration of these interfaces and the application of these interfaces in a technology-independent way. The introduced enhancements with ontologies [38] do not allow ensuring interoperability between different service instances as well [159]. In particular, in this context the Business-to-IT problem is not addressed in an adequate way as well. To this end, existing web service-based composition frameworks require programming activities for constructing the respective orchestration model [17].

With respect to the application of common standards to ensure interoperability is not addressed by common solutions as well. Currently, there are many fuzzy standard definitions available, keeping room for interpretation in implementing the respective standards and thus resulting in incompatible realisations. This is due to the intention of service providers to bind customers to their environments. Due to these non-compatible standard realisations service consumers are forced to cope with the respective implementations and runtime environments, and thus limiting the flexibility of this approach as intended SOA-based technical solution. In particular the missing support to transparently integrate security and accounting mechanisms are limiting factors with respect to dynamic B2B environments, not coping with the essential identified requirement to account service consumers for service consumption and enable the transmission of business critical data in a secured and reliable way. These mechanisms should be integrated in a transparent fashion, not forcing the service developers and users to cope with respective technical details, and thus ensuring the required flexibility. Current frameworks do not support this proceeding by requiring explicit credentials for accounting and security related issues, and the respective service has to cope with the corresponding accounting and security handling. This is due to the existing standards coping with the declarative specification of user credentials are described vague and do not follow a specific schema, thus enforcing a manual inspection and integration in the respective service implementation.

Finally, due to the mentioned gaps in this section the essential requirement to access web services in an effective way is not coped in an adequate way as well. Due to the vast amount of fuzzy web service standard specifications and respective, non-compatible and interoperable realisations, an integration analysis and effort has to be spent when coping with the integration and combination of different services. In particular this forced proceeding delays the finalization and execution of orchestrated service due to manual inspection and adaptation actions. Additionally, the required adaptation actions in the service implementation reduce the performance of the service implementation as well. Altogether, these limitations counteract the key success factor of web services to provide a technological environment for SOA-based environments, namely the effective access to the service infrastructure [134].

3.5 Process description languages

As depicted in section 3.3 and 3.4 two levels have to be taken into account when referring to processes, eBusiness applications and B2B collaborations. One level is the development of specific services and applications, which is referred to as “programming in the small”. The other level describes the so-called “programming-in-the-large” paradigm, which foresees to assemble given functions and services in a transparent way by orchestrating the respective service interfaces. The first one has been coped in section 3.4. In this section, a summary of the specialities of this “programming-in-the-large” paradigm as well as the respective frameworks is given.

In the “programming-in-the-large” paradigm, the following viewpoints have to be distinguished:

- *Conversation*: A conversation is a global sequence of messages exchanged among peers participating in a composite web service [25].
- *Orchestration*: This aspect considers the assembling of existing services to a new one. However, existing web service-based composition frameworks require programming skills and activities in order to realise the intended orchestration model, which is contradictory to the SOA paradigm announced in section 3.3. The well-known Business Process Execution Language (BPEL) applies to this viewpoint.
- *Choreography and coordination*: This aspect takes into account the cross-domain message exchange between collaboration partners. The major difference to the orchestration aspect is that the latter one strongly focuses on a local viewpoint of one collaboration partner, assembling specific resources and services in order to provide a new service. On the choreography level the message exchange between collaboration partners is considered, and thus providing a global viewpoint of a collaboration environment. Related standards to this level are e.g. WS-Coordination [196], WS-CDL [195] and BPSS [23]. WS-Coordination relies on a centralistic approach by describing a framework to support coordination protocols. The main roles in this concept are participants and a coordinator. The participants rely on the coordinator to interact with other participants. WS-Coordination therefore defines a coordination protocol, a coordination type and coordination context enabling the involved participants to interact with each other in the foreseen context and order. However, this quite centralistic approach faces problems when the central coordination instance fails. Therefore, other approaches like WS-CDL focus on a decentralized control flow and coordination. WS-CDL leaves the control by each involved collaboration partner, but describes the order of messages to be exchanged between these partners. So it is every partner’s responsibility to ensure that the respective messages are sent in the foreseen order. WS-CDL does not

foresee a central monitoring and controlling instance, and thus this concept is more robust and stable in case of errors.

As depicted in sections 3.2 and 3.3, the basis for assembling services in a transparent fashion is an inter-connection network. As depicted in section 3.2 the latter one is called enterprise service bus (ESB), however, this concept, and in particular the respective realisations show some significant gaps towards a dynamic, transparent B2B environment.

The combination of ESB environments and automated web service composition was intended to overcome these issues in order to build a flexible, dynamic enterprise integration platform. The solution proposed in [172] refers to an AI-planning system to provide an abstract workflow being derived from users' requirements. This abstract workflow is mapped during runtime to concrete service instances via the ESB by applying a matchmaking engine. Although this approach binds the abstract service entities at runtime to concrete service instances, a significant gap of this solution is that it does not allow an evolution of the workflow and the connected services once the abstract endpoints are bound to concrete service instances.

In addition, it has been shown that already existing business process description languages, e.g. BPEL, BPMN [22] and ebXML [49], are of static nature and as such they are not suitable for dealing with dynamic scenarios and adaptation. This also includes fault handling, monitoring, and recovery [150]. A more detailed overview about the specialities of common workflow description languages is given in [101]. A detailed overview of common business process management environments is given in [20]. The major gap of these traditional approaches relies on the static setups and bindings between collaboration partners, binding collaboration partners permanently on a syntactical level. As depicted in [178], current approaches have to adapt to such more dynamic scenarios in order to cope with the increasing demand on dynamicity and flexibility.

As depicted in section 2.2 there is a concrete need in the simplicity of consumption, provisioning and orchestration of complex IT services in order to cope with today's requirement in common B2B environment, in particular with respect to the engineering domain. Therefore, in the following section 3.5.1 an overview of orchestration possibilities of complex IT environments and services is given, namely the development and handling of processes within the high performance computing domain.

3.5.1 Scientific workflows with common process description languages

In this section an analysis of the requirements for scientific workflows is given. In particular, it is highlighted in how far the existing de-facto standard for workflows in the eBusiness domain, BPEL, can be used for such processes.

Since this kind of workflows differ in many aspects from "classical" workflows, in the following section a detailed analysis of these environments is given. The concrete implementation of this abstract scenario is given in section 5.1.1.

Requirements for scientific workflow environments

Scientists are specialised in a specific domain and are not aware of the underlying IT and service infrastructure they are using for their experiments and simulations. However, since current environments enforces them, in particular within the HPC domain, to have at least a basic knowledge of the relevant IT infrastructure and the software environments, scientists are often forced to gain this basic knowledge as well in addition to their specialised domain knowledge. However, this basic knowledge is typically not sufficient to execute the simulation processes in an optimal way, or to face specific problems within the IT infrastructure, e.g. if the environment behaves not as expected. This requires the support of experts, causing delays and inconveniences for both, the scientists as well as the system administrators. Therefore, it is sensible to allow decoupling the infrastructure from the application logic, which is reflected in the scientist's applications and services. In particular, this decoupling allows a professional and optimised management of the IT infrastructure environment, whilst scientists can concentrate on their research.

Furthermore, scientists are also interested in controlling the workflow execution, since, during the simulation, in particular in case of unforeseen phenomena, it is of interest to suspend, eventually change some parameters and resume the simulation. In addition, in case of such unforeseeable phenomena it is also of interest to extract the respective intermediary results.

Besides this, during execution of a simulation process, a refinement of the workflow structure might be required as well, due to the intermediary simulation results gained during the execution of the simulation. This evolution based on intermediary simulation results is required due to the unforeseeable behaviour of specific materials or environments which are not predictable, and thus building the base for the need of such kind of simulations.

Another issue that have to be considered in the context of scientific simulation environments is that these processes are often characterised as long-running processes. Furthermore, the infrastructure of such HPC environments is replaced periodically due to the fast development and improvement in the IT hardware sector. Since this development allows better computing power by consuming less energy, after approximately 5 years the systems have to be replaced by new architectures, in particular with respect to continuously rising electric power prices and an increasing interest of the public in the energy consumption of such systems. So, the adaptation of IT infrastructures is a common issue, enforcing an evolution of the respective simulation environments due to changed hardware environments (e.g. new IT infrastructures allowing for the same performance for a specific job, but consume less energy to complete it).

This idea of scientific processes and workflows is not new. Actually, there are a lot of workflow management systems (WfMS) available. However, these approaches are specialised and thus limited to specific domains, causing problems in case of collaborations within inter-disciplinary domains, like Taverna [132], Pegasus [40], Triana [158] or GriCol [37].

In particular, in these environments scalability is not well addressed. In addition, choreographies are not considered at all within scientific workflow environments. Choreographies allow for the assembling of different instances of workflows, in particular with respect to cross-domain collaborations [59].

Another peculiarity of scientific workflows is that the relevant results are often achieved by several executions of the simulation process with varying parameters in a trial-and-error manner, a so-called parameter-study [165]. This proceeding is due to the fact that for most real-world problems it is difficult to determine a mathematical model in order to describe the entire problem statement. Therefore, simulations, in particular parameter studies, are executed in order to find the optimal solution by testing and evaluating various different settings. Once a result is achieved, it is also of significant importance to allow the reproducibility of the achieved results, allowing ultimately the traceability of relevant results at a later point in time.

Applying common workflow description languages to the scientific computing domain

Simulation processes follow a well-defined flow, which can be described within common workflow languages [59]. In particular, the de-facto standard for web service-based workflows, namely BPEL, is in general usable for scientific workflows. The major target of common workflow environments is the automation of well-defined processes, which maps theoretically to the principle of parameter studies. These studies are reflected by the execution of the same process with varying input parameters, until a specific goal is achieved.

Although the combination of Kepler and MeDICi allows for an initial abstraction of the underlying infrastructures in order to orchestrate the underlying, relevant services [29], this combination does not support the hierarchical orchestration of services. This aspect if of essential importance in B2B environments in order to allow encapsulating already existing services, their orchestration and finally provisioning via an abstract interface to service consumers.

Within scientific processes large amounts of data have to get processed [112], so the respective handling is a fundamental issue [41]. However, since BPEL mainly concentrates on the control flow but not on the data flow, BPEL shows some gaps with this respect. Namely, BPEL is missing a corresponding support for the handling of large data sets at workflow level, pipelining, the sharing of data between different and potentially distributed instances, as well as different abstraction views on the relevant data sets [163]. In addition, these data sets are often of heterogeneous nature, so, in order to allow for the usage of a workflow engine in this context, the explicit passing of links between these data sets, as well as the potential mapping of the different data types, has to be integrated in a transparent way.

Within such environments, scientists act as modeller, user, administrator and analyst of the process [59]. This development of scientific processes is done in a "trial-and-error" manner following a data-centric viewpoint, whereas classical workflows describe the control flow of an application / process not considering the respective data flow. The major gap here is that common workflow languages do not provide sensible mechanism to handle large datasets, which have to get processed during execution of the scientific workflow [163] [111].

The mentioned workflow technologies are already established in the eBusiness domain, where the major focus relies on the description of well-known processes. Due to the well-structured proceeding within simulation environments, allowing for an adaptation of these common established workflows technologies, the scientific community shows more and more interest in workflow technologies [42][173]. These workflows, in particular in combination with a SOA based environment, allow the execution of experiments within highly heterogeneous environments.

This is due to the fact that specific simulations tend to behave on specific platforms (hardware and software environment) differently, in particular with respect to the execution performance. So, it might be sensible to execute the different phases of a simulation process (which are described in detail for a specific simulation job in section 2.2.2) on different computing nodes with varying optimised configurations. This shows once again that for an optimal usage of the corresponding environments a profound knowledge of the underlying infrastructure is of essential importance. However, most scientists do not have this knowledge and are also not interested in gaining this knowledge. Scientists prefer to focus on the development of their simulation processes and to solve the respective issues, so an environment allowing scientists to concentrate on the design and the flow of the simulation but not on the technical details is needed.

Flexibility is a crucial issue in scientific workflows as well. Due to changes in environments and the unpredictable and dynamic behaviour of the application, an environment for such simulation processes must be able to allow the transparent usage of the relevant, available resources. This required flexibility is twofold: on the one hand, an adjustment of the workflow itself might be necessary, due to received intermediary results, e.g. showing that specific branches of a simulation should be taken into account in more detail. On the other hand, it might be necessary to adjust the service infrastructure, since it has been detected during execution that some processing steps could be performed in a more efficient way when being deployed on specific systems or by adapting the respective runtime environment.

From a technical perspective, this issue can be faced by providing the HPC resources in a SOA-like fashion. Therefore, an interoperable access to resources via web service technologies, namely WSDL and SOAP, would allow in the first instance to enable an interoperable and more abstract access to the relevant resources by hiding, at least to some degree, the technical peculiarities of the underlying infrastructure. In the following, these resources can be orchestrated with state-of-the art workflow languages, like BPEL.

The transparent usage of workflow and service infrastructures requires an automated data transformation and service discovery, since common web service-based environments do not allow this due to the needed technical background (section 3.4). As discussed before, conventional scientific workflow languages are data driven and are provided within a hybrid language.

In summary, scientific environments can benefit from the robustness of proven, common workflow technologies, in particular by taking advantage of the proven and grown integrated mechanisms for recovery, exception handling and flow control. However, common scientific process environments just allow changing the input parameters, but not the server infrastructure.

BPEL is widely accepted in industry and research, enabling also for the integration of legacy applications via web service interfaces. Furthermore, there is a broad tooling support, allowing for the visual design of workflows and thus allows building up complex workflows. In particular the visual support allows also users with little IT background to model respective processes.

As it has been shown in [4] BPEL can be applied to scientific workflows. In particular, it has been proven that free available BPEL enactment environments satisfy the scalability and reliability requirements of scientific workflows [183].

In order to close the remaining gaps with respect to the handling of large data sets in scientific workflow environments, an extension to BPEL has been announced, namely BPEL-D [85]. The BPEL-D extension allows for the description of data dependencies [85] within BPEL processes. However, these dependencies are translated into standard BPEL before execution, so these dependencies are not available during runtime. This makes this approach not usable within dynamic environments at all. Another approach to handle big amounts of data in BPEL processes is described in [191]. This approach allows the integration of references to datasets across web services. This approach enables to refer only the relevant data sets for a specific simulation, and thus allowing transferring only these data sets to the corresponding computing nodes. However, it is not described how these data sets should be transferred, in particular with respect to the transport protocol to be used for the data transfer, as well as how the data source should be accessed. This data has to be protected and thus, in order to access the data sets remotely, this relevant information has to be provided as well.

An architecture for scientific workflow is described in [59]. The corresponding reference implementation currently relies on the Apache AXIS2 framework [7] and does not allow an abstraction of the integrated resources and thus does not permit the dynamic, transparent evolution of scientific workflows. Therefore, the realisation of an extended service bus allowing for the invocation of virtual web services has to be integrated as described in section 4.2.

3.5.2 Summary

Workflow description languages focus on the flow of messages between different service instances. In particular, they describe the order and direction of the respective message flow, and thus the order of the applied actions. Therefore, common workflow description languages rely on flexible messaging infrastructures [101]. In particular, an enhanced messaging and virtualisation environment is required in order to enable workflow based environment to adapt to changing requirements and conditions.

So far, current workflow based environments do not allow an evolution support by enabling the transparent exchange of already integrated services. In particular, future work has to concentrate on adaptation by designing and considering also abstract, virtual entities during the design of a workflow [178].

Other approaches as depicted in [64] provide a conceptual approach towards a model driven architecture, targeting on describing dynamic workflows on task basis. However, the presented approach provides just some abstract concepts. In particular the essential mediation service is only roughly described on a conceptual basis.

Furthermore, there is only a rudimentary support for automatic workflow evolution, also taking into account the respective workflow structure, but not the connected and orchestrated services. As shown in [145] existing approaches focussing on the automated composition and evolution of workflows are not able to cope with this issue due to the high complexity.

Finally, there is only a rudimentary support for the integration and support of human resources in a SOA-like manner within workflow based environments.

3.5.3 Addressed key requirements

As shown in section 3.5.2, current workflow environments, in particular environments coping with process definition and execution in the HPC domain do not support process evolution in a sufficient and adequate way to face the identified requirements in section 2.2. In particular, the discussed approaches provide just a basic support to exchange services with new ones providing the same syntactical interface. This enforces a manual integration and adaptation effort, and thus hinders the required identified flexibility and dynamicity. A transparent evolution of the integrated services is not supported as well, resulting in additional efforts required after integrated services have changed. Due to this high complexity and requirements of manual efforts in case of service evolution and integration of new services, current approaches fail to handle this kind of process evolution automatically.

In particular, scalability is not well addressed by current approaches. Especially choreographies are not considered at all within common scientific workflow environments, which is a significant limiting factor in dynamic collaborative environments.

Finally, the integration of human resources is not supported by current solutions and approaches as well. The respective extensions tackling this aspect are detailed in the following section 3.7.

3.6 IT Service Management – ITIL

The Information Technology Infrastructure Library (ITIL) [74] is a collection of altogether 5 core publications, describing best-practices in the commercial operation and provision of IT infrastructures and services. Therefore, each of these core publications covers a specific aspect of the service management lifecycle, namely

- Service strategies
- Service design
- Service transition
- Service operation
- Continual service improvement

In the following a detailed overview about the named aspects is given.

Currently, the ITIL is published in its third version, released in 2007 and updated in 2011. A general overview of the 5 elements and their interrelation are given in Figure 9. A detailed overview of the corresponding aspects is given in the following.

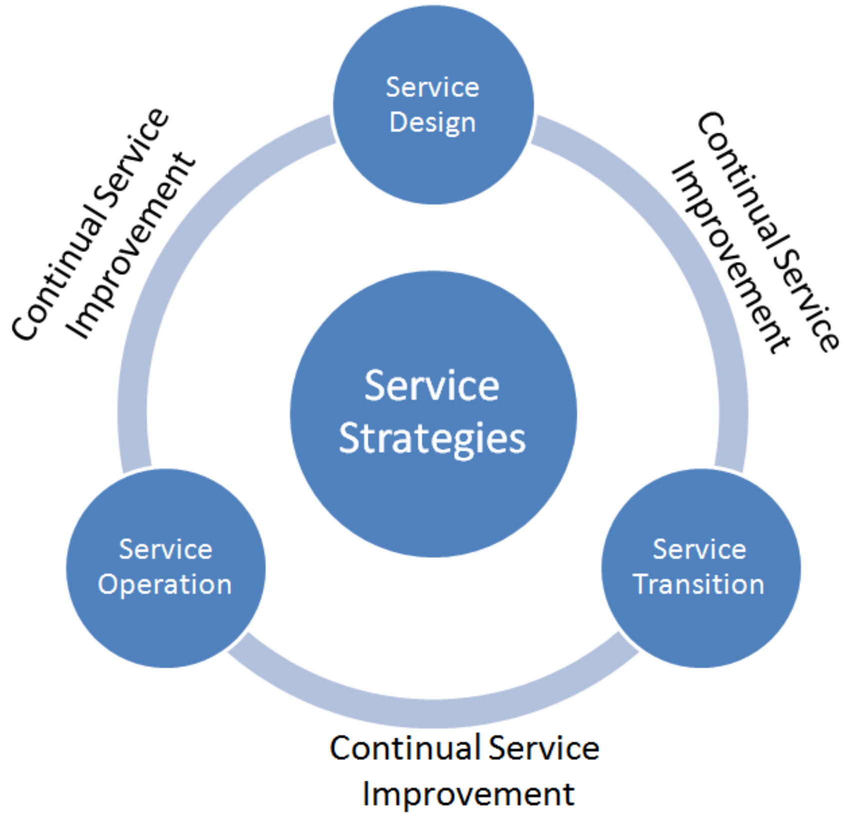

Figure 9 – ITIL lifecycle

3.6.1 Service Strategy

The service strategy volume described the central element of the ITIL service lifecycle description. In particular, this volume provides guidelines about how to develop a long-term service strategy with a strong focus on market-driven analysis. Namely, the following processes are covered within this volume:

- *Strategy management for IT services*: This process describes the necessary steps in order to determine and develop a long-term service management strategy.
- *Service portfolio management:* This process handles all aspects with respect to the management of the provided service portfolio. This ensures that the service provider offers the best suitable mix of services with respect to the required business outcome.
- *Financial management of IT services*: This process ensures a sensible management of the financial issues of the service provider, namely the budgeting, accounting and charging.
- *Demand management*: The objective of this process is to provide a planning methodology in order to manage forecasted demands, also including the determination of potential new customers.
- *Business relationship management*: This process reflects the management of inter-business activities, in particular to gain and maintain knowledge and services via the build-up network.

3.6.2 Service Design

The Service Design volume focuses on guidelines for the design of IT services and processes. It has to be noted that the ITIL service design mainly relates to the technical delivery of services, rather than just relying on the technical design of services and processes. The major focus in this context is the integration of services within large business and technical environments. In particular, the following processes are covered within this volume:

- *Design coordination*: The objective of this process is to steer and manage the coordination of the design process.
- *Service Catalogue*: The service catalogue provides a list of services offered by a service provider. It may also include additional information, like accounting information, persons being in charge of the service, the intended customers of the service, and so on.
- *Service level management*: This process covers all issues with respect to agreed service levels within contracts, including the continuous monitoring and evaluation of the performance of the offered services. This is part of a so called service-level-agreement which can be seen as a contract between a service consumer and a service provider clearly identifying the technical and QoS related aspects and thresholds to be kept. At this level, it is also possible to agree on fees to be paid by the service provider, e.g. in case he does not fulfil the agreed service levels.
- *Availability management*: This process objective is to manage the entire availability of services. This also strongly relates to the service level management. This process also allows for the optimal ratio between required availability of services in comparison to the accrued costs.
- *Capacity management*: The objective of this process is to estimate and manage the required resources to operate and maintain specific services. The target outcome of this process is to determine the optimum amount of required resources to operate specific services, whilst doing this in a cost-effective way.
- *IT service continuity management*: This process mainly focuses on the assessment and management of risks within the IT domain, affecting the operation of the offered and provided services. In particular, this process covers the steps necessary to recover and continue operation during or after a disaster.
- *Information security management system*: Within this process, relevant policies with respect to information security management and IT related risks are identified and applied.
- *Supplier Management*: This process covers all aspects with respect to the management of suppliers being required in order to offer a specific service.

3.6.3 Service Transition

This volume covers the aspect of determining which services are required for a specific business environment. In particular, the following processes are covered within this volume:

- *Transition planning and support*: This process focuses on the planning and the respective support of the service transition process.
- *Change management*: This process reflects standardised methods and procedures to handle changes within an IT service landscape in an efficient way.
- *Service asset and configuration management*: The objective of this process is to provide all (configuration) information necessary to deploy and operate a specific service.
- *Release and deployment management*: This includes the support of the service development team to automatically deploy and distribute software and hardware in a platform-independent way. This also includes the quality control of newly developed and delivered services.
- *Service validation and testing*: This process ensures that the provided and delivered services are fully operational.
- *Change evaluation*: The objective of this process is to ensure that only standardised methods and tools are applied to handle the entire change process.
- *Knowledge management*: This process reflects all efforts required to gain and maintain knowledge of the service transition process.

3.6.4 Service Operation

This volume provides best-practices in order to achieve the agreed service levels during service operation. This part of the ITIL lifecycle is related to the phase when the developed services are delivered and consumed by the corresponding customers. Therefore, this volume handles also the technical management, application management, operations management, service desk and also the staff responsible for the service operation.

In particular, the following processes are reflected within this volume:

- *Event management*: The objective of this process is to define actions to be triggered in case of the occurrence of specific events.
- *Incident management*: This process ensures that the normal operation phase of the service is achieved as soon as possible, thus minimising the impact on the business operation.
- *Request fulfilment*: The objective of this process is to ensure that the requested results of the services fulfil the corresponding requirements.
- *Problem management*: This process defines standardised procedures in case of problems and errors during service operation.
- *Access management*: The objective of this process is to define access schemas, clearly defining who is allowed to access which service.

3.6.5 Continual Service Improvement

This volume focuses on the continuous improvement of IT services with respect to changing business needs. This is done in particular by identifying and implementing improvements supporting the corresponding business processes. This volume follows a seven step process [74]:

1. Identify the strategy for improvement
2. Define what will be measured
3. Gather the corresponding data
4. Process the data
5. Analyse the extracted information and data
6. Present and use the information
7. Implement identified improvement

3.6.6 Addressed key requirements

As depicted in section 2.2 there is a need for service lifecycle to enable complex, dynamic B2B environments. The ITIL specification describes common best practices of IT service management in a business related context. However, the ITIL approach describes a conceptual approach with no direct support to get integrated in existing, operating IT service landscapes. Therefore, additional IT mediation services are required in order to allow an integrated coupling of the presented approaches with the service execution environment. To this end, the ITIL specification does not address the requirement for an executable lifecycle management environment.

3.7 Integration of Human Resources

In the previous sections a detailed technical analysis has been provided about the gaps of current web service and process based environments. In this section the corresponding methods and technologies with respect to the integration of human resources in B2B collaborative environments are analysed. As depicted in section 2.4, there is a concrete need not only in integrating and orchestrating services, but also in integrating human resources as well in this process. In particular within complex scenarios, humans often are directly involved and have to be enabled to collaborate synchronously with each other. For example when a scientist executes a simulation of a cancellous bone in order to determine the optimal position of an implant, a doctor has to get consulted at a specific point of time.

So far, there is no direct support for this kind of collaboration in common collaboration frameworks.

The authors of [70] tackle this issue by presenting a model-driven approach to describe the integration of humans within business processes in a more abstract way. In particular, a role-based view is introduced allowing for the abstraction of concrete users playing a specific role. Although this approach principally enables for the role-based integration of humans within a complex B2B collaboration scenario, it does not allow considering synchronous collaborations at specific points of time within a collaboration workflow.

Another well-known approach with respect to integrating humans within process oriented B2B collaboration environments is the BPEL4PEOPLE extension [2]. The intention of BPEL4PEOPLE is to support a broad range of scenarios involving people within business processes [151]. In particular, BPEL4PEOPLE refers to WS-Human Task providing a notation, state-diagram and API for human tasks as well as a respective coordination protocol. Taking this into account it is possible to assign specific tasks to human roles. The mentioned coordination protocol ensures that the interaction with these human task descriptions is applicable in a more service-oriented fashion, whereby the execution of the respective tasks is controlled autonomously.

Altogether, the BPEL4PEOPLE extensions allows for a broad range of opportunities in which human resources can be represented and grouped: individually, via roles, groups and also as a result of query execution. These strategies can also be used as the basis for work assignments. In addition, there are a number of distinct ways in which manual tasks (i.e. those undertaken by human resources) can be implemented, ranging from inline activities in which both the task definition and the associated work directives represent a part of the same node in the process through to standalone tasks (defined in a distinct process definition) which are coordinated by a PeopleActivity node in a BPEL process. However, there is only a minimal distinction made between tasks and task instances. Whereas this is inconsequential when specifying a static process model, many of the elements in the enhanced BPEL4People/WS-HumanTask proposal require specific addressing e.g. invoking a remote task requires knowledge of the remote endpoint, the process name, task name, the specific process instance and task instance. Similarly, data elements are specific to a process instance (not all process instances), hence they also need to be named respectively. Moreover there seems to be no notion of process instance or task instance identifiers in these naming schemes that facilitate navigation to a specific instance that is currently in progress (e.g. for delivering a notification or updating an associated data element). Finally, there is no support for more detailed definitions of specific resources (e.g. via capabilities) or for the use of resource characteristics when distributing work.

From an organisational perspective the provided model is minimalistic and does not take into account common concepts, such as jobs, reporting lines, organisational groups etc., nor allows using these characteristics for work distribution purposes or for identifying or grouping resources in a generic sense. BPEL4PEOPLE also provides only minimal access to historical information, since it is only possible to refer to preceding work items in the same case. Moreover it is not clear to what extent this can be used for work distribution purposes.

Another weakness of this approach is that there is no provision for imposing an authorisation framework over the tasks in a process to limit the potential range of resource usage. Additionally, there is no clear specification about who is allowed to consume which resource at a specific point of time. Furthermore, the steering mechanisms for restricting the range of actions initiated by a specific resource are limited. This includes e.g. the delegation of tasks as well as resource reallocation. Some of these options can be restricted at task level, but not on a per-resource basis. An additional aspect not considered in this approach is the support for optimising the work item throughput, e.g. by auto-starting specific tasks in case of specific events, piled execution of tasks, etc. Furthermore, it is not clear how the visibility of work items can be restricted. So far, the assigned tasks are visible to all involved persons, which might be a problem in case of cross-domain collaborations where it should not be visible to external collaboration partners who are the persons in charge of the execution of a specific task. Finally, the corresponding task description and the respective management are available only locally; however, this information is needed for remote invocations as well [170].

