

APPLICATION OF THE FUSION APPROACH FOR ASSISTED COMPOSITION OF WEB SERVICES

Spiros Alexakis, Markus Bauer, Albina Pace, Alexa Schumacher
CAS Software AG, Wilhelm-Schickard-Str. 10-12, 76131 Karlsruhe, GERMANY
spiros.alexakis@cas.de markus.bauer@cas.de albina.pace@cas.de
alex.schumacher@cas.de

Andreas Friesen
SAP Research, CEC Karlsruhe, Vincenz-Prießnitz-Str. 1, 76131 Karlsruhe, GERMANY
andreas.friesen@sap.com

Athanasios Bouras
Institute of Communication and Computer Systems (ICCS),
9, Iroon Polytechniou str., Zografou Campus, 15780 Athens, GREECE
bouras@mail.ntua.gr

Dimitrios Kourtesis
South East European Research Centre (SEERC),
17 Mitropoleos Str., 54624 Thessaloniki, GREECE
dkourtesis@seerc.org

The FUSION approach proposes both a conceptual framework and a system architecture that supports the composition of business processes using semantically annotated web services as building blocks. Results will be validated by supporting collaborative commercial proof-of-concept pilots. The FUSION approach will facilitate trans-national pilot cases having operations spanning the enlarged Europe, in particular: integration of transactions of a franchising firm, provision of career and human resource management services, collaboration of companies in a chain of schools of foreign languages. The paper provides an overview on the FUSION approach and illustrates how it can be applied on one of the pilot cases.

1 INTRODUCTION

Service Oriented Architecture allows systems to be implemented using a wide range of technologies. SOA systems are defined as a set of loosely coupled services. In order to interoperate, services are described using formal definitions such as WSDL. High level languages such as BPEL allow us to define the orchestration for the fine grained services exposed by different systems which then can be incorporated into workflows and business processes implemented in composite applications.

Together these technologies lay the grounds for Enterprise Application Integration; however for systems to be interoperable, inconsistencies at the data and functional level need to be overcome.

FUSION addresses these interoperability issues by proposing a conceptual framework and a system architecture that supports semantically enhanced, reusable business processes through the use of semantic annotations of Web services.

2 FUSION APPROACH OVERVIEW

In order to facilitate the resolution of structural and semantic differences of the input and output messages exchanged between interoperable Web services of a defined process, an Enterprise Integration Ontology (ENIO) is introduced. ENIO is a multi-layered and multi-faceted Ontology. Layering defines the level of abstraction and the level of exposure of the Ontology whereas the facets represent the role of the concepts within the solution. The ontology provides a common reference for data semantics through the data facet; enables search and discovery of Web Services through the functional facet and enables process composition via the process facet. The ontology also introduces an upper layer covering domain independent concepts. The domain dependent extensions are then expressed in the facets (Friesen, 2007).

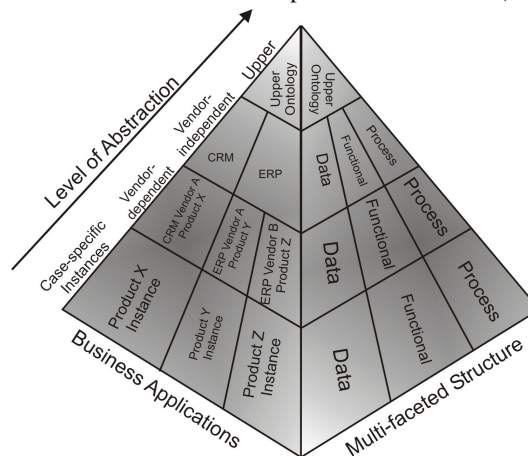


Figure 1 – ENIO Ontology Overview (Bouras, 2006)

2.1 The architecture

The FUSION architecture is made up of the following components:

Semantic Analyzer - Concepts Designer

The Concepts Designer is responsible for handling and managing the multi-layered, multifaceted FUSION Ontology during each phase of the ontology lifecycle.

Semantic Analyzer - Semantic Profiler

The Semantic Profiler is a graphical editor for adding annotations utilizing classes and instances of all the developed facets of the FUSION Ontology through SA-WSDL. It also generates XSLT transformations for up and down-casting functionality, i.e. it provides a mapping from concrete system dependent data types into ontology concepts and vice versa.

