# Optimal Granularity for Service-Oriented Systems
(Extended Abstract)

Saad Alahmari and Ed Zaluska
School of Electronics and Computer Science
University of Southampton, Southampton. SO17 1BJ, UK
saa08r@ecs.soton.ac.uk, ejz@ecs.soton.ac.uk

## Abstract
The improved interoperability and business agility of business systems based on Service-Oriented Architecture (SOA) has created an increased demand for the reengineering and migration of legacy software systems. The wide range of current migration techniques for legacy systems in different implementations technologies does not address important aspects of service granularity, which affect service reusability, governance, maintainability and cohesion. This paper proposes a novel framework for the effective identification of the key services in legacy code. The approach focuses on defining the right services based on standardized modelling languages (UML and BPMN). This framework provides effective guidelines for optimal service granularity for a wide range of possible service types.

**Keywords**: Service-Oriented Architecture, Service granularity, Software Evolution, Legacy Systems, BPMN.

## 1. Introduction
Coping with technology evolvement and rapid business changes significantly affect existing legacy software systems. In the past, software systems were typically developed with embedded business rules and logic, scattered and duplicated code, unstructured modules, and tightly-coupled functions. Such legacy systems nevertheless often represent a considerable investment by the underlying business which will frequently rely on the software for many day-to-day business activities. Service-Oriented Architecture (SOA) is a modern approach to implementing (and re-implementing) such systems as a set of robust and interoperable services. There are many different interpretation of the 'best' approach to build a (SOA) system [1]. It is not merely applying an architectural approach at a high level; it is about making all involved parties in the system (such as business users, analysts, and developers) think in a service-oriented manner. The success of the SOA approach critically depends on the correct identification, presentation and definition of the key services, because the exposed functionalities in a service define the service granularity. It is important to appreciate that achieving an optimal level of service granularity requires a compromise between many elements, both technical and non-technical. In particular, the optimal granularity of key services can be expected to vary at various layers with different service types [2] or service layers [3, 4].

Recent research conducted by Kohlborn [5] on thirty modern service analysis approaches showed that 76% of those approaches introduced two types of services (e.g. business service, software service or generic service). The other approaches depicted between one and eleven types of services. All of the approaches studied applied one of the standard analysis techniques (e.g. top-down, bottom-up and meet-in-the-middle), although the services identified were significantly different from one approach to another. This means that defining service granularity, which is a very challenging task, requires considering not only service characteristics but also provisional service types. For example, an infrastructure service that is concerned with providing heterogeneous underlying capabilities to other services should be fine-grained for high reusability and encapsulation. On the other hand, a business service will typically be implemented as a coarse-grained service for maximizing business value and traceability of business processes.

The adoption of Business Process Modelling Notation (BPMN) and Unified Modelling Language (UML) as the modelling languages in this research is motivated by two factors. Firstly, many research studies agree that process-oriented modelling provides an excellent foundation for deriving the optimal services [6,7,8,9,10,11] in any system. Secondly, while UML 2.0 Activity Diagrams for business process modelling (introduced by the Object Management Group (OMG)) is a useful analysis tool it suffers from limitations on modelling related resources and representing various types of control-flow constructs [12]. BPMN and UML present two different views of complex software systems [13]. BPMN is based on process orientation, whereas

UML is an object-oriented approach. In other words, both approaches will identify services from different prospective; one from the enterprise level and the other from the application level. This provides a mechanism to align business and software-based aspects using the functionality provided and allow service metrics to play a key role in identifying the optimal service granularity.

# 2. The background

Existing research conducted in the area of migrating and integrating legacy systems for SOA has proposed several different approaches from a number of different perspectives: the technical domain, the business domain, the conceptual approach and the detailed analysis approach [5].

Research that focuses on the technical domain starts the migration of legacy systems by reengineering the legacy codes into candidate services and then those services are mapped onto functional requirements and processes [6, 15, 16, 17,18]. This technique is called the bottom-up approach because it converts legacy-system components into key services on the basis of the legacy system functionality [19, 20]. The drawbacks of these approaches are that they identify target services at two different architectural levels without a reconsolidation process (i.e. new services are identified both through domain analysis at an abstract level and also targeted services are synthesised directly from legacy code). They also lack any detailed understanding of the problem domain [17] and require human intervention to assist in determining the optimal service granularity [6, 18]. Such applied-technical-analysis techniques by themselves are insufficient. For example, Chen, et al [15] claims that their feature analysis bridges the gap between the abstract architecture level and source code, whereas business processes are excluded.