Other approaches focus on the formal description of human workflows [199]. The limitations of these approaches are twofold: on the one side, this description just allows for the depiction of human workflows, but not for the integration of services as well. On the other side, only tasks for specific people can be specified. This approach does not foresee to allow for a role-based task assignment, which is of significant importance in case of a specific person who is not available and a substitute has taken over temporarily his role. Furthermore, this model does not allow for the configuration and setup of synchronous collaboration sessions between two or more persons, which is a limiting factor as well.

Assembling and orchestrating tasks in an optimal way can improve the entire execution time of a workflow involving humans. Therefore, [68] depicts how the interaction times with humans and the corresponding task related services influence the entire execution time. However, this is more a formal approach describing the impact of the involvement of humans and IT services to the entire execution time, but does not provide a framework to setup and steer this collaboration as well. As presented in [107] the authors try to overcome this issue by enhancing BPEL4PEOPLE human workflows with constraints allowing realistic modelling of the respective tasks, in particular enabling the optimisation of the entire task flow. However, this approach fails in addressing the essential identified requirement to support synchronous collaborations between human experts as well.

The invocation of web services within human centric workflows has been considered in [27]. Therefore, the authors propose a framework allowing to dynamically binding web services to specific tasks assigned to human resources. However, this approach just allows for the invocation of a corresponding web service, but does not foresee to enable synchronous collaboration sessions between humans as well.

Setting up sessions between two or more participants has been addressed by the Session Initiation Protocol (SIP) [149]. This protocol has been established in the telecommunication sector in order to setup, steer and close communication sessions between two or more participants. In particular, SIP allows the involved parties to negotiate the protocols and conditions for the communication session.

In order to describe the characteristics of the respective multi-media data streams, the Session Description Protocol (SDP) [67] is used. SDP allows defining which codecs, transport protocols and addresses are used within such a session.

However, these established approaches do not apply to setting up a collaborative session between humans. This is mainly due to the fact that the collaboration between humans is not just limited to establish a data stream connection between them, but also to setup and steer the respective, specialised applications and services. This includes also the selection of the relevant applications for a specific collaboration session, as well as the distribution of the required data sets. In particular the latter one requires that the original data owner keeps the control about who is allowed to access which data sets. SIP and SDP do not address these aspects.

3.7.1 Summary

It has been shown that the idea of integrating human resources within common workflow environments is not a new issue. There is already a multitude of existing approaches available, trying to tackle the issue of integrating human resources in B2B processes. However, as depicted in this section, common approaches do not address this topic in an adequate way. In particular there is no support in integrating human experts in a SOA like fashion, thus not addressing the identified key requirements of chapter 2. This aspect is depicted in the following section.

3.7.2 Addressed key requirements

It has been shown that current solutions do not allow for a sufficient support, in particular within dynamic B2B environments, which typically enforce the involvement of several external experts for specific issues. Especially the lack of supporting the setup and steering of synchronous collaboration sessions is the major limiting aspect of current approaches, and thus a corresponding enhancement is necessary enabling for the orchestration of human experts and IT based services within common process descriptions. This procedure enables the planning of complex workflows in one place by ensuring to abstract from the underlying infrastructure. This includes the abstraction from the IT service environment, as well as from the environment being required for the involved human resources. For example, when designing such a workflow, the designer should not be forced to take into account the specific applications required by the human experts being involved in this workflow, but only to specify *who* has to collaborate with *whom* and *when*.

Following the identified requirements in chapter 2 current approaches do not provide an adequate support for these requirements. In particular with respect to supporting the synchronous collaboration between humans, current approaches trying to integrate human resources in workflow based environment do not allow such kind of collaborations. These approaches are limited to the assignment of specific tasks to selected persons only.

In addition, current solutions do not support the assignment of human resources to specific roles, but only for the definition of specific user accounts with respective access rights. This is a limiting aspect in dynamic and flexible B2B environment, since in particular in case of a new allocation of a specific role the entire setup has to be redone in order to allow the new role owner to use the respective services and data. The dynamic assignment of specific roles to selected people is only supported rudimentarily, e.g. if the same account is used by several people. However this results in a loss of control about who is able to access which resources. It is of essential importance that this control remains at the respective company.

Finally, the necessity of invoking human resources remotely in order to set up a synchronous collaborative working session is not supported by current environments, in particular the set up and session control during collaboration execution, including the configuration of the required tooling landscape (sharing CAD information, communication and collaboration tools).

3.8 Addressed key requirements

In this section, an overview is given about how the analysed state of the art concepts and realisation frameworks address the identified key requirements of chapter 2. The respective analysis is depicted in Figure 10.

	Simplicity (Ease of use)			Provisioning			Interoperability	Time Constraints	Reliability and Robustness	Collaboration Support	
	1.1	1.2	1.3	2.1	2.2	2.3	3.1	4.1	5.1	6.1	6.2
B2B - ESB	--	-	-	-	--	--	-	--	-	--	--
SOA	--	-	-	-	-	--	-	--	--	--	--
Webservices	--	--	-	-	--	--	-	--	--	--	--
Process Description Languages	--	--	-	-	--	--	-	--	-	--	--
Service Lifecycle Management - ITIL	--	--	--	--	-	--	--	--	--	--	--
Integration of Human resources	--	--	--	--	--	--	--	--	--	--	-

--	Identified requirement is not addressed
-	Partly address the identified requirement, but does not meet it in an adequate way
+	Meet the identified requirements, but requires additional efforts to address it in an adequate way
++	Fully supports the identified requirement

Figure 10 – Overview of addressed requirements by SotA

4. A dynamic Web Service Interface

In this chapter, a virtualisation environment is introduced addressing the identified gaps in current environments. The concept and realisation of this environment is analysed by taking into account two major roles, namely the perspective of a service provider and a service customer. The specific targets of the roles are depicted in section 2.1. Although the intentions of the user groups seem to be contradictory, there is a major advantage in applying a service virtualisation infrastructure for both groups. First of all, the introduction of a virtualisation abstraction layer enables both profiles for an increased flexibility. The service customer can easily change service providers with minimal efforts, whereas the service provider is also enabled to adjust his service infrastructure without affecting his customers. For example if new hardware or software environments are installed, new communication standards are developed and used by some customers, or if failures occur in the internal infrastructure, the corresponding requests can still be served without affecting the customer at all. Of course this procedure requires considering the current user and service runtime context. So, besides the pure technical replacement of a service by another one, the respective SLAs as well as the service context have to be taken into account. The first issue might apply if there are long running contracts with a specific service provider, and thus a change might go hand in hand with additional, significant costs. The latter one applies for example if, in order to continue the service execution with another service provider, a large amount of data has to be transferred. In particular, in situations where results are required in near-term lengthy data transmissions are not acceptable, even if the potential new service provider would provide a comparable service in better quality. To this end, the decision to replace a service provider has multiple dimensions to be considered. However, as soon as the decision is taken, which can happen quickly as depicted in chapter 2, the replacement process should be executed without enforcing involved experts to take into account technological details.

By applying such an infrastructure, this procedure enables open markets for services, where both the service provider as well as the service customer can benefit from. Service providers can argue to provide a flexible infrastructure enabling the service customer to easily use their services, whilst hiding the respective infrastructure details, and thus providing benefits for themselves and the service customer. The service customer can easily integrate services whereby remaining able to easily change service providers as well, without the burden to take into account all technical details of the service to consume.

Finally, the enhancement of this infrastructure towards an environment allowing for the SOA based integration of human experts as well enables describing holistic, cross-domain collaboration environments. By supporting both, the integration of human experts as well as of complex, specialised services in a transparent fashion, this environment tackles the identified key requirements.

As depicted in section 3.4, the technological basis for service offering and consumption in a SOA fashion are web service technologies. In this context, service WSDLs describes static interfaces for static consumption behaviour not allowing for the necessary flexibility in dynamic B2B scenarios.

The following introduced service virtualisation approach takes an essential step in this direction, by not directly exposing a static interface, but a service contract describing the available functionality as well as how the respective methods can be invoked.

To this end, in the following sections a virtualisation gateway infrastructure is introduced extending the messaging infrastructure by intercepting incoming and outgoing calls to enact transformation and security policies upon them. This allows in particular the conversion of abstract calls into concrete invocations and the routing and distribution of messages to a variety of providers, thus implicitly serving composition and dynamicity.

Therefore, in section 4.1 the architecture of the virtualisation gateway is introduced, in particular highlighting the enhancement capabilities of the introduced approach.

Section 4.2 depicts the design of the virtualisation gateway approach by applying the introduced concept as abstraction layer over IT based environments, in particular by encapsulating the underlying IT infrastructures in a transparent fashion.

In section 4.3 the realisation details of the virtualisation gateway infrastructure are given.

The aspect of integrating human experts in collaborative working environments in a SOA like fashion is addressed in section 4.4. To this end, the introduced virtualisation gateway approach is enhanced towards a dynamic, collaborative session management environment is introduced, depicting in particular the integration and synchronous collaboration support of human experts in complex, SOA based collaboration processes.

The respective service lifecycle management support of the virtualisation gateway approach is depicted in section 4.5.

Finally, section 4.6 summarised the addressed key requirements of the introduced virtualisation gateway approach and the respective collaboration support.

4.1 Virtualisation Gateway Architecture

In this section the architecture of the virtualisation gateway is introduced and described in more detail. As highlighted in section 2.2 and 2.3, there is a need of increased flexibility in dynamic B2B environments, and thus in an abstract view on encapsulated services. To this end, the introduced service virtualisation gateway concept consequently reflects a respective abstraction layer. This abstraction layer operates as an intermediary service between service consumers and the corresponding service logic by intercepting and analysing invocations and mapping them to relevant service instances. This mapping can also involve necessary transformation steps as the virtualisation gateway does not focus on a specific interface description.

Since the gateway acts as a single entry point to not only local service logics, but potentially also to the whole virtual collaboration environment, it also provides the functionality to encapsulate services.

Therefore, the following application perspectives of the virtualisation gateway approach are depicted in detail:

- The service consumer perspective;
- The service provider perspective;
- A holistic viewpoint taking into account service consumers and service providers.

4.1.1 Service Consumer perspective

From a service consumer perspective the virtualisation gateway acts as a local message interceptor, enabling for the invocation of virtual service endpoints. The virtualisation gateway approach acting as local message interceptor ensures the application of local policies of the service consumer which have to be applied, as well as a transparent interaction channel with varying service providers. This procedure is depicted in Figure 11, highlighting the virtualisation gateway approach from a customer perspective. In particular the depicted filters & policy enhancement of the gateway ensures the application of the respective policies and content related filters, e.g. allowing the transformation of specific message parameters. As shown here, the virtualisation gateway acts as a single communication channel for the service consumer by encapsulating the respective, underlying service infrastructure. In this context, the location (local or remote) and invocation details of the invoked service are completely transparent for the service consumer.

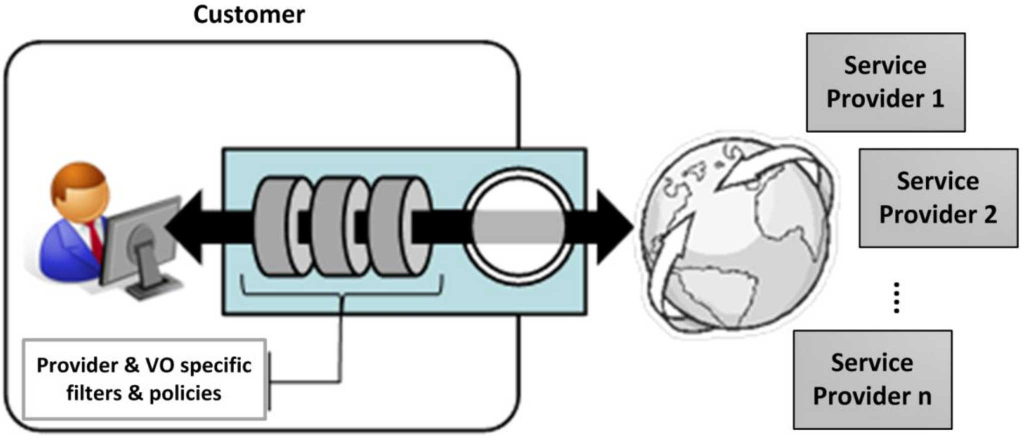

Figure 11 - The Gateway Principle

As a virtual endpoint, the gateway is capable of redirecting messages from virtual addresses to actual, physical or virtual locations (local or in the Internet), thus simplifying endpoint management from client side, e.g. applications / client services are not affected by changes in the collaboration, such as replacement of service providers etc. In particular the capability to transform and redirect messages to virtual service endpoints is depicted in detail in section 4.1.3, allowing for the upmost flexibility for service consumers and service providers. An intrinsic handler in the gateway channel hence consists in an endpoint resolution mechanism that identifies the actual destination of a given message.

4.1.2 Service Provider perspective

In order to announce a service, service providers announce a virtual service interface definition (WSDL). Incoming calls are intercepted by the gateway in order to transform these messages by taking into account the service implementation's requirements. To this end the gateway accesses a knowledge base containing all the necessary information like the mapping of the virtual name to a concrete service endpoint and the transformation of method names and parameters. As the gateway can route to any number of physical or virtual service endpoints, this approach allows service providers to expose multiple WSDLs for one and the same service, as well as the other way round, namely to expose multiple services via a single (virtual) web service interface. This allows not only to distinguish between various (potentially user specific) instances of a single web service, as is currently being exploited by cloud service farms, but it also enables the service provider to dynamically move service instances between resources, as well as to compose new services through combination of existing services, without having to rewrite the corresponding logic. It is hence also possible to provide services dynamically – e.g. by announcing new virtual interfaces, or by altering the service logic without requiring to change the interface.

These aspects simplify service management significantly, as it removes the issue of having to cater for individual instances and their representation. For example if used in a cloud like fashion, updates initiated at the breeding site would automatically spread across the network over time.

Figure 12 shows the structure of this gateway. The concept follows the same approach as depicted in section 4.1.1, but provides more details about the internal structure of the virtualisation gateway concept.

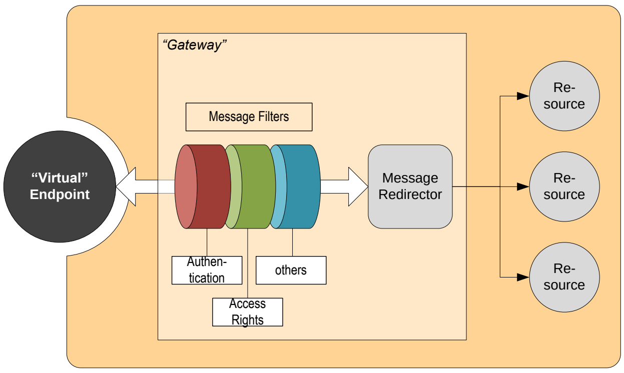

Figure 12 - Gateway Structure

This structure enables service providers to encapsulate and hide their infrastructure, in particular allowing for the provision of virtualised IT service products. With the gateway being extensible, it provides the basis to non-invasively enact security, privacy and business policies related to message transactions. With the strong SOA approach pursued by the virtualisation gateway, the structure furthermore meets the requirements of minimal impact and maximum deployment flexibility. Through its filters, it furthermore allows the standardised messaging support. The gateway is also constructed in a way that allows participation in multiple collaborations at the same time without requiring reconfiguration of the underlying infrastructure.

4.1.3 Application of the virtualisation gateway approach

Through using “double-blind virtualisation” mechanisms, e.g. by deploying the gateway on both consumer and service provider side, it is possible to alter resources and providers without affecting the calling applications. Figure 13 presents a sample deployment of the gateway infrastructure. The dashed line denotes an indirect link (as the gateway extends the message channel).

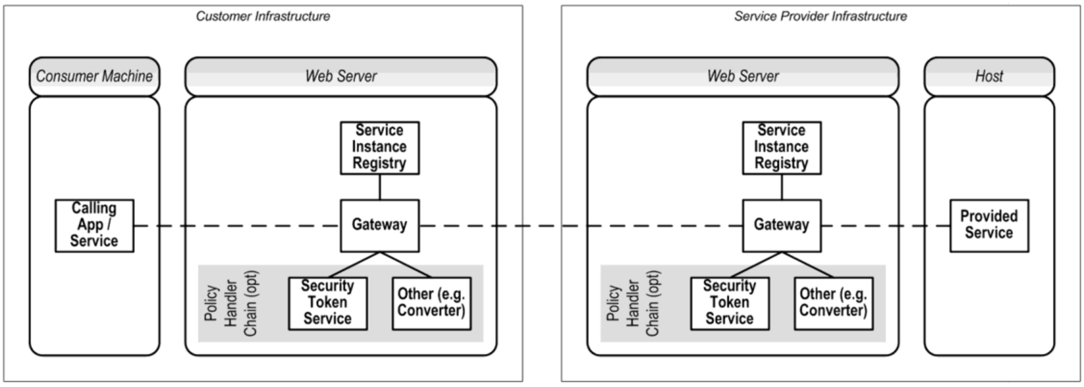

Figure 13 - Sample deployment of the Gateway infrastructure

In effect, this virtualisation technology thus acts as an enhanced messaging infrastructure which can serve the following purposes:

- *Policy enforcement*: The gateway can enforce policies upon message transactions since it allows the definition of criteria that must be fulfilled before a potential consumer is authorized to access a specific service. To this end, policy decision points can be inserted into the message chain (cf. Figure 12). For example, it is possible to differentiate service consumers based on their reputation, so as to estimate the relative trustworthiness, the effort worth investing etc.
- *Message security, identity and access management*: Due to the multitude of existing security related standards and approaches, service providers have to ensure to cover most of them in order to offer a migration path to new customers. Ideally, each client application should be able to fetch security tokens that are necessary for service access, and each deployed service should be able to authorize an incoming request using a claims-based security model with fine-grained authorization. Unfortunately, many applications in production today do not yet adhere to these principles, and the gateway can serve as a migration path towards broader adoption of the claims-based access model. The customer-side gateway can authenticate internal requestors, request security tokens on their behalf and protect (i.e. encrypt) outgoing messages in a transparent way. A service-side gateway can act as a policy-enforcement point to authenticate and authorize incoming calls. For example, the gateway can establish a secure connection to the service consumer while the concrete client application does not support any secure data transmission.
- *Protocol translation*: Not only are web service protocol specification subject to constant changes, but also a broad scope of protocols serve similar needs, thus

leading to recurring interoperability issues between customers and providers. With static interfaces, the consumer has to ensure that his/her service invocation is compatible with the specifications of the provider.

- *Transformation:* Since the gateway acts as a transparent message interceptor for the underlying services, transformation activities can be applied without impacting either the service consumer or the respective service instance. Transformation actions can be processed after the protocol translation, in particular regarding the transformation of service specific parameters.
- *Filtering and information leakage protection:* The gateway can detect and remove private information from a request, offering a hook to install information leakage detection and prevention mechanisms. This can be applied e.g. for business critical data being marked with a respective security class. The gateway environment can apply related business policies to ensure that only data of foreseen security classes is processed in external services.
- *Load balancing & fail over:* The gateway can act as a load balancer. If e.g. one service is currently heavy in use, the gateway may decide to forward requests to this service to an equivalent one.
- *Routing:* Using a similar concept to DNS, the gateway allows for dynamic routing of messages depending on registered and available providers (cf. load balancing). This allows on-the-fly replacement of providers, given that no current request is still in progress (in which case the respective data would be lost). Since the gateway exposes virtual endpoints to the consumer, he does not have to adapt the process' logic, if the providers change.
- *Login monitoring:* Often it is interesting for a service provider to see which version of a service is still used by the customers. Via the gateway this information is also available.

In summary, the gateway of a service provider acts as the virtualisation endpoint of the services exposed by the respective organization. Its main task consists in intercepting incoming and outgoing messages to enforce a series of policies related to access right restrictions, secure authentication, transformation etc., thus ensuring that both provider and collaboration specific policies are maintained in transactions.

The gateway of a service consumer ensures the integration and usage of virtual services in an abstract way, focussing on the respective functionality. To this end, the gateway ensures that service consumers can focus on their specific domain, without the burden to take into account specific technological details. In particular there is no difference any more in the usage of local or remote services.

4.2 Virtualisation Gateway Infrastructure Design

As depicted in chapter 2 the virtualisation gateway infrastructure should allow for a transparent and flexible usage of services and applications being provided via web service interfaces. In this section a conceptual overview of the gateway is given, enhancing the initial concept of so-called service locators depicted in [66].

In addition, it is described how the gateway infrastructure can be applied in order to face the challenging issue about the provisioning of complex IT service products, in particular with respect to the involvement of domain experts not being aware of the

IT infrastructure they intend to use.

4.2.1 Hierarchical layered application of the virtualisation gateway approach

In this section it is shown how the virtualisation gateway infrastructure can be applied to specific environments following a hierarchical layered approach. This procedure allows for a concentration of expertise within knowledge domains by enabling experts from specific domains to consume relevant services of other, related domains in an easy and transparent way. This environment is reflected in Figure 14.

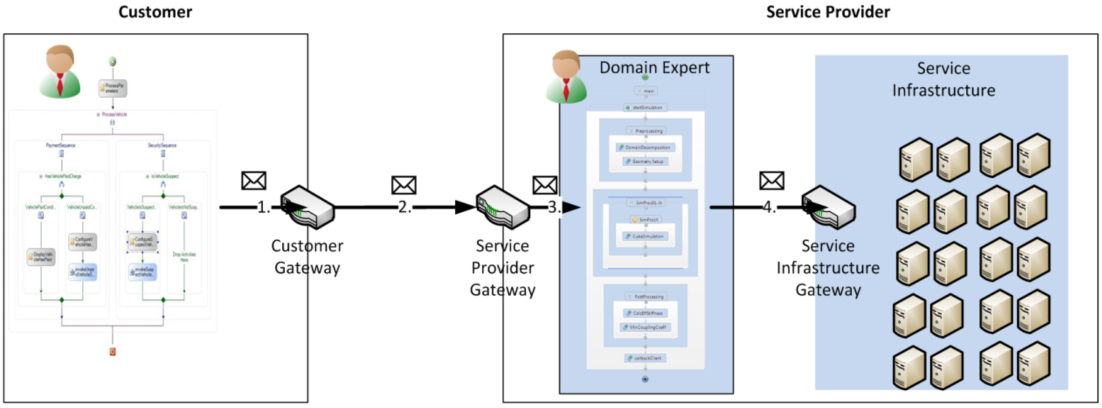

Figure 14 - Applying the virtualisation infrastructures to different domains

As depicted, the gateway infrastructure is applied to the following domains:

- The service customer;
- The service provider;
- The service infrastructure (acting as a local gateway not being foreseen for external usage).

In this environment, the service consumer has designed a local workflow in order to orchestrate different locally as well as remotely provided services. As depicted, all invocations are transferred via a virtualisation gateway by invoking virtual service endpoints. The virtualisation gateway logic decides in the next steps to which destination and in which format the affected message should be transferred. Since these invocations are applied via virtual service endpoints, the change of a concrete service endpoint does not affect the customer's workflow at all. This is due to the fact that, besides the transparent routing of the relevant messages, the gateway allows for a transparent message transformation as well. So, in case of changing the mapped physical service endpoint of a virtual one, the gateway enables the integration of rules in order to map invocations to new service interfaces and instances. In particular, for the designer of the service consumer workflow it does not matter if the service is executed locally or remotely. The process designer can concentrate on the design of the process flow without the need to take into account the concrete service infrastructure.

In the chosen example, the service provider domain is divided into two sub-domains, since the service provider provides both, complex IT services as well as the corresponding infrastructure. This approach allows also for integrating external IT infrastructures or service providers, as depicted in Figure 16. In that case the Service Infrastructure Gateway has to be located as entry point at the respective third party infrastructure service provider. However, this does not affect the workflow orchestration of the domain expert of the service provider, since the invocation of the service infrastructure is also encapsulated within the gateway infrastructure by using the announced virtual service endpoints. It has to be noted that this encapsulation hierarchy is not limited to the mentioned example. The introduced virtualisation gateway approach allows encapsulating any number of domains and combining them with each other in a transparent fashion via virtual service endpoints.

The complex IT service is provided by allowing a domain expert to design a workflow reflecting the necessary steps by orchestrating the relevant services in a transparent way. All these services are integrated in the workflow via virtual service endpoints, reflecting the functionality of a service and the necessary information for a service invocation. No other technical details have to be considered at this stage for the workflow designer.

Finally, the Service Infrastructure Gateway allows for the mapping of requests to the service infrastructure.

In the following, the processing of this hierarchical gateway infrastructure is depicted by following a specific service invocation from a service consumer. This example relies on the cancellous bone simulation use case described in section 2.2.2 showing the encapsulation of specific expertise domains in a transparent way.

1. As part of the customers' workflow reflecting the simulation of a specific region of a human body, a cancellous bone simulation process has to be triggered. The consumer's workflow engine invokes the corresponding virtual service endpoint.
2. The invocation message is intercepted by the consumer's gateway, which forwards this request to the foreseen service provider, holding a contract with the customers' company for this kind of simulations. The workflow designers as well as the execution environment are not aware about which service provider is going to process the corresponding step in his workflow, since service consumers do not want to get bothered with these details. Finally, they are interested in the results in order to proceed with their work.
3. The domain expert of the service provider designed a process for the simulation of cancellous bones by applying the required functions, applications and data flows in the optimal way. By taking advantage of the internal Service Infrastructure Gateway the domain expert does not have to take into account the technical details of the service infrastructure. The domain expert is enabled to orchestrate virtual service endpoints within his simulation process. This process is internally deployed. After having received the request for a cancellous bone simulation from the customer and having checked if this customer is allowed to execute this request, the service provider gateway forwards the invocation to the workflow engine hosting the requested service.

4. Finally, the activities of the local workflow of the service provider are transmitted via the Service Infrastructure Gateway to the best suitable computing resources.

In summary, the application of the virtualisation gateway environment as depicted in this section enables

- Service consumers to use and integrate complex services in their own, abstract process descriptions;
- Domain experts to orchestrate services based on the provided functionality, whereas being able to concentrate on their domain expertise;
- Service consumers and providers to stay flexible with respect to new changing service and IT infrastructures.

4.2.2 Dynamic service infrastructure adaptation

In addition to the expert domain encapsulation capacities of the virtualisation gateway approach, the introduced environment also supports the identified and required flexibility and dynamicity with respect to orchestrating and evolving service infrastructures. Therefore, the following aspects have to be considered:

1. Adapting / replacing service providers and the respective services
2. Adapting the underlying IT and service infrastructure within the domain of a service provider
3. Integrating third-party services in a transparent fashion

The tackling of the first aspect with the virtualisation gateway approach is depicted in Figure 15.

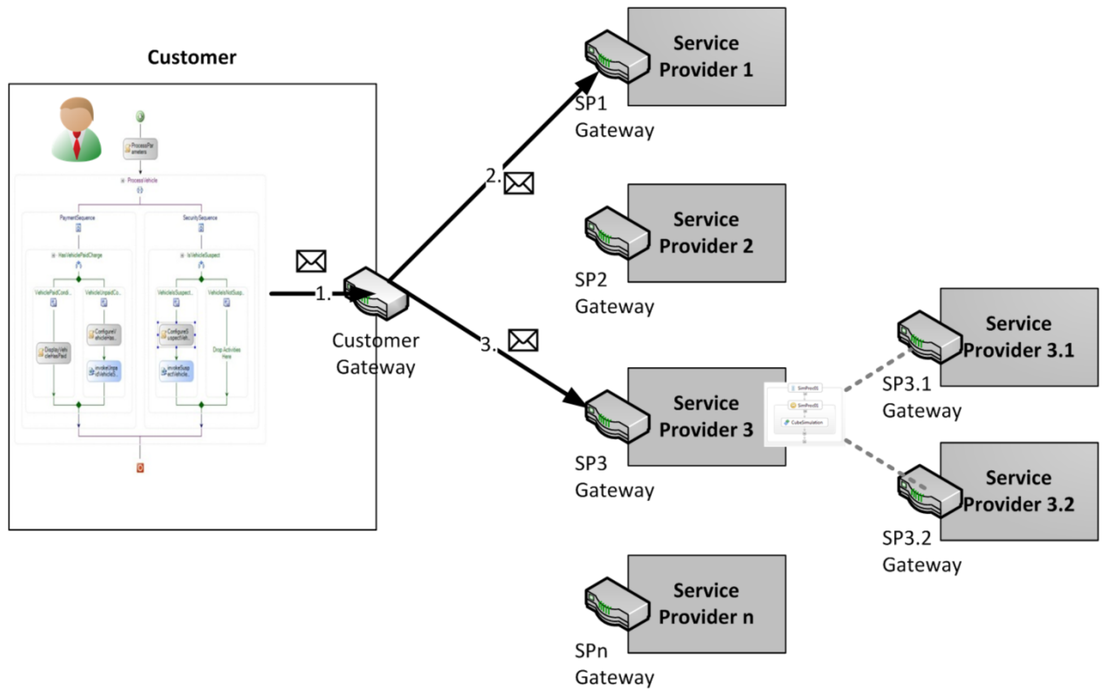

Figure 15 – Multiple Service Providers

As shown here, the service customer virtualisation intercepts the invocation of a virtual service endpoint and forwards the respective message to the corresponding service provider gateway. In the first instance, this is the gateway of Service Provider 1. However, due to the unsatisfactory performance of Service Provider 1, the service customer's company decides to contract another service provider with better conditions for services for this specific issue. So, the next time the service customer's workflow invokes the respective, local virtual service endpoint, the service customer's gateway redirects, after transforming the message to match the new service provider's virtual endpoint specification, to the gateway of Service Provider 3. This proceeding is hidden from the service customer and his process environment, so even in case of changing service providers the required adaptation actions are done within the respective virtualisation gateways, and thus transparent for the service customer. This is due to the fact that service providers are considered in this proceeding as black-boxes, not requiring taking into account the internal structure of the IT and servicing environment.

