

A Software Architecture for Grid Utilisation in Business Workflows

S. Gudenkauf¹, W. Hasselbring¹, F. Heine², A. Höing², O. Kao², G. Scherp¹

¹R&D-Division Business Information Management
OFFIS Institute for Information Technology
Escherweg 2
D-26121 Oldenburg
stefan.gudenkauf@offis.de
hasselbring@offis.de
guido.scherp@offis.de

²Dept. of Telecommunication Systems, Complex and Distributed IT Systems
Technische Universität Berlin
Einsteinufer 17
D-10587 Berlin
felix.heine@tu-berlin.de
andre.hoeing@tu-berlin.de
odej.kao@tu-berlin.de

Abstract: Within an enterprise various information systems have to be run. Enterprise Application Integration (EAI) has become a well-established way to integrate such heterogeneous business information systems and to map business processes to the technical system level. To do so, workflow systems and middleware are employed to constitute SOAs. Thereby, Web Services provide means to enable service orchestration as well as to hide the underlying infrastructure. Also, to allow resource sharing within a single organisation, technologies and concepts from the domain of High Performance Computing (HPC) and Grid computing were adopted to constitute so called Enterprise Grids. Nevertheless, a holistic approach for service orchestration and inter-organisational resource sharing in business information systems (BIS) is not yet available.

The recently started BIS-Grid project, a BMBF-funded project in the context of the German D-Grid initiative, intends to contribute to the realisation of such a holistic approach by not adopting but rather extending Grid technologies as a highly capable means of integrating decentralised BIS. In this paper, we illustrate our efforts to realise our approach. We present the intention of the project, portrayed by the description of two application scenarios to be performed with two of our partners, we present the requirements on which the project is based, and we present the architecture of a Grid-enabled WS-BPEL engine that is able to orchestrate stateful Grid Services.

1 Motivation

Enterprises often have to run various information systems for special purposes. For example, information on enterprise resources, customers, and products have to be stored and accessed. Thereby, information is held redundantly, making their maintenance a complex and expensive task. Enterprise Application Integration (EAI) has become a well-established way to integrate such heterogeneous business information systems [CHKT05], often accomplished via Web Service-based service orchestration in service-oriented architectures (SOA): business processes considered as workflows are mapped to the technical system level, relieving maintenance. Usually, enterprises are structured in departments which often have their own information technology (IT) resources such as servers and storage systems. As enterprises grow, their departments tend to evolve into “silos” containing large amounts of IT resources. Establishing Enterprise Grids is one approach to bridge department boundaries to balance workload and to reduce resource demand of otherwise isolated departments. EAI and Enterprise Grids have become well-established means to address challenges of BIS integration and resource sharing. Nevertheless, these challenges are addressed individually and only enterprise-wide. BIS integration and resource sharing are not well-integrated. Grid technologies such as the Grid middlewares UNICORE 6 and Globus Toolkit 4 are based on the Web Service Resource Framework (WSRF) [Ban06], a standard that allows stateless Web Services to become stateful. Stateful WSRF-based Web Services, also called Grid Services, provide the basis to build SOAs using Grid technologies. Thus, Grid technologies and EAI have much in common since both technologies focus on integration problems within a heterogeneous environment - Grid technologies on resource level and EAI on application level.

In BIS-Grid^{1,2}, a BMBF-funded project of the German D-Grid³ initiative, we intend to enable Grid technologies to be used for the integration of decentralised business information systems, bridging the gap between information system integration and resource sharing, and allowing EAI to dynamically traverse organisation-specific boundaries under consideration of enterprise security. This goes beyond the idea of Enterprise Grids because Enterprise Grids only take parts of Grid technologies and concepts from the Grid domain, and adapt and employ them within a single enterprise.

The project especially addresses small and medium enterprises (SME). BIS-Grid enables these enterprises to design and run workflows realised as Grid Service orchestrations to develop and provide dynamic solutions for information systems integration, resource sharing, and utilisation without having to run own large and expensive computing and storage centres. On the technical side of the project, a WS-BPEL-based [OAS07b] workflow engine that is capable of integrating Grid Services, the *BIS-Grid engine*, shall be developed.