An enhanced response to business agility requirements is one of the key SOA objectives usually cited for a business domain approach [7, 10, 21, 22]. Researchers argue that SOA is not merely integrating the software infrastructure, it also needs to consider the underlying business models (e.g. business process, use case and activity diagrams) [10, 22]. Top-down analysis techniques identify seamlessly services mapping from business processes or use cases [23]. What distinguishes SOA from other software methodologies is that SOA claims to be strategically aligned with the overall business vision [24]. Those approaches neglect fine-grained services and define only coarse-grained services with non-technical descriptions [2]. They do not consider any of the technical aspects of interfaces, behaviour, and service composition [7].

Because of the complexity of most software systems, researchers often adopt abstract models to simplify the descriptions of legacy systems [24, 25, 26]. Conceptual models describe only the high-level activities in core business processes, without any detailed business logic and rules. The resulting candidate services tend to be coarse-grained with redundant functions. Although these approaches were successful in broadly highlighting important architectural aspects, they failed to provide detailed guidance for the SOA re-engineering and in particular the underlying patterns failed to provide any usable guidelines to enhance service granularity.

There has been a continuous failure to identify the "right" services of legacy systems when SOA characteristics such as reusability, loss-coupling, and encapsulation are considered. Researchers have concluded that combining the bottom-up and top-down approaches is essential to define the right services (this is called the "meet-in-the-middle" approach) [8, 9]. Such approaches incorporate broad guidelines to enhancing the service granularity such as reusability, business-alignment, and reduce the effects of subsequent application changes.

A detailed analysis of the prior work demonstrated that none of the previous approaches has enabled an accurate identification of when services should be coarse-grained and fine-grained. While the research studied almost always concludes with similar service design principles, they could not set out well-defined and practical steps to accomplish those principles. However, all of the references agree on the complexity of considering all applicable factors to fulfil both the business and technical aspects. They do not identify the various types of services with optimal granularity in any effective fashion. Not enough detail is included on how to identify key services along with new business requirements and the targeted service characteristics.

# 3. The proposed approach

This paper proposes an SOA architectural framework to assist service identification, definition and realization for migrated legacy systems with optimal service granularity. The framework is based on two technical and process portfolios that are derived from UML and BPMN analyses, in addition to a knowledge-based portfolio (as shown in figure 1). It encompasses both functional and non-functional elements that affect the service identification of migrated legacy systems. The service hierarchy definition by Kulkarni and Dwivedi [2] is enhanced to define a comprehensive service metadata model for enterprise service types and characteristics. This service metadata model provides effective standards for service granularity quantification after capturing potential services from both models (UML-based and BPMN-based).

In many real-world scenarios, the only available information about the legacy system is the code itself (in which case we identify this as one possible scenario). Our overall approach consists of three main stages as follow:

First stage: The analysis and reengineering stage, which has the following activities:

1) If any documentation, expertise or interviews are provided, a analysis model can be created to build the knowledge portfolio (optional)
2) Transforming the legacy system codes into UML models by reverse-engineering techniques to obtain a static model automatically (using modelling tools such as IBM Rational Rose). The reverse-engineering process results in class diagrams that generate activity diagrams of the overall system.
3) Transforming the UML models into BPMN automatically using Model Driving Development (MDD) in order to generate a business processes portfolio. The Relation Definition Language (RDL) describes the transformation rules as part of the Eclipse Modelling Framework (EMF) which utilizes the IBM Model Transformation Framework (MTF) running on the IBM WebSphere Business Modeler. Standards for mapping elements between UML activity diagram and BPMN are identified according to pattern-based analysis of BPMN [12, 14].

Second stage: The services identification elements stage, which has the following activities:

4) Defining functions and data entities from the activities portfolio; applying clustering technique to select candidate business services (coarse-grained services) based on customer-reported unique defects (CRUD) metrics [27].
5) Defining atomic processes and business entities from the processes portfolio; applying clustering metrics at the atomic process level to define fine-grained services.

Third stage: The services evaluation stage, which has the following activities:

6) Determining relative services that can be identified in service components (where the levels of abstraction are varied among services).
7) Evaluating coarse-grained and fine-grained services against the service metadata model outputs.

