

## Data-Driven Adaptation of Heterogeneous Service Processes

George Athanasopoulos, Aphrodite Tsalgaidou  
*Dept. of Informatics and Telecommunications,  
National and Kapodistrian University of Athens, Greece*  
*{gathanas, atsalga} @ di.uoa.gr*

Contemporary instantiations of the Service-Oriented Computing (SOC) paradigm e.g. Web, P2P and Grid services, and information sources such as the emerging Sensor Web and materializations of Web 2.0 and/or Web 3.0 are bustling within the currently forming pervasive computing environment. Within this context the development of adaptable service processes comprising heterogeneous services that utilize available information sources is an emerging trend addressed by this paper. Throughout this document service processes are regarded as systems which operate within a specific environment and perform specified activities, using services, in an orderly manner producing and/or consuming related information during their execution. This emerging trend cannot be addressed by rigid, “static” contemporary approaches such as WS-BPEL [1]. Along the same lines, applied AI planning techniques [2] have primarily focused on the provision of automatically constructed orchestrations (or similarly called process plans) comprising Web services that may be semantically annotated. Similarly, Context Aware Computing (CAC) efforts [3] have addressed neither the interoperability concerns raised by the use of heterogeneous services, nor the utilization of emerging information sources. This gap is being addressed by the work described in this paper.

Specifically, following the principles of the CAC field we argue that within a pervasive computing environment where information sources and services thrive, it is highly probable that information relevant to a process is produced by external systems/sources. Within this frame a process state is not solely depending on the values of its internal parameters but, also on information pertaining to the process environment; therefore, the latter should be taken into account in process development and execution. Information within the context of this paper refers to structured data that have associated meta-information elements and explicitly specified validity time dictated by the information provider.

To leverage the provision of adaptable processes we propose an approach that utilizes contextual information contained within a specific ‘space’ as well as appropriate process adaptation algorithms. This space is considered to be the process environment and, it is open to other processes and systems for information exchange. It is continuously queried for information pertinent to the service process at hand; when relevant and valid information is found, it is utilized by the process execution engine for the adaptation of process instances. The supported adaptation algorithms are based on AI planning techniques which generate specific process plans (i.e. conditional plans which decide which path will be followed upon the value of a condition) that control the execution and adaptation of service processes.

The described approach is realized as a platform comprising three main components: a **Semantic Context Space Engine** (SCS Engine) that is responsible for the collection of contextual information, a **Process Optimizer** responsible for the generation of process plans, and a **Service Orchestration Engine** responsible for the execution, monitoring and adaptation of heterogeneous service processes.

A service process within this context is associated to a set of ‘spaces’ defined at deployment time and managed by the SCS Engine. At least two spaces are associated by default to each service process instance: an instance specific space and a process related space. The former is used for the collection of information related to the service process instance at hand whereas the latter is used for the collection of information related to all instances of the same process. Nonetheless a process instance can only access directly the instance specific space; additional spaces are accessed in an indirect manner through the instance specific space.

Upon the deployment of a semantically enhanced process specification (e.g. BPEL4SWS-based) the Process Optimizer parses the specification at hand and generates a set of information discovery queries which

are fed to SCS Engine, and a process plan that is fed to the Service Orchestration Engine. For the generation of process plans the Process Optimizer implements an AI planner which considers the process adaptation problem as a non-deterministic, partial observability planning problem [4].

The SCS Engine apart from supporting the perpetual execution of information discovery queries provides a semantically enhanced Linda model [5] via which one may execute operations such as insert, read and retrieve of information. As soon as matching results are discovered within the SCS Engine they are pushed to the Service Orchestration Engine.

The Service Orchestration Engine implements a BPEL-based engine that supports the execution of heterogeneous service orchestrations. The need for integrating heterogeneous services has been exemplified in the SODIUM project [6]; our approach towards the accommodation of heterogeneous services has been influenced by the outcomes of this project. Upon the discovery of related information by the SCS Engine the process monitoring component of the Service Orchestration Engine adapts the execution path according to the suggestions specified by the process plan.

The implementation of all three components is based on existing solutions. Specifically, the SCS Engine is based upon the JavaSpace [7] component and accommodates extensions for the semantic annotation of the contained information. Regarding the Process Optimizer our plan is to provide a prototype implementation based on the application of Model Checking techniques [4]. As far as the Process Orchestration Engine is concerned, Apache ODE engine is our prime candidate to be used as a basis for the implementation.

Compared to relevant contemporary approaches [8][9] the innovative aspect of our approach, further to the utilization of heterogeneous types of services, lies in the decoupling of the 'space', the process execution engine and adaptation algorithms. The execution and monitoring of a service process is performed by the execution engine, driven by the process plans generated by the Process Optimizer. Information stemming from external sources is solely managed by the SCS Engine. Interactions among a 'space' and a process instance are performed whenever relevant information is discovered within the SCS Engine. This decoupling enables us to avoid unnecessary message exchanges between a 'space' and a process instance and the delays accruing from the execution of 'complex' coordination (or transformation) rules by the SCS Engine such as in [9].

The proposed approach, along with the platform constitutes a departure from the current state of the art

in the SOC domain, for two main reasons: a) it accommodates the provision of data-driven adaptable heterogeneous service processes, something that, to the best of our knowledge, has not been addressed so far by other approaches and, b) it fosters the opening of service processes to external systems. There are a number of benefits accruing from such an approach, such as: service processes may cooperate with other processes or systems in an implicit manner without enforcing the use of specific interfaces (i.e. expressed in terms of operations) or bindings (i.e. expressed in terms of specific endpoints); this form of cooperation further promotes the decoupling among collaborating parties which is an important prerequisite for the provision of adaptable systems. Therefore, we believe that the presented approach contributes in the provision of service-oriented adaptable systems and paves the way for the provision of autonomic properties into service oriented systems.

**Acknowledgement:** This work has been partially funded by ELKE (contract 70/3/5829) and the European Commission (contract ICT-FP7-249120 ENVISION project)

## 2. References

- [1] A. Alves, et al. Web Services Business Process Execut Language Version 2.0. OASIS Standard Apr. 2007.
- [2] R. Jinghai et al., A Survey of Automated WS Compos. Methods, SWSWPC 2004, LNCS (3387), pp. 43-54.
- [3] Q. Lirong, et al., Context Optimization of AI planning for Services Composition, in 2006 IEEE Int. Conf. on e-Business Engineering, Oct. 2006, Shanghai, China, pp. 610-617.
- [4] P. Bertoli, et al., Planning in nondeterministic domains under partial observability via symbolic model checking, in Proceedings of the IJCAI, Seattle, USA, 2001, pp. 473-478.
- [5] D. Gelernter, Generative communication in Linda, ACM Trans. Program. Lang. Syst., vol. 7(1), 1985, pp. 80-112.
- [6] A. Tsalgatidou, et al., Unified Discovery and Composition of Heterogeneous Services: The SODIUM Approach, in At your service: An overview of results of projects in the field of service engineering of the IST programme, E., Di Nitto et al(Eds.) MIT Press Series on Inf. Systems, 2008, pp. 67-100.
- [7] E. Freeman, et al., JavaSpaces Principles, Patterns, and Practice, Addison-Wesley, 1999.
- [8] L. Nixon, et al., Tuplespace-based computing for the semantic web: A survey of the state-of-the-art, Knowl. Eng. Rev., vol. 23(2), 2008, pp. 181--212.
- [9] A. Mocan, et al., Filling the Gap Extending Service Oriented Architectures with Semantics, in 2nd IEEE International Symposium on Service-Oriented Applications, Integration and Collaboration, 2006, Shanghai, China.