¹ This work is supported by the German Federal Ministry of Education and Research (BMBF) under grant No. 01IG07005 as part of the D-Grid initiative.

² <http://www.bisgrid.de>

³ <http://www.d-grid.de>

2 Application Scenarios

The application scenarios are motivated by our industrial partners: CeWe Color⁴, the number one services partner for first-class trade brands on the European photographic market supplying both stores and internet retailers with photographic products, and KIESELSTEIN Group⁵, global market leader in the field of wire drawing and draw-peeling for the automotive industry. Both partners have needs for enterprise application integration, which they are willing to realise with Grid technologies: CeWe Color to integrate enterprise data for unified access for call centre agents, and KIESELSTEIN Group to improve access to, and retrieval and maintenance of product and project data. For CeWe Color, the impact of digital photography affected requirements to BIS and business processes. Product mass customisation and the need to flexibly respond to market development made demands to BIS that can adapt dynamically. Information about customer relations (CRM), stored in these systems, have to be accessible by call centre agents that use it to provide information to customers, e.g. about order status. This access has to be provided unified and with hard constraints to the quality of the demanded services. For example, system response time and security of data transfer is crucial for the call centre scenario and shall ideally be assured contractually. For KIESELSTEIN Group, the main challenge is to integrate enterprise resource planning (ERP) data and product (CAD/PDM) data that are distributed across two sites. Additionally, the BIS at these sites store information redundantly, since the enterprise grew together from three different producing factories, each providing their own information systems.

The overall goal of the two application scenarios is to investigate the feasibility and expenses of enterprise application integration based on Grid technologies, and to evaluate additional benefits on applying Grid technologies. Thereby, the applied solutions shall be able to allow the enterprise to dynamically switch between enterprise-specific Grid-based EAI (Figure 1a) and Grid utilisation (Figure 1b and 1c). While in the first case resources and application providing are both located within the enterprise (inhouse providing), Grid-enabled solutions allow application providing and/or resources to be outsourced to the Grid (Grid application providing). In this way, it becomes possible to integrate external Grid Services in enterprise-specific workflows and to create specific Virtual Organisation (VOs) that may traverse enterprise boundaries within a defined range without disregarding security issues. Nevertheless, when Grid application providing is applied, beside security issues that have to be addressed further quality of services must be guaranteed (cp. Section 3).

⁴ <http://www.cewecolor.de>

⁵ <http://www.kieselstein-group.com>

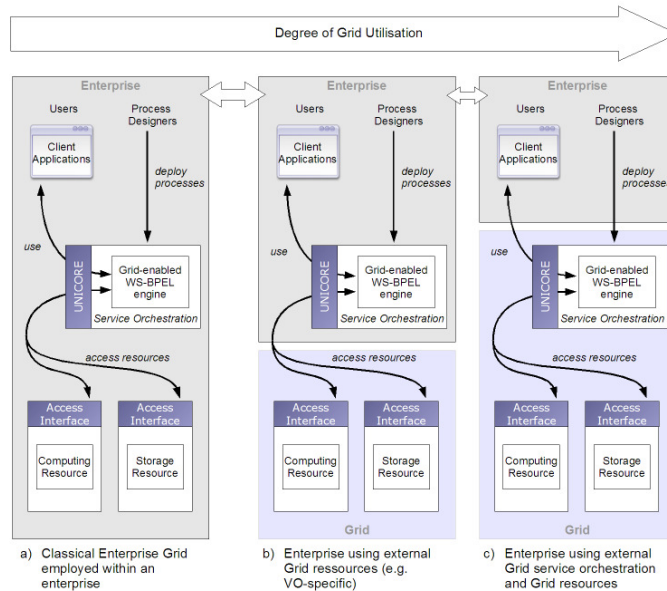


Fig 1: Dynamic utilisation of Grid resources

3 Requirements

To discover the requirements that apply to the intended BIS-Grid solutions, we first started with examining activities of workflow execution in general. This examination is based upon [EBC+06], who inter alia examine the management of scientific workflows. Then we examined demands from our application partners and derived business-related requirements. Also, we collected requirements that originate from demands that the Grid Computing domain makes on workflow-based service orchestration. Typical tasks that are needed for workflow execution are the definition of workflows, workflow deployment, and tasks that enable workflow execution (workflow enactment). This separation provides a framework for requirements discovery. In Table 1, the requirements are presented separated by these tasks: workflow definition (RQ-DF), workflow deployment (RQ-DP), and workflow enactment (RQ-EN). The table presents only the requirements that we regard as definitely to be fulfilled (MUST) and those to be fulfilled with appropriate efforts (SHALL). Requirements that may be fulfilled if project resources allow are not considered due to minor relevance.

