A Model-Driven Architecture Approach to the Efficient Identification of Services on Service-oriented Enterprise Architecture

Saad Alahmari, David De Roure, Ed Zaluska
School of Electronics and Computer Science University Southampton, UK
{saa08r, dder, ejz }@ecs.soton.ac.uk

***Abstract*—Service-Oriented Enterprise Architecture requires the efficient development of loosely-coupled and interoperable sets of services. Existing design approaches do not always take full advantage of the value and importance of the engineering invested in existing legacy systems. This paper proposes an approach to define the key services from such legacy systems effectively. The approach focuses on identifying these services based on a Model-Driven Architecture approach supported by guidelines over a wide range of possible service types.**

Keywords - SOEA, Service-Oriented Enterprise Architectures, Enterprise Engineering, Legacy systems, Architecture modelling, Model-Driven Architecture

I. INTRODUCTION

Service-Oriented Enterprise Architecture (SOEA) is a modern approach to implementing (and re-implementing) software systems as a set of robust and interoperable services. A complete architecture will be based on an underlying structure of resources (e.g. a defined business strategy, appropriate business processes, key data and information, software applications etc.) together with their interdependencies. One of the most vital resources is the existing software (i.e. the legacy systems), often representing a considerable investment by an underlying business which will frequently rely on the legacy software for many day-to-day business activities. In this paper, legacy systems refer to any computer programs inherited from previous software systems that are not service-based.

Research in academia and industry to migrate legacy systems to a SOAE environment has mostly concentrated on:

(a) developing service wrappers for existing business logic

(b) or (as an alternative) an incremental migration process consolidating the existing business logic

(c) or (as an alternative) integrating the legacy code using adapters.

Service identification is a key design stage and forms the initial phase of every SOEA project lifecycle [1]. There has been relatively little research into an effective approach to identifying the 'right' services to be implemented in a SOEA system using approaches such as Model Driven Architecture (MDA).

Our research emphasizes the importance of defining the 'right' service because poor design decisions made here can result in compromises affecting the entire service–oriented enterprise. The 'right' service is assumed to provide an optimal level of granularity that does not interfere with service design objectives (e.g. loose coupling) but also provides the tradeoffs required by the organisation (e.g. in terms of complexity levels and maintainability). It is also important to stress that defining the key services based on legacy systems inevitably requires a compromise between many elements, both technical and non-technical.

In SOEA, the context of a business service is always driven from a business process or function. This means that it is usually appropriate for the business process definition and design to contribute to the process of service identification. To support this objective, the Object Management Group (OMG) has proposed a platform-independent MDA approach for modeling processes and services [2, 3]. At the business level, our research adopts the Business Process Modeling Notation (BPMN) to provide a behavior model of the legacy system representing graphically the interactions and collaboration among the participants. However, a number of existing legacy systems were implemented using an Object Oriented (OO) approach, often modeled with UML. Our research hence starts with an automatic transformation from a static UML model to a BPMN model, i.e. we transform UML activity diagrams to BPMN business processes. We also adopt an SOEA meta-model that defines service types along with the semantics. Unlike existing approaches, our approach incorporates the ability to trace key requirements to ensure that critical business changes are implemented.

The key contribution of this paper is to provide an overall framework and guidelines for the effective identification of the 'right' services from existing legacy software systems. The rest of this paper is organized as follows: section II discusses related work. Section III introduces the classification of service types. Section IV proposes our design approach. Section V presents an example of an application and is followed by section VI which provides an evaluation of this example. Finally, section VII contains the conclusions.

II. RELATED WORK

Prior research in the area of migrating and integrating legacy systems for the support of SOEA has generated a number of different approaches. The majority of these papers pay relatively little attention to the service identification phase; they implicitly consider it as a sub-stage of the design phase. The previous research can be analyzed from several different perspectives: the business domain, the technical domain, and the enterprise domain.

