GAMBUSE: A Gap Analysis Methodology for Business Service Engineering* 

Dinh Khoa Nguyen, Willem-Jan van den Heuvel, Mike P. Papazoglou 
ERISS, Tilburg University, The Netherlands 
{D.K.Nguyen, W.J.A.M.vdnHeuvel, M.P.Papazoglou}@uvt.nl

Extended Abstract

Service Oriented Architecture (SOA) promotes highly standardized, loosely coupled and Web-enabled services to foster rapid, low-cost and easy composition of distributed enterprise applications [3]. SOA development methodologies such as [1][2][3] provide models, best practices, standards, reference architectures necessary to construct a well-defined SOA. Central to any of these methodologies are business service analysis and design techniques for identifying, conceptualizing, profiling and rationalizing service-enabled business processes. Business services capture transactional and value-creating business activities that are delivered by service providers to service clients under strict business conditions, e.g., sending an invoice, checking inventories or creating a purchase order [1]. Business services are intrinsically conceptual and implementation-agnostic. They may be physically realized in several alternative ways; typically however they are constructed with Web service technologies, e.g., SOAP, WSDL and BPEL.

The objective of business service engineering is to identify candidate business services, and provide an in-depth understanding of their functionality, scope, reuse, and granularity. Unfortunately, many of today’s service analysis and design techniques rely only on ad-hoc and experience-based identification of value-creating business services from existing best-in-class case studies and implicitly assume a “green-field” situation focusing on the development of completely new services while offering very limited support for discovering candidate services from a varied inventory of pre-existing software assets. There is a lack of a more rigorous service identification and conceptualization approach that ensures that the resulting business services can contribute high business value and have the right granularity level for maintainability and reusability purposes across the enterprise. Besides, it is important to note that most of the effort in a SOA adoption goes for the task of modifying and integrating existing IT applications than on deploying new ones [5]. However, the migration of the existing systems to a new SOA environment goes far beyond the basic wrapping techniques, which are used to typically implement a thin SOAP/WSDL/UDDI layer on top of existing software functionalities [3]. While relatively simple non-critical applications may be effectively built in this way, designing industrial-strength SOAs ne-

* This is an extended abstract of the paper [4].
cessitates a business service engineering methodology to support the dissecting, decomposing and repurposing existing functionalities for the new business services.

The approach that we introduced in [4] is named “Gap Analysis Methodology for BUusiness Service Engineering” (GAMBUSE) and attempts to address these specific shortcomings by concentrating on the following two inter-related issues:

- **Business Service Identification**: In a service-enabled to-be process, each activity contributes to a well-defined business objective, e.g., order management, inventory management, customer management, etc., and could be realized by a high-level composite service interaction. GAMBUSE introduces a process-driven technique to identify and conceptualize meaningful business services for an SOA project by collectively grouping all activities that contribute to the same course into logically cohesive and loosely coupled business services. A high-quality SOA design must ensure that the resulting business services have the right level of granularity for the maintainability and reusability purposes. Cohesion and coupling are the key design principles in GAMBUSE that help to ascertain reasonable quality of resulting business services during the business service engineering.

- **Gap Analysis**: GAMBUSE adopts a flexible gap analysis approach that detects and assesses the reuse of available (internal and external) software assets (termed as-is services) as (parts of) the newly conceived business services that collectively shape a to-be business process. Functional equivalence and behavioural and policy misfits between the newly developed to-be processes and the as-is services are taken into account to support the reuse strategy of existing software functionalities.

Inputs of the Gap Analysis include the abstract as-is (internal and external) software artefacts, as well as the to-be business process and services. Output is a business service portfolio that not only specifies the new business services in terms of their functionality, behaviour and policies, but also defines several service realization strategies and proposes a decision framework for selecting an appropriate one. A service realization strategy should include design decisions whether (parts of) the new BS functionalities can be obtained by reusing or revamping existing IT assets, or whether they should be re-designed or developed from scratch [3]. This portfolio serves as the basis for further concretization into service interface, endpoint and policy definitions (e.g., using WSDL, WS-Policy and BPEL) during the next physical SOA design phase.

**References**