How aspects 2 and 3 are tackled is shown in Figure 16. As shown there, the integration of third-party services realised in a transparent way, ensuring the provisioning of complex services without affecting service customers with internal organisational details. To this end, encapsulated services and processes are enabled to invoke via virtual service endpoints locally as well as remotely provided services in the same and transparent way. The exposed services here consist of a complex process orchestrating some specialised services in order to provide a service to customers. To this end, some functionality cannot be provided by the service provider itself, so the respective invocation is redirected to an external third-party service provider. This approach ensures that the required service infrastructure required for an optimal execution of the process can be adapted and evolved on the fly. The respective realisation details are depicted in section 4.3.

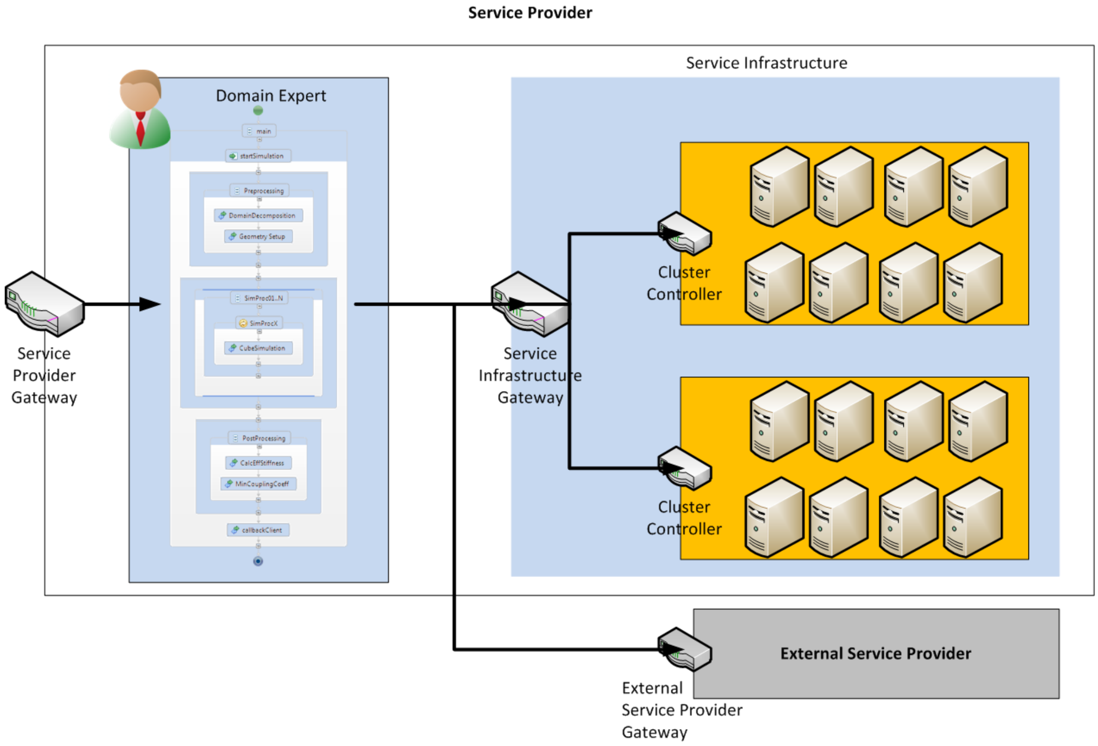

Figure 16 – Applying Gateway to handle local Service Infrastructure

4.3 Realisation of the virtualisation gateway

The reference implementation of the virtualisation gateway approach has been realised as transparent communication infrastructure. This communication infrastructure is acting as a message interceptor and as a message transformer. In particular for web services, an HTTP router is needed enabling this message interception and transformation without affecting the client calling the corresponding service as well as the underlying service implementation. This approach is depicted in Figure 17.

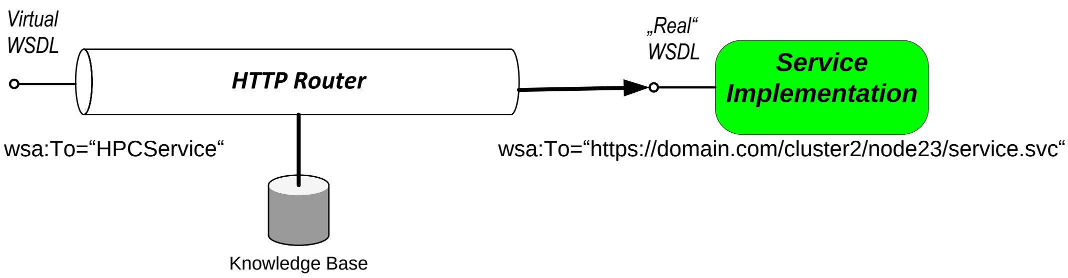

Figure 17 - Technical Realisation

In particular, the HTTP router tunnels requests from a virtual WSDL to a concrete service call of a "real" service interface. Therefore in this example the virtual WSDL provides a virtual endpoint with the name wsa:To='HPCService'. The HTTP router now maps this web service call to the physical service endpoint "https://domain.com/cluster2/node23/service.svc" of the corresponding service implementation.

This is done completely transparent to the invoking client as well as to the service implementation.

The gateway itself can be seen as an extensible chain of message handlers (so called Policy Enforcement Points (PEPs)) querying customizable Policy Decisions Points (PDPs). Anyone can thus define and register own policies which can take full control over the message handling in a plugin like fashion. Notably, the plugins need to follow a logical sequence, as e.g. decryption and authentication should be executed at the very beginning of the chain, whereas forwarding the respective message to the physical endpoint should be executed as last operation in the chain. The processing logic itself can be controlled through passing additional metadata across plugins.

The prototype realisation referred to in the following has been realised on the basis of the .NET 3.5 framework. Compared to other common web service based environments, like AXIS [7] or Synapse [8], the .NET 3.5 environment provides enhanced extension capabilities by applying WCF filters. The application of the WCF filter concept to the virtualisation gateway concept implementation is given in Figure 18. Taking this into account, the .NET 3.5 framework allows a target-oriented extension and optimisation of the prototype implementation. To this end, the entire routing and transformation logic has been realised based on the .NET 3.5 framework.

However, some IT infrastructure environments do not allow for the deployment of .NET 3.5 services, so in order to steer the invocation of application and services on the IT Service Infrastructure, an additional Cluster Controller has been realised, as depicted in Figure 16. The cluster controller is realised as a light-weighted gSOAP[5] application, allowing for the execution of local system operations which can be triggered via a web service interface, whereas consuming only a little amount of IT resources. It has to be noted that this cluster controller has to provide a reduced functionality compared to the virtualisation gateway, and thus can be realised within such an environment. Furthermore, the Cluster Controller allows for passing references to data sets to the relevant computing nodes and clusters respectively. These references include

- The data transfer protocol and the location of the required data set;
- Optionally a query allowing for the selection of specific elements of the dataset. This proceeding allows for the reduction of the amount of data that have to get transferred to the compute nodes;
- The accounting information being required to access the data set.

These references allow an intelligent management of large data sets, whereby following a SOA-based approach.

In this context, a service provider can use the virtualization gateway to decide which resources / services / workflows are exposed via a specific virtual service endpoint – all other services either remain hidden or are exposed without a virtualisation

[5] http://www.cs.fsu.edu/~engelen/soap.html

gateway intermediary. Notably, the service provider has full control over the exposed services, e.g. which methods are available for specific customers only (cf. Figure 18), whereas being fully compliant to already existing WSDL interfaces. Due to the plugin approach based on WCF filters the gateway can be enhanced with any kind of filter, thus allowing for any kind of needed message transformation or additional required functionality. The reference implementation focuses on virtual service endpoints by integrating filters in order to modify the relevant addressing information and adjust the respective parameters.

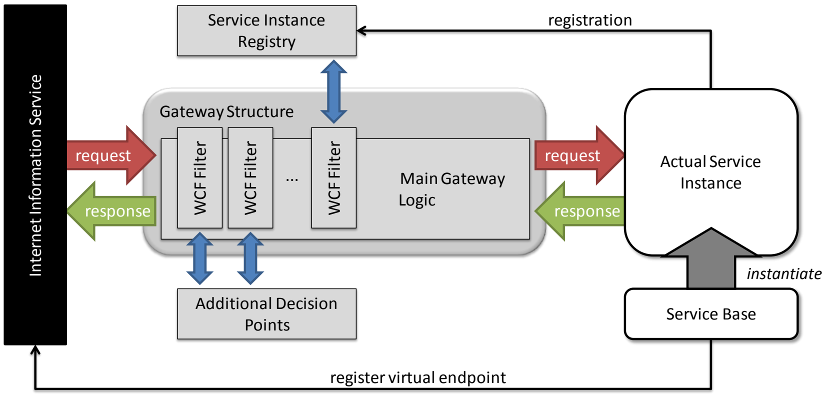

Figure 18 - Gateway Structure and its Relationship to IIS and Service Instances

Policy handlers can be registered at the virtualisation gateway using the management interface and the identifier of the specific gateway structure. Each service instance can thus principally be associated with its own gateway and policy handler chain, allowing for maximum flexibility.

The Service Instance Registry is a specific type of policy decision point that identifies the physical service endpoints on the basis of the transaction metadata (sender, addressed endpoint etc.). It will instruct the message chain about the next endpoint to forward the message to. In particular, the service instance registry provides a web service interface, allowing for the adaptation of the respective setup information at run-time. I.e. via this interface the routing logic of the virtualisation can be adapted, whereas the respective virtual service endpoints are not affected. By allowing adapting the routing functionality via the service instance registry interface, different external optimisation mechanisms can be applied to the virtualisation gateway approach, ensuring the extensibility of the entire framework for enhancements reflecting local trends and requirements.

In the following section this virtualisation gateway infrastructure is going to be enhanced towards an infrastructure allowing also for the consideration of human resources within workflows. This enhancement is applied to the virtualisation gateway approach via an additional filter realising the required collaboration setup and execution.

4.4 Enhancement of the Virtualisation Gateway towards a Dynamic Session Management (DSM)

In this section a further enhancement of the virtualisation gateway infrastructure is introduced. This enhancement extends the gateway towards a dynamic session management (DSM), allowing for the role-based, synchronous collaboration between humans. The enhancement of the gateway towards a DSM environment has been realised in the context of the research project CoSpaces [35], realising a holistic framework for different kind of IT supported workspaces. The integration of the introduced DSM enhancement of the virtualisation gateway approach within the CoSpaces framework is described in detail in section 5.1.3. It also describes how data management and information exchange within such a dynamic environment is handled whereby addressing security needs of different partners involved [11]. Specifically, the framework will support users in the on-demand selection of participants, and ensure that data required for the session is made available.

In order to enable the required environment for dynamic, synchronous ad-hoc collaborations as depicted in section 2.4, the respective framework has to be based on a distributed engineering environment for remote teams to work together efficiently, securely and reducing the frequency of travelling to a single location. In addition, both planned and ad-hoc meetings within a distributed virtual environment with appropriate decision making and communication tools have to be supported in order to ensure a smooth collaboration between the involved people. The key technical challenges therefore undertaken are:

- Dynamic integration of decision making and communication tools within the distributed environment
- Secure data sharing
- Multi-user visualization and interaction metaphor for collaborative working
- Creation of co-located, distributed and mobile workspaces for multi-functional teams
- Role-based access models.

The approach presented here follows the Grid concepts [55], namely that participants (similar to resources) are selected and integrated on the basis of their capabilities and availability, and that the available resources are configured on-the-fly according to the collaborative needs. The DSM gateway extension has been developed in close collaboration with industrial partners to assess real business needs and requirements in the context of collaborative working environment. To this end, in particular security concerns and complexity issues were raised: business critical data is shared between various types of collaborators requiring only extracts of the entire information; also configuration, maintenance and execution should be simple, with little technical knowledge required by the user(s).

The setup and maintenance of ad-hoc collaborations as depicted in section 2.4 is time-consuming and currently in particular in time-critical situations impossible – consequently, an intelligent framework is needed that realizes all these tasks automatically. As key infrastructure for such purposes the DSM enables to setup and steer the respective, available human resources in a role-based, SOA like fashion.

4.4.1 Dynamic Session Management Design

The DSM allows for the dynamic ad-hoc role-based setup and steering of participants, machines, and application to support the described dynamic, ad-hoc collaborative working session.

The DSM provides an infrastructure for dynamic ad-hoc collaboration sessions. In particular, the DSM consists of two components, both being realised as an extension of the virtualisation gateway, namely

- The DSM manager component, and
- The DSM gateway component.

The DSM manager provides a central interface to handle global collaboration setups, affecting all involved (distributed) resources. The DSM manager allows for the centralised steering of the collaboration setup and acts also as a central contact point so that the involved collaboration partners can inform about context changes, e.g. when a mobile user is faced with decreasing network bandwidth, and thus requires a different setup to display the respective CAD models. The respective setup and adjustment steps of a collaborative session are depicted in detail in section 4.4.5. After the setup of a collaborative working session, the respective DSM partner gateway interact directly with each other, thus ensuring to provide a flexible, scaling environment without potential centralised bottlenecks. To this end, the DSM Manager is tackling with administrative issues, whereas the DSM partner gateways apply to the steering and support of the respective, local human resources and services.

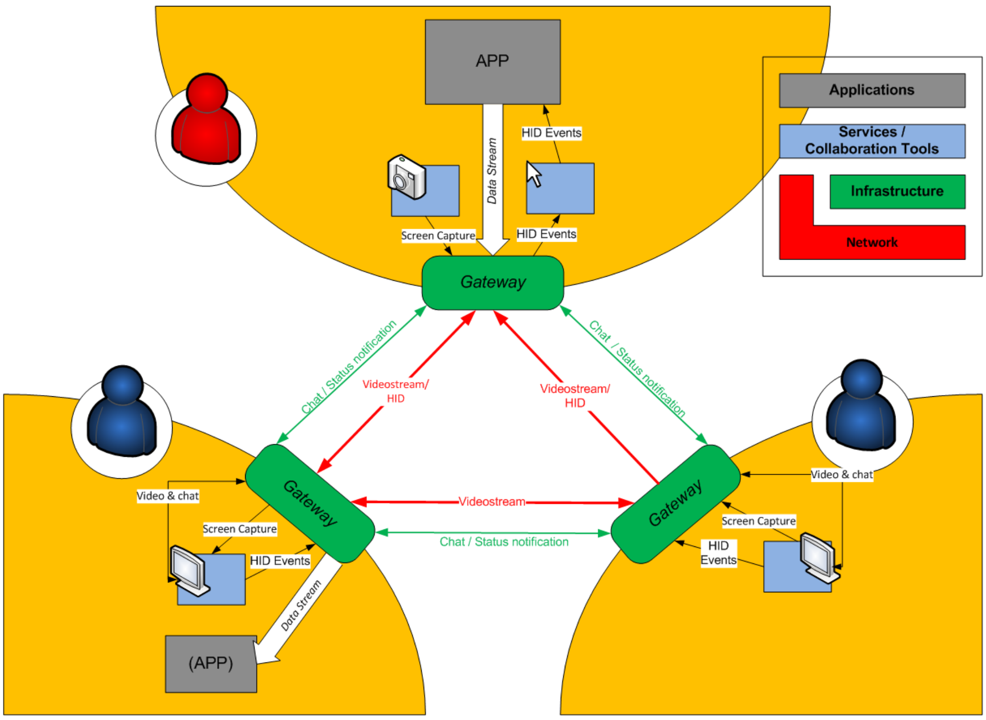

Figure 19 - Dynamic Session Management Overview

To allow for an interoperable configuration between the parties involved, each partner has to provide a corresponding DSM gateway. The DSM gateway therefore has been enhanced with a plugin allowing processing of respective collaborative working session configurations to configure the hosts of the involved roles. Figure 19 depicts this proceeding in an overview of the entire system design.

As depicted in section 5.1.3, the DSM gateway also exposes a frontend to a local controlling instance at each partner's machine, which allows the instantiation and the steering of different applications – all machines participating in a collaborative working session with shared applications hence need to provide such an application controller.

4.4.2 Architecture of the DSM Gateway-Extension

In this section the architecture of the DSM gateway extension is introduced.

To allow interoperable access to the DSM components the DSM manager as well as the gateway are realised with web service technologies [58] [190]. This provides an interoperable opportunity to setup a partner's infrastructure for a specific collaboration, following the extensible virtualisation gateway approach introduced in section 4.1. The respective DSM architecture is depicted in Figure 20.

Since the DSM gateway extension has to handle control and configuration messages as well as proprietary data streams between applications, there are two communication buses described in this approach:

The web service bus is used to configure and setup the DSM gateway as well as for sending out notifications. However, the network bus is used for the application specific data and video/audio stream. The network bus is accessed via the proxy, who is a local representative of the network bus allowing the application controller as well as the corresponding application to handle data stream to the collaboration partners in a secure and transparent way, since the respective application interacts with the proxy only, but not directly with other applications and services.

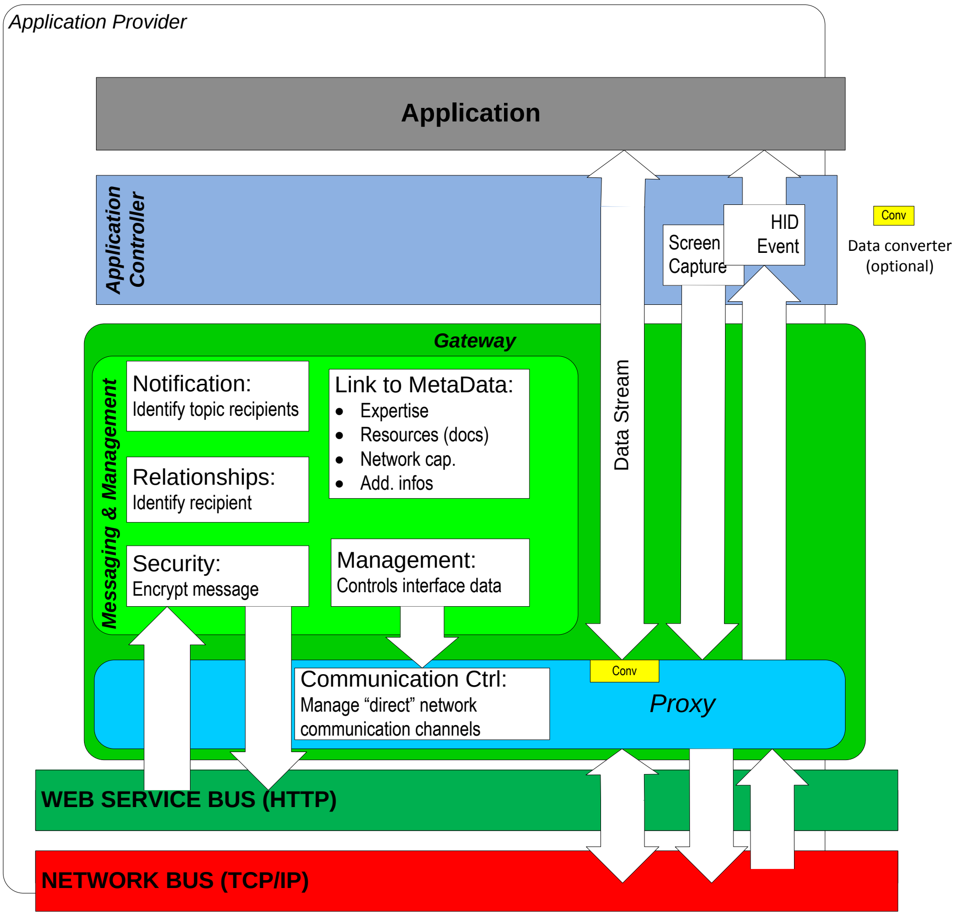

Figure 20 – Gateway DSM extension architecture

To this end, the DSM gateway provides, beside the instantiation and configuration capacities, also:

- *A notification subsystem*: this notification support realizes a publish-subscribe system, allowing both the propagation of events, e.g. that changes have been done in a document, or a participant is no longer available, as well as the exchange of chat messages between users and application-specific commands, enabling the bilateral communication between applications on an event-basis.
- *A relationship management subsystem*: every DSM gateway holds a list of references to all involved partner DSM gateways. This information is needed in cases when partners become unavailable or their context changes in such a way that they can no longer join the collaboration as originally intended. If one partner recognizes such a change, this status-change will be propagated via the notification sub-system.
- *Security issues*: since the DSM gateway allows the configuration of local services hosted by the respective collaboration partners, it is of essential importance that only explicitly allowed parties and roles are allowed to access these services and data. This subsystem guarantees that only authenticated and authorized users and services are able to set up the services and access the respective data sets. In addition, this component also verifies the integrity of configuration and collaboration steering messages.

- *Links to metadata*: During the collaboration, references to resources such as the expertise of current participants, and other users, as well as access to documents, network capabilities, etc. are needed. The DSM gateway also manages such information which is provided within the collaboration setup. In the case of the loss of a collaboration partner or a context change, the DSM gateway is informed of the issue. By propagating the lost capacities the environment is able to e.g. assign a new partner with the corresponding expertise for that specific task, or reconfigure the collaboration session by including a new expert with a respective expertise.
- *A proxy*: The proxy provides a local entry point for the data communication of the application controller as well as for the application itself. The proxy can be configured by the DSM gateway in such a way that the communication is redirected to another proxy of a collaboration partner. This communication considers both the data stream of the application as well as the transfer of screenshots for the application remote control and the corresponding Human Interface Device (HID) events. So, for example, in case that one partner is no longer available for a specific collaborative working session or a new partner has been added to an already running session, no reconfiguration of the local application has to take place, since the proxy encapsulated the respective data streams in a transparent way whereby ensuring the respective access rights and policies. In particular the latter one ensures that newly added collaboration partners do not get access to data they are not authorised to. The proxy component furthermore supports transparent message encryption as well as setting up and linking external data converters – this functionality is required when partner with different back end systems collaborate with each other, enforcing a data transformation between these applications. This proceeding ensures that also heterogeneous environments at the application layer are supported in a transparent way.

4.4.3 Addressed security issues in dynamic collaboration environments

With an increasing number of distributed and dynamic eBusiness collaborations, security, user authentication and particularly data protection play a significant role. Business secrets, including confidential data, need to be well-protected against the abuse of third parties using sophisticated security mechanisms, for example strong encryption and decryption algorithms to protect the information exchange between the partners. Nowadays, a multitude of security principles and procedures are available and approved for daily usage, but most are limited by predefined settings and static configurations. In order to create ad-hoc collaborations and on-demand data sharing, a dynamic security infrastructure must be considered and introduced that allows the greatest possible freedom of action together with the best available security level [95] [10].

To overcome the problems of dynamic and on-demand business collaboration, in particular in order to allow the role-based authentication and authorisation of users, the security framework Shibboleth [175] has been integrated within the DSM gateway extension. Shibboleth was essentially developed to protect online resources across and/or within organizational boundaries by applying a role-based access model. Shibboleth provides a federated Web Single Sign On (SSO) and attribute exchange framework. It also provides extended privacy functionality, allowing the users and their home site to control the attribute information being released to each service provider. Using Shibboleth-enabled access simplifies the management of identity and access permissions for both identity and service providers.

Therefore, a so-called three step hand-shake protocol was designed, enabling an efficient solution without requiring specific configuration adaptations during the creation of ad-hoc collaboration sessions. This protocol enhances the basic authentication procedures of the Shibboleth framework and ensures a flexible management of user/partner identification and access control for certain resources. The protocol follows three steps, namely:

1. After receiving a configuration from the DSM Manager, the gateway, which acts as a Shibboleth service provider and is accessible only for specific users, sends an authentication request to the corresponding DSM partner gateways, including the current setup ID, the address of the users IDP address as well as the corresponding shibboleth handle. Therefore, each partner site must host a Shibboleth identity provider typically coupled with the local identity management system to handle user authentication inside the organization.
2. The partner gateway receiving such an authentication requests queries the corresponding IDP for the users' credentials and validates these results with its own PDP plugin in order to determine if this user role is allowed to setup a collaboration session using the determined resources of the setup.
3. If the user is allowed to consume the resources being announced in the corresponding setup, an acknowledgement is returned. Otherwise, a corresponding error message is sent out and the setup for this partner is stopped.

This three step authentication procedure is not limited to specific user roles, but can also be applied at company level, e.g. allowing the access to specific services for all roles within a company. This processing ensures that only foreseen partners are allowed to consume specific resources, thus keeping the full control of the resources at the resource providers' site. In particular, this proceeding allows for

- The ability to identify and link up with remote experts on demand,
- accessing to context-based information,
- secure data access,
- context-related interfaces for collaboration,
- dynamic composition of persons, services and data.

A detailed overview of a shared dataspace for eBusiness collaborations following this approach is given in [94].

4.4.4 DSM gateway extension collaboration schema

As depicted in section 4.3, the virtualisation gateway allows the integration of plugins, enabling a flexible and expandable infrastructure. The DSM extension has been developed correspondingly by realising the required functionality as a plugin, namely a respective WCF filter. In addition, the virtualisation gateway has been enhanced with a web service interface enabling the invocation from external infrastructure components, like the collaboration broker of the CoSpaces framework. The detailed processing of the session setup is discussed in detail in section 4.4.5.

In this section, the schema is described reflecting the required information to be processed by the DSM extension in order to provide all necessary information to the underlying infrastructure, in particular with respect to setup synchronous collaborative working sessions. The corresponding information is provided within an XML schema. The following elements reflect the corresponding XML Schema definition, graphically represented by Altova XMLSpy environment[6].

Collaboration Setup Schema

The collaboration setup schema reflects the setup information required to setup a synchronous collaborative working session, including the setup information for all involved collaboration partners. As depicted in section 3.7 current approaches do not address the required capabilities to set up synchronous collaborative session between humans. To this end, a corresponding collaboration setup schema is introduced in this section. Figure 21 depicts the corresponding schema:

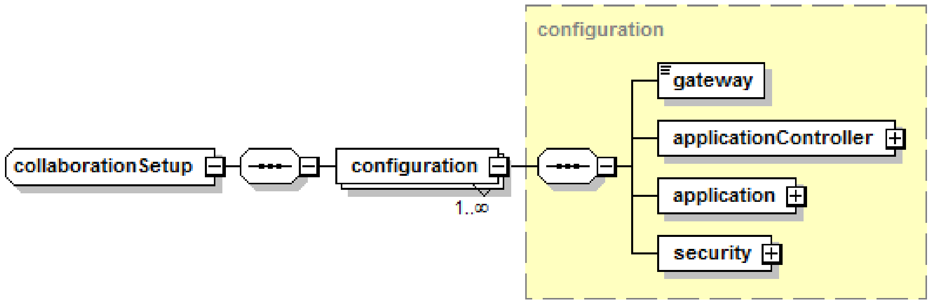

Figure 21 – Collaboration Setup Schema

In particular, the following elements are reflected in the setup information:

- For every collaboration partner, a *configuration* element is provided. Since at least one partner has to be involved, at least one configuration element is mandatory.
- The *gateway* field reflects the address of the corresponding partner DSM gateway.
- The *applicationController* field reflects the reference to the local application controller of a specific user.
- The *application field* depicts the setup information required to setup the corresponding application at the user's device.
- *Within the security* element the corresponding information about collaboration partners and their security level required are reflected respectively.