From the business domain perspective, reference [4] proposes a graph-based framework to discover service granularity. Reference [5] focuses mainly on how to define the right services in the analysis phase, on the

basis of business change factors and goals. Reference [6] introduces an approach that depends on exploring the purposes of a business process in order to identify a service, i.e. considering business goals, together with any pre- and post-conditions. Reference [7] notes the importance of granularity in defining web services without suggesting any particular solution. Reference [8] attempts to address the gap between service provider and requester regarding to service agreements. A Unified Service Model (USM) is proposed along with a service operational model to specify business services from a business people perspective. Although the authors of this approach claim that the USM defines business services at multiple levels of granularity, no metrics or guidelines are provided to identify the service granularity

In the technical domain, in reference [9] the authors develop adapters using a CORBA wrapper (CORBA IDL, SOAP, WSDL, and UDDI), and in reference [10] the authors use reverse engineering techniques with a Java Native Interface (JNI) wrapper to encapsulate code with the Commerce eXtensible Markup Language (CXML). The drawback of this approach is that it lacks an understanding of the problem domain and therefore migrates independent blocks of code directly into services that cannot interoperate effectively. Reference [11] discusses the transforming of the legacy systems developed with Object-Oriented Design (OOD) or Component Based Design (CBD) into SOEA applications using feature analysis. These service operations are then exposed by class delegations using a tool called a Web Service Wrapper. Reference [12] proposes an architecture-based service-oriented reengineering approach that uses a hierarchical clustering method. Reference [13] proposes a hierarchical clustering algorithm to extract independent services from procedural software systems into an object-oriented (OO) model.

To be able to achieve traceability through various SOEA lifecycles, some researchers have focused on the enterprise domain. Reference [14] utilizes specific enterprise service hierarchy patterns for selected business processes to determine the service granularity. Reference [15] outlines sets of activities in the analysis phase that lead to an adequate broad foundation for service identification. References [1, 16, 17] propose a comprehensive approach with a high-level service architectural classification and iterative processes. Reference [17] extends the service design concepts of the SOAF framework in [17] with a business-driven approach based on a meta-model to define service granularity. Reference [19] attempts to categorize services based on operational state of services and logical presentations, i.e. differentiating between application and business services. Reference [20] introduces a semi-automated approach to identify services on process-oriented systems. It converts the UML-based business process models into XMI. The XMI reader "NSUML" is used to produce the MOF (Meta Object Facility) for mapping XMI meta-model. The algorithm used runs over an XMI meta-model developed a statistic based approach which helped in creating APIs to query candidate services.

An analysis of this related work demonstrates that none of the previous approaches has enabled accurate service identification, in terms of when services should be coarse-grained and when they should be fine-grained, in effect ignoring the interdependencies between different service types. Although the approaches studied usually conclude with proposed service design principles, these do not provide well-defined and effective mechanisms to accomplish these principles. However, the references all agree on the complexity of considering every applicable factors to fulfill all of the different business and technical enterprise aspects. This paper concentrates on service granularity in (SOEA) legacy systems because we believe that service granularity is a critical component of service design with a significant impact on other key design aspects such as reusability, maintainability, performance and flexibility.

III. DEFINING SERVICE TYPES

A service should accomplish certain goals which can correspond to a business or a technical requirement. Our classification is concerned primarily with defining the "optimal" level of granularity for every service type. This classification will guide the service identification framework to define roles and properties for each service type. It will be also used (along with a proposed metric) in identifying the 'right' services with optimal granularity. The closest service type classification in the literature to that in our approach identifies eight generic service types from a hierarchical perspective [20, 21]. In contrast, our classification focus is on defining the purpose of the service, i.e. what the service is expected to expose to the service consumer. The functional scope of different services is a key element in constructing a service taxonomy. The purpose of the service can be defined either as 'CRUD' functions (create, retrieve, update, delete), or as business logic, or as specific-domain functions (e.g. a service for customer credit check) or infrastructure capabilities.

