# Dynamic Adaptive Service Architecture – Towards Coordinated Service Composition

Claus Pahl*

Lero and School of Computing, Dublin City University
Dublin, Ireland
cpahl@computing.dcu.ie

**Abstract.** With software services becoming a strategic capability for the software sector, software architecture needs to address integration problems to help services to collaborate and coordinate their activities. The increasing need to address dynamic and automated changes can be answered by a service coordination architecture with event-based collaboration that enables dynamic and adaptive architectures. Intelligent service and process identification and adaptation techniques are suitable solutions for event-driven and on-demand service architectures. We define an architectural solution space and identify research challenges.

## 1 Introduction

Service-oriented architecture (SOA) is a methodological framework for software architectures, supported by Web services as the platform technology. Particularly scalability and suitability for collaborative applications are limited due to the restrictive nature of current composition, collaboration and interaction techniques such as orchestration and choreography languages as the core principles.

Interoperation and coordination of services is a major challenge service-oriented architecture in the context of on-demand scenarios - as the emergence of cloud computing as a form of service architecture virtualisation demonstrates [4]. Today, hand-crafted service architectures are in place and provide support for software systems in classical sectors such as finance or banking. However, their inherent structural inflexibility makes changes and evolution difficult.

## 2 A Changing Architectural Landscape – SOA Challenges

The vision behind recent initiatives such as cloud and on-demand computing is to enable collaboration of service communities [4]. These exhibit a more dynamic nature of interaction, which requires novel software architecture techniques for the identification of needs and behaviours and the adaptation and customisation of provided services to requested needs. The coordination of activities between

* This work was supported, in part, by Science Foundation Ireland grants 03/CE2/I303_1 (Lero) and 07/RPF/CMSF429 (CASCAR).

communities of users and providers needs to be supported [5]. Architecture-based solutions for these evolving and software-intensive systems are sought.

Currently, orchestration and choreography approaches describe business interaction protocols that coordinate and control collaborating services [8]. Challenges for architectural configuration to support future needs are [7]:

– Dynamic and adaptive processes. Services and processes need to provide adaptive capabilities in order to respond to evolving demands and changes without compromising operational and financial efficiencies. A challenge is to provide self-management support for dynamic service compositions.
– QoS-aware service compositions. Service compositions must be QoS-aware - including business regulations, performance levels, reliability requirements or service-level agreements (SLA).

## 3 Architecture Implications – Coordination

The changing architectural landscape requires flexible composition techniques such as event-driven and decentralised coordination instead of tightly coupled synchronous and centralised compositions [9] – resulting in three objectives:

– *Objective 1: provide a technology framework (platform + methodology) that allows flexible composition of services for dynamic service architectures.* The core solution can be built around a notion of a coordination space. This coordination space acts as a passive infrastructure to allow communities of users and providers to collaborate through the coordination of requests and provided services. The coordination space can be governed by event-driven principles: tasks to perform some activity on an object occur are requested, services collaborative and coordinate their activities to execute these tasks.
– *Objective 2: provide flexible infrastructure mechanisms to support dynamic, changing service architectures.* Dynamic selection and adaptive, process-centric composition of services to meet user needs requires a considerable degree of flexibility: user requests might be incomplete or incorrect and need to be corrected, individual requests can be part of an ongoing process that can be derived from the context and the execution history, and provided service might need to be adapted and customised to meet user needs.
– *Objective 3: provide a solution to support future Internet objects and applications.* Users are concerned with the processing of objects. In classical enterprise scenarios these objects are electronic documents passing through business processes, but within the Future Internet, the notion of objects will broaden, capturing any dynamic, evolving entity.

The central concepts are objects and processes. Evolving objects are dynamic entities that represent an end-to-end view. The process notion refers to business processes on these objects. States of the process are points of variation for objects: data evolves as it passes through a process. Process-centricity is the first aspect that characterise this new architecture [3]; the second is a paradigm shift from a pull- to a push-model of communication. Instead of requesting services directly (pull), requests are published (push) and responded to independently.

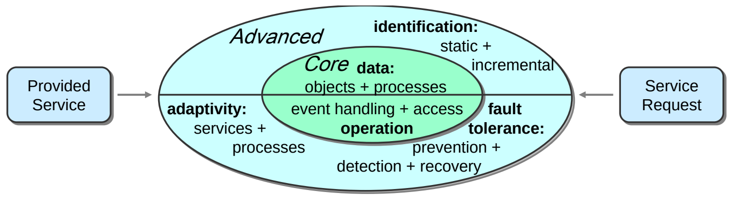


**Fig. 1.** An Architectural Challenges Framework.

## 4 A Challenges Framework and Architectural Solutions

Fig. 1 represents an architectural solution framework that we use here to locate and describe specific research challenges. The conceptual framework is goal-driven, event-based collaboration of services. We can identify two *core* facets: the data representing processable objects and the processes themselves and the operation of coordination based on event handling and self-management.

