MODEST

Model-Driven Agents for Semantic Web Services

BMB+F Project Proposal

DFKI GmbH
Department of Deduction and Multi-Agent Systems
German Research Centre for Artificial Intelligence
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany

Dr. Klaus Fischer, Dr. Matthias Klusch, Christian Hahn, Ingo Zinnikus

November, 2007

Modest: Model-driven Agents for Semantic Web Services

1 Project Objectives

The project Modest aims at cross-disciplinary research and development of innovative means on model-driven agent-based coordination of Semantic Web services with limited proof-of-concept demonstration. Modest integrates model-driven software development, service-oriented architectures, and Semantic Web service technologies for an innovative approach to platform independent coordination of Semantic Web services by intelligent agents.

It is planned to apply the expected results (e.g. modeling tools) developed in Modest for the modeling of collaborative business processes at different sites of Saarstahl AG. With a longer term perspective one can expect that the results of Modest can be commercialized in the area of service-oriented architectures.

Research in Modest is structured into five work packages where the focus is on the development of a platform independent metamodel for Semantic Web services (WP3). This metamodel is integrated with an already existing platform independent metamodel for multiagent systems which, however, will need some modifications to take into account the new requirements (WP2). The integrated metamodel will be extended with innovative approaches to semantic service matchmaking to support the integration of existing Semantic Web services at system design time (WP4). The two prototypical pilot applications in steel production and in eHealthcare are used to demonstrate the usefulness of the developed approach (WP5).

1.1 Driving Vision

Today's enterprises operate in a dynamic environment which is characterized by global outsourcing, shrinking product life-cycles, and unstable demand. To prosper in this environment, enterprises face a growing need to share information and to collaborate at all levels of the value chain. As organizations are gradually transforming into "networked organizations", interoperability becomes the main challenge to realize the vision of seamless business interaction across organizational boundaries.

Service-oriented architectures are today's favorite answer to solve interoperability issues. As various kinds of systems can be used to implement the service-oriented architecture, the recent trend is to apply the principles of model-driven development by (i) modeling the service-oriented architecture in a more abstract manner and (ii) providing model transformations that define mappings between this more abstract specification and the underlying platform specific systems. In principle, the model-driven development approach allows specifying a custom-tailored workflow which incorporates the practical experience of business analysts and developers. Despite the rather static nature of many business scenarios, points or situations of choice occur, where the specific partners which deliver a specific task, e.g. a carrier service, can be selected at design or even at execution time, thus allowing a certain degree of flexibility in business service provisioning if semantically-enriched information is available. However, since the existing standards for business services lack semantics, the meaningful integration of services exclusively relies on human business domain experts. In contrast, Semantic Web service technology adds expressivity to existing Web service standards by introducing well-formed semantics that simple Web service descriptions are lacking and envisages the automated and meaningful composition of complex business services through logic-based reasoning upon their semantic annotations. However, in many real-world cases of business process modeling among contracted and trusted business partners, the fully automated coordination of partly unknown business Web services is neither adequate nor efficient in practice. When service composition is concerned the Semantic Web service approach can be compared to planning from first principles while the model-driven approach can be compared to planning from second principles if the platform specific engine for executing the models is powerful enough.

This short discussion already reveals that both approaches, model-driven development as well as Semantic Web services, have their pros and cons when used to integrate external, outsourced business services. However, the commonality that both approaches envision intelligent agents to be key enabler for service composition and coordination results in the following research questions that have to be answered in Modest.

Can agent technology make both concurrent technologies, model-driven development and Semantic Web services, more convergent to realize efficient and meaningful agent-based coordination of pervasive business services in practice? Is it possible to harness the benefits of both for this purpose?

Our research hypothesis is that the combination of a top-down, agent-based model-driven development approach with Semantic Web service technology for business service discovery and, eventually, ad-hoc service invocation is not only feasible but enables a seamless integration of agent systems into semantically enhanced service-oriented architectures.

1.2 Project Goals - Overall, Scientific

The main objective of the project MODEST is to provide a model-driven engineering tool-supported methodology for the seamless integration of intelligent agents into service-oriented architectures in a constantly changing pervasive environment of Semantic Web services. In consequence, the project's core operational objectives are:

- Design of a new methodology and best practices for the model-driven development of platform independent Semantic Web services.
- Developing new solutions to the integration of agents into service-oriented architectures based on both platform independent metamodels for multiagent systems and Semantic Web services.
- Design of a model-driven Semantic Web services matchmaker agent that discovers semantic services independent of selected description formats (in model-driven development terms: platforms) like OWL-S (Ontology Web Language for Services), WSML (Web Services Modeling Language) and SAWSDL (Semantic Annotation of WSDL and XML Schemas)[1].
- New solutions to the hybrid Semantic Web service selection problem with focus on the standard SAWSDL.
- Test and evaluate the technologies produced in order to identify pros and cons of the technological approach for future extensions and exploitation.