6 http://www.altova.com/xml-editor/

Application Capabilities

In order to setup the user interface for a collaboration partner a corresponding application has to be started and configured, reflecting the current requirements within the collaboration context. This information is provided within the application capabilities field. The respective structure is depicted in Figure 22.

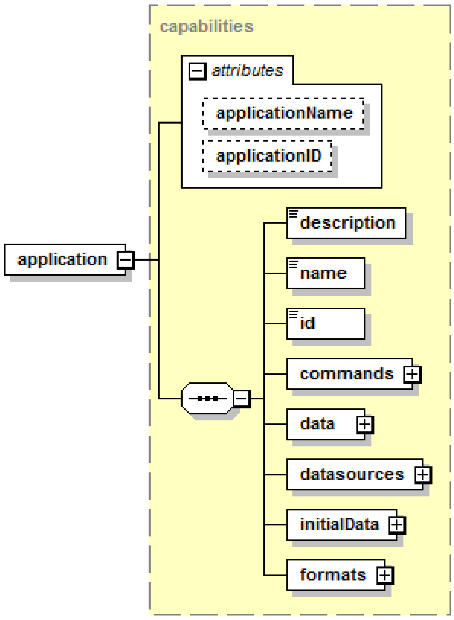

Figure 22 – Application Capabilities

The following elements reflect the setup information for one application being involved within a specific collaboration session:

- The attributes *applicationName* and *applicationID* reflect the information required to identify the corresponding application at the user's side.
- *description:* This element contains a textual, human-readable description of the application.
- *name:* This element contains the name of the application being displayed to the user.
- *id:* If already available this element describes an application instance identifier. This information is required since some instances of the same application might be executed at once.
- *commands:* This element reflects the commands being executable for the application. For the registration of an application this field describes all available functionality of an application. In the setup information for a concrete session this field contains the functionality of the application being required for a specific session, thus providing the capability to reduce the provided functionality depending on the collaboration context.
- *data:* This field describes the corresponding data operations. In particular in distributed, heterogeneous collaboration environment different services and environments might be used. So, during setup, it has to be described how the corresponding data can be accessed.
- *datasources:* This element describes the corresponding data locations.

- *initialData:* This field describes the data being required for the setup of the collaboration session. In particular, this data referred here is going to be loaded right after the start of the session by the corresponding application.
- *formats:* This element describes the data formats being supported by this application. In particular, during the application registration process this field contains all supported data formats, whereas during the collaboration session setup this field contains the data formats being supported for this specific collaboration session only.

Commands

This field represents the commands being available for the corresponding application. During the application registration process this field summarises all available commands for this application. During the collaboration setup this field describes the required functionality of the application for the respective collaboration session, so only the required ones are activated. This proceeding allows e.g. to disable specific functionalities for participants except one, enabling the collaboration session administrator to steer the session correspondingly. The included elements are depicted in Figure 23.

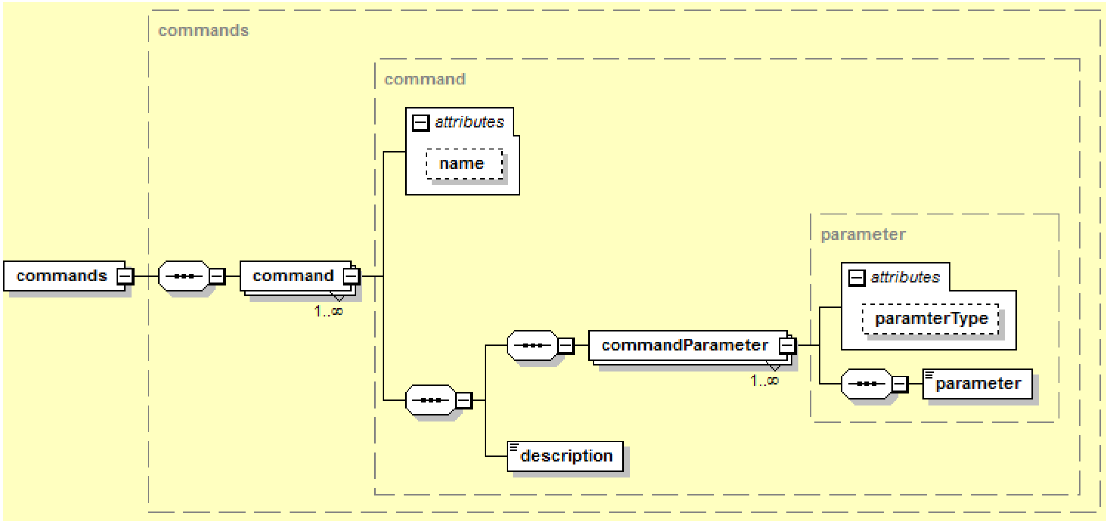

Figure 23 – Commands

In particular, the commands field consists of at least one command element. The command element contains the following elements:

- The attribute *name* describes the name of the corresponding command.
- *commandParameter:* This field describes one command parameter. In detail, the included attribute *parameterType* depicts the type of the corresponding parameter, while the element *parameter* contains the respective value.
- *description:* This element contains a textual, human readable description of the corresponding parameter.

Data

This element depicts the commands to be executed in order to load and save data sets within a collaborative working session. The structure of the *data* element is depicted in Figure 24.

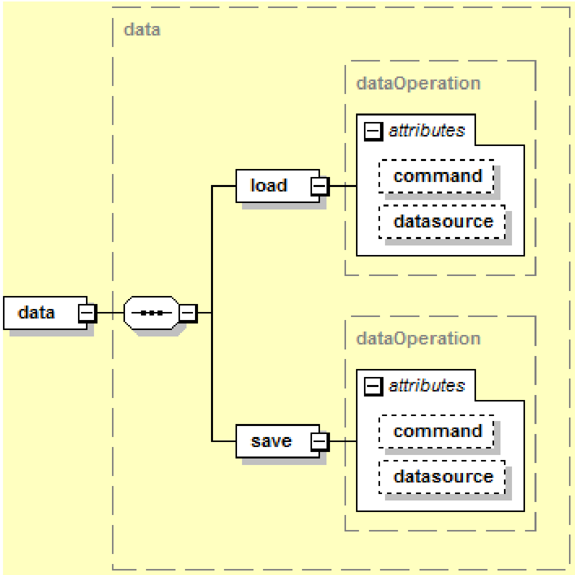

Figure 24 – Data

In detail, for every *datasource* the corresponding *load* and *save* commands are given. These commands are intended to allow executing these commands remotely via the DSM gateway extension. This proceeding enables to share collaboration results between the involved parties, whilst ensuring that only foreseen participants are allowed to apply changes to specific datasets.

Datasources

This element describes the data sources providing the required data for a specific collaboration session. The corresponding structure is given in Figure 25.

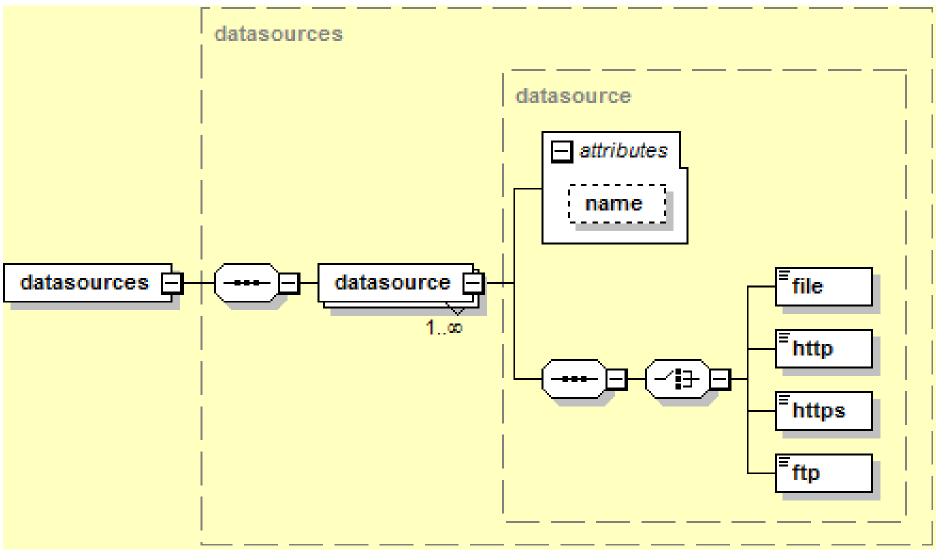

Figure 25 – Datasources

The *datasources* element consists of at least one *datasource* element. The datasource element contains

- The attribute *name* reflecting the name of the corresponding data source.
- The corresponding *access protocol* and the *location* of the corresponding data source. In combination with the commands provided within the *data* element the user is enabled to modify the foreseen datasets correspondingly.

Formats

This element depicts the supported formats of an application. During the registration process of an application this element describes all supported formats of an application, whereas during the collaboration setup process this field depicts the supported formats for this particular collaboration session. The specific structure of the *formats* element is given in Figure 26.

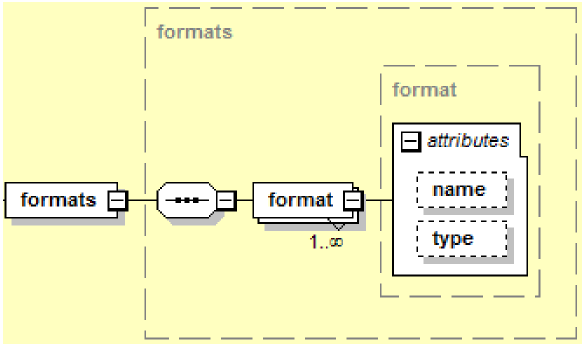

Figure 26 – Formats

In particular, the formats element consists of at least one format element, containing the attributes

- *name*: describing the human readable name of the supported format
- *type*: describing the concrete type of the supported format, e.g. *.pptx for Microsoft PowerPoint 2007 file format.

InitialData

This field provides links to data sources to be loaded right after the start-up of the corresponding application. The respective structure is given in Figure 27.

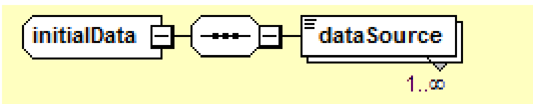

Figure 27 - InitialData

Security

This field depicts all security related information for every collaboration partner, as reflected in Figure 28.

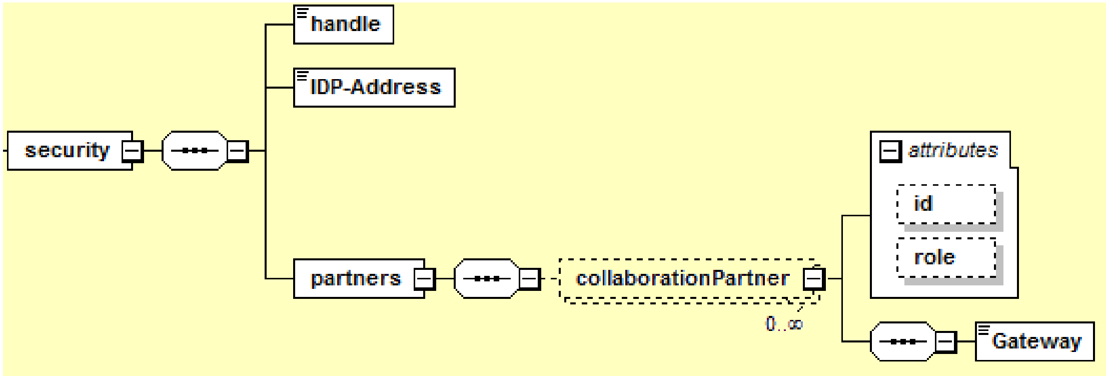

Figure 28 – Security

In detail, the security element provides for every collaboration partner

- *handle*: This field contains the corresponding Shibboleth handle
- *IDP-Address*: This field contains the Shibboleth IDP Address of the corresponding user
- *partners*: this field contains for every collaboration partner the corresponding partner identifier, the role of the partner as well as the address of the partner's DSM gateway address.

4.4.5 Collaboration Setup

In order to setup a collaborative working session, the DSM gateway provides an interface to receive configuration data from the DSM-Manager, as depicted in section 4.4.2.

To enable the dynamic setup of collaborative sessions some information has to be made available in advance. The current context of a user is described with respect to his current location, available hardware, the currently available applications of the user, as well as the corresponding capabilities. In particular the latter information is of significant importance in cases where dynamic and ad-hoc collaboration sessions are to be set up. To provide this data to the knowledge support, every application is registered once being available via the DSM enhanced virtualisation gateway, making this information available for the setup of collaborative working session. This registration includes, in addition to the application type, the specific capabilities of the application. The corresponding details are depicted in section 4.4.4.

To sum up, the DSM extension allows configuring and executing (via the Application Controller) dynamic collaboration sessions. In particular, Figure 29 shows the steps being executed for the setup of a participant.

Collaboration Setup Processing

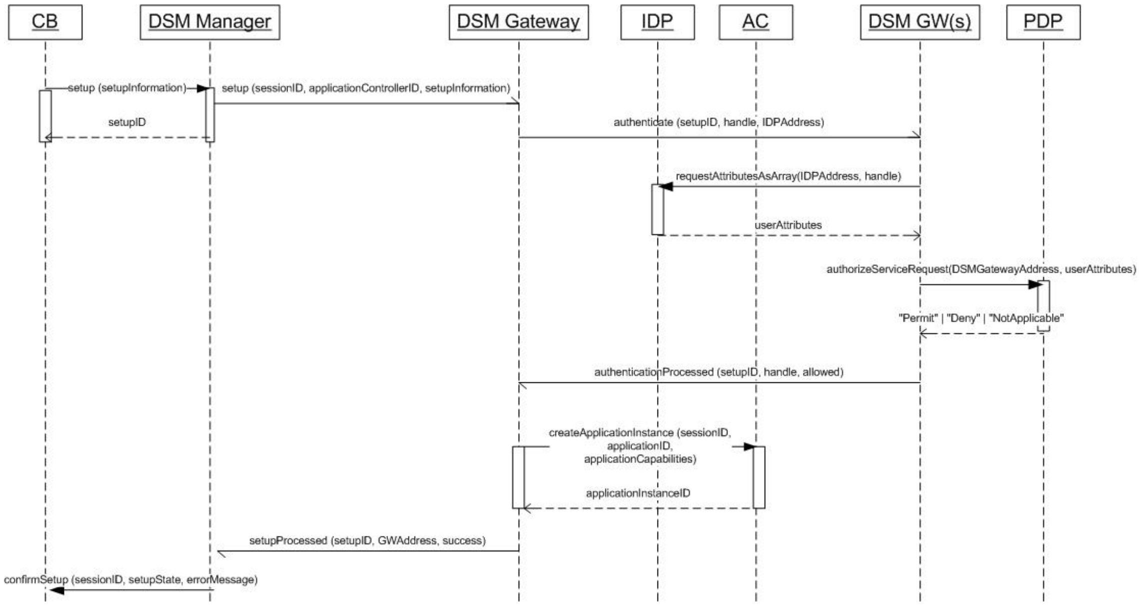

Figure 29 - DSM Setup

1. The collaboration broker (CB) provides the setup information (cf. section 5.1.3) of a specific collaboration session to the DSM-Manager.
2. The DSM-Manager determines the partners involved and sends the corresponding configuration information to the specified DSM partner gateways.
3. After having received the setup information from the DSM Manager the corresponding DSM gateway initiates the authentication procedure by sending an authorization request to all involved DSM partner gateways, as depicted in section 4.4.3.
4. After successful authentication of all involved partners within the corresponding collaborative working session, the DSM gateway configures the corresponding proxy acting as an intermediary for data streams between all collaboration partners. If needed, a data converter will also be configured to allow a data transfer between heterogeneous environments. The finalisation of the configuration is then propagated to the DSM-Manager.
5. After the configuration of all partners has been finished, the collaboration session can now be started by the DSM-Manager. This is done by an explicit start-call to each of the involved partners. After having received a start-call from the DSM-Manager each DSM gateway starts the application via the application controller.

Application Registration

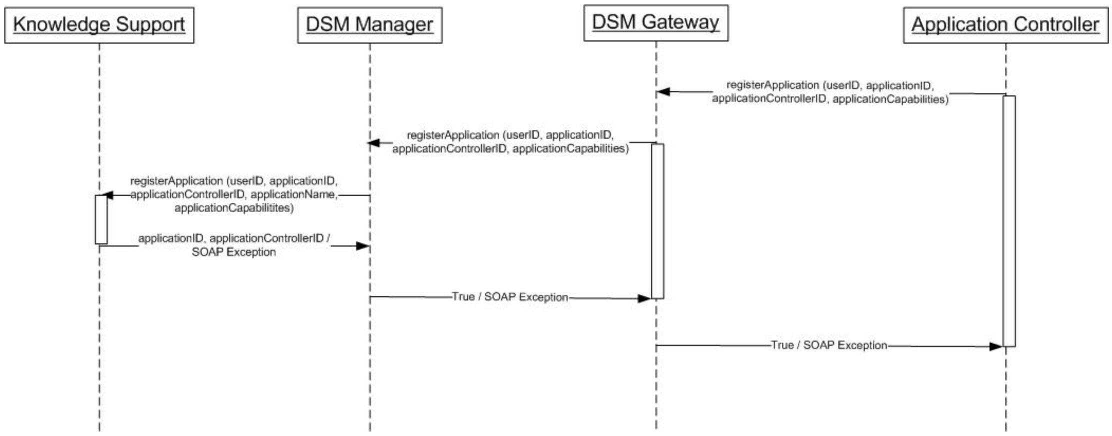

Figure 30 – Application Registration

In order to interact with a user an overview of the available applications has to be provided, via which the interaction with the user can take place. Therefore, whenever a user is available via a specific device, this device registers itself to the corresponding knowledge base by providing all required information being necessary to collaborate with the respective user.

1. The application controller registers every available and for collaboration issues foreseen application via the DSM gateway. In particular, the corresponding user ID, the application ID, the application controller ID as well as the corresponding application capabilities (cf. section 4.4.4) are provided.
2. The DSM gateway transfers this information to the DSM Manager. This way, the application controller just has to be aware of his DSM gateway, and not all the other remaining infrastructure components.
3. Finally, the DSM Manager registers the corresponding applications with the respective credentials at the knowledge support. After this step, the user is attachable for authorized partners to collaborative working sessions, following the credentials provided by the user via the application controller. For example it is possible to restrict the amount of available functions of specific applications during the registration process.

Application Unregistration

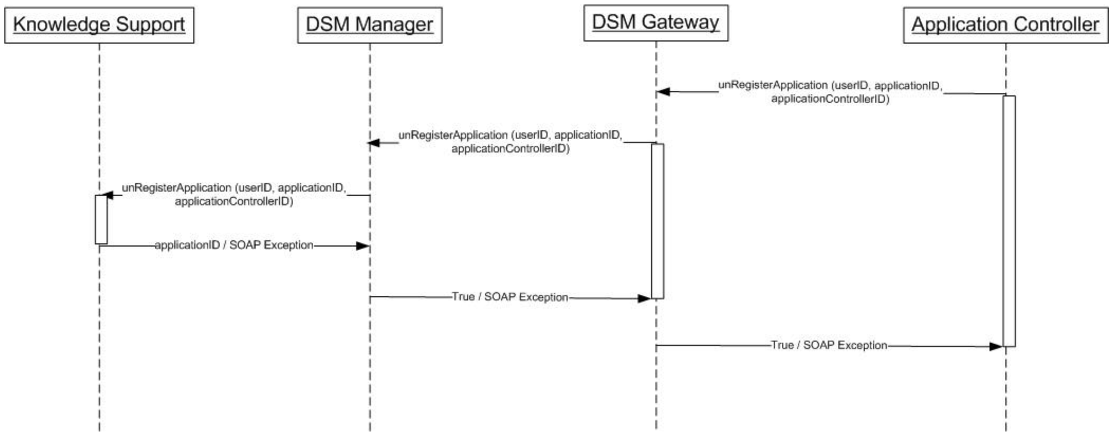

Figure 31 – Application Unregistration

In order to signalise that a specific user or a specific application of the user is no longer available, e.g. in case the user shuts down his registered device, the corresponding application have to be removed from the knowledge support. This signalises for future collaboration sessions that this application is no longer available.

1. The application controller invokes the unregistration process via his local DSM Gateway by providing the corresponding userID, applicationID as well as applicationControllerID.
2. The DSM Gateway forwards this request to the DSM Manager
3. The DSM Manager finally removes the corresponding entries within the knowledge support component.

Application Controller Registration

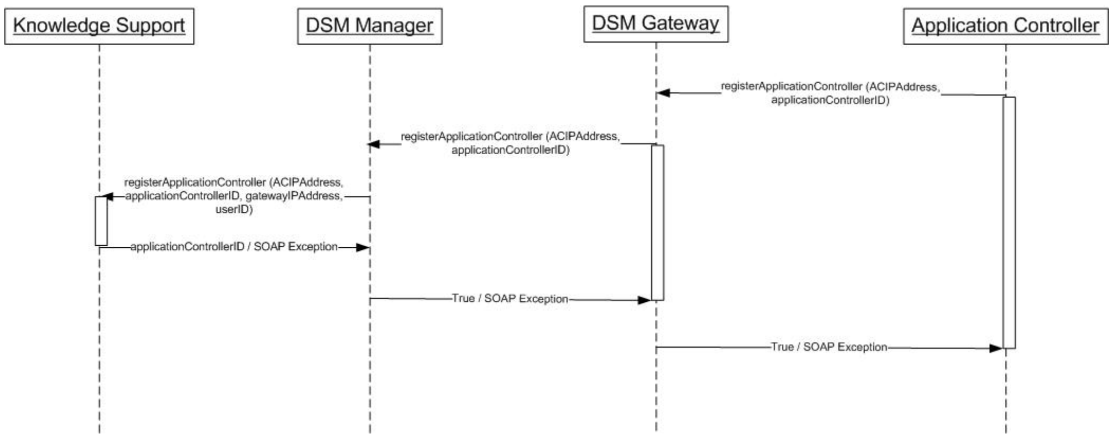

Figure 32 – Application Controller Registration

In order to enable the setup of a collaborative working session, a reference to the corresponding application controller has to be provided, via which the respective applications can be started and steered for a specific user. The application controller is a separate component of the CoSpaces framework allowing for the remote instantiation and steering of applications [103].

1. Right after start-up the application controller registers itself via his local DSM gateway with his applicationControllerID and IP-Address respectively.
2. This information is transferred from the DSM gateway to the DSM Manager component.
3. Finally, the DSM Manager registers the applications controller as well as the respective DSM gateway. The registration of the respective DSM gateway address is needed in order to determine via which DSM gateway a specific application controller is accessible.

Application Controller Unregistration

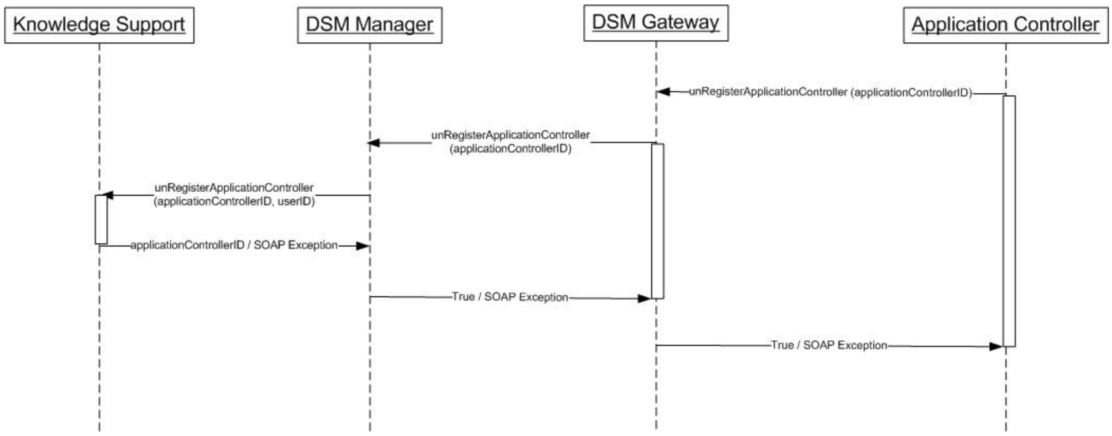

Figure 33 – Application Controller Unregistration

In order to signalise to the collaborative working environment that a specific application controller, and thus user, is no longer available, the following application controller unregistration process has to be processed:

1. The application controller sends an unregistration request to his local DSM Gateway by providing its own application controller ID.
2. The DSM gateway forwards this request to the respective DSM Manager component.
3. Finally, the DSM Manager removes the corresponding entry from the knowledge support. Afterwards, the corresponding application controller, and thus user, is no longer available for future collaborative working sessions. In order to participate again in such a setup, the corresponding application controller has to trigger the application controller registration process again.

Session Reconfiguration

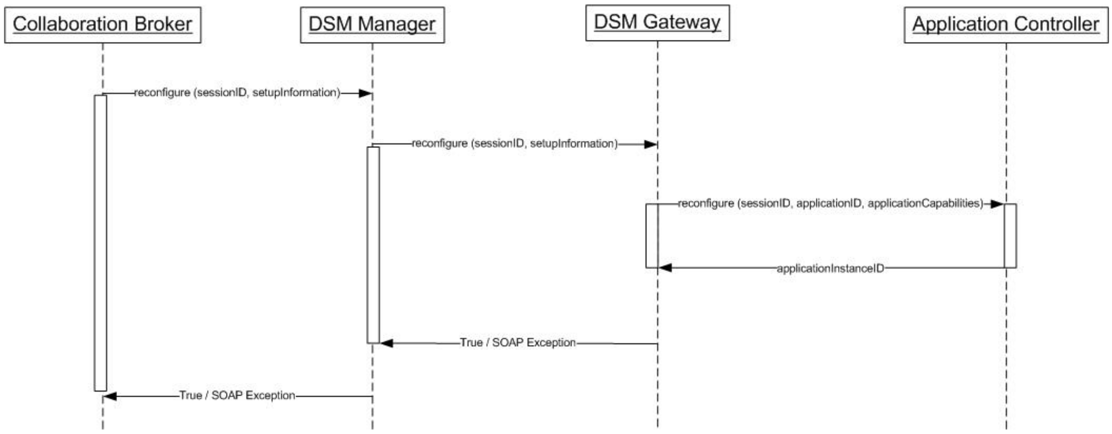

Figure 34 – Session Reconfiguration

During the execution of a collaborative working session it might become necessary to reconfigure the session setup, e.g. since during the session it has been determined that an additional external expert is needed, or that another application / CAD dataset has to be used in order to resolve an issue. Therefore, the session reconfiguration process has to be executed in the following way:

1. Initiated by the collaboration broker a new setup information schema (cf. section 4.4.4) with the respective sessionID is transferred to the DSM Manager.
2. The DSM Manager in the next instance re-initiate the current collaborative working session by signalizing to the corresponding DSM gateway components to reconfigure the collaborative working session with the given session ID and the relevant setup information. In case of participants not yet being part of the running collaborative working session the corresponding authentication process for these participants is executed as depicted in the collaboration setup process.
3. Finally, the DSM Gateway reconfigures the corresponding application controllers according to the changed setup information provided by the collaboration broker.

Collaboration Session Start

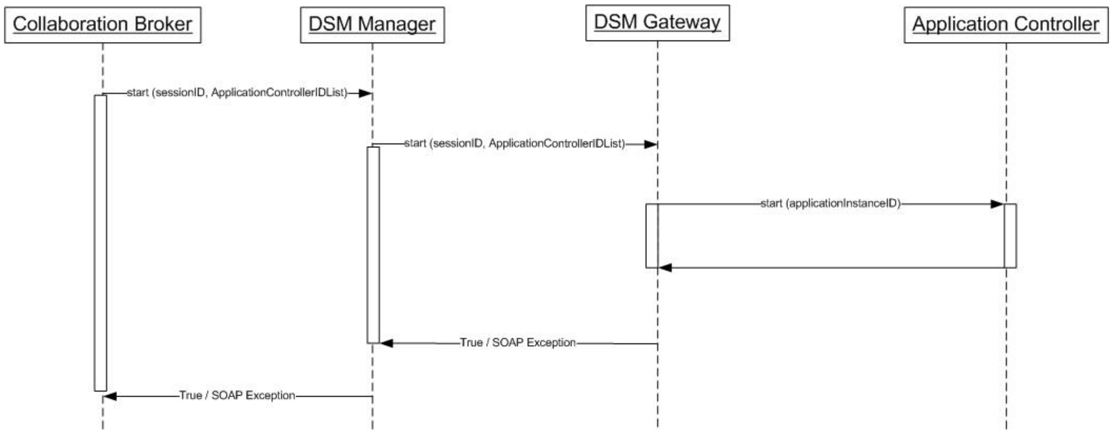

Figure 35 – Collaboration Session Start

After a successful setup of a collaborative working session the session has to be started separately. Therefore, the following process has to be followed:

1. After receiving a starting-message, including the corresponding sessionID from the previous setup process as well as a list of application controllers which should be part of the setup, the DSM Manager initiates the synchronous collaboration session by signalising to every involved DSM gateway (which might control several involved application controllers) to start the respective applications and link to the foreseen datasets. To this end, every involved DSM gateway receives the respective session ID as well as a list of the application controllers to be set up. This proceeding allows starting the session with a selected number of participants in the first instance, and allowing attaching other participants during the execution of the collaborative working session on demand.
2. The DSM Manager distributed the relevant information about the involved partners to the corresponding DSM gateway components, including the list of application controllers being registered via each DSM gateway. So every DSM gateway gets the list of application controllers being registered via this DSM gateway and being foreseen to take part in the current collaboration setup.
3. Finally, the DSM Gateway triggers the session start by signalizing the respective application controllers to start the corresponding applications being initiated for this specific session.