<i>Requirement</i>	<i>Level</i>	<i>Description</i>
RQ-DF-1	MUST	To maintain state information, workflows must have a type system and must be able to import message types specified in WSDL-based service interfaces
RQ-DF-2	MUST	It must be possible to validate service orchestrations against the orchestrated services' WSDL interfaces
RQ-DF-3	MUST	Workflows must support service invocation and synchronisation, message content assembly, response information extraction, fault indication [EBC+06]

RQ-DF-4	MUST	Workflow definitions must support modelling control flow as well as data flow
RQ-DF-5	MUST	Workflow definitions must have the ability to temporarily store information to be exchanged between orchestrated services
RQ-DF-6	MUST	Workflow definitions must be hierarchically composable in order to allow high level service composition and workflow distribution
RQ-DF-7	MUST	Workflow definitions must support failure handling
RQ-DF-8	SHALL	Defining a business workflow shall be domain oriented; technology-specific definition details shall be concealed
RQ-DF-9	SHALL	It shall be possible to model transactional behaviour
RQ-DF-10	MUST	It must be possible to integrate human interaction within a service orchestration
RQ-DF-11	SHALL	It shall be possible to provide interaction roles (such as “administrator”) that enable authorisation for providing information within human interactions
RQ-DF-12	MUST	Workflows must handle the orchestration of stateful Web Service resources
RQ-DF-13	MUST	WSRF-compliance must be fulfilled
RQ-DP-1	SHALL	Before deployment, workflows shall be tested to assure functional correctness
RQ-DP-2	MUST	The workflow engine must support <i>hot deployment</i>
RQ-DP-3	MUST	Deployment security must be ensured when deploying on external resources
RQ-EN-1	MUST	A WSDL interface must be provided for each workflow so that it can be invoked remotely by sending a SOAP message to its WSDL interface
RQ-EN-2	MUST	It must be possible to execute workflow instances concurrently
RQ-EN-3	MUST	The workflow engine must be designed to provide workload scalability
RQ-EN-4	MUST	Workflows must be observable (e.g. observability of internal variable values, subtask status, and failure occurrences and exceptional conditions)
RQ-EN-5	MUST	Quality of services must be assured (e.g. system response time, security)
RQ-EN-6	MUST	It must be ensured that a third party is not able to manipulate, or interfere neither running workflow instances nor service calls initialised by workflow instances
RQ-EN-7	MUST	The workflow engine must provide means to enable humans to provide needed information, including an authorisation mechanism for these interactions
RQ-EN-8	MUST	The state of active workflow instances must be persisted, e.g. for crash recovery
RQ-EN-9	MUST	The costs of external service utilisation must be acquired and cumulated dynamically, and must be examinable by the owner of the workflow instance

Tab. 1: Requirements

4 Basic Technologies and Standards

Based on the preceding analysis, we considered technologies and standards that meet the identified requirements as well as possible. We focussed on technologies and standards that are already well-established in the Grid domain or in the business domain, or in both. The following technologies and standards are selected to be applied in BIS-Grid.

The *Business Process Modeling Notation* (BPMN) [OMG06b] is a standardised flow-chart based notation maintained by the Object Management Group (OMG). It has been developed by the Business Process Management Initiative (BPMI). BPMN has been created to model understandable graphical representations of business processes. The intend is to provide a standard visual language that process modellers recognise and understand. Thereby, the focus is set on modelling the control flow of activities.

Main aspects of the BPMN specification are conformance to visual appearance of the BPMN graphical elements, conformance to the semantics of BPMN elements, and the possibility to exchange BPMN diagrams between conformant tools. However, the current adopted specification does not contain a standard mechanism for diagram exchange, nor a formal metamodel. With the upcoming *Business Process Definition Metamodel* (BPDM) [OMG06a] an explicit metamodel and serialisation mechanism for BPMN concepts will be provided.