1.3 Expected Outcome

The main expected outcomes of the MODEST project include the following three interlinked results that are produced in respective work packages:

- Platform independent metamodel for multiagent systems that draws upon related work in the project ATHENA[2]. This includes the development of modeling tools for the creation and editing of service-based models by means of relevant technologies such as the Eclipse Modeling Framework[3] and the Graphical Modeling Framework[4] (cf. work package WP2).
- Platform independent metamodel for Semantic Web services together with model transformations to selected individual metamodels of Semantic Web service formats (OWL-S, WSML, SAWSDL) that allow to transfer information from one platform to the other. This includes the development of appropriate modeling tools by means of the Eclipse Modeling Framework and the Graphical Modeling Framework (cf. work package WP3).
- Hybrid Semantic Web service matchmaker for SAWSDL and experimental evaluation of its proof-of-concept implementation over an initial test collection to be built as well as initial version of a model-driven Semantic Web service matchmaker (cf. work package WP4).
- Demonstrator of service metamodeling and matchmaking for selected steel production and eHealthcare use case scenarios (cf. work package WP5).

[1] www.w3.org/TR/sawsdl

[2] Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications, `http://www.athena-ip.org`

[3] www.eclipse.org/emf/

[4] http://www.eclipse.org/gmf/

2 State of the art and preparatory work

In this section we provide a brief overview of the state of the art in the related research area, building on the results of the ATHENA and the SCALLOPS[5] projects. Two papers [4, 18] that present the previous work Modest builds on were accepted for full presentation at the conference on Autonomous Agents and Multiagent Systems (AAMAS) 2008 which gives evidence that, on the one hand, the description actually is state of the art and, on the other hand, the previous work is of major interest in agent-related research.

2.1 State of the art

2.1.1 Agent-based integration of Semantic Web services

Modest project will research the support of a model-driven seamless integration of Semantic Web services into agent-based systems. For this purpose, we focus in work package WP2 on the extension of our previously developed platform independent metamodel for multiagent systems. This extension could for instance include an ontology metamodel that serve as a reference ontology for the platform independent metamodel for agents and Semantic Web service. In the context of ATHENA, the integration of Web services into business processes using principles of model-driven development was one of the core topics. We demonstrated in [19], how to integrate Web services into the agent-oriented programming language JACK. However, the integration is done semi-automatically by using syntactical service descriptions. A fully automated integration could be achieved using the semantic information provided by the Semantic Web services in combination with the reference ontology.

Several approaches already exist to integrate Web services into agent systems (e.g. [11, 14, 10]). However, they neither deal with an adequate metamodel for Semantic Web services nor provide a model-driven based integration. The authors of [6] discuss a model-driven integration of Semantic Web services into agent systems. However, the proposed metamodel to describe Semantic Web services neither bases on any of the prominent description formats like OWL-S, WSML, or the standard SAWSDL nor allows defining a model transformation to them which makes the approach less practical.

2.1.2 Metamodeling of Semantic Web Services

We want to design a new methodology and best practices for the model-driven development of platform independent Semantic Web services (PIM4SWS). For this purpose, we focus on (i) the development of a platform independent metamodel for Semantic Web services; and (ii) the definition of model transformations from the platform independent metamodel to the existing metamodels for OWL-S, WSML, and SAWSDL. Please note that in the context of model-driven development the term *platform independent metamodel* has the same meaning as the term *common Semantic Web service description language* in the context of Semantic Web service technology.

Recently, metamodels were defined for two of the three prominent Semantic Web service descriptions languages. In [15, 3], the authors describe a metamodel for OWL-S and [15] discusses a metamodel for WSML. However, to the best of our knowledge, a platform independent metamodel for Semantic Web services that abstracts from the prominent examples OWL-S, WSML, and SAWSDL does not exist yet. The development of this metamodel will be done in work package WP3.

We take the metamodels of OWL-S and WSML as a base to define a model transformation from the PIM4SWS into these representations. However, in the case of SAWSDL, we have to define our own metamodel that will be the target of the SAWSDL model transformation. The model transformations as well as the definition of a metamodel for SAWSDL will be done in work package WP4.

2.1.3 Semantic Web Service Selection

The Modest project tries to bridge the gap between model-driven service architectures and Semantic Web services. For this purpose, we focus on the development of (i) a matchmaker agent for selecting registered Semantic Web services in SAWSDL that semantically best match a given user query; and (ii) model-driven Semantic Web services matchmaker agent that discovers semantic services independent from selected description formats.

According to the project vision, the automated selection of semantically relevant Web services by matchmaker agents strongly supports the semi-automated design of business processes in service-oriented

[5] http://www-ags.dfki.uni-sb.de/~klusch/scallops/

architectures. There are quite a few Semantic Web service matchmaker agents available for the semantic service description languages OWL-S and WSML, and others, but none for the standard SAWSDL yet. In work package WP4, we will develop and evaluate the performance of an all new hybrid semantic matchmaker agent for semantic services in SAWSDL.