The data nature is also considered as a function of the frequency of data modification. Our analysis is designed to facilitate data exposure and process access functions through flexible and reusable services [16]. Based on the concept of the Business Intelligence (BI) meaning of data, we define two new types of services: a master-data service and a transactional-data service. These new two service types reside at the same level of data services abstraction. For example, a master-data service (such as the initial definitions of customer name and address) is not likely to change on a frequent basis, hence few data parameters are manipulated, and there is therefore no verification or compensation mechanism required [22]. With a transactional data service (such as "items ordered") there will be always a data change (e.g. when the customer places the new order), hence such a service requires more communications. Table 1 shows our classification of service types with different levels of granularity from the functional perspective. We define seven different service types; process service, business service, composite service, transactional-data service, master-data service, utility service, and infrastructure service. The services are defined from the most granular

(e.g. a process service) to the least granular (e.g. an infrastructure service).

TABLE I. THE CLASSIFICATION OF SERVICE TYPES

Service Type	Functional Scope
Process Service (most granular)	Processing sequenced tasks in a pre-defined flow of business process (coarse-grained).
Business Service	Presenting business logic of a transaction or a business entity (coarse-grained).
Composite Service	Aggregating several services in the enterprise (coarse-grained).
Transactional-data Service	Processing CRUD functions of transitional data (fine-grained).
Master-data Service	Processing CRUD functions of master-data (fine-grained).
Utility Service	Providing domain functionalities to other services (fine-grained).
Infrastructure Service (least granular)	Providing essential technical functionalities for other services and the architectural enterprise (fine-grained).

IV. PROPOSED APPROACH

A service definition is always derived from an original business process or a business function [23]. In this context, our framework is based on process portfolios derived from the UML and BPMN standards in addition to an optional knowledge portfolio (as shown in fig. 1). Legacy code is transformed to a static model (e.g. UML class diagrams) and then a behavior model (e.g. BPMN diagrams). In the case of new system requirements, business processes in BPMN models are updated. Cluster metrics are then used to generate services based on the classification of service types. The SOEA meta-model provides a comprehensive description of service types integrating the semantic of business processes. It provides effective standards for service granularity quantification after the capturing of candidate services. In many practical legacy system migrations, there is no available information about the legacy system apart from the code. Assuming that only the legacy code is available (as an extreme scenario); our approach consists of three main stages as follows:

- First stage: the analysis and reengineering stage. This includes the following activities:
  1. If any documentation, or staff interviews are available, an analysis model can be produced to provide a knowledge portfolio.
  2. Transforming the legacy system codes into a formal representation in UML models (e.g. class diagrams).
  3. Transforming the UML models into BPMN models automatically using MDA techniques in order to define the business processes portfolio.

- Second stage: the services elements identification stage. This consists of the following activity:
  4. Defining atomic processes and business entities and then applying clustering metrics to the atomic processes to identify candidate services (e.g. coarse and fine-grained services).

- Third stage: the services evaluation stage. This consists of the following activity:
  5. Evaluating candidate services against the SOEA meta-model to specify the service with optimal level of granularity.

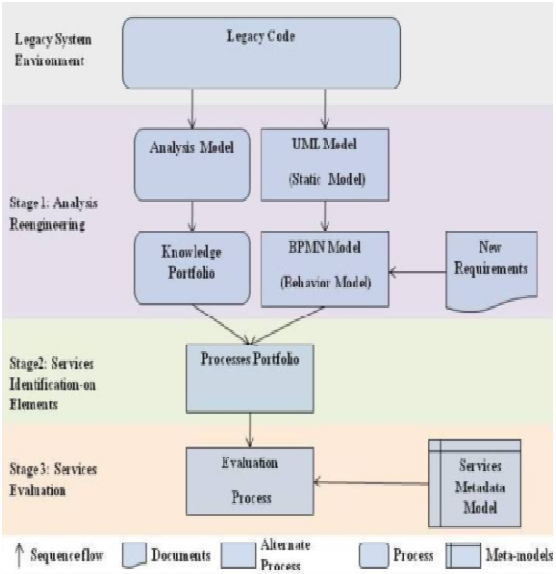

Figure 1. Service identification framework

These three main stages are now considered in greater detail.