The *Web Service Business Process Execution Language* (WS-BPEL) [OAS07b] is an XML-based OASIS standard to describe business workflows whose activities are implemented as external Web Services. Also, the WS-BPEL business workflows are offered via Web Service interfaces as well. WS-BPEL makes use of existing standards such as XML Schema, WS-Addressing and XPATH, and offers comprehensive elements to model the control flow (sequences, loops, parallel flows, etc.) of a business workflow as well as for data manipulation. More complex features like fault and error handling are considered, too. Together with the concept of compensation (roll back failed activities) transaction-similar behaviour is realisable in the presence of Web Service invocations. The model of a business workflow in WS-BPEL only uses abstract parts (port types and operations) of a Web Service WSDL interface to describe the external message communication. Binding these abstract interfaces to concrete service locations is obliged to the executing WS-BPEL engine. Since WS-BPEL is a widely adopted standard we have chosen it to model business workflows that are capable of orchestrating Grid Services. Furthermore, several engines are available to execute WS-BPEL-based business workflows. In BIS-Grid we decided to use ActiveBPEL⁶, because it comprehensively implements the current WS-BPEL 2.0 standard, is available as open source, and is well-established in the commercial domain.

The *Web Services Resource Framework* (WSRF) [Ban06] is an XML-based OASIS standard that emerged from the Grid context. It extends stateless Web Services, so that stateful interactions with so-called Web Service resources are possible in a standard manner. WSRF is a family of five standards each describing a specific aspect needed for stateful Web Services (better known as Grid Services), for example, the management of a (WS-) resource lifetime. The WSRF standard is well-established in the Grid context and was adopted by the Grid middlewares Globus Toolkit 4⁷ and UNICORE 6⁸. They use Grid Services for basic Grid functionalities such as file transfer and resource allocation that require stateful interaction. In BIS-Grid we decided to use UNICORE 6 for our purposes. It provides features offered as so-called UNICORE Atomic Services (UAS) implemented as WSRF-conform Grid Services, providing basic Grid functionalities not available by ActiveBPEL. Furthermore, it provides authentication and authorisation mechanisms based on X.509 certificates. The implementation of self designed Grid Services called Additional Services is possible, too. A direct contact to the developers of UNICORE enables us to sustainably integrate our extensions into UNICORE 6.

⁶ <http://active-endpoints.com/active-bpel-engine-overview.htm>

⁷ <http://www.globus.org>

⁸ <http://www.unicore.eu>

We intend to use UNICORE 6 as a transparent Grid-layer upon the ActiveBPEL engine as described in Section 6. Table 2 presents an overview on the technologies and standards with a description which requirements presented in Section 3 they address.

<i>Technology/Standard</i>	<i>Addressed Requirement</i>	<i>Description</i>
BPMN	RQ-DF-8	Focuses on modelling business processes (control flow-oriented), omitting technical details (such as explicit modelling of data flow)
WS-BPEL	RQ-DF-1	Provides a type system that relies on WSDL message type import
	RQ-DF-3	Supports basic activities (invocation, synchronisation, message content assembly, information extraction from responses)
	RQ-DF-4	Requires to explicitly model control flow as well as data flow to provide an executable business process description
	RQ-DF-5	Provides a temporary information storage mechanism by variables
	RQ-DF-6	Hierarchically composable; service orchestrations are regarded as services
	RQ-DF-7	Provides failure handling mechanisms by fault and compensation handlers
	RQ-DF-9	Supports modelling transactional behaviour
	RQ-DF-13	Already addresses most of the WSRF requirements [Ley06]
RQ-EN-1	Orchestrations can be described by WSDL interfaces	
ActiveBPEL	RQ-DP-2	Supports hot deployment
	RQ-EN-2	Supports concurrent execution of workflow instances
	RQ-EN-3	Scalability of ActiveBPEL is “ <i>limited only by the available hardware resources</i> ” [EBC+06] ⁹
	RQ-EN-4	Provides simple workflow monitoring capabilities
	RQ-EN-8	Can be configured to use relational databases for workflow persistence
WSRF	RQ-DF-12	Defines a generic framework for modelling and accessing stateful resources using Web Services
UNICORE 6	RQ-DF-13	Is WSRF-compliant Grid middleware platform
	RQ-EN-6/-7	Supports X.509 certificate-based authorisation and authentication