For this purpose, we will particularly draw inspiration from the fields of ontology alignment, information retrieval, and our previous work on hybrid semantic service matchmakers for semantic services written in OWL-S and WSML, that are OWLS-MX [9] and WSMO-MX [7]. The majority of Semantic Web service matchmakers performs logic-based semantic service matching. That is, they are keeping with the original idea of the Semantic Web to determine semantic relations between resources (including services) on the basis of logical inferencing using their annotations grounded in description logics and/or rules. Non-logic-based semantic matchmakers do not perform any logic-based reasoning but compute the degree of semantic matching of given pairs of semantic service descriptions based on, for example, syntactic similarity measurement, structured graph matching, or numeric concept distance computations over given ontologies. Both matchmakers, OWLS-MX and WSML-MX are hybrid in the sense that they combine text similarity measurement with logic-based reasoning on semantic annotations defined in formal ontologies. For a comprehensive survey of semantic Web service matchmakers, we refer to [8].

2.2 Preparatory Work

2.2.1 Service Integration into the Agent-oriented Programming Language JACK

In the context of the integrated EU FP6 project ATHENA the multiagent systems research group developed a model-driven approach for BDI agents based on the JACK [12] development environment. The main idea for ATHENA was to demonstrate how models which were defined according to the platform-independent metamodel for service-oriented architectures (PIM4SOA) can be transformed into models that can be compiled into executable code using a metamodel definition for JACK and the JACK development environment (see [2, 5] for a detail discussion of the transformations). Furthermore, in order to use a Web service within plans of JACK agents, we defined a second transformation that maps the concepts of a metamodel for WSDL to particular concepts of the JACK metamodel (e.g. Capability). Detailed information on the model-driven framework for the integration of services into agent systems can be found in [19].

2.2.2 Development of a Platform Independent Metamodel for Multiagent Systems

In our ongoing work, we offer a proposal on how to exploit the ideas and techniques of model-driven development in agent-oriented software engineering. We developed a platform independent metamodel for multiagent systems and defined model transformations from this metamodel to the metamodels of JACK [12] and JADE [1]. The core of the platform independent metamodel for multiagent systems was divided into seven views, i.e. agent, organization, interaction, behavior, role, environment, and multiagent system that are discussed in more detail in [4].

2.2.3 Supply Chain Management in Steel Production

For many years, the MAS research group cooperates closely with Saarstahl AG. In this cooperation agent technologies and other innovative approaches are investigated whether they can be applied in supply chain management in steel production. The supply chain starts with hot metal production. After the hot metal is converted into steel in the steel works, the steel is transformed into steel products (wire rod, bar steel, beams, semi-finished products, and products of open die forgings) for customers at the masticator sites. Saarstahl sees its core competence in the production of specialized steel products, for which the knowhow and the efficient use of resources are key success factors. In Saarland Saarstahl has about 5000 employees, which produced in 2006 roughly 2,6 mio. tonnes of steel. Saarstahl AG is therefore a major economic factor of Saarland.

3 Detailed Description of the Workplan

3.1 Work Packages of Modest

Modest is structured in five work packages (WP). WP1 deals with project management. In WP2 and WP3 the metamodels for multiagent systems and for Semantic Web services (SWS) are investigated, where the focus of Modest lies on WP3. WP4 investigates service matchmaking for Semantic Web services described in the standard SAWSDL. In WP5 the developed concepts are applied in two application pilots.

In the following we describe the individual work packages and the effort regarding PMs.

WP1: Management (1PM)

This work package deals with project management and is open for the whole duration of Modest. The tasks to deal with are: (i) Organization of meetings and coordination between the different work packages; (ii) Controlling of the project budget; (iii) Coordination of reporting and publication of results at scientific conferences and journals; and (iv) Maintenance of the exploitation plan and early identification of potential products from Modest results.

R1.1: Interim report

R1.2: Final report

WP2: Platform Independent Metamodel for Multiagent Systems

The objective of this work package is to investigate how to extend the metamodel for multiagent systems (PIM4Agents) with respect to the seamless integration of Semantic Web services in a model-driven manner.

3.1.1 Problem Description

Several agent-oriented programming languages (e.g. JACK, JADE, etc.) exist to design and implement multiagent systems. The platform independent metamodel for multiagent systems allows to define agent systems on a very abstract[6] level. Besides the general benefit to improve (i) quality by allowing the reuse of models and the mappings between models and (ii) software maintainability by favoring a better consistency between models and code, this metamodel (i) establishes interoperability among various agent systems and other information technologies, and (ii) identifies common agent-oriented concepts to increase the interoperability between agent-oriented programming languages.

3.1.2 Solution Approach

For the seamless integration of Semantic Web services, we (i) extend the core metamodel using the Eclipse Modeling Framework (as done for specifying the core of PIM4Agents) as well as (ii) define a formal semantics for the PIM4Agents using Object-Z [16]. In combination with work package WP3, it will be possible to design and model semantically-enriched service-oriented architectures where agents are used as mean for service composition and coordination.

3.1.3 Preconditions, Prerequisites

This work package has no internal preconditions, however, the metamodel for multiagent systems as well as parts of the graphical editor are based on our previous work.

3.1.4 Description of Work

Task 2.1 (Extending the PIM4Agents Metamodel) The core of the PIM4Agents is structured into several MAS-specific aspects that are extended in this work package to guarantee the seamless integration of Semantic Web services into MAS. These extensions include for instance the specification of an ontology metamodel that enhances the description of semantically-enriched information within the PIM4Agents. The additional aspects as well as the concepts are—like the core of the PIM4Agents—specified using the Eclipse Modeling Framework (1 PM).