Collaboration Session Stop

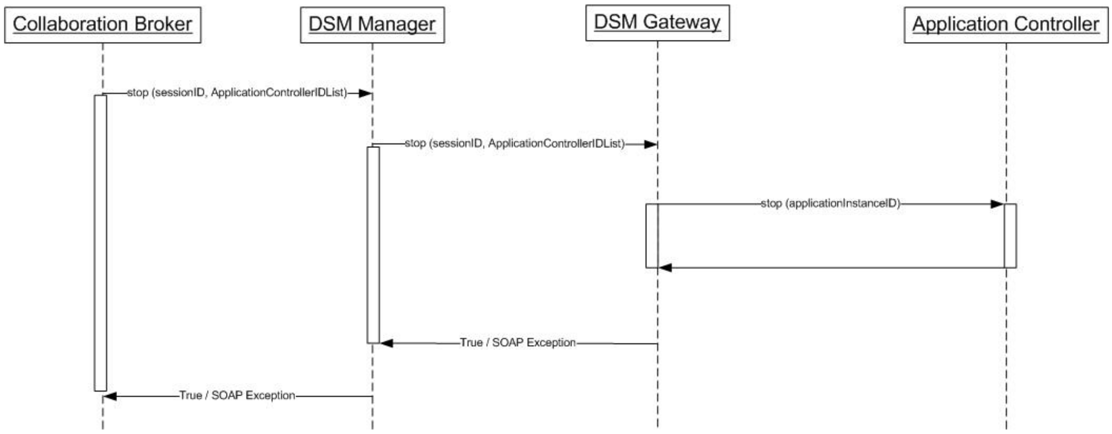

Figure 36 – Collaboration Session Stop

In order to stop a collaborative working session in a controlled way, the following process has to be followed:

1. The collaboration broker initiates the stopping process of a collaborative working session by sending out a corresponding request to the DSM Manager, including the session ID of the session to be stopped, as well as a list of application controllers for which the session should be stopped. This proceeding allows also to stop the collaborative working session only for a selected number of participants, e.g. if some experts are not needed for the entire collaborative working session.
2. The DSM Manager initiates in the next instance the stopping of the corresponding applications being setup for this session by forwarding the stopping-command to the DSM gateways via which the listed application controllers are registered.
3. Finally, the corresponding DSM gateway initiate the termination of the applications being setup for this collaborative working session by signalising the relevant application instance ID from the setup process to the application controller.

4.5 Applying the Virtualisation Gateway approach to the ITIL lifecycle

As depicted in section 3.6 the ITIL lifecycle model reflects the entire lifecycle of an IT service. In this section the presented gateway approach in section 4.2 is going to be mapped to the corresponding ITIL lifecycle phases. In particular it is shown how specific phases of the ITIL lifecycle are supported. It has to be noted that this application of the ITIL lifecycle relies on IT services and infrastructures only, but not to the role-based integration of human resources in a collaborative context.

In the following, the respective virtualisation gateway capabilities are reflected with respect to how far they address the corresponding ITIL application.

4.5.1 Service Strategy

- *Strategy management for IT services*: By applying the virtualisation capabilities of the gateway infrastructure this process is supported by enabling adapting and modifying the corresponding service implementations without affecting the service consumers. This enables the service provider to do all these modifications, e.g. in case of new available hardware and software environments, without the burden to adjust the corresponding client applications.
- *Service portfolio management:* The gateway environment directly supports this process by enabling a transparent management of the available service implementations. In particular the gateway environment allows hiding the specific service implementation from the customer, and thus enables the service provider to steer the routing of incoming invocations in an intelligent way.
- *Financial management of IT services*: By applying the gateway environment as single entry point to the service environment, the service provider can transparently integrate corresponding authentication, authorisation and accounting mechanisms.
- *Demand management*: The gateway environment supports the integration and development of new functionality by encapsulating and thus separating the service implementation from the service interface. This way, new functionality can be integrated in existing or new services, whereas the corresponding virtual service endpoint does not change. This enables the service provider to realise a transparent evolution of the underlying service infrastructure.
- *Business relationship management*: By relating to the virtualisation capabilities of the gateway environment, an enhanced, transparent management of business relationships is possible. In particular, the virtualisation approach enables the integration of virtual service endpoints within services and applications, which can be changed during runtime, facing the demanding aspect of dynamicity in B2B environments.

4.5.2 Service Design

- *Design coordination*: By taking advantage of the virtualisation capabilities of the gateway environment the design process can be distributed by integrating virtual endpoints and finally configure the gateway environment to map the respective messages to concrete service instances as soon as they are available.
- *Service Catalogue*: The service instance registry (SIR, cf. section 4.3) of the gateway environment enables to provide, in addition to the mapping information for virtual service endpoints, additional information about the services, e.g. specific accounting information.
- *Service level management*: Since the gateway environment intercepts all incoming messages for a service provider, at this level the gateway environment can be used to parameterise the invocation of services in such a

way, that, depending on the agreed SLA, different service variations are invoked.

- *Availability management*: The gateway environment provides all required mechanisms in order to steer the service instance invocations, and thus enables to steer also the payload on the corresponding servers.
- *Capacity management*: By monitoring the invocations being processed by the gateway environment this information can be used to support the capacity management process.
- *IT service continuity management*: Due to the transparent interception of messages by the gateway environment new invocations can be easily transferred to new services and IT environments in case of failures and disaster recovery.
- *Information security management system*: By taking advantage of the gateway environment plugin mechanism the infrastructure enables to transparently integrate different policy handlers and security mechanisms. As depicted in section 5.1.2 within the BREIN evaluation scenario the gateway environment has been enhanced with the STS component, allowing for transparent authentication and encryption mechanisms for cross-domain interactions.
- *Supplier Management*: By applying the gateway environment on both, customer and service provider side, and thus using the "double-blind virtualisation" capabilities of the gateway environment as depicted in section 4.1, different suppliers can easily and transparently be integrated or replaced in services and applications during runtime.

4.5.3 Service Transition

- *Transition planning and support*: By using virtual service endpoints the gateway environment provides a significant support in service transition.
- *Change management*: Due to the capabilities of the gateway environment to also change the mapping information from virtual service endpoints to concrete ones the gateway infrastructure enables a dynamic and flexible support of the change management process.
- *Service asset and configuration management*: By applying the gateway environment the configuration process can be optimised, since specific configuration related issues can be encapsulated within a virtual service endpoint, e.g. to use only specific addressing or security mechanisms.
- *Release and deployment management*: By announcing only virtual service endpoint the corresponding service implementations can be changed and adapted in a transparent fashion.
- *Service validation and testing*: By using the gateway environment also different setups of a specific service can be easily integrated and compared.
- *Change evaluation*: By applying the gateway environment a smooth change of adjusted services is doable. In particular if a service has been changed and adapted to face new requirements, this new service is used by all customers

by adjusting the corresponding entries within the SIR.

- *Knowledge management*: The SIR can be used to monitor and store all relevant information about the transition process.

4.5.4 Service Operation

- *Event management*: Since the gateway environment can be configured to intercept all messages, a corresponding event handler can easily get integrated in this environment to trigger the corresponding, foreseen actions for specific events.
- *Incident management*: By taking advantage of the virtualisation capabilities of the gateway environment the normal operation mode can be achieved quickly. In particular if the service is still under development, the gateway environment can also handle request to not yet finalised services by sending default replies to these operations. As soon as the service is fully implemented, the corresponding invocations are forwarded to the service implementation.
- *Request fulfilment*: By applying specific policy handler within the gateway environment, the corresponding service invocation results can be evaluated.
- *Problem management*: In case of errors specific operations can be triggered by the gateway environment by applying corresponding policy handlers.
- *Access management*: By applying corresponding policy handler, as shown in section 5.1.2, security and access management operations can be integrated within the gateway environment in a transparent way.

4.5.5 Continual Service Improvement

As depicted in section 3.6.5 this volume focuses on the continuous improvement of IT services with respect to changing business needs. This is done in particular by identifying and implementing improvements supporting the corresponding business processes. All these steps are directly supported by the virtualisation gateway environment.

4.6 Addressed key requirements

In this section the identified key requirements derived from the scenarios in chapter 2 are confronted with the capabilities of the introduced virtualisation gateway environment. A summary of the addressed key requirements in comparison to current approaches and solutions is given in Figure 37.

4.6.1 Simplicity (Ease of use)

Simplicity of service usage is one of the identified key requirements to be addressed in order to allow flexible, dynamic SOA based environments. As depicted in section 3 current approaches support this aspect only in a limited way. The tackling of these identified gaps by the introduced virtualisation gateway approach are depicted now in the following sections.

Support of different user types

As depicted in section 2.2 the support of users with varying level of expertise is of significant importance. This relies mainly on the amount of integrated knowledge within a service provided by a specific service provider, i.e. a non-experienced user is not able to configure complex IT service environments and thus relies on the capacities of the service provider to provide a service facing his needs without bothering him with the internal details of the respective service. On the other hand, experienced users knowing their problem domain and the respective IT service environment in detail are able to apply problem specific optimisations to the service environment, and thus can ensure better and faster results. Additionally, the service provider just has to provide a basic service for this kind of user, and thus can offer services in an easier way.

As highlighted in section 3.1 the following user types should be considered:

Casual users are supported by the introduced virtualisation gateway approach by encapsulation services and IT infrastructures in a transparent fashion, and thus allowing users of this group to consume complex services and IT infrastructures as well. The virtualisation gateway supports the respective service providers with its capacities to hide the organisational background and domain expertise needed to provide a complex IT service to customers with only little IT knowledge.

The virtualisation gateway approach supports *experienced users* with the integrated configuration capacities, allowing providing services with varying configuration and abstraction opportunities. In particular, the virtualisation gateway environment allows providing configuration capabilities required by the user, whereas hiding the ones not being necessary for the user. The respective degree of configurability can be chosen by the respective service provider.

Expert users are supported by the virtualisation gateway environment by also allowing for providing extensive configuration and setup capabilities to users, whereas the service provider has the full control about the amount of configuration capabilities given to the user. In particular, service providers can thus provide varying cost models with respect to the configuration capabilities.

Dynamic service coupling and orchestration

Increasing the dynamicity in B2B environments is of significant importance. In particular the adjustment of service and IT infrastructures with respect to changing business goals and environments requires a flexible and adjustable environment (cf. section 2.2 and 2.3). In particular the integration of new service providers in running processes, as well as the replacement of existing service providers during application execution has to be supported in order to allow service consumers to adjust their services dynamically. To this end, the virtualisation gateway environment allows invoking services via virtual service endpoints, which are mapped on demand to concrete service instances. This proceeding ensures that the underlying service and IT infrastructure is encapsulated and thus hidden from the users, allowing for a transparent adaptation of the underlying service and IT infrastructure.

In particular, the virtualisation gateway enables for the transparent integration and dynamic replacement of integrated third party service providers without affecting the user's application or workflow.

Additionally, the virtualisation gateway approach also allows the transparent handling of errors by acting as a transparent message interceptor. In particular the flexible enhancement capabilities of the virtualisation gateway environment with respect to dynamically integrate additional message processing filters, the virtualisation gateway allows thus for integrating message filters catching error messages and trigger respective corrective actions. Corrective actions are e.g. to redirect the initial request to a service providing comparable functionalities, or even modify and redirect the initial request to an external service provider. The entire proceeding is encapsulated in the virtualisation gateway approach and thus transparent for the service consumer.

Finally, the virtualisation gateway tackles the identified issue about process evolution as well. To this end, the virtualisation gateway approach supports the invocation of virtual service endpoints, allowing the dynamic adjustment of the underlying service and IT infrastructure without affecting the original process description. The introduced approach also supports the process evolutions itself by enabling software agent based environments to integrate services and IT infrastructures via virtual service endpoints, and thus decoupling the process logic from the respective technical infrastructure.

Support the integration of heterogeneous application, service and IT environments

As shown in chapter 2 the integration and coupling of specialised services and environments is of significant importance to ensure best suitable outcomes. In particular in the engineering domain the focussed integration of specialised services and expert knowledge is an essential success factor.

To this end, the virtualisation gateway approach enables the transparent integration of heterogeneous service environment by encapsulating the respective service and IT infrastructure with virtual service endpoints in a transparent fashion. In particular the business-to-IT problem is tackled by the virtualisation gateway environment by decoupling the different views, namely of the involved domain experts and the respective IT infrastructure and service providers, as depicted in section 4.2. Especially since the virtualisation gateway allows providing different abstraction levels of the provided services, namely enabling the provision of specific configuration and optimisation opportunities for domain experts, and thus providing the best suitable support for the respective experts, this results in an optimal usage of the available expertise and service performance. This proceeding ensures the required decoupling of domain knowledge and technical IT infrastructures. Additionally, the technical complexity of the underlying service and IT infrastructures is encapsulated and thus allows for the transparent integration of complex service environment into processes and applications by following a functional viewpoint.

Finally, the virtualisation gateway enables as well to integrate internally as well as externally provided services in a transparent fashion, avoiding that service consumers have to take into account specific contracting issues when integrating a transparent service description within their processes and applications. The virtualisation gateway approach allows for providing centralised policies describing the determination process for the best suitable service for a specific issue in a transparent fashion. In this way, the service consumer does not have to be aware if he is using an internal or an external service, since the respective invocation is handled transparently within the virtualisation gateway environment.

4.6.2 Provisioning

In this section, the addressed key requirements by the virtualisation gateway approach with respect to service provisioning and realisation are summarised.

Enable the "Programming in the small" and "Programming in the large" paradigm

The SOA paradigm itself, as depicted in section 3.3, provides a conceptual basis in the sense to couple services in a technology independent way. However, current approaches realising a SOA-based environment lack in supporting the required dynamicity and technology independence. To this end, the introduced virtualisation gateway environment provides an enhanced realisation of this paradigm, in particular by also considering human resources in a SOA like fashion. In particular, service developers and process designers are enabled by applying the virtualisation gateway environment to concentrate on the realisation of the required and intended functionality. Service providers are also enabled to assemble and combine existing services from a business perspective in order to provide new and more complex IT services and products to external and internal customers. Additionally, applying the introduced approach allows domain experts and service consumers to concentrate on their specific expertise domain and consume provided services in an easy-to-use and technology-independent fashion.

This proceeding ensures to follow the well-established paradigm "Programming in the small" by supporting developers to concentrate on the realisation of specific functionality and provide them in a re-usable, technology independent way. "Programming in the large" is supported by the virtualisation gateway approach by enabling service consumers and process designers to concentrate in assembling the needed functionality for a specific goal by assembling the provided services.

Service Lifecycle Management Support

As depicted in detail in section 4.5, the virtualisation gateway approach supports all relevant aspects of the ITIL service lifecycle.

Transparent and integrated accounting support

The introduced virtualisation gateway approach allows for integrating specific messaging filter in order to transparently provide accounting functionality without affecting the respective service implementation. If the service consumer also uses a virtualisation gateway at his place (using the double-blind invocation approach as depicted in section 4.1.3) this accounting information can also be integrated in a transparent fashion without affecting the service consumer or his application / service at all.

4.6.3 Interoperability

As depicted in section 3.4.2 interoperability is mainly related to the application of common standards. The introduced virtualisation gateway approach can be applied as a single service entry point. Therefore the virtualisation gateway is the only involved component that has to support the relevant standards, but not the coupled services.

As depicted in section 3.4 there are currently a lot of different, partially incompatible web service-based standards available. However, in order to allow interoperable access to services and infrastructures, a common understanding about message transmission and formation is of essential importance.

The effort to support different standards, including also different realisations of these standards, is thus reduced to one component, the virtualisation gateway, whereas the underlying service infrastructure is not affected by these standards. This is particularly the case when different service consumers or new service consumers invoke a specific service, applying different standards and the respective realisations within their own services and processes. To this end the virtualisation gateway can be applied to handle all varieties of standards and the respective realisations, whereby ensuring that this "wave" of different, partially incompatible, realisations do not affect the underlying service infrastructure. In particular, by applying respective message filters for each standard to be supported, the virtualisation gateway provides a flexible, extensible environment, also being able to cope with new standards and respective implementations. This integrated, extensible filter mechanism also allows coping with changes in internal standards and realisations in the same, transparent way. Additionally, this ensures that the respective services can be executed target-oriented, providing optimal performance with respect to the provided functionality.

This procedure ensures on one hand the required flexibility and dynamicity, since service providers are able to cope with varieties of different standards and realisations whereby providing standardised services. On the other hand, service consumers do not have to cope with specific standards being supported by specific service providers. This applies in particular to the application of the virtualisation gateway approach following the "double-blind virtualisation" approach. In this deployment scenario the service consumer virtualisation gateway takes over the required standard related transformations.

4.6.4 Time Constraints

Ensuring effective access to web services is depicted as an essential key requirement in section 2.5. To this end, considering the virtualisation gateway as a single entry point to a service provider's service landscape, the available resources on which a specific service is deployed can be enhanced on-demand and as needed in order to ensure the availability of a service within a specific timeframe. Additionally, by increasing the available resources in a transparent fashion the timeframe needed to provide the requested service results can also be steered to some degree. Therefore, the virtualisation gateway environment can also act as a load-balancer by routing service invocations to different service instances, by taking into account the current payload of the respective service environments. This proceeding enables to provide better scaling services and IT environments.

By applying the virtualisation gateway in a hierarchical manner, as depicted in [138], enables the entire environment additionally to provide large, complex and heterogeneous service environments in an efficient and scaling way.

4.6.5 Reliability and Robustness

Reliability and robustness of the service infrastructure ensure the depicted encapsulation of domain expertise in different layers (cf. section 2.2 and 4.2). In particular the introduced domain specific abstraction views on the underlying service infrastructures enforce a stable, reliable and robust provisioning of the respective services.

The introduced virtualisation gateway environment fully supports this aspect by encapsulating the relevant environments in a transparent fashion. In case of failing or replaced services, the virtualisation gateway can cope with these issues transparently (e.g. by re-invoking another service providing the same results as the failing one) by keeping the entire process, from a customer point of view, stable.

Additionally, the virtualisation gateway approach allows also for an enhanced, transparent support for security mechanisms, which are of essential importance ensuring reliable and secure communication between the involved parties. To this end, the flexible enhancement capabilities of the virtualisation gateway environment enable to integrate respective security handling services in a transparent fashion. Therefore the virtualisation gateway approach has been enhanced within a concrete application scenario with a security token service (cf. section 5.1.2), transparently allowing for the secure interaction between service consumers and service providers, whereas ensuring that only foreseen users are allowed to access the respective service environment. In particular, this enhancement allows also for the dynamic, on-demand adaptation of the security settings during runtime, thus ensuring that even after changes in the service landscape the entire environment is still operating in a stable, reliable and secure way.

Finally, in order to ensure a stable and reliable operation of user processes, in particular with respect to the integration of human experts, the respective service consumer context has to be considered as well. As depicted in section 4.4.4 this includes e.g. the location of a service consumer and the devices being available for this user. Based on this information, the user context can be derived and applied to respective user credentials, e.g. when the user is currently connected with a mobile device, business critical data should not be transmitted to this device due to potential industrial spying issues.

The virtualisation gateway allows to support the processing of this context related information as well, thus ensuring a stable and reliable operation.

4.6.6 Collaboration Support

In this section the addressed key requirements by the virtualisation gateway approach, namely the DSM extension, are summarised.

Synchronous collaboration support

The synchronous collaboration between human experts is of significant importance, as depicted in section 2.4, in order to ensure the integration of the respective domain expertise, which is particularly required for the construction of complex engineering products. To this end, as depicted in section 4.4 the DSM extension of the virtualisation gateway environment allows to setup, configure and steer synchronous collaborative session between human experts, whereby taking into account the respective available service environment ensuring the best suitable tooling support for the involved experts in a transparent fashion. In particular the DSM extension enables to setup synchronous collaborative working sessions by providing an infrastructure coping with providing and processing all required configuration and context information, as depicted in detail in section 4.4.4, to ensure synchronous collaborations with collaboration tools in a SOA like manner.

Finally, the DSM extension of the virtualisation gateway approach underlines the flexible enhancement capabilities of this approach to new application fields. This enhancement supports in particular the remote invocation of human resources.

Role-based collaboration support

As depicted in section 4.4.3 the DSM extension of the virtualisation gateway approach supports the role-based integration of human experts in complex processes in a transparent fashion. This proceeding ensures the required flexibility, in particular with respect to the used services required for specific issues, dynamic changes in the service landscape and required expertise to tackle a specific issue. Namely, during the development of a complex engineering product, issues might arise not being foreseeable from the very beginning. To this end, specific expertise is required to resolve this issue, which has to be integrated in the development process. Therefore, the introduced DSM gateway extension allows for setting up and enhancing synchronous collaboration sessions by enabling integrating new human experts in a role-based fashion following a SOA based approach (cf. section 4.4.3) and new services on the fly, whereby ensuring a secure data transmission and sharing in a transparent fashion and ensuring that every involved collaboration partner keeps the control of their data.

4.6.7 *Summary*

This section summarises the addressed key requirements by the introduced virtualisation gateway approach in section 4.6. To this end, Figure 39 depicts the addressed key requirements, allowing a comparison with the identified gaps of current approaches.

	Simplicity (Ease of use)			Provisioning			Interoperability	Time Constraints	Reliability and Robustness	Collaboration Support	
	1.1	1.2	1.3	2.1	2.2	2.3	3.1	4.1	5.1	6.1	6.2
B2B - ESB	--	-	-	-	--	--	-	--	-	--	--
SOA	--	-	-	-	-	--	-	--	--	--	--
Webservices	--	--	-	-	--	--	-	--	--	--	--
Process Description Languages	--	--	-	-	--	--	-	--	-	--	--
Service Lifecycle Management - ITIL	--	--	--	--	-	--	--	--	--	--	--
Integration of Human resources	--	--	--	--	--	--	--	--	--	--	-
Virtualisation Gateway Approach	++	+	++	++	+	++	+	+	+	++	++

--	Identified requirement is not addressed
-	Partly address the identified requirement, but does not meet it in an adequate way
+	Meet the identified requirements, but requires additional efforts to address it in an adequate way
++	Fully supports the identified requirement

Figure 37 – Summary of addressed key requirements

5. Evaluation of the gateway approach

In this chapter the virtualisation gateway approach is applied to concrete use case scenarios, realised within various research projects. In these use cases, the virtualisation gateway approach capabilities are evaluated with respect to tackling the addressed gaps of common approaches (cf. section 3.8).

The application of the virtualisation gateway to specific use cases is depicted in section 5.1.

In section 5.2 the evaluation results of the virtualisation gateway approach are presented.

5.1 Application of the gateway approach to use cases

The introduced virtualisation gateway approach provides a new abstraction layer for SOA-based infrastructures addressing the needs of dynamic eBusiness environments. The main benefit is a significantly increased of flexibility from both the technical as well as the business perspective. From a technical point of view it is now possible to bind services statically in application and services whereas the corresponding implementation can be adapted or even migrated. Additionally the service provider can announce the available services independently from the protocol the potential service consumers are going to use to interact with the respective service environment. This way of announcing services allows its providers to use and re-use already existing services in a very easy way. Besides the composition of services within a process has also been improved: depending on the target outcome of a process services can now get combined focussing on the announced functionality in a technology independent way. The service provider is consequently able to provide “new” services depending on the currently available resources, services and their current payload.

Finally, the DSM extension of the virtualisation gateway approach (cf. section 4.4) enables the integration of human resources by allowing synchronous, tool-supported collaboration between individuals in a SOA like fashion.

The main benefits of this approach can be summarised as follows:

- Role-based integration of human resources in eBusiness environments.
- Increased customers satisfaction: service providers are now able to adapt very fast to different and changing customers’ needs.
- Easy and improved maintenance of provided services.
- Efficient development since the customers’ technical point of view does not need to be considered within a concrete service implementation.
- Easy adaptation of provided services to changing web standards via the virtualisation gateway as a single entry point for service invocations. Since web standards in the area of security, addressing, reliable message transfer, etc. are continuously under development and improvement, the corresponding service provider has to support as most of these standards as possible.

- Decreased costs due to the increased flexibility in choosing and integrating different service providers in a transparent and dynamic fashion.
- Loose coupling of services can be realised in a straight forward way by applying the virtualisation gateway (cf. section 4.1)
- Monitoring and logging in abstraction layer: enables detailed analysis of used services, interactions protocols, etc.
- Public governance guidelines force the realisation of specific functionality which is often not conforming with the current service realisation.

In order to evaluate the conceptual approach of the introduced virtualisation infrastructure the realisation of the gateway prototype (cf. section 4.3) has been integrated within the research projects CoSpaces, BREIN and GAMES.

Therefore, the virtualisation gateway approach being presented within this thesis is used to transparently integrate an authentication and authorization infrastructure within these frameworks without affecting the underlying steering and coordination infrastructure components as well as the corresponding shared services and data. Consequently, the service consumers as well as the service providers do not have to consider security aspects within their tasks whereby the framework ensures that only foreseen partners are allowed to access the corresponding services and data sets.

5.1.1 Offering and consumption of complex IT services

As described in section 3.5.1, BPEL can be used to describe scientific workflows as well; however, the realisation of this kind of workflows with BPEL is still facing some major gaps. Therefore, it has been depicted in this section how the introduced virtualisation infrastructure can enable the usage of BPEL for such workflows in order to close these gaps. In the following, an example of a workflow description of the introduced initial HPC usage scenario is depicted.

At this stage, it is not reflected how the relevant activities have to get modelled for a scientific environment. This is described in detail in [163]. Within the HPC usage case, BPEL has been used in order to control the setup and execution of a scientific workflow management system, as presented in [164].

The following BPEL process has been developed using the integrated eclipse BPEL designer[7]. This is an example of how standard graphical editors can be used to orchestrate such complex processes and make them available as easy-to-use services for external customers. By proceeding like this, scientist can concentrate on the simulation itself without the need to take into account the underlying infrastructure. Figure 38 depicts the graphical representation of this BPEL process from the eclipse BPEL designer, reflecting our presented cancellous bone simulation.

[7] http://www.eclipse.org/bpel/

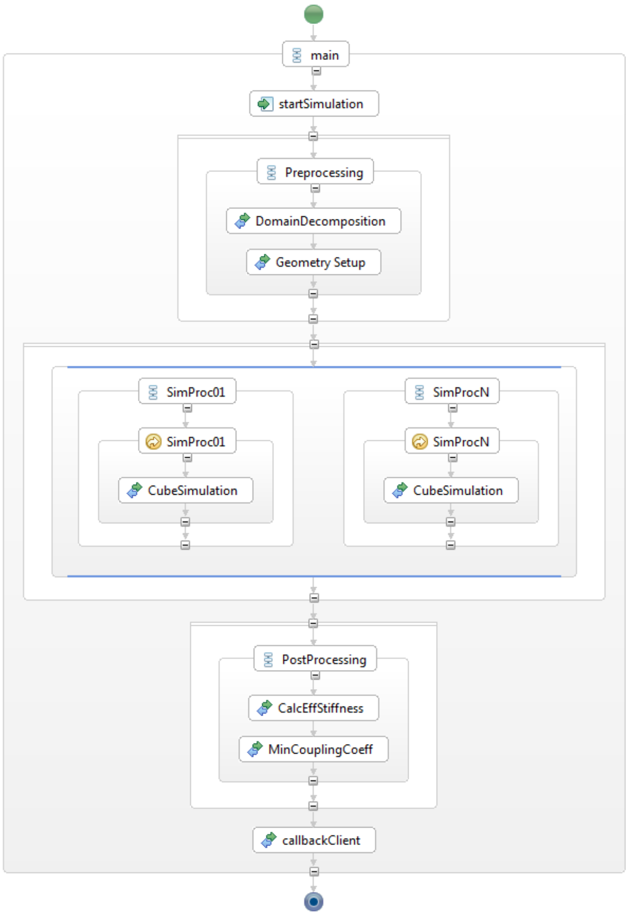

Figure 38 – BPEL Process for a cancellous bone simulation

The process is started by receiving the invocation message via the respective web service interface by triggering the startSimulation method. In the following the defined activities of the simulation process are executed, namely

- The preprocessing
- The simulation itself
- The postprocessing of the simulation results

Therefore, the required functionality of the respective steps is provided by separate applications, which can be invoked via a virtual service interface. These invocations are mapped in the next step to concrete service endpoints, or might be forwarded to an external party. However, the scientist designing this process does not have to be aware of the location of these services. He can just orchestrate the required functions in the BPEL process and the virtualisation infrastructure takes care that the relevant invocations are sent to the correct services. The process remains stable, whereas the respective underlying service infrastructure may change.