Tab. 2: Technologies and standards to be used and addressed requirements

Requirements that are not or not fully covered have to be addressed directly within the project. We assume that WS-BPEL can be used in a way that fulfils the requirements RQ-DF-10 and RQ-DF-11 (modelling role-enabled human interaction), regarding our application scenarios, and RQ-DF-12 (orchestration of stateful Web Services) without needing to extend or adapting the language. According to [Ley06], BPEL4WS [BPE06], which is the predecessor of WS-BPEL, already fulfils many of the requirements of the WSRF, thus addressing RQ-DF-12 and RQ-DF-13. The workflow engine must recognise the WS-BPEL-compliant modelling and must execute the workflow as intended by the modelling. Also, the engine has to provide secure deployment and execution of the workflows (RQ-DP-3, RQ-EN-5, and RQ-EN-6) which possibly can be achieved by utilising UNICORE’s authorisation and authentication mechanisms. These mechanisms can also be used to fulfil requirement RQ-EN-7, authorisation-based human interaction.

⁹ Nevertheless, to “*achieve scalable workflows it is important that orchestrated services need to be built to scale too*” [EBC+06]

When regarding RQ-EN-5, quality of service assurance, using Service Level Agreements (SLA) may be a convenient way to provide a contractual basis for workflow and service utilisation in the business domain (cp. [MMM06]). Solutions for RQ-EN-9, accounting and billing, must be based upon adequate business models to be identified. This is subject to our project as well but is not in the focus of this paper. RQ-DF2, RQ-DF-8, RQ-DP-1, and RQ-DP-3 may be fulfilled by a tool that allows to model workflows domain-oriented, to refactor the modelling to be executable by the workflow engine, to validate workflows against WSDL interfaces of the orchestrated services, to test workflows for functional correctness, and to deploy workflows in a secure way by providing authorisation and authentication to the workflow engine and to UNICORE. Such a tool is also not in the focus of this paper.

5 BIS-Grid Approach

The approach that we follow in BIS-Grid is based on three main tasks: modelling processes that shall be executed (semi-)automatically, preparing the modelled workflows to be executable by the BIS-Grid engine, and developing services that shall be orchestrated. The first part of the approach is that business processes of an enterprise are modelled with means that are appropriate to the business domain. We consider the BPMN as such a means since it is oriented to the business domain and it provides an adequate degree of abstraction from technical process definition details. The result of business process modelling is BPMN business process diagrams that illustrate the enterprise's workflows with focus on control flow and activities designation. The role of employees that execute this task is the *domain expert*, which, for example, may be a business analyst. Grid Services to be orchestrated have to be developed by *service developers*. When these services are deployed, information about the services, for example about its functions, data and data types, and protocols, is provided by a WSDL interface. The services are deployed in a UNICORE 6 Service Container. The next step is that the previously created processes are refined to create workflows that can be executed by the BIS-Grid engine. Thereby, the BPMN business process diagrams are used to create WS-BPEL service orchestrations on basis of the WSDL interfaces.

A conceivable procedure to do so may be to use the WSDL interfaces and a model representation of the BPMN business process diagrams, possibly based upon the upcoming BPDm [OMG06a], as input to generate initial WS-BPEL service orchestrations. These are finally refined by *workflow experts* who, for example, refine data flow. The WS-BPEL Grid Service orchestrations are then deployed in the BIS-Grid engine. In Section 6, we present a preliminary architecture of the BIS-Grid engine that executes WS-BPEL-based Grid Service orchestrations. A tool that covers business process modelling, workflow refinement and deployment is envisaged in Section 8.