[6] abstract regarding the agent-oriented programming languages on the platform specific level, e.g. JACK, JADE, etc.

Task 2.2 (Semantics) As a metamodel contains only little information about what the concepts in a language actually means, we develop a complete formal specification of the PIM4Agents using Object-Z [16]. Using Object-Z, we refine the syntax of the individual concepts in the PIM4Agents by formalizing their attributes and invariants. The semantics is defined by specifying a denotational and operational semantics. The denotational semantics is defined in terms of introducing additional semantic variables and invariants. The operational semantics is defined in terms of class operations and invariants restricting the operation sequences that are specified using the timed trace notation of the timed refinement calculus [17] (2 PM).

Task 2.3 (Graphical Editor) Basing on the Eclipse's Graphical Modeling Framework, the syntax of the PIM4Agents is mapped to a concrete graphical syntax. The Graphical Modeling Framework provides the fundamental infrastructure and components for developing visual design and modeling surfaces in Eclipse. To ensure that the developed models conform to the semantics specified in Task 2.2, we extend the generated diagrams by an additional semantics check. This will be done by translating the Object-Z specification into rules defined using the Object Constraint Language[7] (as explained in [13]) that are used to validate and check the created models at design time to ensure that the models can be mapped to the different agent-oriented programming languages (1 PM).

3.1.5 Expected Outcome

R3.1 Definition of extended PIM4Agents.

R3.2 Semantics for PIM4Agents specified in Object-Z.

R3.3 Graphical editor for PIM4Agents implemented.

WP3: Platform Independent Metamodel for Semantic Web Services

The objective of this work package is to define a metamodel for Semantic Web services (PIM4SWS) that covers the core platform independent concepts of Semantic Web services to facilitate a mapping between them and the concepts used by the currently most prominent description formats OWL-S, WSML, and SAWSDL.

3.1.6 Problem Description

Currently, the integration of services into a business process is done semi-automatically by using syntactical service descriptions (cf. Section 2.1.1). For achieving fully automated integration of a service into an executable process, these purely syntactical service descriptions are not sufficient and have to be extended by additional information for service discovery, composition and ad-hoc invocation. The existing Semantic Web service standards OWL-S, WSML and SAWSDL support these three steps in a slightly different manner. For abstracting from these differences and for supporting a model-driven integration of existing services, a more general description of Semantic Web services is necessary.

3.1.7 Solution Approach

Following the model-driven approach, a more general layer for the existing standards (OWL-S, WSML, SAWSDL) is introduced which is expressive enough to encompass these standards. This layer is formally defined by a metamodel for Semantic Web services (PIM4SWS) and supplemented by a formal semantics. For facilitating visualization and development of a PIM4SWS service description, a graphical editor is developed. For deriving an OWL-S, a WSML or a SAWSDL service description from a PIM4SWS model, transformations from PIM4SWS to these three standards have to be specified. These descriptions can then be used for service discovery and matchmaking (work package WP4).

3.1.8 Preconditions, Prerequisites

This work package has no internal preconditions.

[7] http://www.omg.org/technology/documents/formal/ocl.htm

3.1.9 Description of Work

Task 3.1 (Specification of Concepts) The concepts needed for the description of Semantic Web services will be listed and their semantics will be specified in natural language. The specification will define individual properties of the concepts as well as their relations to each other.

The concepts identified in Task 3.1 are utilized to design the metamodel for Semantic Web services (PIM4SWS) using the Eclipse Modeling Framework (3 PM).

Task 3.2 (Semantic Specification of Concepts) The semantics of the concepts described in Task 3.2 is formally defined. For this purpose, we again – as in the case of the PIM4Agents – use Object-Z and explore its adequacy with respect to defining the semantics of the PIM4SWS. However, it is an open research question how much of the semantics specified in Task 3.1 can actually be captured with Object-Z (5 PM).

Task 3.3 (Graphical Editor) Based on the Graphical Modeling Framework (GMF), we develop a graphical editor that can be used to model Semantic Web services on a more abstract level. GMF provides the fundamental infrastructure and components for developing visual design and modeling surfaces in Eclipse (3 PM).

Task 3.4 - (Model Transformations): The metamodel defined in Task 3.1 contains concepts for describing models of Semantic Web services on a more abstract level. For using these models in a matchmaker for the different standards (OWL-S, WSML, SAWSDL), a transformation from the PIM4SWS to each of these standards will be specified. We use the Atlas Transformation Language[8] to define the model to model transformation between the PIM4SWS and the metamodels of the aforementioned description formats (3 PM).

3.1.10 Expected Outcome

R3.1 PIM4SWS specified.

R3.2 Semantics for PIM4SWS specified in Object-Z.

R3.3 Graphical editor for PIM4SWS implemented.

R3.4 Transformations from PIM4SWS to OWL-S, WSML and SAWSDL specified.

WP4: Hybrid Semantic Web Service Selection

This work package is devoted to the (i) development and experimental performance analysis of a hybrid semantic matchmaker for Semantic Web services described in the standard SAWSDL and (ii) development of a platform independent, plugin-based matchmaker.