Based on a core architecture, we look into challenges for the information architecture, the operational aspects of dynamic and adaptive service architecture and quality considerations – forming the advanced layer in our framework.

*Information Architecture.* The architecture needs to process object and process information in many ways:

- static and incremental process identification aiming to determine individual tasks, i.e. steps of a larger process, to achieve goals based on an abstract, user-centric and object-based goal specification. Static determination can identify processes based on static knowledge, e.g. in the form of common structural patterns. Incremental determination based on mining approaches can incrementally identify behaviour patterns based on historical data.
- adaptation of provided services to meet the needs of identified process and requestor via mediation between client and provider. Based on identified processes that should enable an object goal to be achieved, adaptations of existing services or subprocesses might be necessary to bridge the gap between requirements and actual services: service-level adaptation as data-centric mediation based on identified process patterns and process-centric adaptation to adapt processes locally to include user profile and context aspects.

*Operation through Coordination.*

- Event handling is the challenge. A variety of coordination models has been proposed [1],[6],[2], e.g. based on tuple spaces. Domain- and application context-specific solutions and approaches based on semantic extensions need to be further investigated and applied to service composition and mediation.
- Self-management is a requirement in dynamic systems. A critical aspect is fault-tolerance [10]. The classical security aspects prevention, detection and recovery can be applied to define challenges. Correction of incorrect or incomplete input is a fault prevention technique; constraint monitoring is detection or remedial strategies can be defined for recovery purposes.

– Governance is a management-related aspect that also bridges into quality. Compliance with not only technical constraints is needed for self-management, but also wider regulatory and business constraints are of importance for virtualisation environments such as the cloud that bridge organisational boundaries and therefore need to reconcile different regulatory needs.

The coordination models selected to support a solution determines notational aspects we would expect an architectural description language to deal with.

*Quality Reflections.* We have discussed different technology challenges for dynamic, adaptable service coordination architectures. As quality is a central concern of software architecture, the respective techniques need to be considered from a quality perspective. The infrastructure techniques suggested here require specifically qualities related to the dynamic context in which they need to be provided, i.e. performance and reliability are central challenges. The services (i.e. applications themselves) are subject to varying qualities as required by the context, but need to be dealt with dynamically, i.e. efficiency and reliability are again critical requirements. Accountability through governance is another quality aspect of importance.

## References


1. D. Balzarotti, P. Costa and G.P. Picco. The LighTS tuple space framework and its customization for context-aware applications. Web Intelligence and Agent Systems 5(2): 215-231. 2007
2. E.-E. Doberkat, W. Franke, U. Gutenbeil, W. Hasselbring, U. Lammers and C. Pahl. PROSET - Prototyping with Sets, Language Definition. Software-Engineering Memo 15, Universitt GH Essen, 1992.
3. V. Gacitua-Decar and C. Pahl. Automatic Business Process Pattern Matching for Enterprise Services Design. 4th International Workshop on Service- and Process-Oriented Software Engineering (SOPOSE-09). IEEE Press. 2009.
4. B. Hayes. Cloud computing. Communications of the ACM 51(7):9-11. 2008.
5. B. Johanson and A. Fox. Extending Tuplespaces for Coordination in Interactive Workspaces. Journal of Systems and Software 69(3), 243-266. 2004.
6. L. Nixon, O. Antonechko and R. Tolksdorf. Towards semantic tuplespace computing: the semantic web spaces system. Symp on Appl Computing SAC'07. 2007.
7. M.P. Papazoglou, P. Traverso, S. Dustdar, F. Leymann. Service-Oriented Computing: State of the Art and Research Challenges. Computer, 38-45, Nov. 2007.
8. J. Rao and X. Su. A Survey of Automated Web Service Composition Methods. Intl. Workshop on Semantic Web Services and Web Process Composition 2004. Springer LNCS 3387, Pages 43-54, 2005.
9. C. Utschig-Utschig. Architecting Event-Driven SOA: A Primer. Oracle. http://www.oracle.com/technology/pub/articles/oraclesoa_eventarch.html. 2008.
10. M. Wang, K. Yapa Bandara and C. Pahl. Integrated Constraint Violation Handling for Dynamic Service Composition. IEEE International Conference on Services Computing SCC 2009. IEEE. 2009.