APPLICATION OF THE FUSION APPROACH
FOR TOOL ASSISTED COMPOSITION OF WEB SERVICES IN CROSS ORGANISATIONAL ENVIRONMENTS

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

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

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

Kourtesis Dimitrios
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 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 in 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 FUSION Semantic Analyzer, a design time environment,
- the FUSION Ontology Repository
- the FUSION Integration Mechanism, a runtime execution environment

The sub components involved are described in the following sections.

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. The tool supports and facilitates the creation, development and maintenance the FUSION Ontology, which constitutes a well-defined representation and concept model of business applications and services and is stored in the Ontology Repository.

Semantic Analyzer - Semantic Profiler
Semantic Profiler is a graphical environment that allows the user to create the semantic profiles of the available Web Services that represent business services. It is an editor for adding annotations utilizing classes and instances of all the developed facets of the FUSION Ontology through SAWSDL (Semantically Annotated Web Service Description Language). 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, search and discover candidate services automatically, check for and resolve data incompatibilities and, finally, ground the Abstract Process Models to executable BPEL process code using the previously discovered services.

Two tools will assist manual process design and semi-automatic process design. The semantical manual process designer allows for customising an Abstract Process Model (abstract BPEL), invoke discovery and grounding and deploy it on a BPEL engine. The semi-automatic designer generates the orchestration out of a list of Web services and a composition goal, consisting of the primary goal and the recovery goal. From these a composition plan is created, containing the control and data flow of the process. This composition is then transformed to executable BPEL.

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. It supports registration and publication of semantically annotated Web Services, semantically-enriched categorization, search in the services registry and the deployment and the execution of the grounded business process models. The integration mechanism includes a) a Semantic Registry, b) a native BPEL engine and c) a generic deployment and administration component (FUSION, 2006).
2.2 System Users in the FUSION Lifecycle

The users involved in the FUSION System lifecycle phases are the following:
- an IT Consultant responsible for extending the ontology with necessary concepts and annotating the Web services with ENIO concepts.
- a Business Consultant (Business Process expert) for creating generic processes that can be customized for more specific installations.

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

Germanos is a franchising company operating a chain of retail stores for telecommunication goods and services. In each country the company has established one national headquarters to coordinate the logistics and retail activity at the national level. Their IT landscape consists of different IT systems, as for instance WMS, ERP, CRM, delivered from different vendors and on different platforms.

Interjob is a network of SMEs active in the provision of executive search services. The members of the network are acting as typical human resource (HR) consultants in their regional markets. For international requests, i.e. recruiting skilled engineers from new member states for vacancies in old member states, they co-operate acting like a single entity. Each network member is independently operating CRM and HR systems that suit his needs.

Pharos is a chain of schools of Foreign Languages and Computing. From its headquarters in Sofia the network’s Regional Directorate monitors and coordinates all activities. In each country, the Country Headquarters supervise and coordinate educational and business activities of the chain of schools at national level.

![Figure 2 – FUSION Pilots: Tailoring degree vs. process complexity](image-url)
The pilots represent different interoperability aspects, in order to be typical for European SMEs. While in the Pharos use case all systems have been individually tailored, Germanos deploys only standard software. At the same time, processes of 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 elaborated in more detail.

4. COLLABORATIVE PROCESS BEFORE AND AFTER: PHAROS EXAMPLE

Pharos is the biggest chain of schools of Foreign Languages and Computing in the SEE region. From its headquarters in Sofia, Bulgaria, the network’s Regional Directorate monitors and coordinates the activities of ca. 40 schools and Country Headquarters in Bulgaria, Albania, FYROM and recently in Ukraine.

In each country the Country Headquarters supervise and coordinate educational and business activities of the chain of schools at national level.

The Regional Headquarters administer the whole Pharos network and are responsible for its development on national and international level. They coordinate pricing policy and logistics for and plan marketing activities for each country, set the educational plan and guidelines (course syllabus, didactic method etc.) and control quality of services in each country.

The Regional Directorate at Sofia (which operates as County Headquarter, too) supervises all financial transactions at Schools and Country Headquarters, and monitors the quality of the educational services offered. Monitoring is achieved through periodic or ad-hoc reporting procedures that are requested and initiated by the Regional Directorate. Country Headquarters collect and forward the reporting data from individual schools.

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.

F97 systems deal with operational data related to customers (students), financial transactions, and school resources, while the MIS system deals with analytical data needed to support decision making, financial activities and quality assurance. Operational data of interest could be student personal data, student administrative data or student financial data. School administrative data provide input for further statistical analysis by MIS.
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 Headquarter 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 Headquarter’s 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’ 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 are 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 three core and two supporting phases:

1. **Web Service Enablement and System Installation**: expose the functionality required for implementation of the business processes as Web Services
2. **Ontology Engineering Phase**: extensions and instantiations to customise the FUSION EAI Ontology
3. **Semantic Uplifting Phase**: semantic annotation of Enterprise Services and publication of Semantic Profiles

4. **Process Design Phase**: manual or semi-automatic composition of Enterprise Services into Business Processes

5. **Process Execution Phase**: execution of EAI Business Processes.