3.1.11 Problem Description

The process of semantic selection covers both semantic matching and ranking of SAWSDL services; hybrid selection combines both non-logic based and logic based semantic matching means to select a Web service that is most semantically relevant to a given query. Major problem of hybrid semantic SAWSDL service matching is that SAWSDL has no formal semantics but offers syntactic elements with which one can refer to ontologies that specify the semantics of each element of the Web service description outside the language. There are three main problems we have to solve in this work package: (i) Semantic matching of service descriptions each containing references to multiple ontologies that differ in their specification language ranging from description logics, to structured XML to unstructured text; (ii) Combination of non-logic-based and logic-based concept similarity means for hybrid semantic matching of SAWSDL services with a given query that performs reasonably well with respect to classic retrieval performance criteria; (iii) There does not exist any service retrieval test collection for Semantic Web services in SAWSDL.

[8] http://www.eclipse.org/m2m/atl/

3.1.12 Solution Approach

The above mentioned problems can be solved by exploiting appropriate means of ontology alignment in combination with text and structured XML similarity means from information retrieval. Semantic annotations in different ontology languages with formal grounding in description logics can be converted to the standardized common DIG (DL Implementation Group) format for further processing by description logic reasoners with DIG interface such as Pellet and RACER, and other concept similarity means. Development of the algorithmic matching scheme can draw from the approaches of the matchmakers OWLS-MX and WSMO-MX. Performance testing and false positives/false negatives analysis of the resulting SAWSDL-MX matchmaker can iteratively be performed over a given service retrieval test collection to provide feedback to the development of improved versions. The required test collection can be semi-automatically built from existing collections and manual extension.

Because of the limited resources in Modest, we do not develop a full-fledged matchmaker agent but a preversioned one, called MDS-MX, for the selection of Semantic Web services written in the common Semantic Web service description language, that is the platform independent metamodel language (PIM4SWS). That is, the MDS-MX still does not perform platform independent Semantic Web services matching. Instead, the MDS-MX agent first converts registered Semantic Web services and given query in PIM4SWS to those description languages accepted by the plugged in language-specific matchmakers (e.g., OWLS-MX, WSMO-MX, SAWSDL-MX). Second, it concurrently calls these matchmakers to find services that semantically best match with the given query. Finally, it returns all service candidates that are (re-)converted to the common description language PIM4SWS to the business process analyst and service-oriented architecture designer who issued the original query.

3.1.13 Preconditions, Prerequisites

The development of SAWSDL-MX and MDS-MX builds on hybrid semantic Web service matchmakers OWLS-MX and WSMO-MX we developed in the project SCALLOPS. The performance testing of the matchmaker exploits an extension of the automated evaluation environment (SME2[9]) we developed in SCALLOPS. The development of the SAWSDL service retrieval test collection exploits our conversion tool OWLS2WSDL.

3.1.14 Description of Work

Task 4.1 (Hybrid Semantic Web Service Matchmakers) Work of this task is devoted to the design and proof-of-concept implementation of a hybrid Semantic Web service matchmaker agent, called SAWSDL-MX, for Semantic Web services specified in the standard format SAWSDL, as well as a preversion of a model-driven Semantic Web service matchmaker MDS-MX that makes use of the SAWSDL-MX and other language-specific matchmakers. The SAWSDL-MX matchmaker combines non-logic-based with logic-based semantic matching means and is capable of dealing with referenced ontologies that are specified in different formats like text, XML, OWL, and description logics. The MDS-MX matchmaker accepts Semantic Web service descriptions in the common Semantic Web service description format PIM4SWS (cf. WP3), provides a conversion for PIM4SWS, and makes use of the plugged in language-specific hybrid Semantic Web service matchmakers (SAWSDL-MX, OWLS-MX WSMO-MX) to discover relevant services on the more abstract level of description (PIM4SWS). (5 PM)

Task 4.2 (Experimental Performance Evaluation) Work of this task is devoted to the evaluation of the performance of the all new SAWSDL-MX matchmakers developed in Task 4.1. For this purpose, we build an initial test collection SAWSDL-TC1 from scratch, and extend the existing collections OWLS-TC2. The performance evaluation of SAWSDL-MX is conducted with the respectively extended SME2 evaluation environment over SAWSDL-TC1 by measuring the recall/precision, F1 (i.e. a combination of recall and precision), accuracy, average query response time, and resource consumption of the matchmaker. It is further planned to organize a follow-up event of the first S3 (Semantic Web Service Selection) contest for OWL-S service matchmakers with focus on SAWSDL service matchmaking (cf. www-ags.dfki.uni-sb.de/˜klusch/s3/), if possible (that is, if SAWSDL

[9] Software tool for measuring the performance of service retrieval tools (including matchmakers) in terms of precision/recall, average query response time, accuracy, etc.; performance is measured with respect to a given service retrieval test collection

matchmakers other than the SAWSDL-MX are publicly available for participating in the event). For reasons of resource limitation, the initial version of the model-driven Semantic Web service matchmaker MDS-MX will not be evaluated in this project. (5 PM)

3.1.15 Dependencies

WP3 The work of Task 4.1 and 4.2 partially depends on the results achieved in Task 3.1 and 3.4: The development of MDS-MX in 4.1 and the building of the test collection in 4.2 is influenced by the specification of the metamodel for Semantic Web services (PIM4SWS) and the corresponding transformations to SAWSDL.