A. Analysis and Re-engineering Stage

To construct a sufficient foundation for service identification, the analysis and re-engineering stage consists of three models. First, the analysis model provides high-level system understanding and support for the business requirements. Secondly, the activity portfolio provides the structure of the legacy system. Thirdly, the process portfolio describes the behavior of the legacy system.

The analysis model is based on understanding what the system does because the legacy code itself is not sufficient. Legacy software systems were typically developed with embedded business rules and logic, scattered and duplicated code, unstructured modules, and tightly-coupled functions. The objective of the analysis model is to discover the system main functions and components via workshops, questionnaires, interviews and available documents. The analysis model will complement the modeling of the atomic processes in BPMN and also capture additional information of non-functional requirements such as performance and reusability.

The UML model ensures decomposed units are relative and coarse-grained. Here our aim is to partition the class diagrams in terms of business functions in order to determine primitive functions; the class diagrams are derived from the legacy code using IBM Rational Software Architect and reverse-engineering techniques. Business functions are defined from the knowledge portfolio. We describe the fundamental structural and behavior of the legacy system using UML class diagrams and activity diagrams respectively.

Firstly, the reverse-engineering process results in a class diagram that shows the defined classes of the legacy code providing a static model. The activity diagrams are then defined manually. We stress that it is important to guarantee consistency, completeness, and correctness when devising a behavioral diagram (e.g. activity diagram) from a structural diagram (e.g. class diagram); otherwise all subsequent design representations would inherit the same issues. Several suggested approaches have been proposed to attempt to check the correctness of UML diagrams using different techniques such as verification rules and meta-models [24], profiling inconsistency and evaluation [25]. However, to the best of our knowledge, no tool is currently available from any industry vendor to address these issues which are always affected by model changes [25]. The knowledge portfolio can assist to resolve this issue by reviewing any available business and design documents (including interviews with the system experts).

The BPMN model is used to represent the business processes of the legacy system and also any new requirements. The adoption of process modeling using BPMN as a modeling language is motivated by several factors. Firstly, the existing prior research confirms that process-oriented modeling provides a good basis for SOEA [6, 17, 18]. Secondly, the UML Activity Diagrams for business process modeling introduced by the OMG suffer from significant limitations when modeling related resources and representing various types of control-flow constructs [26]. It is also important to note that BPMN has important advantages over UML when describing complex scenarios including the availability of richer constructs [27].

The use of MDA is an important growing trend, as it provides capabilities to transform a model notation to other models. Our transformation uses the MDA concept to transform UML activity diagrams automatically into BPMN diagrams. The transformation is based on the classification of four categories of patterns that map UML activity diagram elements to BPMN 2.0 elements. The first pattern category is 'Element-To-Element' mapping which has similar direct correspondent semantics in both models (e.g. mapping a *ControlFlow* element in a UML activity diagram to a *SequenceFlow* element in a BPMN diagram). The second pattern category is 'Element-To-Null' mapping which has elements that exist only in UML activity diagram semantics. The third pattern category is 'Element-To-Elements' mapping in which a single UML element corresponds to more than one BPMN elements. The final pattern category is 'Elements-To-Element' mapping which has many UML elements that correspond to a single BPMN element. In our research, at this stage we implement only the transformations for the first pattern category (we will incorporate the other patterns in future work). Fig. 2 shows the entire process of the transformation using Relation Definition Language (RDL) to control the model transformation.