6 Software Architecture of the BIS-Grid Engine

One technical goal in BIS-Grid is the development of a WS-BPEL-based workflow engine that is capable of integrating Grid Services, the *BIS-Grid engine*. This engine is a composition of ActiveBPEL and the Grid middleware UNICORE 6. Thereby, ActiveBPEL is used to execute WS-BPEL-based workflows. Upon the ActiveBPEL engine a Grid-layer is created on basis of UNICORE 6 so that each workflow can both use Grid Services and can be offered as a Grid Service as well. Technical issues that are relevant for Grid Services, in particular security, are completely managed by UNICORE 6 and are transparent to ActiveBPEL. Thus, our extensions will only affect UNICORE 6 and not the underlying ActiveBPEL engine which could therefore be replaced by any other WS-BPEL engine. A more detailed view of the architecture is shown in Figure 2. The core elements are the *BIS-Grid Service*, running in the UNICORE 6 service container, and one or more ActiveBPEL engines, running on cluster nodes for scalability reasons. For each *BIS-Grid workflow* a WS-BPEL workflow is deployed in an ActiveBPEL engine and a corresponding Grid Service, called *Workflow Service*, is deployed in the UNICORE 6 service container. The BIS-Grid Service provides the functionality for a *workflow expert* to deploy and undeploy BIS-Grid workflows. A deployment package for a BIS-Grid workflow consists of a WS-BPEL workflow description and additional information, for example, endpoint references used for dynamic binding of Grid Services, security policies, and billing information. After the BIS-Grid Service has received a deployment package two tasks are executed. The first calls the deployment service of the ActiveBPEL engine which adds the WS-BPEL workflow to its workflow repository. Possible errors, e.g. caused by an invalid description or incorrect deployment information, are reported to the workflow expert.

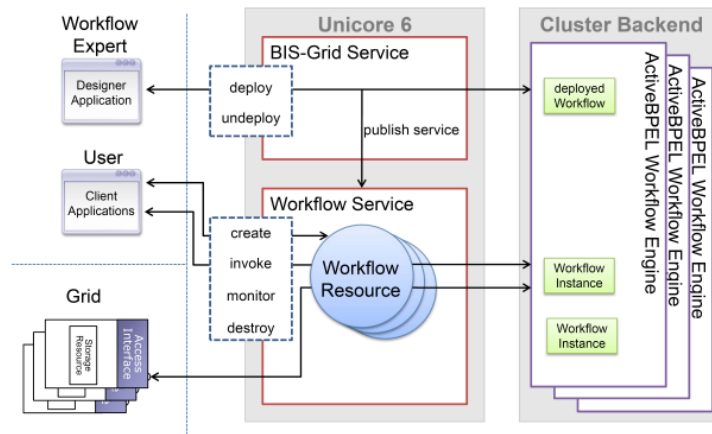


Fig 2: Architecture overview

If the first task succeeds the second creates a new Workflow Service and deploys it in the UNICORE 6 Service Container. This Workflow Service provides the same interface as the original WS-BPEL workflow supplemented by additional Grid-specific functions, e.g. to create or destroy a *workflow instance*, to monitor workflow execution, and to retrieve current accounting and billing information.

After the creation of a *workflow resource* via the Workflow Service method *create* each call to a workflow resource method (e.g. *invoke*) that originates from the WS-BPEL workflow interface is forwarded to the WS-BPEL workflow in the ActiveBPEL engine. The first message which is sent to a WS-BPEL workflow creates a new workflow instance in the ActiveBPEL engine. This instance is used by the workflow resource in further calls. If the workflow resource is no longer used, e.g. when the workflow terminates, it can be destroyed via the *destroy* method whose call also destroys the corresponding workflow instance in the ActiveBPEL engine. Using a BIS-Grid workflow is usually hidden to the user by a client application. A very strong requirement in the business domain is security. Thus, among others the architecture addresses secure conversation (SSL, encrypted SOAP) as well as authentication and authorisation of users, which is partly provided by UNICORE 6. This ensures a basic secure access to workflow resources and workflow instances. We intend to extend these security features by integrating a role-based access system into UNICORE 6 to allow the definition of fine-grained access policies at method-level on the workflow resource interface. Summarising, we introduced a scalable and secure architecture for a Grid-enabled workflow engine based on the workflow definition language WS-BPEL. We build on standardised and well-established technologies in the business and Grid domain, namely ActiveBPEL and UNICORE 6 to obtain interoperability and sustainability. In order to fulfil the requirements presented in Section 3, we plan to develop UNICORE 6 Grid Services (the BIS-Grid Service and the Workflow Service) and the integration of a fine-grained role-based access system. However, the current state of work is that the BIS-Grid engine's architecture described in this section is preliminary and the definition of an in-depth architecture is ongoing. Implementation start is envisaged for early 2008.