WP5 The work of Task 4.1 affects the work of Task 5.2: The matchmaker SAWSDL-MX will be demonstrated in the MODEST demonstrator D2 (eHealthcare pilot).

3.1.16 Expected Outcome

R4.1 Prototypical implementation of a hybrid semantic SAWSDL service matchmaker (SAWSDL-MX) and an initial version of a model-driven Semantic Web service matchmaker (MDS-MX).

R4.2 Initial SAWSDL retrieval test collection (SAWSDL-TC1) for testing the performance of SAWSDL service matchmakers. Performance evaluation environment (SME2) for automated testing of SAWSDL service matchmakers for given test collection. Experimental results of testing the performance of the SAWSDL-MX matchmaker over SAWSDL-TC1 using the environment SME2. Scientific publication(s) on the results achieved in this work package.

WP5: Modest Demonstrator

The practical usefulness of the concepts developed in Modest will be demonstrated in two applications: Industrial steel production (Modest demonstrator D1) and eHealthcare (Modest demonstrator D2).

3.1.17 Preconditions, Prerequisites

Modest **Demonstrator D1:** In a close cooperation with Saarstahl AG a shop floor control system for steel production was developed for the steel work in Völklingen. Building on the experience in the development of this system, it is planned to support Saarstahl AG in the development of a system for the complete supply chain from hot metal supply over steel production to the production of customer products at the mesticator sites.

Modest **Demonstrator D2:** In the BMB+F project Scallops a prototypical application for emergency medical assistance was developed on the basis of Semantic Web services. In this work OWL-S was used to describe the services. This implementation will be the starting point for Modest. However, in contrast to Scallops where the coordination of services was done from the point of view of a mobile user, the idea is now to take the point of a view of a locally fixed centralized coordination point.

3.1.18 Description of Work

Task 5.1 (Saarstahl Pilot) The Saarstahl case is a proof of concept for designing the main process within the supply chain based on the results of Modest. Several challenges have to be addressed when it comes to a service-oriented design of the complete supply chain. From the viewpoint of Saarstahl it is fundamental that (i) business requirements that are specified by Saarstahl can easily be translated into a running system and (ii) existing systems (e.g. data bases) can be re-used within the SOA to maintain the high product quality. Both requirements match nicely with our approach.

The Saarstahl pilot will consist of the following steps: (i) Specification of models and requirements formulated with the PIM4Agents' graphical editor, (ii) matchmaking on the PIM-level using plugin matchmaker in accordance to the specified service requirements (defined with the PIM4SWS's graphical editor), and (ii) applying model transformations on the agent models to generate code that will invoke the Semantic Web services. (4 PM)

Task 5.2 (eHealthcare Pilot) The results of Modest will be demonstrated for the emergency medical assistance use case of the Health-Scallops demonstrator which originally was implemented in the predecessor project Scallops by means of Semantic Web service technology. In opposite to Scallops, in Modest, we are now taking the rather static business perspective of the central coordination entity on this use case by presuming a fixed business process workflow for which appropriate existing Web services have to be discovered yet.

The Modest eHealth pilot will consist of the following steps: (i) Transforming the OWL-S services of the Health-Scallops use case into both SAWSDL and the new Semantic Web service metamodel format (partial support by the PIM4SWS's graphical editor), (ii) focusing the overall Health-Scallops use case scenario into task oriented business process workflow patterns, and (iii) using the new hybrid matchmaker SAWSDL-MX to assign relevant services to individual use case workflow patterns on demand. (3 PM)

3.2 Timetable and Milestones

Figure 1 shows the timetable for all five work packages of the Modest project.