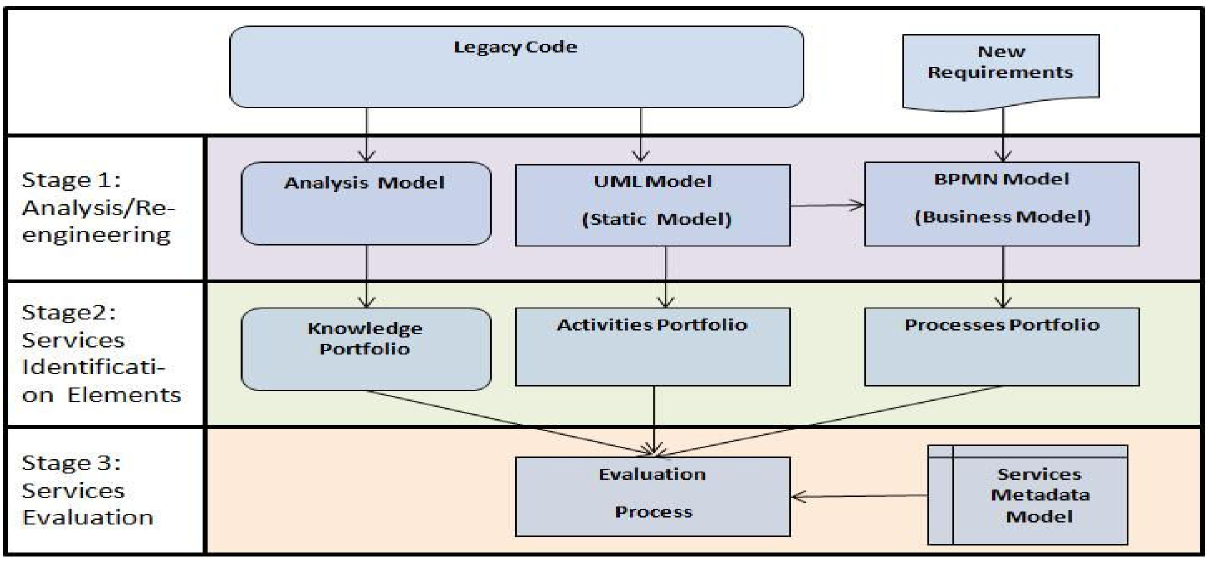


Figure1: The overall service identification process

## 4. Conclusions

Our proposed framework is currently at the second stage of implementation where a transformation model to convert from UML activity diagram to BPMN diagrams is required. To evaluate our approach, various components of legacy code will be used along with our built metrics. The success of this novel approach will contribute an automated transformation process between UML and BPMN, a standardized service metadata model for enterprise services and also a general-purpose framework for service identification in SOA systems. Preliminary work on a legacy system code has suggested that our migration approach can be successful and viable for many object-oriented systems.

## References


[1] K. Kontogiannis, G.A. Lewis, D.B. Smith, M. Litoiu, H.Muller, S.Schuster, E.Stroulia, “The landscape of service-oriented systems: a research perspective,” Proc. 2007 International Workshop on Systems Development in SOA Environments, IEEE, 2007, pp. 1-6.

[2] N. Kulkarni, V. Dwivedi, “The Role of Service Granularity in A Successful SOA Realization - A Case Study,” Proc. IEEE Congress on Services, IEEE Computer Soc, 2008, pp. 423-430.

[3] P. Reldin, P.Sundling, “Explaining SOA Service Granularity: How IT-strategy shapes services ” Master’s thesis, Dept. Management and Engineering Industrial Economics, Institute of Technology Linkoping University, 2007.

[4] Ramollari, E., Dranidis, D. & Simons, A.J.H. “A Survey of Service Oriented Development Methodologies”, 2nd European Young Researchers Workshop on Service Oriented Computing, 2007. Available at www.dcs.shef.ac.uk/~ajhs/research/papers/soasurvey.pdf

[5] T. Kohlborn, “A consolidated approach for service analysis,” Master’s thesis. Westfalische-Wilhelms University Munster,2008.

[6] Z.P. Zhang, H Yang, “Incubating services in legacy systems for architectural migration,” Proc. 11th Asia-Pacific Software Engineering Conference (APSEC 2004), IEEE Computer Soc, 2004, pp. 196-203.

[7] C. Rolland , R.S. Kaabi, “An intentional perspective to service modeling and discovery,” Proc. 31st Annual International Computer Software and Applications Conference, IEEE Computer Soc, 2007, pp. 455-460.

[8] A. Erradi, S. Anand, N. Kulkarni , “SOAF: An architectural framework for service definition and realization,” Proc. IEEE Services Computing Workshops, IEEE Computer Soc, 2006, pp. 151-158.

[9] A. Erradi, N. Kulkarni, P. Maheshwari, “Service design process for reusable services: Financial services case study,” Proc. 5th International Conference on Service-Oriented Computing (ICSOC 2007), Springer-Verlag Berlin, 2007, pp. 606-617.