Semantic Analyzer - Process Designer

The Process Designer enables the user to reuse Abstract Process Models, discover candidate services automatically, check for and resolve data incompatibilities and

finally, ground the Abstract Process Models to executable BPEL process code using the discovered services.

Ontology Repository

The Ontology Repository module constitutes a fully functional file system, where concepts, classes and instances of the facets of the FUSION Ontology are stored.

Integration Mechanism

The integration mechanism is the execution environment of the FUSION system.

3 PILOT USE CASES

The FUSION approach will be validated in the frame of collaborative commerce proof-of-concept pilots. Each pilot has operations spanning the Enlarged Europe, in particular:

- Germanos: integration of transactions of a franchising firm (Greece, Poland, Romania, Bulgaria, Ukraine, Cyprus and FYROM)
- Interjob: provision of career and human resource management services (Hungary and Germany)
- Pharos: collaboration of companies in a chain of schools (Bulgaria, FYROM, Albania).

These pilots represent different interoperability aspects. While in the Pharos use case all systems have been individually tailored, Germanos only deploys standard software. Processes at Germanos are already automated while the processes of Pharos are performed by mail, phone or fax. Other aspects of complementarity are complexity and intercultural focus. Germanos processes are rather complex, while the exchange of HR information in the Interjob case needs harmonisation due to intercultural differences. In the following, the Pharos case will be explained in more detail.

4 COLLABORATIVE PROCESS BEFORE AND AFTER: PHAROS EXAMPLE

4.1 Scenario

The IT infrastructure of the Pharos network is not homogeneous. Schools in Bulgaria, FYROM and Albania use different variants of *F97*, a custom-built system offering certain core CRM and ERP business functions, customised to support the specific needs of Schools in different countries. On the other hand, the Regional Directorate at Sofia and all Country Headquarters rely on *MIS*, an information management and decision-support system customised specifically for the needs of the Pharos network.

4.2 Demonstration process: student transfer to another school

For the purpose of demonstration of the FUSION approach we have selected the process of a student transfer between schools, as it involves software systems of two schools and one headquarter with bidirectional information exchange.

Before FUSION

A student can be transferred to another school, for example if he moves or if he wishes to visit a different course. In order to change from the old school to the new school, he has to contact the Country Headquarters to initiate the transfer. The Country Headquarter then has to make sure that:

- the student's school and course specific data is retrieved from the old school and forwarded to the new school
- the student gets properly registered at the new school (provided there are still free places available in the courses the student wants to attend)
- the Country Headquarters internal bookkeeping is updated.

The information exchange necessary for the execution of the steps above is performed by phone, fax or email. System records are edited and updated by hand.

With FUSION:

To improve this situation, an automated process has to be introduced. Its building blocks are Web Services that will be introduced at the schools F97 systems and at the Country HQ MIS. The process consists of the following steps:

1. The user issues a transfer student request, specifying the student's global Customer ID and the IDs of the old school and the new school.
2. Using the Customer ID, the Student's Customer Record is provided by the Country HQ MIS.
3. From the Customer Record, the student's school-specific Student ID (as given and maintained by the old school) is extracted. Using this ID, the school-specific registration data is retrieved.
4. The Student's Registration is then forwarded to the new school. The new school may reject the student, e.g. if there are no places available.
5. If the student is accepted the Country HQ's Customer Record for the student is updated with the new registration information received from the new school.
6. As a last step, the student is unregistered from the old school and the process terminates successfully.

In the next section, we show how Pharos' systems can be prepared to run this reworked process and how the FUSION approach simplifies its implementation.

5 FUSION APPROACH APPLIED TO PHAROS

The FUSION System Lifecycle consists of the following phases: (The process execution is beyond our scope, as it mainly consists of the execution of standard BPEL code.)

5.1 Web service enablement of the involved applications

In FUSION, an enterprise application is called Web Service Enabled, if the relevant functionality is exposed in the form of industry standard web services. FUSION restricts the concept of Web Services to those services that have their interfaces described in WSDL and use SOAP-formatted XML messages (Mitra, 2003).