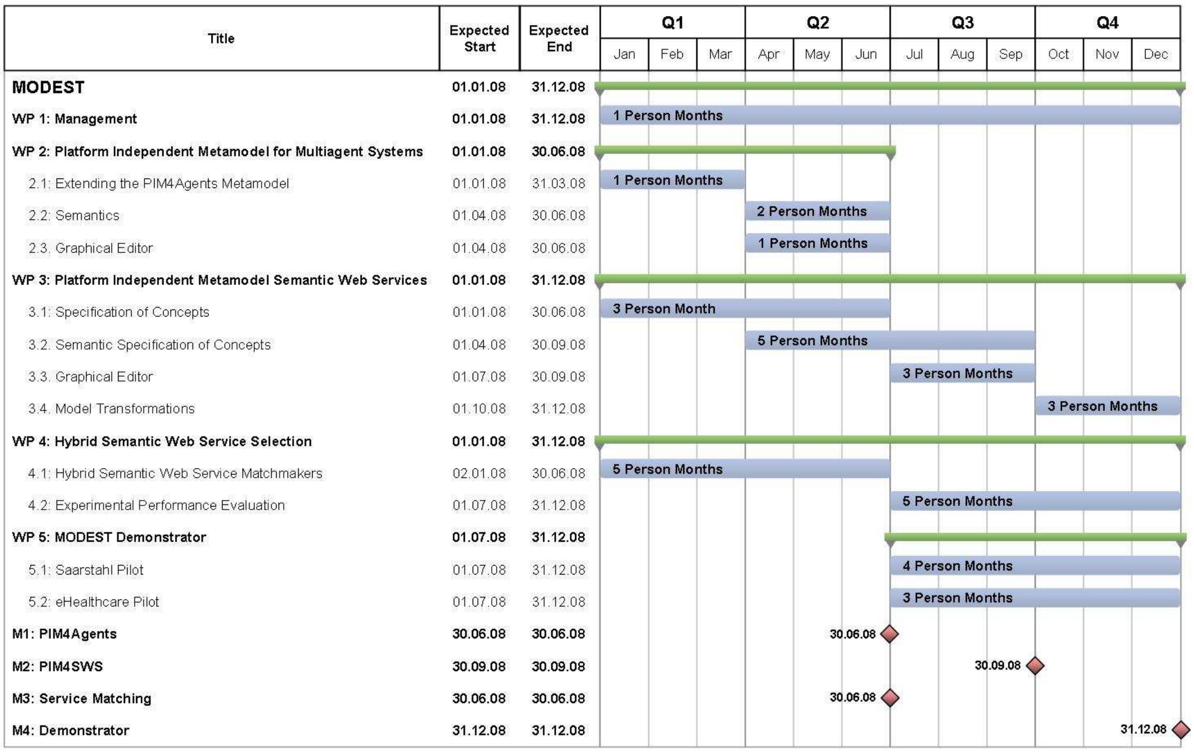

Figure 1: The Gantt Chart of the Modest project. The numbers in the bars show the effort in person months. The effort is spent in the time interval that is marked by the respective bar.

Milestones

The following milestones are defined for Modest:

M1: PIM4Agents (June 2008) Modified PIM4Agents including semantic specification and graphical editor is available (WP2).

M2: PIM4SWS (September 2008) The metamodel for Semantic Web services including semantic definition and graphical editor is available (WP3).

M3: Service matching (June 2008) Hybrid semantic matchmaker for SAWSDL Web services and initial version of a model-driven Semantic Web service matchmaker are available (WP4).

M4: Demonstrator (December 2008) Demonstrator for Saarstahl and eHealthCare pilot is available (WP5).

Collaboration with other Projects and Scientists

Joint research and development within this project is carried out with the following not funded collaborators.

Saarstahl AG in Völklingen, Saarland: Joint work on the development of the MODEST demonstrator "Saarstahl Services Pilot" in work package WP5. The collaboration is executed through advisoral input (in WP3 and WP5) for practical business process modeling and the engineering of Semantic Web services for the selected use case. Collaborators: Walter Pressmann (Saarstahl AG), Dr. Fischer (DFKI)

Swinburne University in Melbourne, Australia: Partial joint work on the design of Semantic Web service matchmaker for the standard SAWSDL (work package WP4). Collaborators: Prof. Kowalczyk (Swinburne U/CITR), Dr. Klusch (DFKI)

Universidad Rey Juan Carlos in Madrid, Spain: Partial joint work on testing the performance of developed semantic service matchmaker agent in the selected eHealth use case (work packages WP4 and WP5). Collaborators: Prof. Ossowski (URJC), Dr. Klusch (DFKI)

The inter-project links of MODEST are as follows.

SHAPE (Semantically-enabled Heterogeneous Service Architectures): SHAPE is an European FP7 funded research project that investigates a new standard metamodel for semantically-enabled heterogeneous service architectures. Contact to SHAPE: Dr. Fischer (DFKI)

COIN (COllaboration and INteroperability for networked enterprises): COIN is an European FP7 funded integrated project. The main tasks to work on in COIN are the integration of agents into the newly developed open source service platform and the design of innovative services for negotiation models in a business context for small and medium enterprises. Contact to COIN: Dr. Fischer (DFKI)

4 Resources

The MODEST project is planned for a period of one year.
Personnel: 1 senior researcher, 2 researchers, 4 research assistants (18 hrs/week), 0.5 secretary.
Hardware Equipment: sufficient support to provide an efficient working environment for the staff of MODEST.

References

[1] F. Bellifemine, F. Bergenti, G. Caire, and A. Poggi. JADE - a java agent development framework. In R. Bordini, M. Dastani, D. J., and A. El Fallah Seghrouchni, editors, *Multi-Agent Programming: Languages, Platforms and Applications*, volume 15 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*, pages 125–147, Berlin et al., 2005. Springer.

[2] K. Fischer, C. Hahn, and C. Madrigal-Mora. Agent-oriented software engineering: a model-driven approach. *International Journal of Agent-Oriented Software Engineering*, 1(3/4), 2007.

[3] R. Grønmo, M. C. Jaeger, and H. Hoff. Transformations between UML and OWL-S. In *ECMDA-FA*, pages 269–283, 2005.

[4] C. Hahn. A platform independent agent-based modeling language. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2008. (accepted as full paper).

[5] C. Hahn, C. Madrigal-Mora, K. Fischer, B. Elvesæter, A.-J. Berre, and I. Zinnikus. Meta-models, Models, and Model Transformations: Towards Interoperable Agents. In *Proceedings of the 4th German Conference on Multiagent System Technologies*, volume 4196 of *Lecture Notes in Computer Science*, Berlin/Heidelberg, 2006. Springer.