[10] M.P. Papazoglou , W.J. van den Heuvel, “Service-oriented design and development methodology” , International Journal of Web Engineering and Technology (IJWET), vol. 2, no. 4, 2006, pp. 412-442

[11]P. Jamshidi, M. Sharifi, S. Mansour , “To Establish Enterprise Service Model from Enterprise Business Model,” Proc. IEEE International Conference on Services Computing, IEEE Computer Soc, 2008, pp. 93-100.

[12] N. Russell, W.M.P. van der Aalst, A.H.M. ter Hofstede, P. Wohed, “On the Suitability of UML 2.0 Activity Diagrams for Business Process Modelling”. BPM Center Report BPM-06-03, BPMcenter.org, 2006.

[13] BPMI.org. Business Process Modelling Notation (BPMN) Information: “Frequently Asked Questions”. Available at http://www.bpmn.org/Documents/FAQ.htm.

[14] P. Wohed, W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, N. Russell. “Pattern-based Analysis of BPMN - An extensive evaluation of the Control-flow, the Data and the Resource Perspectives “ (revised version). BPM Center Report BPM-06-17 , BPMcenter.org, 2006.

[15] F. Chen, S. Li, H. Yang, C. Wang, W. Chu, “Feature analysis for service-oriented reengineering,” Proc. 12th Asia-Pacific Software Engineering Conference, IEEE Computer Soc, 2005, pp. 201-208.

[16] Y. Zou , K. Kontogiannis, “Towards a Web-centric Legacy System Migration Framework”, the 3rd International Workshop on Net-Centric Computing (NCC): Migrating to the Web, International Conference on Software Engineering (ICSE'01),Toronto, Canada, May 2001.

[17] L. Jianzhi, Z. Zhang, H. Yang, “A grid oriented approach to reusing legacy code in ICENI framework,” Proc. Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration, IEEE Computer Soc, 2005, pp. 464-469.

[18] Z. Zhang, L.Ruimin, H. Yang. “Service identification and packaging in service-oriented reengineering”. In: Chu WC, Juzgado NJ, Wong WE, editors. Proceedings of the 17th international conference on software engineering and knowledge engineering, Taipei, Taiwan, July 14–16, 2005. Los Alamitos (CA): IEEE Computer Society Press; 2006. p. 219–26

[19] M. Perepletchikov, C. Ryan, Z. Tari , “The impact of software development strategies on project and structural software attributes in SOA” Proc. On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshop. OTM Confederated International Workshops and Posters AMeSOMe, CAMS, GADA,

MIOS+INTEROP, ORM, PhDS, SeBGIS, SWWS, and WOSE 2005. Proceedings, Springer, 2005, pp. 442-451[20] T.D. Meijler, C. Ryan, Z. Tari, “Top down versus bottom up in service-oriented integration: an MDA-based solution for minimizing technology coupling,” Proc. Service-Oriented Computing - ICSOC 2006. 4th International Conference. Proceedings, Springer-Verlag, 2006, pp. 484-489.

[21] M. Galster, E. Bucherer, “A Business-Goal-Service-Capability Graph for the Alignment of Requirements and Services,” Proc. IEEE Congress on Services, IEEE Computer Soc, 2008, pp.399-406.

[22] K. Yukyong and D. Kyung-Goo, “The service modeling process based on use case refactoring,” Proc. Business Information Systems. 10th International Conference, BIS 2007. Proceedings, Springer, 2007, pp. 108-120.

[23] N. Bieberstein, R. Shah, K. Jones, S. Bose, and M. Fiammante, “Service-Oriented Architecture (SOA) COMPASS: Business Value, Planning, and Enterprise Roadmap”, Prentice Hall PTR, Upper Saddle River, NJ (2005)

[24] A. Arsanjani, “Service-oriented modeling and architecture: How to identify, specify, and realize services for your SOA”, Available at www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/.

[25] W. Xiaofeng, S. Hu, E. Haq, H. Garton, “Integrating legacy systems within the service-oriented architecture,” Proc. 2007 IEEE Power Engineering Society General Meeting, IEEE, 2007, pp.7.

[26] A. Arsanjani and A. Allam, “Service-oriented modeling and architecture for realization of an SOA,” Proc. 2006 IEEE International Conference on Services Computing, IEEE Comput. Soc, 2006, pp. 1.

[27] Ph. Teale, and R. Jarvis, "Business Patterns for Software Engineering Use", The Architecture Journal, Available at http://msdn2.microsoft.com/enus/arcjournal/aa480036.aspx, 2004.