7 Related Work

There are several papers on the orchestration of stateful services via BPEL4WS [BPE06]. Leymann [Ley06] describes the appropriateness of using BPEL4WS as a basis for Grid Service orchestration since it already fulfils many requirements of the WSRF standard. He states some missing aspects especially concerning monitoring capabilities, and concludes that a Grid-specific extension of BPEL4WS is more appropriate than creating new Grid-specific standards. The appropriateness of BPEL4WS for Grid Service orchestration is also confirmed in [EBC+06], [Slo06], and [CYG+04]. Emmerich et al. [EBC+06] illustrate their experience in orchestrating scientific workflows gained within an extensive case study for the automation of polymorph prediction application, using BPEL4WS [BPE06]. They describe the extent with which BPEL4WS supports the definition of scientific workflows, and describe the evaluation of reliability, performance, and scalability of ActiveBPEL on executing a complex scientific workflow. In [DFH+07], Dörnemann et al. discuss composing Grid Services by using BPEL4WS, too. They present a solution that is conceptually based on extending the BPEL4WS specification. Also, they present an Eclipse-based workflow modelling tool they have developed. Dörnemann et al. see the seamless integration of security mechanisms, for example WS-SecureConversation [OAS07a] as well as Virtual Organisation management in the BPEL4WS engine and the modelling tool, as a topic of further research.

Orchestrating stateful services is in the focus of many German and international projects. For example, the German D-Grid projects Text-Grid [Tex06], and InGrid [ING06], European projects such as A-WARE [AWA06], Chemomentum [CHE06], and EGEE [EGE06], and international projects such as g-Eclipse [GEC06], myGrid [MYG06], and caGrid [caG06]. These projects, as well as the previously described papers, mainly focus on scientific workflows instead of business workflows that are relevant to BIS-Grid. Concerning the use of BPEL (BPEL4WS or WS-BPEL), the described papers focus on extending or adapting the language, thus creating BPEL dialects. In BIS-Grid, we instead focus on proper BPEL modelling by using existing language elements. We are in contact with some of the mentioned projects to work together on common problems and to use other synergy effects.

8 Conclusion

In this paper we have presented the workflow-based approach of the project BIS-Grid to utilise Grid technologies for the integration of decentralised business information systems (BIS) as an enhancement of classical Enterprise Application Integration (EAI) in service-oriented architectures (SOA). We introduced two application scenarios with commercial partners that shall evaluate the feasibility and determine the expense and benefit of this approach. Afterwards, we described identified requirements to realise a Grid-enabled workflow execution and separated them by workflow definition, workflow deployment, and workflow enactment. The standards BPMN, WS-BPEL and WSRF as well as the WS-BPEL engine from ActiveBPEL and the Grid middleware UNICORE 6 were identified as an appropriate technical basis to be build on that already covers many of the requirements. Requirements which were not fulfilled are focussed directly in BIS-Grid. Furthermore, we presented a procedure to transfer abstract business domain-oriented workflows to concrete and executable technology-based Grid workflows utilising the technologies mentioned above. Based on ActiveBPEL and UNICORE 6 we developed an architecture for a WS-BPEL based workflow engine that is capable of integrating Grid Services.

Besides refining the architecture of the so-called BIS-Grid engine, especially focussing on quality of services, we focus on further challenges in future. One challenge is to use WS-BPEL for interactions that are not recognised in the standard explicitly, for example, the invocation of (stateful) Grid services and human interaction. We plan to create a catalogue of WS-BPEL patterns for these type of interactions that are solely based on existing WS-BPEL elements and thus compatible to each WS-BPEL-compliant engine. These patterns must be supported by a WS-BPEL modelling tool to hide their complexity. This tool must also support modelling and refactoring business domain-oriented workflows to be executable by the BIS-Grid engine. Furthermore, validation and testing of workflows must be possible, too. At last, we have to focus on commercial issues such as accounting and billing as well as to develop appropriate business models for Grid providing.