[6] G. Kardas, A. Goknil, O. Dikenelli, and N. Y. Topaloglu. Model transformation for model driven development of semantic web enabled multi-agent systems. In P. Petta, J. P. Müller, M. Klusch, and M. P. Georgeff, editors, *MATES*, volume 4687 of *Lecture Notes in Computer Science*, pages 13–24. Springer, 2007.

[7] F. Kaufer and M. Klusch. Wsmo-mx: A logic programming based hybrid service matchmaker. In *Proceedings of the 4th IEEE European Conference on Web Services (ECOWS 2006), IEEE CS Press, Zurich, Switzerland*, 2006.

[8] M. Klusch. Semantic web service coordination. In *M. Schumacher, H. Helin, H. Schuldt (Eds.): CASCOM - Intelligent Service Coordination in the Semantic Web. Chapter 4. Birkhäuser Verlag, Springer*, 2008.

[9] M. Klusch, B. Fries, and K. Sycara. Automated semantic web service discovery with owls-mx. In *Proceedings of 5th International Conference on Autonomous Agents and multiagent Systems AAMAS*, Hakodate, Japan, 2006.

[10] X. T. Nguyen and R. Kowalczyk. Ws2jade: Integrating web service with Jade agents. In *Proceedings of Service-Oriented Computing: Agents, Semantics, and Engineering, AAMAS 2007 International Workshop, SOCASE 2007*, pages 147–159, 2007.

[11] L. Padgham and W. Liu. Internet collaboration and service composition as a loose form of teamwork. *Journal of Network and Computer Applications, Special edition on Teamwork*, 30(3):1116–1135, August 2007.

[12] M. Papasimeon and C. Heinze. Extending the UML for designing JACK agents. In *Proceedings of the Australian Software Engineering Conference (ASWEC 01)*, 2001.

[13] D. Roe, K. Broda, and A. Russo. Mapping UML models incorporating OCL constraints into Object-Z. Technical Report 2003/9, Imperial College, 180 Queen's Gate, London, 2002.

[14] B. T. R. Savarimuthu, M. Purvis, M. Purvis, and S. Cranefield. Agent-based integration of web services with workflow management systems. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1345–1346, New York, NY, USA, 2005. ACM.

[15] D. Skogan, R. Grønmo, and I. Solheim. Web service composition in UML. In *EDOC*, pages 47–57, 2004.

[16] G. Smith. *The Object-Z Specification Language. Advances in Formal Methods.* Kluwer Academic Publishers, 2000.

[17] G. Smith and I. J. Hayes. Structuring real-time Object-Z specifications. In *Proceedings of the Second International Conference on Integrated Formal Methods*, volume 1945 of *Lecture Notes In Computer Science*, pages 97–115, London, 2000. Springer.

[18] I. Zinnikus, C. Hahn, and K. Fischer. A model-driven agent-based approach for the integration of services into a collaborative business process. In *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, May 2008. (accepted as full paper).

[19] I. Zinnikus, C. Hahn, M. Klein, and K. Fischer. An agent-based, model-driven approach for enabling interoperability in the area of multi-brand vehicle configuration. In *Proc. of the Fifth International Conference on Service-Oriented Computing (ICSOC)*, volume 4749 of *LNCS*, pages 330–341. Springer Verlag, 2007.

A Glossary

BDI: Belief, Desire, Intention

DL: Description Logic (http://dl.kr.org)

GMF: Graphical Modeling Framework (http://www.eclipse.org/modeling/gmf/)

MAS: Multiagent Systems

MDS-MX: Model-driven Semantic Web service Matchmaker

OWL-S: Ontology Web Language for Services (http://www.daml.org/services/owl-s/1.1/)

OWLS-MX: Hybrid semantic matchmaker for services written in OWL-S (http://projects.semwebcentral.org/projects/owls-mx/)

OWLS-TC2: OWL-S service retrieval test collection, version 2 (http://projects.semwebcentral.org/projects/owls-tc/)

PIM: Platform-Independent Model

PIM4Agents: Platform Independent Metamodel for Multiagent Systems

PIM4SOA: Platform Independent Metamodel for Service-oriented Architectures

PIM4SWS: Platform Independent Metamodel for Semantic Web Services

SAWSDL: Semantically Annotated WSDL and XML Schemas; Standard description language for semantic Web services, since August 2007 (http://www.w3.org/TR/sawsdl/)

SAWSDL-MX: Hybrid semantic matchmaker for services written in SAWSDL

SAWSDL-TC1: SAWSDL service retrieval test collection, version 1

SME2: Semantic Service MatchMaker Evaluation Environment

SOA: Service-Oriented Architecture

SWS: Semantic Web Services

WSDL: Web Services Description Language (http://www.w3.org/TR/wsdl)

WSML: Web Service Modeling Language (http://www.wsmo.org/wsml/wsml-syntax)

WSMO: Web Service Modeling Ontology (http://www.wsmo.org/)

WSMO-MX: Hybrid semantic matchmaker for services written in WSML (http://projects.semwebcentral.org/projects/wsmomx/)