In the following, the application of the first four phases is demonstrated. In this paper, we focus on a manual process composition. 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). They are supposed to form coarse grained Enterprise Services which may be used in different business process scenarios with a high degree of reusability. The following rules define a certain degree of self-containment for Enterprise Web Services:

1. The operations of an Enterprise Web Service can be expressed in the following form: $op_i(P_i) : R_i$, $P_i$ denotes the input message (request) type for the operation $op_i$, $R_i$ the output message (response) type.\(^1\) The public interface $I$ comprises the set of all operations $op_i$.

2. The Web Service’s implementation has to guarantee a certain postcondition for each operation upon responding with the output message, if a corresponding precondition has been satisfied upon receiving the input message. This principle is called Programming by Contract and is introduced in (Meyer, 1997).

3. If the precondition of an operation requires some specific internal state of the Web Service’s implementation, that state must be always reachable by executing one or more operations from $I$.

The third part of the rule ensures that operations are either atomic or, in case of complex operations, protocol dependencies (execution order dependencies) between operations are kept within the same Web Service interface $I$.

In other words, all operations related to a business process step, should be aggregated in one single Web Service interface. All operations required to execute and influence a coherent Private Process in the Service’s implementation should be grouped together in this manner.

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.

---

\(^1\) Note that $P_i$ and $R_i$ can contain an arbitrary number of message parts. Therefore, a Web Service operation can accept an arbitrary number of input parameters and return an arbitrary number of output parameters.
These adapters are components that run in a Web server or Web Application container, provide the Web service interface and delegate all Web Service requests to the legacy components. The following figure depicts both solutions: in the left a Delphi server component has been enhanced with additional Web Service interfaces, in 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](http://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 we 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, Output and Fault Data Facet Concepts that it relates to.

4. If the Data Facet does not entity exist then it must be created. In the Pharos case `registerStudent` requires as an input a `#Student` class. A Student can be considered as a subclass of FUSION’s standard concept `#Person` but additional data type 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. This entails the creation and publication of Semantic Profiles (annotations of a service’s operations, inputs/outputs, and internal behaviour), the creation of transformation rules for resolving message heterogeneities during process execution, and the generation and deployment of mediator services to perform the actual transformations. This procedure is supported by the FUSION Semantic Profiler, a tool facilitating service annotation with respect to the Functional, Data, and Process Facets defined in 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

   `<wsdl:message name="SetStudentRegistration">
   <wsdl:part element="impl:inputSetStudentRegistration" name="parameters"/>
   <sawsdl:attrExtensions
   sawsdl:modelReference="http://extranet.pharos.bg/ontologies/
   PharosOntology.owl#StudentRegistration"/>
   </wsdl:message>`

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

   `<wsdl:message name="SetStudentRegistration">
   <wsdl:part element="impl:inputSetStudentRegistration" name="parameters"/>
   <sawsdl:attrExtensions
   sawsdl:modelReference="http://extranet.pharos.bg/ontologies/
   PharosOntology.owl#StudentRegistration"/>
   RegistrationRecord2Ont.xslt
   sawsdl:loweringSchemaMapping=http://extranet.pharos.bg/ontologies/
   Ont2RegistrationRecord.xslt />
   </wsdl:message>`

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"/>
```

Optional invoke activities can be annotated further with “optional=true” in the comment tag.

- 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. The OWL snippet below shows the process model entry in the ontology.

```
<StudentTransferCBPmanual rdf:ID="Student_School_TransferCBPmanual"
    <hasProcessModel rdf:datatype="&xsd;string">
      http://extranet.pharos.bg/processes/StudentTransfer.bpel
    </hasProcessModel>
    <upt:hasCategory rdf:resource="#Inbound_Outbound_Logistics_Instance"/>
</StudentTransferCBPmanual>
```

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://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. EFlow (Casati, 2000) uses static workflow generation methods where a composite process is modelled as a graph manually and may be updated dynamically. The tasks in the workflow are however not semantically annotated. Automatic discovery of Web services is based on a definition contained in each service node in the graph. Other research in dynamic workflows such as (Davulcu, 1999) and (Mueller, 2004) are based on homogenous environments and require no mediation amongst services.
6. 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.

- Data and message level heterogeneities between different systems and Web Services result in increased development and integration costs.
- Insufficient service publication and discovery techniques complicate the selection of available Web Services during the design & implementation phase of business processes.
- Inadequate modelling and composition techniques for Web Service based processes result in hand-coded process implementations which prevent the adaptation to new business requirements or changes in the information systems.

FUSION addresses these issues by defining methodology, architecture and tools that extend the concept of SOA by utilizing Semantic Web Service technologies. The FUSION ontology defines semantic concepts that allow for

- tool-supported semantically-assisted data mediation to overcome data level heterogeneities,
- a more efficient business process implementation by semantically-assisted Web Process composition based on reusable Abstract Process Models,
- semantically-assisted search, discovery, and selection of the appropriate Web Services during the design and implementation of business processes.

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.

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

7. REFERENCES

7. F. Casati, S. Ilnicki, L. J. Adaptive and Dynamic Service Composition in EFlow, 2000