In the Pharos use case both involved legacy systems do not expose web services. After the identification of relevant business functionality, there are two possibilities for exposing web services. The first approach includes the restructuring of the corresponding source code parts in order to make them run and deploy as Web Servers or as components within Web servers or Web Application containers. Alternatively, adapter components for the legacy functions have to be created.

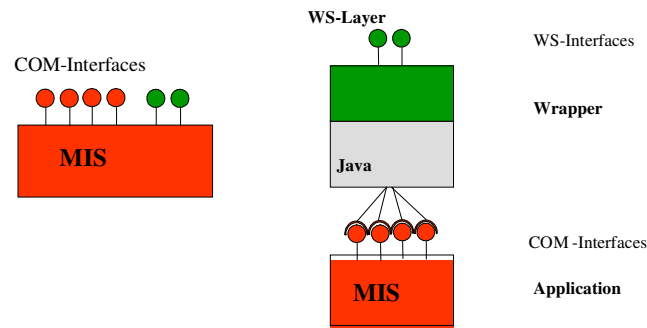


Figure 2 - Possible migration strategies for the Pharos use case

The above figure depicts both solutions: on the left a Delphi server component has been enhanced with additional Web Service interfaces, on the right part a Web Service layer in Java communicates with the unchanged server component of MIS through RPCs (remote procedure calls) via DCOM.

For Pharos, the first option has been chosen, as both systems involved have been programmed in Delphi 6. The toolkit from INDY (www.indyproject.org) provides the technical basis for publishing SOAP based Web Services from Delphi Code. Web Services to be published include:

1. *transferStudent(CustomerID, NewSchoolID)*
2. *getCustomerRecord(CustomerID)*
3. *getStudentRegistration(StudentID, SchoolID)*
4. *registerStudent(studentRegistration, NewSchoolID)*

5.2 Ontology Engineering phase

The Ontology Engineering phase comprises all activities needed for customising the shared semantic model that constitutes the cornerstone of a FUSION EAI solution; the FUSION EAI Ontology. The software tool supporting the user's activities within this procedure is the FUSION Concepts Designer, a visual tool facilitating the creation, extension, customisation, and maintenance of multi-layered and multi-faceted ontologies. In the concrete case the following steps have to be followed:

1. Define a physical location (Web Server Directory with read/write capabilities) in order to deploy the Ontology needed for Pharos, for instance *<http://extranet.pharos.bg/ontologies/PharosOntology.owl>*
2. For the Functional Facet, examine the operations in the Enterprise Service WSDL file, in order to determine what kind of functional semantics are needed. In the PHAROS case the *setStudentRegistration operation* raises the need for adding a new concept *registerStudent* under *FunctionalFacetEntity*
3. For each Functional Facet concept that is created we must also specify the Input and Output Data Facet Concepts that it relates to.
4. If the Data Facet entity does not exist then it must be created. In the Pharos case *registerStudent* requires as input a *#Student* class. A Student can be considered as a subclass of FUSION's standard concept *#Person* but additional data types and object properties must be added (e.g. *hasGrade*)

5.3 Semantic Uplifting phase

In this phase Web Service descriptions are lifted from the syntactic, to the semantic level, through references to the common semantic model, the FUSION EAI Ontology. The phase consists of the following steps:

1. For each `wsdl:portType` or `wsdl:operation`: annotate with the respective functional facet concept, best describing the function being performed


```
<wsdl:operation name="inputSetStudentRegistration">
  <wsdl:input name="inputSetStudentRegistrationRequest"
    message="impl:SetStudentRegistration"/>
  <sawSDL:attrExtensions
    sawSDL:modelReference="http://extranet.pharos.bg/ontologies/
PharosOntology.owl#registerStudent"/>
</wsdl:operation>
```
2. For each `wsdl:part`: annotate with the respective data concept, best describing the data being transported by the message part
3. For each `wsdl:part`: define XSLT transformations for translating the data from XSD to the OWL data concept in the previous step, and vice versa
4. Provide the location of the target mediator Web Server, and any other information required for generating and deploying the mediator service
5. Publish the Semantic Profile to the FUSION Integration Mechanism

5.4 Process Design phase

During the process design phase, EAI scenarios involving several “semantically uplifted” business applications are realised. This is supported by the Process Designer which allows instantiating the Abstract Process Models by discovering appropriate services and generates a grounded BPEL4WS model, to be deployed for execution. For the Pharos example in manual composition the phase consists of the following steps:

- A process model is created in abstract BPEL (using a BPEL editor) and annotated (via comments) with concepts from the functional facet of the ontology. The snippet below shows an invoke activity in BPEL which is annotated with a comment containing a link to the concept *GetCustomerRecord*.
- `<!-- function="GetCustomerRecord"`

- `concept="http://.../FusionOntology#GetCustomerRecord" -->`
- `<invoke inputVariable="CustomerID" name="GetCustomerRecord"`
- `outputVariable="CustomerRecord"`
- `partnerLink="HeadQuarters"/>`
- This process model is uploaded to a repository and an instance is created in the process facet of the ontology, with a link to the process model in the repository. This process model may now be reused by other Business Consultants by:
- Browsing through the ontology process facet for the best suited process model. In this case, under the category *Inbound_Outbound_Logistics*, the process model *#Student_School_Transfer_CBPManual*, which has a property *hasProcessModel* containing a link to the BPEL file <http://extranet.pharos.bg/processes/StudentTransfer.bpel>
- Loading the BPEL file in the process model customizer, which allows the user to view the process graphically, remove optional tasks and invoke discovery, grounding and deployment.

6 RELATED WORK

FUSION aims at simplifying Web service composition with the use of abstract process templates and improves Web service discovery by taking functional semantics into consideration. The idea of replacing abstract functions by executable Web services during runtime was proposed in (Mueller, 2004). FUSION implements this idea by adding semantic annotations to Web services for discovery.

The METEOR-S framework (Verma, 2005) makes use of semantics to describe functional and non functional capabilities of Web services and allows binding of Web services to abstract processes. Data heterogeneities are resolved with the use of proxies. FUSION employs a similar approach, however introduces the concept of optional functions so that abstract processes are customisable and hence reusable. The FUSION ontology also allows for annotation of services such that they may be composed automatically (Friesen, 2007) using planning techniques.

Other research in dynamic workflows such as (Davulcu, 1999) and (Mueller, 2004) are based on homogenous environments and require no mediation amongst services.

7 SUMMARY AND BENEFITS

FUSION addresses the need of many enterprises to implement business processes that involve a number of business partners and information systems. Current software architecture trends (SOA, Web Services and BPEL) support the implementation of these business processes, as they provide a technological solution for the communication between independent systems. However, implementing business processes on the basis of a number of different underlying information systems – even if these expose suitable Web Services – is still a costly and error prone task.

FUSION addresses these issues by defining methodology, architecture and tools that extend the concept of SOA by utilizing Semantic Web Service technologies.

The applicability of the FUSION Approach is demonstrated in the frame of pilots from different application domains, involving organisations of different size and structure. Each pilot benefits by the introduction of automated business processes, which can be cost-effectively implemented by the FUSION EAI Ontology.

Acknowledgements

This paper is supported by the **FUSION FP6-027385 Project**.

8 REFERENCES

1. Friesen A, Alazeib A, Balogh A, Bauer M, Bouras A, Gouvas P, Mentzas G, Pace A. Towards semantically-assisted design of collaborative business processes in EAI scenarios, INDIN, 2007.
2. FUSION Consortium. FUSION Deliverable 6.1 -- FUSION Initial Exploitation Plan, 2006.
3. Meyer B. Object-Oriented Software Construction, 2nd ed., Prentice-Hall, 1997.
4. Mitra N. SOAP -- W3C Recommendation, June 2003. <http://www.w3.org/TR/soap>
5. K. Verma, K. Gomadam, A. P. Sheth, J. Miller, Z. Wu, The METEOR-S Approach for Configuring and Executing Dynamic Web Processes, 2005
6. R. Mueller, U. Greiner, E. Rahm, Agentwork: A Workflow System Supporting Rule-Based Workflow Adaptation, 2004
7. H. Davulcu, M. Kifer, L. Pokorny, C. R. Ramakrishnan, I. V. Ramakrishnan, S. Dawson, Modelling and Analysis of Interactions in Virtual Enterprises, 1999
8. A. Bouras, P. Gouvas, K. Kalaboukas, G. Mentzas, A Semantic Service Oriented Infrastructure for EAI, 2006