The depicted BPEL process in Figure 38 has been applied within the GAMES HLRS HPC testbed evaluation to execute the cancellous bone simulation. The introduced virtualisation gateway enabled the abstract orchestration of the cancellous bone simulation process via virtual service endpoints in BPEL, allowing also non-IT affine users to develop and consume such complex services. To this end, the virtualisation gateway approach enabled to develop a simulation process at from an abstract viewpoint, whereby ensuring an optimal resource usage during execution. In addition, the virtualisation gateway environment has been coupled with different energy-consumption optimisation components [90], thus enabling also for the high-level orchestration and usage of complex IT services and resources, whereby allowing for the transparent optimisation of different aspects (like in this case the energy consumption).

5.1.2 Inter-Organisational Collaboration in a Multi-Agent environment

The IP BREIN tackled the challenge that in today's world, enterprises, independent of their size, have to cooperate closely with other companies to keep their competitiveness, as no company is capable of fulfilling all requirements alone. Especially for SMEs these collaborations are not really cost-efficient, as they have to put in high efforts to be able to compete on the market with other players. Therefore BREIN enables service providers to reduce costs whilst maximizing profit by providing a framework that will automatically adapt to changes in individual business needs and/or environment in an intelligent manner. Cost and effort for service provisioning will be greatly reduced by simplifying business goal definition, intelligent optimization and decision making support. Therefore, the introduced virtualisation gateway approach has been applied within the BREIN framework to allow the framework to integrate virtual resources in workflows to achieve a higher degree of flexibility. This approach allows for both an easier and more abstract usage of resources (e.g. a customer just invokes a "simulation" service without considering technical details) as well as an increased support of dynamism in such environments by easing the replacement of service providers (e.g. the customer still invokes a "simulation" service whilst his own company gateway redirects this request to a new service provider).

The existing WSDL approach would affect in such a dynamic environment that every client of a specific service provider has to adapt their applications to new service interfaces in case of any modification of the corresponding service provider infrastructure or in the case of a service provider change. Additionally a lot of added effort has to be spent for the corresponding service setup. But with the new virtualisation gateway the client does not need to update his code, although the syntactical interface may have changed, since the messages of the calls via the old interface are mapped or, if needed, transformed, to the interface of the new service.

With the virtualisation gateway approach the service provider is now able to implement any adoption needed, even regarding changes in inter-communication standards. Now it is possible to provide several interfaces for the same service, each adapting to another interface. For example, one customer needs a secure connection to the service because sensible data have to be transferred while another one uses another version of WS-Addressing [21] or WS-Security [128].

The prototype realisation of the virtualisation gateway as depicted in section 4.3 has been evaluated and applied within the IP BREIN final airport logistics demonstrator. In the following section a short overview of the scenario implementation is given. A detailed overview of the respective evaluation environment is depicted in [81].

Scenario implementation

In order to face the raised issues, multi-agent systems are coupled with semantic web technologies in order to enable the respective agents to act autonomously on reaching their specific target, whereby enabling them to adjust their own strategy and linked services. The virtualisation gateway environment enables in this context for a decoupled interaction with the respective software agents, whereby ensuring interoperable, reliable and secure communication channels. To this end, the virtualisation gateway environment has been enhanced with a specific security policy filter. This has been done by taking advantage of the respective virtualisation gateway enhancement capabilities (cf. section 4.3) in order to integrate the STS component [125] allowing a transparent handling of all security related functionalities.

With the semantic technology enhancement of the agents they are enabled to take advantage of the interface intermediation of the virtualisation gateway environment, by allowing them to interact with the corresponding partner agents without the burden to take into account specific implementation details and respective security issues. Figure 39 depicts how the gateway environment enables for the realisation of this scenario.

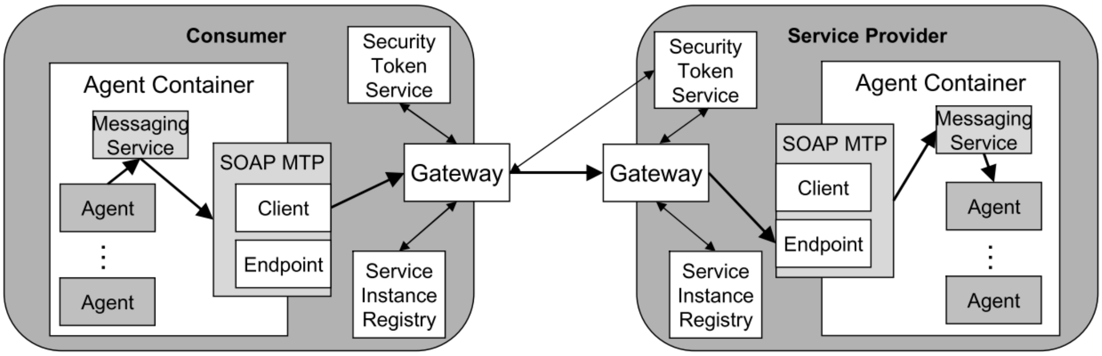

Figure 39 – Coupling Gateway Infrastructure with agent-based environments [81]

5.1.3 Distributed Workspaces – Integration of Human Resources

A prototype realisation of the DSM gateway extension has been integrated and evaluated within the CoSpaces project. In this subsection, a short overview of the scenario is given. More demonstrations of the entire CoSpaces environment can be found at the project website[8].

CoSpaces framework

In this section a short introduction about the CoSpaces software framework is given, showing the integration of the DSM enhancement of the introduced virtualisation gateway approach in a specific scenario.

As shown in Figure 40, the CoSpaces framework consists of several components, namely the Portal, Knowledge Support, Group Management, Positioning and Identification, Resource Management, Dynamic Session Management and

8 http://www.cospaces.org/demonstrators

Application Controller. A brief description of these components is given in the following:

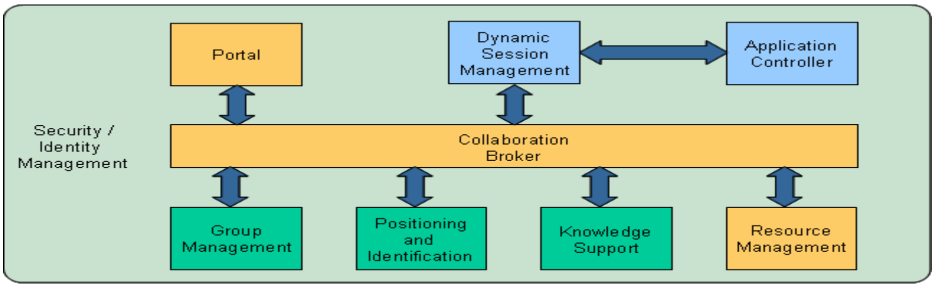

Figure 40 - Conceptual View of the CoSpaces Framework [94]

- *Portal:* This module allows the initiator of the collaboration to plan and execute a collaborative session by selecting people and resources.
- *Knowledge Support*: This module maintains information on experts and necessary documents to provide context support, basing on the BSCW environment [14].
- *Group Manager*: This module maintains the information about people including personal details, access rights, roles etc.
- *Position and Identification*: This module tracks the activities of the people using various tracking technologies.
- *Resource Management*: After successful authentication and authorisation via the DSM, this module allows transferring relevant information to local service instances, in particular in order to increase performance.
- *Application controller*: This module allows for the instantiating and steering of locally running applications.
- *Collaboration Broker*: This module acts as an intermediary service between all other modules. Beside others, the collaboration broker provides the required setup information for the DSM.

Scenario implementation

A first prototype of the DSM gateway extension has been demonstrated at the CoSpaces Framework demonstration at the Supercomputing Conference 2007 in Reno..

In this demonstration, a collaborative session has been set-up, demonstrating the CoSpaces environment at the High Performance Computing Centre booth by applying a digital mock-up scenario, showing a distributed discussion of a prototype between partners in Stuttgart, Reno and Cologne. The data set under discussion relates to a new landing gear of an aircraft together with its tubing. In this scenario there have been three virtualisation environments involved in order to enable the involved partners to work collaboratively on the mock-up, namely:

- A powerwall at the HLRS booth in Reno
- The CAVE of HLRS in Stuttgart
- A powerwall in Cologne.

The partners in Stuttgart and Cologne represented internal engineers at different manufacturing sites, whereas the partner in Reno represented an external expert being involved in this discussion.

The discussion focussed on two issues:

- A clash in the struts
- The problem, that the wheels of the airplane are apparently too heavy.

During the discussion, the raised issues have been solved and annotated collaboratively. Figure 41 shows a snapshot of the respective demonstration session.

Figure 41 – CoSpaces Demonstration Setup

5.2 Evaluation

In this section, the evaluation results of the virtualisation gateway approach as well as the introduced DSM extension are summarised. Therefore, in section 5.2.1 the qualitative evaluation of the virtualisation gateway approach is presented, taking into account functional aspects and measurements of the introduced approach.

In section 5.2.2 a quantitative assessment of the virtualisation gateway approach is given, depicting in detail measurements about the performance and scalability of the introduced approach.

5.2.1 Qualitative Evaluation

The gateway toolkit supports dynamic and extensible message policy enforcement – this ranges from routing (virtualisation & encapsulation) over authorisation and access restriction (in combination with e.g. the STS [125]) to message transformation (interoperability) and collaboration support. This toolkit enhances the messaging channels of (web service) based transactions, allowing to enforce any amount of policies upon in- and outgoing messages (cf. section 4.1). Through using “double-blind virtualisation” mechanisms, i.e. by deploying the gateway on both consumer and service provider side (cf. section 4.1.3), it is possible to alter resources and providers without affecting the calling applications. In order to evaluate the gateway toolkit in a qualitative way the following aspects have been considered:

- Simplicity (ease of use), cf. Table 2,
- Reliability & Robustness, cf. Table 3,
- Provisioning, cf. Table 4,
- Interoperability, cf. Table 5,
- Time Constraints, cf. Table 6,
- Collaboration Support, cf. Table 7

Table 2 – Evaluation - Simplicity (Ease of use)

Simplicity (ease of use)	
Description	Managing the dynamicity and individual messaging requirements per participant in a virtual organisation, including authentication mechanisms, physical location etc. is unnecessarily complicated. In particular maintenance of dynamic endpoints in environments such as dynamic virtual organisations or cloud infrastructures would require the end-user to configure an endpoint that can easily be changed (potentially from external) or he has to constantly update it himself. The principle of virtualisation simplifies such maintenance and should allow end users to have a constant virtual communication endpoint, as well as providers to easily expose dynamic services (see also provisioning). What is more, it should allow for addressing different messaging requirements, such as authentication, encryption and message conversion.
Measurement	Difficulty to change a physical endpoint and impact on the overall infrastructure Complexity to achieve enhanced policy enforcement, such as security related requirements
Assessment	The gateway infrastructure extends common virtualisation means to the general web service world and enhances it with two specific capabilities: (1) by “double blind invocation”, meaning that both customer and provider maintain a virtualisation interface and (2) by a policy filter channel which can be easily extended. The double blind invocation allows that endpoints can change (at least) at two levels: within the VO (change of service provider) and within an individual provider (change of resource) without affecting the endpoint of the customer’s application. Adaptation is performed by a simple management invocation of the virtualisation gateway affected (i.e. for VO level on the end-user side and for resource-level on the provider side), thus automatically changing all related communication.

Since this can be regarded as an enhanced messaging channel, it was extended with the capabilities to host a dynamic number of policy filters within its channel, allowing both customer and provider to enforce potentially any kind of policy on incoming and / or outgoing messages, including authentication and even transformation. The policy filters can be principally written in any language and can act upon the whole message easily, thus allowing for custom transformation even between invoking methods. They are individually registered in the messaging infrastructure.

Table 3 – Evaluation - Reliability and Robustness

Reliability & Robustness	
Description	Messaging between participants in a dynamic virtual organisation must be reliable and robust. This does not imply that any transaction must conform to the reliable messaging specifications, but more importantly, that no endpoints are lost, no channel broken during mid-transaction etc.
Measurement	Are transactions via the messaging infrastructure as reliable as without it? Can the messaging infrastructure compensate for dynamicity in the virtual organisation?
Assessment	The gateway infrastructure shows about the same reliability as the .NET framework, in other words messages are not more likely to be lost than using the web service communication without the gateway. Notably, the messaging infrastructure does increase the latency of messaging (see section 5.2.2), which implies that the overall throughput decreases. What is more important and what is the specific focus of the virtualisation gateway infrastructure is that the virtual organisation as a whole becomes more reliable. In particular, dealing with dynamic endpoints is complicated without the virtualisation gateway. In principle, any endpoint can move during transaction without leading to errors in the overall transaction – yet practically, this will lead to loss of web service state and thus leads to errors on the actual content level.

Table 4 – Evaluation - Provisioning

Provisioning	
Description	Virtualisation provides a means to simplify provisioning in so far as that it allows exposing services more easily, without having to cater for all the messaging details. In particular this affects dynamic services that typically require a lot of maintenance in order to sustain availability. Related to that, virtualisation is a means to effectively convert interfaces in order to hide the complexity of the underlying interface, thus simplifying usage of the corresponding resources.
Measurement	Can services be more easily exposed via the messaging infrastructure? Can dynamic services be exposed? Can service interfaces be converted? Can information be selectively made visible, respectively hidden?
Assessment	The gateway infrastructure cannot expose services that do not provide means for interaction and the current reference implementation focuses still mainly on exposing web services in an enhanced fashion. However, principally it is possible to convert messaging between different protocols, thus e.g. executing applications upon incoming web service invocations etc. This proceeding extends the service provisioning scope respectively. It has to be noted that this is similar to the capabilities of exposing an aggregated workflow. The specific capability of the messaging infrastructure however relies on dynamic service provisioning in a “virtually stable” fashion. Besides the encapsulated location of a specific service together with the extension capabilities of the virtualisation gateway approach allows for interface conversion as well by applying additional policy filters. This procedure allows in particular for selective, individualized exposure of capabilities. Notably, the current messaging infrastructure allows that both location of the endpoint, as well as service logic changes on the fly (i.e. during VO enactment) without affecting the customer and / or requiring that the service needs to be exposed anew.

Table 5 – Evaluation – Interoperability

Interoperability	
Description	Service provisioning in general is a major source for interoperability issues, as the interface and the messaging protocol differ between services and, more importantly, between frameworks. Interoperability issues can principally be decreased through virtualisation techniques.
Measurement	Scope of specifications supported by the messaging infrastructure Mapping of protocols and / or interfaces
Assessment	The gateway infrastructure is principally protocol and specification agnostic, even though it is (currently) restricted in its implementation to web service transactions and hence to the HTTP protocol. Since the gateway does not necessarily interact with either message header or body, any type of messages can be passed through it – however, to actually employ the policy filters of the messaging channel, the message structure should follow the recommendations of the relevant filters. In particular the Service Instance Registry (SIR, cf. section 4.3), and hence the routing capability of the virtualisation gateway reference realisation requires that the WS-Addressing specification is followed. In addition, the gateway can host enhanced filters that convert between different protocols, e.g. to ensure that other filters or the service receive the information in their respective format. As such, the filter can e.g. convert between the WSRF and the WS-Transfer specification (with some information loss), so that the interoperability scope of the service is extended.

Table 6 – Evaluation - Time-Constraints

Time-Constraints	
Description	Real time constraints in particular apply for the execution of the overall distributed environment and the individual service logic, but less so for messaging itself. Nonetheless, timing plays an important role in message handling, as latency can lead to unwanted behaviour in time-critical situations. Latency of the SOAP protocol is already high and should not be extended by the messaging infrastructure.
Measurement	Latency
Assessment	See corresponding measurements in section 5.2.2.

Table 7 – Evaluation - Collaboration Support

Collaboration Support	
Description	Supporting synchronous collaboration between human resources reflects the requirement to collaboratively take advantage of different fields of expertise, whereby abstracting from the respective geographical and technological context. In particular when constructing complex products or facing difficult, even potentially perilous issues requiring the synchronous interaction between different experts, the introduced DSM extension of the virtualisation gateway provides means to a tool supported, synchronous, dynamically adjustable collaboration and exchange between human resources.
Measurement	In order to measure the improvements of the DSM extension of the virtualisation gateway, the DSM extension has been applied to the following workspaces. These workspaces have been developed and evaluated in cooperation with industrial partners within the CoSpaces project [35]: • *Distributed Design Workspace* empowers ad-hoc and scheduled collaboration between distributed, multi-functional design teams. Collaboration through fixed installations and mobile environments has been considered in this workspace. • *Co-located Design Workspace* addresses how ad-hoc and planned meetings between co-located, multi-functional design teams can be supported, providing enhanced immersion, visualisation, interaction, mobility and flexibility. • *Mobile Service Workspace* addresses challenges in supporting mobile site workers to collaborate with remote experts during the realisation or the support phase.
Assessment	The DSM has been successfully integrated in all of the mentioned workspaces [94][95][98]. The interested reader is referred to the CoSpaces Website [35] for additional information of the mentioned workspaces.

5.2.2 Quantitative Evaluation

Since the virtualisation gateway infrastructure principally acts like an enhancement to other servlet containers, such as Tomcat, Apache web server and the Internet Information Service (IIS), it should not limit the capabilities of such containers drastically. Notably, not all services will always be exposed with the full virtualisation gateway enhancements and in some service provisioning cases the “normal” provisioning via e.g. Tomcat is sufficient for the provider’s needs, so that the virtualisation gateway does not need to fulfil the same industrial performance requirements. In addition, it cannot exceed (improve) the performance of these standards, as it is effectively working in close relationship with the IIS and thus cannot show better performance. With the additional processing overhead of the virtualisation gateway infrastructure (address resolution, routing, policy chain etc.), a performance reduction of factor 10 is to be expected and considered acceptable given the typical usage scenarios that require complex message policy enforcement.

In addition, it should be noted that the performance of the messaging infrastructure obviously directly depends upon the registered policies and their complexity, as well as the system the gateway is running on (bandwidth and performance limitations). With this respect all measurements should and will be restricted to plain local (i.e. on board) message passing and parsing, without accounting for bandwidth restrictions. Also, in order to provide meaningful results, delays by policy processing will not be accounted for (as every user can register his / her own policies and hence mechanisms), and the only active policy will be related to the message routing functionality (Service Instance Registry), as it is essential for the correct functioning of the virtualisation gateway.

Thus, the two main criteria of interest for measuring the performance of the gateway system are listed in Table 8.

Table 8 - Quantitative Evaluation Criteria

Criteria	**Metric for measurement**	**Notes**
#1 Scalability	General: relationship between the number of concurrent services and processing speed	The gateway should not perform much lower than the Internet Information Service (IIS6.0). Expected factor: 1:10
	#1a) amount of services that can be hosted via the gateway	IIS6.0: 20.000 static-content sites
	#1b) amount of concurrent messages that can be handled	IIS6.0: 25.000 requests per second. Taking into account the test system (cf. section Evaluation Results) 2400 requests per second has been considered.
	#1c) amount of policy handlers registered in the gateway	There is no comparable product. At least 10.
#2 Latency	General: related to scalability, but focusing on the messaging overhead	Reference product is again the IIS version 6.0. Expected factor: 1:10
	#2a) Time to process a message with a simple policy chain	IIS6.0: 25.000 requests per second. Taking into account the test system (cf. section Evaluation Results) 2400 requests per

		second have been considered.
	#2b) Time per policy handler (pure passing, without processing)	Similar to IIS: roughly 25.000 requests per second. Taking into account the test system (cf. section Evaluation Results) 2400 requests per second have been considered.

Evaluation Results

All tests were executed on Intel Quad Core 2.66 GHz machines with 8 GB RAM. Figures of the IIS (as above) have been provided directly by Microsoft [124] and are thus expected to be slightly higher than what can be achieved by the proposed system.

#1 Scalability

As noted, scalability directly depends on the system's performance (bandwidth, memory restrictions etc.).In the reference statistics from Microsoft no information about the used system was provided, so it has to be assumed that some system tuning has been done before publishing these values. Since the environment has to deal with messaging overhead in the gateway, a scalability to be roughly factor 1/10 of the pure IIS is expected. It has to be noted that the gateway has been evaluated "as is" without applying specific tuning mechanisms in advance.

#1a) Amount of concurrent registered services that can be exposed via the gateway:

In the applied tests it has been figured out that the number of services being exposed is just limited by the amount of memory of the hosting machine of the SIR. Regarding the performance of the message throughput via the gateway the messaging overhead of the Gateway/SIR interaction is the dominant delaying part here. As the test results show the throughput levelled off at around 670 invocations per second with a variance off about 5% on the used test system without being significantly affected by the number of registered services. The inspected variance is due to the background and scheduling processes of the operating system.

#1b) Amount of concurrent messages that can be handled by the messaging infrastructure:

In order to have a reasonable comparison with the performance of the IIS, a simple test service has been developed receiving and returning a given string. This service is invoked via a web service interface. For this performance measurement, the service has been invoked directly as well as via the gateway in order to determine the overhead of the messaging infrastructure. These tests have shown that a direct invocation of the service without interconnected virtualisation gateway infrastructure allows for about 2400 invocations / second. This value has been taken as reference value for the virtualisation gateway evaluation, since the IIS benchmark provided by Microsoft does not indicate on which kind of system this benchmark has been performed.

The invocation of the service with interconnected gateway has shown a maximum throughput of approximately 700 invocations / second, which is about 30% of the "pure" service invocation. This is due to the fact that for the routing via the messaging infrastructure three additional web service requests have to be processed:

1) The SIR has to be invoked in order to resolve the virtual endpoint

2) The transformed message is forwarded to the corresponding service endpoint

3) The response is routed back to the invoking client.

Future realisations of the virtualisation gateway approach could tackle this issue by realising a local cache avoiding the need to query the SIR for every request. The prototype realisation of the virtualisation gateway approach in this thesis has been used to evaluate and validate the introduced concept. To this end, the author did not investigate in specific technology specific optimisations of the reference realisation.

#1c) Amount of policy handlers that can be registered in the gateway without affecting #1a or #1b

The virtualisation gateway allows registering any kind of policy handler, thus the complexity of the corresponding policy handler has a significant impact on the payload of the system directly impacting the entire message throughput. Therefore, in order to ensure a reasonable measurement, empty policy handlers have been registered to the Gateway. This evaluation showed that the number of registered policy handlers does not significantly impact the performance of the virtualisation gateway. The dominant part here is again the messaging overhead between the gateway and the SIR.

#2 Latency

Besides for the obvious impact of system performance, the processing duration of the policy handlers, as well as the bandwidth have an impact on latency. Therefore “empty” policy handlers (except for the SIR) and local invocations for measurement are assumed.

#2a) Time to process a message with a simple policy chain

The delay caused by the additional message passing was expected to diminish the IIS factor by roughly 1/10. As the results show (cf. *#1b*) this factor is currently at 1/3 compared to the direct invocation of a service.

#2b) Time per policy handler (pure passing, without processing)

Effectively, message passing via a policy handler is identical to pure message handling with the factor 2, as handling the result is part of the overall message execution, i.e. the incoming as well as the returning message have to be processed. On the mentioned evaluation system up to 2.500 policy handlers could be handled per second. Notably, as policy handlers are executed in sequence, and not in parallel, the measurement directly depends on the amount of concurrent gateway instances, i.e. the number of messages that can be handled in parallel (#1b). As shown in *#1c* the amount of registered policy handlers does not significantly impact the entire message throughput of the messaging infrastructure.

Assessment

Primary objective towards message handling was related to increase ease of use and enhance the ways that messages can be handled with. With the specific requirement of using the respective infrastructure in business scenario contexts, it has been assumed to reach a performance that is not too far away from the capabilities of the Internet Information Service.

As the results show, 1/3 of the corresponding message throughput can be reached with the current prototype realisation of the virtualisation gateway infrastructure without taking into account complex policy handlers.

The achieved benefits of the virtualisation approach have to be confronted with the additional overhead of the reference realisation. It has to be noted that the reference realisation has been used to evaluate the concept, and thus has not been optimised from a technology perspective to achieve better quantitative results.

It has been shown that most of this messaging overhead is due to the additional invocation of the SIR for every invocation being routed via the gateway. In order to minimize this overhead future releases of the messaging infrastructure have to particularly consider optimising the integration of the SIR. As mentioned before, the introduction of caching mechanisms reducing the amount of required web service invocations will increase the entire performance of the reference realisation significantly. Furthermore, the close integration of the SIR within the virtualisation gateway realisation will obviously increase the quantitative performance as well. I.e. the SIR is no longer a stand-alone component interacting with the virtualisation gateway realisation via web service invocations, but is integrated as an assembly directly in the gateway.

Since future iterations tend to extend on semantic matching capabilities and improved error handling for further improvement on ease of use, an additional reduction of performance is to be expected, as this increases the messaging overhead correspondingly. Specific measurements and respective caching mechanisms have to be taken into account to ensure that policy handlers only become active when needed and result in as little additional overhead as possible. This procedure ensures that the average performance does not drop significantly below current measurements.

Finally, the benefits of the virtualisation gateway clearly justify and compensate this additional overhead. In particular it has to be considered that the virtualisation gateway approach allows for the integration of complex IT services and environments. These services are typically long-running ones and thus relativise the additional overhead, which typically occurs only once during the service invocation.

6. Conclusions and outlook

Dynamic and flexible markets require dynamic and flexible market players. In this context, more and more complex knowledge and services are available which every market player has to cope with.

The management and development of these services requires both profound technical understanding and respective domain specific knowledge. In particular, in scenarios with continuous increasing complexity, requirements and services, a decoupling of process knowledge and the corresponding technical realisation is of essential importance, because otherwise there is no way to cope with the increasing complexity. Furthermore, this increasing complexity requires also for specialisation of individuals and service providers to tackle specific issues in the best suitable way, and thus allowing for building-up specialised knowledge, and be able to easily provide and contribute with this knowledge in complex environments in a SOA like fashion.

6.1 Conclusions

The developed concept in this thesis addressed the gaps of existing environments with this respect, and provided a solution to cope with the analysed and identified key requirements for such dynamic environments.

To underline the capacity of the presented approach, the concept has been evaluated in concrete usage scenarios with respective prototype realisations. It has been shown that the concept realisation of a flexible service virtualisation environment enables the orchestration of such complex environments whereby considering heterogeneous human and IT resources.

In particular, the development of an enhancement of the introduced service virtualisation concept in order to integrate human resources in a SOA like fashion enables to tackle one of the essential requirements, namely to allow for the synchronous collaboration of humans in a given context, whereas avoiding the burden of considering specific technologies.

This innovation enables process designers to describe complex B2B scenarios in common process description languages, whereas being able to abstract from the underlying service and resource infrastructure and supporting the definition of required interactions between human resources / experts in a transparent way. In particular the latter one allows clarifying *which roles* have to communicate *when* with each other, whereby providing the required data and collaboration tooling setup in time and in a transparent fashion.

Finally, a prototype realisation of the proposed environment and collaboration schema has been provided and applied to relevant application scenarios. In particular, the realised concept has been applied and evaluated in demonstrators within the EU research projects BREIN, CoSpaces and GAMES.

6.2 Outlook

It has been shown that the introduced concept can cope with the increasing demands of flexible and dynamic B2B environment. However, the introduced concept and prototype reference implementation is not limited to the issues and requirements identified in this thesis. In the following, two aspects are highlighted allowing for an additional enhancement of the introduced concept.

6.2.1 Increased flexibility

The introduced virtualisation gateway environment provides an infrastructure to cope with a variety of different requirements and dynamically changing conditions. The flexible enhancement capabilities by adding additional message filters to the virtualisation gateway environment enables to address also further, concrete optimisation issues in a transparent way.

For example it would be sensible to investigate an additional enhancement of the virtualisation gateway environment enabling it to cope with errors occurring during service execution in a transparent fashion. Therefore, new rules have to be investigated coping with occurred errors. The virtualisation gateway environment can be enhanced with these specific rules to handle errors in service execution, e.g. from a service provider perspective to automatically resend a failed service invocation to a similar service providing the same results, whereas ensuring the respective message and data transformations are done in a transparent way.