References

- [AWA06] A-WARE. <http://www.a-ware-project.eu/>, 2006. Last Visited: 2006-11-22.
- [Ban06] Tim Banks. Web Services Resource Framework (WSRF) - Primer v1.2. <http://docs.oasis-open.org/wsrp/wsrp-primer-1.2-primer-cd-02.pdf>, May 2006.
- [BPE06] Business Process Execution Language for Web Services Version 1.1. <ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf>, 2006. Last Visited: 2006-11-15.
- [caG06] caGrid/caBIG. <http://www.cagrid.org/>, 2006. Last visited: 2007-09-26.
- [CHE06] Chemomentum. <http://www.chemomentum.org>, 2006. Last Visited: 2006-11-22.
- [CHKT05] S. Conrad, W. Hasselbring, A. Koschel, and R. Tritsch. Enterprise Application Integration. Elsevier, 2005.
- [CYG+04] Kuo-Ming Chao, Muhammad Younas, Nathan Griffiths, Irfan Awan, Rachid Anane, and C-F Tsai. Analysis of Grid Service Composition with BPEL4WS. In Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04), volume 01, page 284, Los Alamitos, CA, USA, 2004. IEEE Computer Society.
- [DFH+07] T. Dörnemann, T. Friese, S. Herdt, E. Juhnke, and B. Freisleben. Grid Workflow Modelling Using Grid-Specific BPEL Extensions. German e-Science Conference 2007, May 2007.
- [EBC+06] Wolfgang Emmerich, Ben Butchard, Liang Chen, Sarah L. Price, and Bruno Wassermann. Grid Service Orchestration Using the Business Process Execution Language (BPEL). In Journal of Grid Computing (2006), number 3, pages 283–304. Springer, 2006.
- [EGEE06] EGEE. <http://www.eu-egee.org/>, 2006. Last Visited: 2006-11-22.
- [GEC06] g-Eclipse. <http://www.geclipse.org/>, 2006. Last Visited: 2006-11-22.
- [ING06] InGrid. <http://www.ingrid-info.de/>, 2006. Last Visited: 2006-11-22.
- [Ley06] Frank Leymann. Choreography for the Grid: towards fitting BPEL to the resource framework: Research Articles. *Concurr. Comput. : Pract. Exper.*, 18(10):1201–1217, 2006.
- [MMM06] Philipp Masche, Paul Mckee, and Bryce Mitchell. The Increasing Role of Service Level Agreements in B2B Systems. In WEBIST 2006, Proceedings of the Second International Conference on Web Information Systems and Technologies: Society, e-Business and e-Government / e-Learning, Setúbal, Portugal, April 11-13, 2006, pages 123–126, 2006.
- [MYG06] myGrid. <http://www.mygrid.org.uk/>, 2006. Last Visited: 2006-11-22.
- [OAS07a] OASIS Web Services Secure Exchange TC. WS-SecureConversation 1.3. <http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/wssecureconversation-1.3-os.pdf>, March 2007.
- [OAS07b] OASIS WSBPEL Technical Committee. Web Services Business Process Execution Language (WSBPEL) Primer. <http://www.oasisopen.org/committees/download.php/23974/wsbpel-v2.0-primer.pdf>, May 2007.
- [OMG06a] OMG Object Management Group. Business Process Definition MetaModel (BPDM), Beta 1. <http://www.omg.org/docs/dtc/07-07-01.pdf>, February 2006. OMG Adopted Specification.
- [OMG06b] OMG Object Management Group. Business Process Modeling Notation (BPMN) Specification. <http://www.bpmn.org/Documents/OMG-02-01.pdf>, February 2006. OMG Final Adopted Specification.
- [Slo06] Aleksander Slomiski. On using BPEL extensibility to implement OGSi and WSRF Grid workflows: Research Articles. *Concurr. Comput. : Pract. Exper.*, 18(10):1229–1241, 2006.
- [Tex06] Text-Grid. <http://www.textgrid.de/>, 2006. Last Visited: 2007-09-26.