In addition, the gateway environment can be enhanced in order to get coupled with an extended SLA management environment, as depicted in [188]. This enables the virtualisation gateway environment to change the routing preferences according to the best suitable service available with respect to the agreed SLAs. This would allow, e.g. in case of changing business rules to invoke the best suitable services following the respective business targets and SLAs in a transparent way. Another trend heads towards the integration of social network technologies and mechanisms in order to determine the best suitable service for a specific user [115]. The virtualisation gateway environment allows in this context to transparently integrate the respective mechanisms.

Another interesting and challenging aspect is the interrelation of the virtualisation gateway approach with the cloud computing paradigm, in particular with respect to the integration of human resources, which is currently not addressed at all. The Software as a Service (SaaS) paradigm already allows for the usage of standard applications, hosted on servers within the cloud via thin clients, typically a web browser. In this cloud service provisioning model it would be sensible to investigate further in applying the double-blind virtualisation approach (cf. section 4.1.3), allowing service consumers to transparently integrate this kind of services in their applications and processes as well. Currently, SaaS foresees that the service consumers have to access the service directly, e.g. by following a specific URL.

Combining the Platform as a service (PaaS) paradigm with the introduced virtualisation gateway approach would make the respective environments more flexible, in particular with respect to the integration of external services and libraries. Currently, service developers have to rely on the provided libraries by the corresponding service provider. A first attempt in combining the virtualisation gateway approach with the Infrastructure as a service (IaaS) paradigm has been undertaken in [90] by enabling an IaaS environment to benefit from the context information of the virtualisation gateway approach in order to optimise the setup and deployment of virtual machines, whereby ensuring the required flexibility.

Taking into account the increasing importance of the Internet of Things (IoT) paradigm, it should be investigated in how far services and devices can be integrated in complex collaboration scenarios. The introduced virtualisation gateway approach enables also the integration of any kind of service and device, and thus provides an extensible infrastructure for such kind of scenarios as well.

Furthermore, there is a trend to usage of mobile devices [167]. In order to allow a smooth integration of this kind of devices as well, the respective services have to adapt to this kind of devices as well. The introduced virtualisation gateway environment can be enhanced in order to cope with these devices and specific requirements of these user types as well.

Further work is also required in investigating in the automation of message and data transformation. The current prototype realisation requires a manual adaptation and implementation of the respective rules.

6.2.2 Energy efficiency labels for IT services

As it is already a common practice to label electronic devices used every day with the respective energy efficiency level[9], a corresponding labelling of IT service is also thinkable. Therefore a detailed knowledge about the energy consumption of applications and services in a given context is of essential importance. In particular the execution context of an application directly impacts the respective energy consumption, e.g. with respect to execution time and amount of used resources.

Assuming that such an energy-efficiency labelling is introduced, service providers selling services with a respective energy-efficiency label have to ensure to follow the corresponding rules and thresholds. In particular, changes in the setup and configuration of services can change the energy-efficiency significantly, and thus the announced labelled energy-efficiency class would no longer be valid. Therefore, an encapsulation of the underlying infrastructure is of essential importance. This encapsulation can be coped with the introduced virtualisation gateway environment, enabling to provide respective services with a defined energy-efficiency class to customers, not forcing them to take into account the corresponding technical details of the service, whereas being ensured that the announced energy efficiency criteria are met. This is in particular of significant importance in case of integrated third-party services. For example, in order to provide a service with a specific energy-efficiency class, all involved services are not allowed to be labelled with a worse energy-efficiency class as the service being sold.

It is also thinkable that this approach can be combined with an enhanced SLA management framework as depicted in section 6.2.1, allowing specifying concrete energy-efficiency restrictions to be followed and thus enabling the virtualisation

[9] http://ec.europa.eu/energy/efficiency/labelling/labelling_en.htm

gateway environment to automatically select the best suitable service for a specific issue.

First steps towards this approach have been taken within the GAMES project [102] by

- defining respective metrics to monitor and analyse the energy efficiency of specific services, as well as to determine the optimisation effects when adapting the services [87] [88];
- coupling the introduced gateway environment with respective runtime system steering components [32][90];
- setting-up a respective monitoring and evaluation infrastructure [97] [92];
- focussing on the behaviour of application and services [30].

In this context, first best practices are available and are very promising [89]. In addition, a concrete application of this enhanced virtualisation gateway environment has been applied to concrete HPC scenarios [91]. The respective capabilities to provide energy aware HPC services by coupling the gateway environment with the respective energy-aware controllers are depicted in detail in [90].

Further steps towards a holistic, energy-consumption aware environment will be taken into account in the context of the CoolEmAll project [100], taking advantage of the extracted application profiles from the mentioned monitoring environment and bringing this knowledge in simulated environments to also physically optimize the IT infrastructure (e.g. by determining hot spots). These optimisations can be applied in a transparent fashion as well by encapsulating the respective services with the virtualisation gateway environment.

7. Bibliography

[1] Ackerman MS, Cranor LF, Reagle J. Privacy in e-commerce: examining user scenarios and privacy preferences. In: *1st ACM conference on Eletronic commerce*. Denver, Colorado, USA: ACM, New York, USA; 1999:1-8.

[2] Agrawal A, Amend M, Das M, et al. *WS-BPEL Extension for People*.; 2007:1-52. Available at: http://xml.coverpages.org/BPEL4People-V1-200706.pdf. Retrieved December 22, 2012.

[3] AkoGrimo - EU IST Project (IST-004293). http://www.mobilegrids.org. Retrieved December 22, 2012.

[4] Akram A, Meredith D, Allan R. Evaluation of BPEL to Scientific Workflows. *Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)*. 2006:269-274. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1630828. Retrieved December 22, 2012.

[5] Alonso G, Casati F, Kuno H, Machiraju V. *Web services: concepts, architectures and applications*. Heidelberg: Springer; 2004.

[6] Alonso G. Myths around Web Services. *IEEE Bulletin on Data Engineering*. 2002;25(4).

[7] Apache Axis. http://axis.apache.org/. Retrieved December 22, 2012.

[8] Apache Synapse. http://synapse.apache.org/. Retrieved December 22, 2012.

[9] Ardagna D, Comuzzi M, Mussi E, Pernici B, Plebani P. Paws: A framework for executing adaptive web-service processes. *IEEE SOFTWARE*. 2007;(December):39–46. Available at: http://doi.ieeecomputersociety.org/10.1109/10.1109/MS.2007.174. Retrieved December 22, 2012.

[10] Assel M, Kipp A. A Secure Infrastructure for Dynamic Collaborative Working Environments. In: Arabnia HR, ed. *Proceedings of the International Conference on Grid Computing and Applications*. Las Vegas, Nevada, USA: CSREA Press; 2007:212-216.

[11] Assel M, Wesner S, Kipp A. A security framework for dynamic collaborative working environments. *Identity in the Information Society*. 2009. Available at: http://www.springerlink.com/index/10.1007/s12394-009-0027-1. Retrieved December 22, 2012.

[12] Astrova I, Koschel A, Kruessmann T. Comparison of enterprise service buses based on their support of high availability. *Proceedings of the 2010 ACM*

Symposium on Applied Computing - SAC '10. 2010:2495. Available at: http://portal.acm.org/citation.cfm?doid=1774088.1774605. Retrieved December 22, 2012.

[13] Barry, DK. *Web Services and Service-Oriented Architecture: The Savy Manager Guide.* Morgan Kaufmann Publishers, 2003

[14] Basic Support for Collaborative Work (BSCW). http://public.bscw.de/. Retrieved December 22, 2012.

[15] Baur T, Breu R, Kálmán T, et al. An Interoperable Grid Information System for Integrated Resource Monitoring Based on Virtual Organizations. *Journal of Grid Computing*. 2009;7(3):319-333. Available at: http://www.springerlink.com/index/10.1007/s10723-009-9134-3. Retrieved December 22, 2012.

[16] Benatallah B, Casati F, Toumani F, Hamadi R. Conceptual Modeling of Web Service. *Lecture Notes in Computer Science*. 2010;2681/2010(Advanced Information Systems Engineering):449-467. Available at: http://www.springerlink.com/content/mw1c3gtcc99u7w6j/. Retrieved December 22, 2012.

[17] Benatallah B, Sheng Q, Dumas M. The Self-Serv environment for Web services composition. *IEEE Internet Computing*. 2003;7(1):40-48. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1167338. Retrieved December 22, 2012.

[18] Bieber N, Schubert L, Koller B. (2006) The Requirement for an Interoperability Service: Power-Distribution in an eBusiness World. In: Cunningham P, Cunningham M *Exploiting the Knowledge Economy: Issues, Applications, Case Studies*.Vol 3. IOS Press; 2006.

[19] Booth D, Haas H, McCabe F, et al. Web Services Architecture. 2004. Available at: http://www.w3.org/TR/ws-arch/. Retrieved December 22, 2012.

[20] Born A, Diercks J. BPM: Zukunftsfähig mit flexiblen Abläufen. Definitionsfrage. *iX*. 2005;12:110-119.

[21] Box, D., et.al. WS-Addressing. Available at http://www.w3.org/Submission/ws-addressing/. Retrieved December 22, 2012.

[22] BPMN (Business Process Model And Notation). http://www.omg.org/spec/BPMN/2.0/. Retrieved December 22, 2012.

[23] BPSS (Business Process Specification Schema). www.ebxml.org/specs/ebBPSS.pdf. Retrieved December 22, 2012.

[24] BREIN - EU IST Project (IST- 034556). http://www.gridsforbusiness.eu. Retrieved December 22, 2012.

[25] Bultan T, Su J, Fu X. Analyzing conversations of Web services. *IEEE Internet Computing*. 2006;10(1):18-25. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1580410. Retrieved December 22, 2012.

[26] Bussler C. *B2B Integration: Concepts and Architecture*. 1 ed. Berlin: Springer; 2003:400.

[27] Canfora G, Penta M Di, Lombardi P, Villani ML. Dynamic composition of web applications in human-centered processes. In: *2009 ICSE Workshop on Principles of Engineering Service Oriented Systems*. IEEE; 2009:50-57. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5068819. Retrieved December 22, 2012.

[28] Chappel DA. *Enterprise Service Bus: Theory in Practice*. 1 ed. (Hendrickson M). O'Reilly Media; 2004:352.

[29] Chase J, Gorton I, Sivaramakrishnan C, et al. Kepler + MeDICi Service-Oriented Scientific Workflow Applications. *2009 Congress on Services - I*. 2009:275-282. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5190669. Retrieved December 22, 2012.

[30] Chen D, Liu J, Kipp A, et al. Usage centric green performance indicators. *ACM SIGMETRICS Performance Evaluation Review*. 2011;39(3):92. Available at: http://dl.acm.org/citation.cfm?doid=2160803.2160868. Retrieved December 22, 2012.

[31] Christensen E, Curbera F, Meredith G, Weerawarana S. Web Services Description Language (WSDL) 1.1. 2001. Available at: http://www.w3.org/TR/wsdl. Retrieved December 22, 2012.

[32] Cioara T, Anghel I, Salomie I, et al. Energy Aware Dynamic Resource Consolidation Algorithm for Virtualized Service Centers Based on Reinforcement Learning. In: *2011 10th International Symposium on Parallel and Distributed Computing*. IEEE; 2011:163–169. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6108269. Retrieved December 22, 2012.

[33] Clement L, Hately A, von Riegen C, Rogers T. UDDI Version 3.0.2. Available at: http://uddi.org/pubs/uddi-v3.0.2-20041019.htm. Retrieved December 22, 2012.

[34] Conti M, Kumar M, Das S, Shirazi B. Quality of service issues in Internet Web services. *IEEE Transactions on Computers*. 2002;51(6):593-594. Available at:

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1009145. Retrieved December 22, 2012.

[35] CoSpaces – EU IST Project (IST-5-034245). http://www.cospaces.org. Retrieved December 22, 2012.

[36] CoSpaces Project Report D20 – Initial case study scenarios for Distributed Design Workspace, August 2010.

[37] Currle-Linde N, Adamidis P, Resch M, Bos F, Pleiss J. GriCoL: A Language for Scientific Grids. In: *2006 Second IEEE International Conference on e-Science and Grid Computing (e-Science'06)*. Amsterdam, Netherlands: IEEE; 2006:62-62. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4031035. Retrieved December 22, 2012.

[38] D'Mello DA, Ananthanarayana VS, Salian S. A Review of Dynamic Web Service Composition Techniques. In: Meghanathan N, Kaushik BK, Nagamalai D, eds. *Proceedings of the First International Conference on Computer Science and Information Technology*.Vol 133. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011:85-97. Available at: http://www.springerlink.com/index/10.1007/978-3-642-17881-8. Retrieved December 22, 2012.

[39] Dabhi VK, Prajapati HB, Doshi V, Chokshi K. Developing Enterprise Solution with Web Services Integration. *International Journal of Web Services Practices*. 2009;4(1):11-17. Available at: http://nwesp.org/ijwsp/2009/vol4i1/ijwsp2009-vol4-no1-02.pdf. Retrieved December 22, 2012.

[40] Deelman E, Blythe J, Gil Y, et al. Pegasus: Mapping Scientific Workflows onto the Grid. *Lecture Notes in Computer Science*. 2004;3165/2004:131-140.

[41] Deelman E, Chervenak A. Data Management Challenges of Data-Intensive Scientific Workflows. *2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*. 2008:687-692. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4534284. Retrieved December 22, 2012.

[42] Deelman E, Gannon D, Shields M, Taylor I. Workflows and e-Science: An overview of workflow system features and capabilities. *Future Generation Computer Systems*. 2009;25(5):528-540. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0167739X08000861. Retrieved December 22, 2012.

[43] Degenring A. Enterprise Service Bus. *JavaSPEKTRUM*. 2005;4:16-18. Available at: http://www.sigs-datacom.de/fachzeitschriften/javaspektrum/archiv/artikelansicht.html?tx_mwjournals_pi1[pointer]=0&tx_mwjournals_pi1[mode]=1&tx_mwjournals_pi1[showUid]=1612. Retrieved December 22, 2012.

[44] DeRemer F, Kron H. Programming-in-the-Large Versus Programming-in-the-Small. *IEEE Transactions on Software Engineering*. 1976;SE-2(2):80-86. Available at: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1702345. Retrieved December 22, 2012.

[45] Dimitrakos T, Golby D, Kearney P. Towards a Trust and Contract Management Framework for dynamic Virtual Organisations. In: *eAdoption and the Knowledge Econ- omy: eChallenges 2004*.; 2004. Available at: http://epubs.cclrc.ac.uk/bitstream/701/E2004-305-TRUSTCOM-OVERVIEW-FINAL%5B1%5D.pdf. Retrieved December 22, 2012.

[46] Ding Y. The semantic web: yet another hip? *Data & Knowledge Engineering*. 2002;41(2-3):205-227. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0169023X02000411. Retrieved December 22, 2012.

[47] Djordjevic I, Dimitrakos T, Romano N, Macrandal D, Ritrovato P. Dynamic security perimeters for inter-enterprise service integration. *Future Generation Computer Systems*. 2007;23(4):633-657. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0167739X06001798. Retrieved December 22, 2012.

[48] Dogac A., Laleci G., Kabak Y., Cingil I. Exploiting Web Service Semantics: Taxonomies vs. Ontologies. *IEEE Bulletin on Data Engineering*. 2002;25(4).

[49] ebXML (Eletronic Business using eXtensible Markup Language). http://www.ebxml.org/. Retrieved December 22, 2012.

[50] Erl T. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall International; 2005:760.

[51] Fensel D, van Harmelen F, Horrocks I, McGuinness D, Patel-Schneider P. OIL: an ontology infrastructure for the Semantic Web. *IEEE Intelligent Systems*. 2001;16(2):38-45. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=920598. Retrieved December 22, 2012.

[52] Fensel D. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. 2 ed. Berlin: Springer; 2003:168.

[53] Fensel D. The Web Service Modeling Framework WSMF. *Electronic Commerce Research and Applications*. 2002;1(2):113-137. Available at: http://linkinghub.elsevier.com/retrieve/pii/S1567422302000157. Retrieved December 22, 2012.

[54] Ferrer AJ, Hernández F, Tordsson J, et al. OPTIMIS: A holistic approach to cloud service provisioning. *Future Generation Computer Systems*. 2012;28(1):66–77. Available at:

http://linkinghub.elsevier.com/retrieve/pii/S0167739X1100104X. Retrieved December 22, 2012.

[55] Foster I, Kesselman C. *The Grid. Blueprint for a New Computing Infrastructure.* 2 ed. Morgan Kaufmann; 2003:748.

[56] Foster I. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *Lecture Notes i Computer Science*. 2001;2150/2001(EURO-PAR 2001 PARELLEL PROCESSING). Available at: http://www.springerlink.com/content/flddyth66lrqdk36/. Retrieved December 22, 2012.

[57] Geer D. Taking steps to secure web services. *Computer*. 2003;36(10):14-16. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1236464. Retrieved December 22, 2012.

[58] Golby D, Wilson MD, Schubert L, Geuer-Pollmann C. An Assured Environment for Collaborative Engineering Using Web Services. In: Ghodous P, Loureiro G, Dieng-Kuntz R *Leading the Web in Concurrent Engineering. Next Generation Concurrent Engineering, Proceedings of the 13th ISPE International Conference on Concurrent Engineering (ISPE CE 2006)*.Vol 12. IOS Press; 2006:111-119.

[59] Görlach K, Sonntag M, Karastoyanova D, Leymann F, Reiter M. Conventional Simulation Workflow Technology for Scientific Conventional Workflow Technology for Scientific Simulation. In: Yang, Xiaoyu; Wang, Lizhe; Jie W, ed. *Guide to e-Science*.; 2011:0-31.

[60] Gosain S, Malhotra A, El Sawy OA. Coordinating for Flexibility in e-Business Supply Chains. *Journal of Management Information Systems*. 2004;21(3):7-45.

[61] Graf S, Hommel W. Organisationsübergreifendes Management von Föderations-Sicherheitsmetadaten auf Basis einer Service-Bus-Architektur. In: *Informationsmanagement in Hochschulen*. 4 ed.; 2010:233-246. Available at: http://dx.doi.org/10.1007/978-3-642-04720-6_20. Retrieved December 22, 2012.

[62] Grant D, Hall R, Wailes N, Wright C. The false promise of technological determinism: the case of enterprise resource planning systems. *New Technology, Work and Employment*. 2006;21(1):2–15. Available at: http://doi.wiley.com/10.1111/j.1468-005X.2006.00159.x. Retrieved December 22, 2012.

[63] Gravano L, Papakonstantinou Y. Mediating and Metasearching on the Internet. *IEEE Bulletin on Data Engineering*. 1998;21(2):28-36.

[64] Grossmann G, Schrefl M, Stumptner M. Conceptual modeling approaches for dynamic web service composition. In: *Proceedings of the Seventh Asia-Pacific*

Conference on Conceptual Modelling (APCCM 2010); 2010:10. Available at: http://portal.acm.org/citation.cfm?id=1862338. Retrieved December 22, 2012.

[65] Gudgin M, Hadley M, Mendelsohn N, et al. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). 2007. Available at: http://www.w3.org/TR/soap12-part1/. Retrieved December 22, 2012.

[66] Haller J, Schubert L, Wesner S. Private Business Infrastructures in a VO Environment. In: Cunningham P, Cunningham M *Exploiting the Knowledge Economy: Issues, Applications, Case Studies*.Vol 3. IOS Press; 2006:1064-1071.

[67] Handley M, Jacobson V, Perkins C. SDP: Session Description Protocol. *RFC 4566.* Available at: http://tools.ietf.org/html/rfc4566. Retrieved December 22, 2012.

[68] Hiesinger C, Fischer D, Stefan F, Herrmann K, Rothermel K. Minimizing Human Interaction Time in Workflows. In: *Proceedings of the Sixth International Conference on Internet and Web Applications and Services (ICIW 2011)*. St. Maarten, Netherlands; 2011:22-28. Available at: http://www.thinkmind.org/index.php?view=article&articleid=iciw_2011_1_40_20083. Retrieved December 22, 2012.

[69] Hollister SJ, Brennan JM, Kikuchi N. A homogenization sampling procedure for calculating trabecular bone effective stiffness and tissue level stress. *Journal of Biomechanics*. 1994;27(4):433–444. Available at: http://linkinghub.elsevier.com/retrieve/pii/0021929094900191. Retrieved December 22, 2012.

[70] Holmes T, Tran H, Zdun U, Dustdar S. Modeling Human Aspects of Business Processes – A View-Based, Model-Driven Approach. *Model Driven Architecture – Foundations and Applications*. 2010;5095/2010:246-261. Available at: http://dx.doi.org/10.1007/978-3-540-69100-6_17. Retrieved December 22, 2012.

[71] Huhns M, Singh M. Service-oriented computing: key concepts and principles. *IEEE Internet Computing*. 2005;9(1):75-81. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1407782. Retrieved December 22, 2012.

[72] Hull R, Su J. Tools for composite web services. *ACM SIGMOD Record*. 2005;34(2):86. Available at: http://portal.acm.org/citation.cfm?doid=1083784.1083807. Retrieved December 22, 2012.

[73] Irani Z, Themistocleous M, Love PED. The impact of enterprise application integration on information system lifecycles. *Information & Management*. 2003;41(2):177–187. Available at:

http://linkinghub.elsevier.com/retrieve/pii/S0378720603000466. Retrieved December 22, 2012.

[74] ITIL website. Available at http://www.itil-officialsite.com/home/home.aspx. Retrieved December 22, 2012.

[75] Iwasa K. Web Services Reliable Messaging TC WS-Reliability 1.1. *OASIS*. 2004:72. Available at: http://docs.oasis-open.org/wsrm/ws-reliability/v1.1/wsrm-ws_reliability-1.1-spec-os.pdf. Retrieved December 22, 2012.

[76] Jähnert J, Mandic P, Cuevas A, et al. A prototype and demonstrator of Akogrimo's architecture: An approach of merging grids, SOA, and the mobile Internet. *Computer Communications*. 2010;33(11):1304-1317. Available at: http://linkinghub.elsevier.com/retrieve/pii/S014036641000112X. Retrieved December 22, 2012.

[77] Jeckle, Mario. Collection of definition attempts for "Web Services". http://www.jeckle.de/webServices/#def. Retrieved December 22, 2012.

[78] Ji-chen J, Ming G. Enterprise Service Bus and an Open Source Implementation. *2006 International Conference on Management Science and Engineering*. 2006:926-930. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4105027. Retrieved December 22, 2012.

[79] Juric MB, Sasa A, Brumen B, Rozman I. WSDL and UDDI extensions for version support in web services. *Journal of Systems and Software*. 2009;82(8):1326-1343. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0164121209000478. Retrieved December 22, 2012.

[80] Kaminski P, Müller H, Litoiu M. A design for adaptive web service evolution. *Proceedings of the 2006 international workshop on Self-adaptation and self-managing systems - SEAMS '06*. 2006:86. Available at: http://portal.acm.org/citation.cfm?doid=1137677.1137694. Retrieved December 22, 2012.

[81] Karaenke P, Schuele M, Micsik A, Kipp A. Inter-organizational Interoperability through integration of Multiagent, Web Service, and Semantic Web Technologies. In: Fischer K, Müller JP, Levy R, eds. *Agent-based technologies and applications for enterprise interoperability*. 98th ed. Springer; 2012:55–75. Available at: http://rd.springer.com/chapter/10.1007/978-3-642-28563-9_4. Retrieved December 22, 2012.

[82] Keller G, Nüttgens M, Scheer A-W. Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)." *Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), Universität des Saarlandes, Heft 89*. 1992:32.

[83] Keller R. Analyse und Optimierung der Softwareschichten von wissenschaftlichen Anwendungen für Metacomputing. *Dissertation.* 2008;:144. Available at: http://elib.uni-stuttgart.de/opus/volltexte/2008/3826/. Retrieved December 22, 2012.

[84] Kenney L, Plummer D. *Magic Quadrant for Integrated SOA Governance Technology Sets, 2007*. 2007:18. Available at: http://www.expertise.com.br/eventos/cafe/gartner_governanca.pdf. Retrieved December 22, 2012.

[85] Khalaf R. *Note on Syntactic Details of Split BPEL-D Business Processes Institut für Architektur von Anwendungssystemen*. Stuttgart; 2007:13. Available at: http://elib.uni-stuttgart.de/opus/volltexte/2007/3230/index.html. Retrieved December 22, 2012.

[86] Kim MJ, Rodriguez JL, Woods JM, Yee ETC. Virtual Web Service. 2009:12. Available at: http://www.google.de/patents?id=B3PIAAAAEBAJ&printsec=abstract&zoom=4&source=gbs_overview_r&cad=0#v=onepage&q&f=false. Retrieved December 22, 2012.

[87] Kipp A, Jiang T, Fugini M, Salomie I. Layered Green Performance Indicators. *Future Generation Computer Systems*. 2011. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0167739X11000860. Retrieved December 22, 2012.

[88] Kipp A, Jiang T, Fugini M. Green Metrics for energy-aware IT systems. In: *Proceedings of the Fifth International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2011)*. Seoul; 2011:241-248. Available at: http://dx.doi.org/10.1109/CISIS.2011.42. Retrieved December 22, 2012.

[89] Kipp A, Jiang T, Liu J, et al. Applying Green Metrics to optimise the energy-consumption footprint of IT service centres. *International Journal of Space-Based and Situated Computing*. 2012;2(2):in press. Available at: http://www.inderscience.com/info/inarticle.php?artid=48897. Retrieved December 22, 2012.

[90] Kipp A, Jiang T, Liu J, et al. Energy-Aware provisioning of HPC services with virtualised web services. In: Khan SU, Kolodziej J, eds. *Evolutionary based Solutions for Green Computing*. Berlin / Heidelberg: Springer Verlag; 2012:29–51. Available at: http://www.springer.com/engineering/computational+intelligence+and+complexity/book/978-3-642-30658-7. Retrieved December 22, 2012.

[91] Kipp A, Liu J, Jiang T, et al. Approach towards an energy-aware and energy-efficient high performance computing environment. In: *IEEE 7th International Conference on Intelligent Computer Communication and Processing*. Cluj-Napoca: IEEE; 2011:493-499. Available at:

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6047921. Retrieved December 22, 2012.

[92] Kipp A, Liu J, Jiang T, et al. Testbed architecture for generic , energy-aware evaluations and optimisations. In: *Proceedings of the First International Conference on Advanced Communications and Computation (INFOCOMP 2011)*. Barcelona; 2011:103–108. Available at: http://www.thinkmind.org/index.php?view=article&articleid=infocomp_2011_5_30_10071. Retrieved December 22, 2012.

[93] Kipp A, Schneider R, Schubert L. Encapsulation of complex HPC services. In: Rueckemann C-P, ed. *Integrated Information and Computing Systems for Natural, Spatial, and Social Sciences*. IGI Global; 2012:in press. Available at: http://www.igi-global.com/book/integrated-information-computing-systems-natural/67413. Retrieved December 22, 2012.

[94] Kipp A, Schubert L, Assel M, Fernando T. Dynamism and Data Management in Distributed, Collaborative Working Environments. In: *Proceedings of the 8th International Conference on the Design of Cooperative Systems*.; 2008. Available at: http://coop.wineme.fb5.uni-siegen.de/proceedings2008/1_02_aKipp_al_16-22.pdf. Retrieved December 22, 2012.

[95] Kipp A, Schubert L, Assel M. Supporting Dynamism and Security in Ad-Hoc Collaborative Working Environments. In: *Proceedings of the 12th World Multi-Conference on Systemics, Cyvernetics and Informatics (WMSCI)*.; 2008. Available at: http://www.iiis.org/CDs2008/CD2008SCI/SCI2008/Abstract.asp?myurl=S367QZ.pdf. Retrieved December 22, 2012.

[96] Kipp A, Schubert L, Geuer-Pollmann C. Dynamic Service Encapsulation. In: *Proceedings of the First International Conference on Cloud Computing*. Munich; 2009:217-230. Available at: http://www.springerlink.com/content/t41672n1728172h4/. Retrieved December 22, 2012.

[97] Kipp A, Schubert L, Liu J, et al. Energy Consumption Optimisation in HPC Service Centres. In: *Proceedings of the Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*. Ajaccio, Corsica, France; 2011:1-16. Available at: http://www.ctresources.info/ccp/paper.html?id=6281. Retrieved December 22, 2012.

[98] Kipp A, Schubert L. eBusiness Interoperability and Collaboration. In: Kajan E, ed. *Electronic Business Interoperability: Concepts, Opportunities, and Challenges*. IGI Global; 2011:153-184. Available at: http://www.igi-global.com/bookstore/chapter.aspx?titleid=52153. Retrieved December 22, 2012.

[99] Kipp A, Wesner S, Schubert L, et al. *A new approach for classifying Grids*; 2007:13. Available at: http://www.it-tude.com/grid_classification.html. Retrieved December 22, 2012.

[100] Kipp A, Wössner U. CoolEmAll - Platform for Optimizing the Design, Operation and Cooling of modular configurable IT Infrastructures. *inSiDE (Innovatives Supercomputing in Deutschland)*. 2011;9(2):44–49. Available at: http://inside.hlrs.de/pdfs/inSiDE_autumn2011.pdf. Retrieved December 22, 2012.

[101] Kipp A. *Ablösung von WS-CDL durch BPEL und WSFL Global Model.* 2006:133. Available at http://elib.uni-stuttgart.de/opus/volltexte/2006/2896/. Retrieved December 22, 2012.

[102] Kipp A. GAMES - Green Active Management of Energy in IT Service centres. *inSiDE (Innovatives Supercomputing in Deutschland)*. 2010:40-43. Available at: http://inside.hlrs.de/pdfs/inSiDE_autumn2010.pdf. Retrieved December 22, 2012.

[103] Kopecki A. Collaborative Integration of Classic Applications in Virtual Reality Environments. *Journal of Systemics, Cybernetics, and Informatics*. 2011;9:24–29. Available at: http://www.iiisci.org/journal/CV$/sci/pdfs/CN11004.pdf. Retrieved December 22, 2012.

[104] Kuppuraju S, Kumar A, Kumari GP. Case Study to Verify the Interoperability of a Service Oriented Architecture Stack. In: *IEEE International Conference on Services Computing (SCC 2007)*. IEEE; 2007:678-679. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4278723. Retrieved December 22, 2012.

[105] Langdon C. IT systems perspective - The state of web services. *Computer*. 2003;36(7):93-94. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1212695. Retrieved December 22, 2012.

[106] Laskey K, Estefan JA, McCabe FG, Thornton D. Reference Architecture Foundation for Service Oriented Architecture Version 1.0. *OASIS*. 2009. Available at: http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-cd-02.html. Retrieved December 22, 2012.

[107] Leymann F, Unger T, Wagner S. On designing a people-oriented constraintbased workflow language. In: *Proceedings of the 2nd Central-European Workshop on Services and their Composition*. Berlin, Germany; 2010:25-31. Available at: http://ceur-ws.org/Vol-563/paper3.pdf. Retrieved December 22, 2012.

[108] Li G, Muthusamy V, Jacobsen H. A distributed service-oriented architecture for business process execution. *ACM Transactions on the Web*. 2010;4(1):1-33.

Available at: http://portal.acm.org/citation.cfm?doid=1658373.1658375. Retrieved December 22, 2012.

[109] Lindner P. Management von verteilten Anwendungen in heterogenen Grid Umgebungen. 2006:149. Available at: http://elib.uni-stuttgart.de/opus/volltexte/2007/3241/. Retrieved December 22, 2012.

[110] Lins FA, dos Santos Júnior JC, Rosa NS. Adaptive web service composition. *ACM SIGSOFT Software Engineering Notes*. 2007;32(4):6. Available at: http://portal.acm.org/citation.cfm?doid=1281421.1281428. Retrieved December 22, 2012.

[111] Ludäscher B, Altintas I, Berkley C, et al. Scientific workflow management and the Kepler system. *Concurrency and Computation: Practice and Experience*. 2006;18(10):1039-1065. Available at: http://doi.wiley.com/10.1002/cpe.994. Retrieved December 22, 2012.

[112] Ludäscher B, Weske M, McPhillips T, Bowers S. Scientific Workflows: Business as Usual? *Lecture Notes in Computer Science*. 2009;5701/2009:31-47.

[113] Ludwig U, Mertin A, Schmid B. Giftige Geschäfte. *Der Spiegel 16/2011*. 2011:45-46. Available at: http://wissen.spiegel.de/wissen/image/show.html?did=78076148&aref=image048/2011/04/16/CO-SP-2011-016-0044-0046.PDF&thumb=false. Retrieved December 22, 2012.

[114] Luftman J. Assessing It/Business Alignment. *Information Systems Management*. 2003;20(4):9-15. Available at: http://www.informaworld.com/openurl?genre=article&doi=10.1201/1078/43647.20.4.20030901/77287.2&magic=crossref||D404A21C5BB053405B1A640AFFD44AE3. Retrieved December 22, 2012.

[115] Maamar Z, Hacid H, Huhns MN. Why Web Services Need Social Networks. *IEEE Internet Computing*. 2011;15(2):90-94. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5731594. Retrieved December 22, 2012.

[116] Mahmoud QH ed. *Middleware for Communications*. Chichester, UK: John Wiley & Sons, Ltd; 2005. Available at: http://doi.wiley.com/10.1002/0470862084. Retrieved December 22, 2012.

[117] Malik Z, Bouguettaya A. Preserving Trade Secrets between Competitors in B2B Interactions. *International Journal of Cooperative Information Systems (IJCIS)*. 2005;14(2-3):265-297.

[118] Manuali C, Laganà A. GriF: A new collaborative framework for a web service approach to grid empowered calculations. *Future Generation Computer Systems*. 2011;27(3):315-318. Available at:

http://linkinghub.elsevier.com/retrieve/pii/S0167739X10001494. Retrieved December 22, 2012.

[119] Margaria T, Steffen B. Service Engineering: Linking Business and IT. *Computer*. 2006;39(10):45-55. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1707634. Retrieved December 22, 2012.

[120] Medjahed B, Benatallah B, Bouguettaya A, Ngu A, Elmagarmid A. Business-to-business interactions: issues and enabling technologies. *The VLDB Journal The International Journal on Very Large Data Bases*. 2003;12(1):59-85. Available at: http://www.springerlink.com/Index/10.1007/s00778-003-0087-z. Retrieved December 22, 2012.

[121] Medjahed B, Bouguettaya A, Elmagarmid AK. Composing Web services on the Semantic Web. *The VLDB Journal The International Journal on Very Large Data Bases*. 2003;12(4):333-351. Available at: http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1007/s00778-003-0101-5. Retrieved December 22, 2012.

[122] Melzer I. *Service-orientierte Architekturen mit Web Services: Konzepte - Standards - Praxis*. 4 ed. Spektrum Akademischer Verlag; 2010:415.

[123] Menge F. Enterprise Service Bus. In: *Free and Open Source Software Conference*.; 2007:6. Available at: http://programm.froscon.de/2007/attachments/15-falko_menge_-_enterpise_service_bus.pdf. Retrieved December 22, 2012.

[124] Microsoft Internet Information Services (IIS). http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/8962d6c9-d2ee-4bb2-8a2f-cd31858edb09.mspx?mfr=true. Retrieved December 22, 2012.

[125] Microsoft Security Token Service (STS). http://msdn.microsoft.com/en-us/library/aa480563.aspx. Retrieved December 22, 2012.

[126] Modafferi S, Mussi E, Maurino A, Pernici B. A framework for provisioning of complex e-services. In: *IEEE International Conference onServices Computing, 2004. (SCC 2004). Proceedings. 2004*. IEEE; 2004:81-90. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1357993. Retrieved December 22, 2012.

[127] Moser O, Rosenberg F, Dustdar S. Non-intrusive monitoring and service adaptation for WS-BPEL. *Proceeding of the 17th international conference on World Wide Web - WWW '08*. 2008:815. Available at: http://portal.acm.org/citation.cfm?doid=1367497.1367607. Retrieved December 22, 2012.

[128] Nadalin, A., Kaler C., Monzilo, R., Hallam.Baker, P. WS-Security. 01.02.2006. http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf. Retrieved December 22, 2012.

[129] Nash A. *Service Virtualization - Key to Managing Change in SOA*.; 2006. Available at: http://www.technologyexecutivesclub.com/Articles/webservices/virtualization.php. Retrieved December 22, 2012.

[130] Nesbit KJ, Laudon J, Smith JE. *Virtual Private Machines: A Resource Abstraction for Multi-Core Computer Systems*; 2007.

[131] Nielsen HF, Thatte S. Web Services Routing Protocol (WS-Routing). 2001. Available at: http://msdn.microsoft.com/de-de/library/ms951249.aspx. Retrieved December 22, 2012.

[132] Oinn T, Greenwood M, Addis M, et al. Taverna: lessons in creating a workflow environment for the life sciences. *Concurrency and Computation: Practice and Experience*. 2006;18(10):1067-1100. Available at: http://doi.wiley.com/10.1002/cpe.993. Retrieved December 22, 2012.

[133] Oliveira Jr. FGA de, Oliveira JMP de. QoS-based Approach for Dynamic Web Service Composition. *Journal of Universal Computer Science*. 2011;17(5):712-741. Available at: http://dx.doi.org/10.3217/jucs-017-05-0712. Retrieved December 22, 2012.

[134] Ouzzani M, Bouguettaya A. Efficient access to web services. *IEEE Internet Computing*. 2004;8(2):34-44. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1273484. Retrieved December 22, 2012.

[135] Padala P, Shin KG, Zhu X, et al. Adaptive control of virtualized resources in utility computing environments. *ACM SIGOPS Operating Systems Review*. 2007;41(3):289. Available at: http://portal.acm.org/citation.cfm?doid=1272998.1273026. Retrieved December 22, 2012.

[136] Papazoglou M, van Den Heuvel W. Web Services Management: A Survey. *IEEE Internet Computing*. 2005;9(6):58-64. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1541947. Retrieved December 22, 2012.

[137] Papazoglou MP, Dubray J. A Survey of Web Service Technologies. *Technical Report*. 2004;(June):73.

[138] Papazoglou MP, Krämer BJ, Yang J. Leveraging Web-Services and Peer-to-Peer Networks. *Lecture Notes in Computer Science*. 2010;2681/2010(Advanced Information Systems Engineering).

[139] Petrie C, Bussler C. Service agents and virtual enterprises: a survey. *IEEE Internet Computing*. 2003;7(4):68-78. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1215662. Retrieved December 22, 2012.

[140] Plummer D, McCoy D, Abrams C. *Magic Quadrant for the Integrated Service Environment Market*. 2006:13. Available at: http://www.gartner.com/id=488721. Retrieved December 22, 2012.

[141] Preuss T, Syrbe J, Konig H. *Virtual private resources. An approach for long-term binding of services*. IEEE Comput. Soc; 1997:216-226. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=628363. Retrieved December 22, 2012.

[142] Projected global revenue of service oriented architecture (SOA) from 2008 to 2020. Available at: http://www.statista.com/statistics/169303/projected-revenue-of-soa-worldwide/. Retrieved December 22, 2012.

[143] Puschmann T, Alt R. Enterprise application integration systems and architecture – the case of the Robert Bosch Group. *Journal of Enterprise Information Management*. 2004;17(2):105–116. Available at: http://www.emeraldinsight.com/10.1108/17410390410518754. Retrieved December 22, 2012.

[144] Rahm E, Bernstein P a. A survey of approaches to automatic schema matching. *The VLDB Journal*. 2001;10(4):334-350. Available at: http://www.springerlink.com/index/10.1007/s007780100057. Retrieved December 22, 2012.

[145] Rao J, Su X. A Survey of Automated Web Service Composition Methods. *Lecture Notes in Computer Science*. 2005;3387/2005:43-54. Available at: http://www.springerlink.com/content/4m6w37g0jffk9bv4/. Retrieved December 22, 2012.

[146] Rezgui A, Ouzzani M, Bouguettaya A, Medjahed B. *Preserving privacy in web services*. New York, New York, USA: ACM Press; 2002:56. Available at: http://portal.acm.org/citation.cfm?doid=584931.584944. Retrieved December 22, 2012.

[147] Rezgur A, Bouguettaya A, Eltoweissy M. Privacy on the web: facts, challenges, and solutions. *IEEE Security & Privacy Magazine*. 2003;1(6):40-49. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1253567. Retrieved December 22, 2012.

[148] Romero D, Rouvoy R, Seinturier L, et al. Enabling Context-Aware Web Services: A Middleware Approach for Ubiquitous Environments. In: *Enabling Context-Aware Web Services: Methods, Architectures, and Technologies*.;

2010:113-135. Available at: http://hal.archives-ouvertes.fr/inria-00414070/. Retrieved December 22, 2012.

[149] Rosenberg J, Schulzrinne H, Camarillo G, et al. SIP: Session Initiation Protocol. *RFC 3261.* 2002. Available at: http://tools.ietf.org/html/rfc3261. Retrieved December 22, 2012.

[150] Russel N, ter Hofstede AH. new YAWL: Towards Workflow 2.0. *Transactions on Petri Nets and Other Models of Concurrency II*. 2009;5460/2009:79-97. Available at: http://dx.doi.org/10.1007/978-3-642-00899-3_5. Retrieved December 22, 2012.

[151] Russell N, Aalst WM. Work Distribution and Resource Management in BPEL4People: Capabilities and Opportunities. *Advanced Information Systems Engineering*. 2008;5074/2008:94-108. Available at: http://dx.doi.org/10.1007/978-3-540-69534-9_7. Retrieved December 22, 2012.

[152] Schmidt M, Hutchison B, Lambros P, Phippen R. The Enterprise Service Bus: Making service-oriented architecture real. *IBM Systems Journal*. 2005;44(4):781-797. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5386706. Retrieved December 22, 2012.

[153] Schubert L, Kipp A, Koller B, Wesner S. Service-oriented operating systems: future workspaces. *IEEE Wireless Communications*. 2009;16(3):42-50. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5109463. Retrieved December 22, 2012.

[154] Schubert L, Kipp A, Wesner S. Above the Clouds: From Grids to Service-oriented Operating Systems. In: Tselentis G, Domingue J, Galis A, et al., eds. *Towards the Future Internet - A European Research Perspective*. Amsterdam: IOS Press; 2009:238 - 249. Available at: http://www.booksonline.iospress.nl/Content/View.aspx?piid=12028. Retrieved December 22, 2012.

[155] Schubert L, Kipp A, Wesner S. From Internet to Cross-Organisational Networking. In: *Collaborative Product and Service Life Cycle Management for a Sustainable World - Proceedings of the 15th ISPE International Conference on Concurrent Engineering (CE2008)*. Springer; 2008:63–75. Available at: http://www.springerlink.com/content/k70m97r68417m600/. Retrieved December 22, 2012.

[156] Schubert L, Wesner S, Dimitrakos T. (2005) Secure and Dynamic Virtual Organizations for Business. In: Cunningham P, Cunningham M *Collaborative Product and Service Life Cycle Management for a Sustainable World - Proceedings of the 15th ISPE International Conference on Concurrent Engineering (CE2008)*.Vol 2. IOS Press; 2005.

[157] Schubert L, Wesner S, Kipp A, Arenas A. Self-Managed Microkernels: From Clouds Towards Resource Fabrics. In: *In Proceedings of the First International Conference on Cloud Computing*. Munich; 2009:167-185. Available at: http://www.springerlink.com/content/xq1138613t36p077/. Retrieved December 22, 2012.

[158] Shields M, Taylor I. Programming Scientific and Distributed Workflow with Triana Services. *Concurrency and Computation: Practice & Experience - Workflow in Grid Systems*. 2006;18(10):1-10.

[159] Shvaiko P, Euzenat J. Ten Challenges for Ontology Matching. In: *7th International Conference on Ontologies, Databases, and Applications of Semantics (ODBASE)*.; 2008:17.

[160] Simon B, László Z, Goldschmidt B, Kondorosi K, Risztics P. Evaluation of WS-* Standards Based Interoperability of SOA Products for the Hungarian e-Government Infrastructure. *International Conference on the Digital Society*. 2010;0:118-123. Available at: http://doi.ieeecomputersociety.org/10.1109/ICDS.2010.28. Retrieved December 22, 2012.

[161] Singh MP, Huhns MN. *Service Oriented Computing Semantics, Processes and Agents*. New York: Whiley; 2005:588.

[162] Singh MP. Physics of service composition. *IEEE Internet Computing*. 2001;5(IEEE Internet Computing):6-7.

[163] Sonntag M, Currle-Linde N, Görlach K, Karastoyanova D. TOWARDS SIMULATION WORKFLOWS WITH BPEL : DERIVING MISSING FEATURES FROM GRICOL. In: *21st IASTED International Conference on Modelling and Simulation MS 2010*. ACTA Press; 2010.

[164] Sonntag M, Karastoyanova D, Deelman E. BPEL4Pegasus : Workflows Combining Business and BPEL4Pegasus : Combining Business. *Lecture Notes in Computer Science*. 2010;6470/2010:728-729.

[165] Sonntag M, Karastoyanova D. Next Generation Interactive Scientific Experimenting Based on the Workflow Technology. In: *21st IASTED International Conference Modelling and Simulation (MS 2010)*. ACTA Press; 2010.

[166] Sotomayor B, Keahey K, Foster I. *Overhead Matters: A Model for Virtual Resource Management*. IEEE; 2006:5-5. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4299350. Retrieved December 22, 2012.

[167] Srirama S, Vainikko E, Šor V, Jarke M. Scalable Mobile Web Services Mediation Framework. *2010 Fifth International Conference on Internet and Web Applications and Services*. 2010:315-320. Available at:

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5476722. Retrieved December 22, 2012.

[168] Stuckenholz A. Component evolution and versioning state of the art. *ACM SIGSOFT Software Engineering Notes*. 2005;30(1):7. Available at: http://portal.acm.org/citation.cfm?doid=1039174.1039197. Retrieved December 22, 2012.

[169] Szyperski C, Gruntz D, Murer S. *Component Software: Beyond Object-Oriented Programming*. Amsterdam: Addison-Wesley Longman; 2002:450.

[170] Takayama Y, Ghiglione E, Wilson S, Dalziel J. Human Activities in Distributed BPM. *Business Process, ServicesComputing and Intelligent Service Management P-147*. 2008:139-154. Available at: http://subs.emis.de/LNI/Proceedings/Proceedings147/gi-proc-147-009.pdf. Retrieved December 22, 2012.

[171] Tan W, Fong L, Bobroff N. BPEL4Job : A Fault-Handling Design for Job Flow Management. In: *Service-Oriented Computing - ISOC 2007*.; 2007:27-42. Available at: http://www.springerlink.com/content/x1053848p6765484/. Retrieved December 22, 2012.

[172] Tang X, Sun S, Yuan X, Chen D. Automated Web Service Composition System on Enterprise Service Bus. In: *2009 Third IEEE International Conference on Secure Software Integration and Reliability Improvement*. IEEE; 2009:9-13. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5325398. Retrieved December 22, 2012.

[173] Taylor IJ, Deelman E, Gannon DB, et al. *Workflows for e-Science - Scientific Workflows for Grids*. Springer; 2007.

[174] Taylor S, Laria G, Ritrovato P, Schubert L. Business Collaborations in Grids: The BREIN Architectural Principals and VO Model. In: Altmann J, Buyya R, Rana OF *Grid Economics and Business Models*.Vol 5745/2009. Lecture Notes in Computer Science. Berlin / Heidelberg: Springer; 2009:171-181.

[175] The Shibboleth Project. http://shibboleth.net/. Retrieved December 22, 2012.

[176] TrustCoM – EU IST Project (IST-2003-01945). http://www.hlrs.de/organization/av/isis/research/past-projects/trustcom/. Retrieved December 22, 2012.

[177] Tsalgatidou A, Pilioura T. An Overview of Standards and Related Technology in Web Services. *Distributed and Parallel Databases*. 2002;12:135-162.

[178] Vaculín R, Neruda R, Sycara K. Process Mediation: Requirements, Experiences and Challenges. In: *Electronic Business Interoperability: Concepts, Opportunities,*

and Challenges. IGI Global. In print. Available at: http://www.vaculin.com/downloads/Vaculin-2010-EBI-chapter-final.pdf. Retrieved December 22, 2012.

[179] Vadivelou G, Ilavarasan E. Solution to dynamic web service composition related to QoS. In: *2011 3rd International Conference on Electronics Computer Technology*. IEEE; 2011:351-355. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5942018. Retrieved December 22, 2012.

[180] Vaughan-Nichols S. Web services: beyond the hype. *Computer*. 2002;35(2):18-21. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=982908. Retrieved December 22, 2012.

[181] Vernadat F. Interoperable enterprise systems: Principles, concepts, and methods. *Annual Reviews in Control*. 2007;31(1):137-145. Available at: http://linkinghub.elsevier.com/retrieve/pii/S1367578807000132. Retrieved December 22, 2012.

[182] Vinoski S. Web services interaction models. Current practice. *IEEE Internet Computing*. 2002;6(3):89-91. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1003137. Retrieved December 22, 2012.

[183] Wassermann B, Emmerich W, Butchart B, et al. Sedna: A BPEL-Based Environment for Visual Scientific Workflow Modeling. In: *Workflows for E-Science, Part III*.; 2007:428-449.

[184] WCF – Windows Communication Foundation. http://msdn.microsoft.com/wcf/. Retrieved December 22, 2012.

[185] Weerawarana S, Curbera F, Leymann F, Ferguson DF. *Web services platform architecture : SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more*. Upper Saddle River, NJ ; Munich [u.a.]: Prentice Hall PTR; 2005:416.

[186] Wesner S, Gallizo GM, Kipp A, Assel M. Realizing context aware collaborations using Grids. In: *Communications in Computer and Information Science*. Springer Verlag; 2009:84-88. Available at: http://www.springerlink.com/content/m743j383741q0384/. Retrieved December 22, 2012.

[187] Wesner S, Schubert L, Dimitrakos T, et al. Towards a platform enabling Grid based Application Service Provision. In: Cunningham P, Cunningham M *Exploiting the Knowledge Economy: Issues, Applications, Case Studies*.Vol 1. IOS Press; 2004.

[188] Wesner S. Integrated Management Framework for Dynamic Virtual Organisations. 2008:151. Available at: http://elib.uni-stuttgart.de/opus/volltexte/2009/3868/. Retrieved December 22, 2012.

[189] Wesner S. Towards an Architecture for the Mobile Grid (Architektur für ein Mobiles Grid). *it - Information Technology*. 2005;47(6_2005):336-342. Available at: http://www.oldenbourg-link.com/doi/abs/10.1524/itit.2005.47.6.336. Retrieved December 22, 2012.

[190] Wesner, S., Schubert, L, Dimitrakos, T. Dynamic Virtual Organisations in Engineering. *2nd Russian-German Advanced Research Workshop on Computational Science and High Performance Computing*, March 14 - 16, 2005

[191] Wieland M, Gorlach K, Schumm D, Leymann F. Towards Reference Passing in Web Service and Workflow-Based Applications. *2009 IEEE International Enterprise Distributed Object Computing Conference*. 2009:109-118. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5277706. Retrieved December 22, 2012.

[192] Winter R, Schelp J. Enterprise Architecture Governance : The Need for a Business-to-IT Approach. In: *2008 ACM symposium on Applied computing*.; 2008:548-552. Available at: http://portal.acm.org/citation.cfm?id=1363820. Retrieved December 22, 2012.

[193] WS-Agreement Specification. http://www.ogf.org/documents/GFD.107.pdf. Retrieved December 22, 2012.

[194] WS-BPEL - Business Process Execution Language. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel. Retrieved December 22, 2012.

[195] WS-CDL (Web Service Choreography Description Language). http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/. Retrieved December 22, 2012.

[196] WS-Coordination. http://docs.oasis-open.org/ws-tx/wstx-wscoor-1.2-spec-os/wstx-wscoor-1.2-spec-os.html. Retrieved December 22, 2012.

[197] WS-Policy specification. http://www.w3.org/TR/ws-policy/. Retrieved December 22, 2012.

[198] Yu Q, Liu X, Bouguettaya A, Medjahed B. Deploying and managing Web services: issues, solutions, and directions. *The VLDB Journal*. 2006;17(3):537-572. Available at: http://www.springerlink.com/index/10.1007/s00778-006-0020-3. Retrieved December 22, 2012.

[199] Zhao X, Qiu Z, Cai C, Yang H. A Formal Model of Human Workflow. *2008 IEEE International Conference on Web Services*. 2008;(60603033):195-202. Available

at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4670176. Retrieved December 22, 2012.

Annex

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema              xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:complexType name="parameter">
  <xs:sequence>
   <xs:element  name="parameter"  type="xs:string"  minOccurs="1"
maxOccurs="1"/>
  </xs:sequence>
  <xs:attribute name="paramterType" type="xs:string"/>
 </xs:complexType>
 <!--command-->
 <xs:complexType name="command">
  <xs:sequence>
   <xs:sequence>
    <xs:element      name="commandParameter"      type="parameter"
maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:element name="description" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"/>
 </xs:complexType>
 <xs:complexType name="commands">
  <xs:sequence>
   <xs:element            name="command"            type="command"
maxOccurs="unbounded"/>
  </xs:sequence>
 </xs:complexType>
 <!-- data -->
 <xs:complexType name="initialData">
  <xs:sequence>
   <xs:element           name="DataSet"           type="xs:anyURI"
maxOccurs="unbounded"/>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="datasource">
  <xs:sequence>
   <xs:choice>
    <xs:element name="file"/>
    <xs:element name="http"/>
    <xs:element name="https"/>
    <xs:element name="ftp"/>
   </xs:choice>
  </xs:sequence>
  <xs:attribute name="name"/>
```

```
 </xs:complexType>
 <xs:complexType name="datasources">
  <xs:sequence>
   <xs:element         name="datasource"         type="datasource"
maxOccurs="unbounded"/>
  </xs:sequence>
 </xs:complexType>
 <xs:complexType name="dataOperation">
  <xs:attribute name="command"/>
  <xs:attribute name="datasource"/>
 </xs:complexType>
 <xs:complexType name="data">
  <xs:sequence>
   <xs:element name="load" type="dataOperation"/>
   <xs:element name="save" type="dataOperation"/>
  </xs:sequence>
 </xs:complexType>
 <!-- environment -->
 <xs:complexType name="environment">
  <xs:sequence>
   <xs:choice>
    <xs:element name="desktop"/>
    <xs:element name="mobile"/>
   </xs:choice>
  </xs:sequence>
 </xs:complexType>
 <!-- desktop -->
 <xs:complexType name="desktop">
  <xs:sequence>
   <xs:choice>
    <xs:element name="monitor"/>
    <xs:element name="_3d"/>
   </xs:choice>
  </xs:sequence>
 </xs:complexType>
 <!-- supported formats -->
 <xs:complexType name="format">
  <xs:attribute name="name" type="xs:string"/>
  <xs:attribute name="type" type="xs:string"/>
 </xs:complexType>
 <xs:complexType name="formats">
  <xs:sequence>
   <xs:element            name="format"            type="format"
maxOccurs="unbounded"/>
  </xs:sequence>
 </xs:complexType>
 <!-- capabilities -->
 <xs:complexType name="capabilities">
```

```
  <xs:sequence>
   <xs:element name="description" type="xs:string"/>
   <xs:element name="name" type="xs:string"/>
   <xs:element name="id" type="xs:string"/>
   <xs:element name="commands" type="commands"/>
   <xs:element name="data" type="data"/>
   <xs:element name="datasources" type="datasources"/>
   <xs:element name="initialData" type="initialData"/>
   <xs:element name="formats" type="formats"/>
  </xs:sequence>
  <xs:attribute name="applicationName" type="xs:string"/>
  <xs:attribute name="applicationID" type="xs:string"/>
 </xs:complexType>
 <!-- collaborationSetup-->
 <xs:complexType name="collaborationSetup">
  <xs:sequence>
   <xs:element      name="configuration"      type="configuration"
maxOccurs="unbounded"/>
  </xs:sequence>
 </xs:complexType>
 <!-- configuration -->
 <xs:complexType name="configuration">
  <xs:sequence>
   <xs:element name="gateway" type="xs:anyURI"/>
   <xs:element name="applicationController">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="applicationController" type="xs:anyURI"/>
     </xs:sequence>
     <xs:attribute name="id" type="xs:string"/>
    </xs:complexType>
   </xs:element>
   <xs:element name="application" type="capabilities"/>
   <xs:element name="security">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="handle" type="xs:base64Binary"/>
      <xs:element name="IDP-Address" type="xs:string"/>
      <xs:element name="partners">
       <xs:complexType>
        <xs:sequence>
         <xs:element   name="collaborationPartner"   minOccurs="0"
maxOccurs="unbounded">
          <xs:complexType>
           <xs:sequence>
            <xs:element name="Gateway" type="xs:anyURI"/>
           </xs:sequence>
           <xs:attribute name="id" type="xs:string"/>
```

```
          </xs:complexType>
         </xs:element>
        </xs:sequence>
       </xs:complexType>
      </xs:element>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
  </xs:sequence>
 </xs:complexType>
</xs:schema>
```