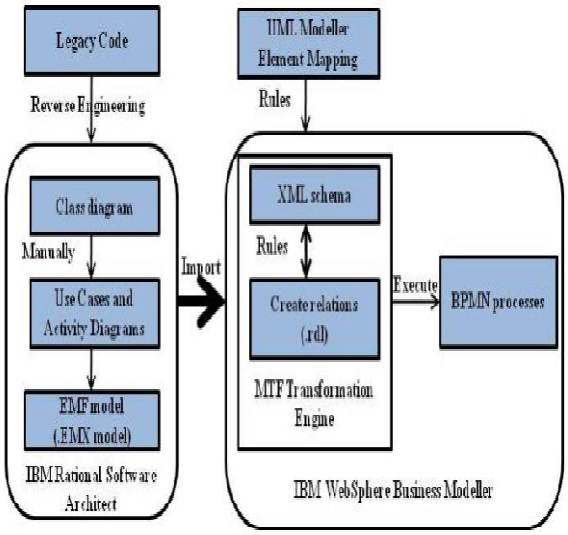

Figure 2. Transformation method between UML and BPMN

The IBM WebSphere Business Modeler tool includes a general XML schema which supports the mapping of an XML file to business processes [28]. Our research utilizes the structure of the XML schema provided with different interpretations of the business processes. After defining our Eclipse Modeling Framework (EMF) model using the Rational Rose tool (as an .emx file), the transformation rules are developed in Relation Definition Language (RDL). The IBM Model Transformation Framework (MTF), which is a set of tools facilitating transformations between Eclipse Modeling Framework (EMF) models, parses the written relations rules. The transformation engine then executes the rules to match both models: the source (UML) model and target (BPMN) model.

B. Service Elements Identification Stage

The service element identification stage is an essential stage for the architecture of service-oriented systems. The term 'optimal service' or 'right service' refers to a service that offers an acceptable size of functionality without interfering with the pre-defined service design concepts. We need to deduce the underpinning elements that assist in identify the 'optimal' services using the three models discussed above (e.g. UML, BPMN and the analysis model). The inputs of this stage are atomic processes and the relative business entities (objects). A bottom-up analysis is conducted to assist in identifying functions and data structures as part of the knowledge portfolio. BPMN models help to describe granular business processes and detailed sub-processes implying different levels of abstractions. For example, the "pool" notation in BPMN represents a business entity or business role in a process which is linked to another entity with "message flow" arrows. In SOEA, this can help to identify initially the size of exchange messages among participants in a service or different services.

In order to identify candidate services, a set of distinct metrics is adopted based on the number of atomic processes and relative business entities (objects). Reference [29] uses only the CRUD functions to identify

the relation between assumed elementary business processes and business entities. Our metrics use executed business logic in addition to the distinct CRUD functions. The clustering metric is also supported by the SOEA meta-model and the classification of service types. Eventually, this classification will provide guidance to an intensive description of each service (e.g. exchanged messages data types). In other words, we focus on what the actual atomic processes are rather than how they are performed. Gathering similar-behavior atomic processes together leads to high cohesion and loose coupling among services. Encapsulating atomic processes and relevant data entities increases reusability and eliminates redundancy. The aim is to disunify groups of atomic processes and business entities into services according to specific rules. The outputs of this stage will be a set of services implementing various service types derived from the activity and process portfolios respectively. These rules are as follow:

- A service can have CRUD functions and executed business logic for either one entity and any number of processes, or any number of business entities and one process.
- A process that has CRUD functions for many entities can be encapsulated in a service with other similar processes and entities.
- A service can not cover CRUD functions and executed business logic for one entity if they are provided by different business processes. Those processes which provide CRUD functions will be defined by a service.
- Every data entity must have at least one creation operation and each atomic process presents a coherent functionality. Every an atomic process invokes at least one entity.

The CRUD functions and business logic identify the relationship between an atomic process and an entity as following:

- "C" means this atomic process CREATES an instance of this business entity.
- "R" means this atomic process READS an instance of this business entity.
- "U" means this atomic process UPDATES an instance of this business entity.
- "D" means this atomic process DELETES an instance of this business entity
- “BL” means this atomic process has business logic belonging to an instance of this business entity.

C. Service Evaluation Stage

The classification of service types establishes a basis to define service granularity. The input of this stage is a set of services with various level of granularity as identified in the previous stage. Elements such as a service contract, an implementation, and an interface can all contribute to this classification and definition [23]. Every service type is explicitly associated with a specific abstract level with implementation description. Because the SOEA view of the enterprise permits a wide variety of potential service types, our design methodology needs to support all possible service scenarios to be complete. We have developed a SOEA meta-model (fig. 3) which presents the relationship between business process characteristics and different service types. We have assumed that the optimal granularity level of the service can be effectively identified based on the combination of our classification and the clustering metric.

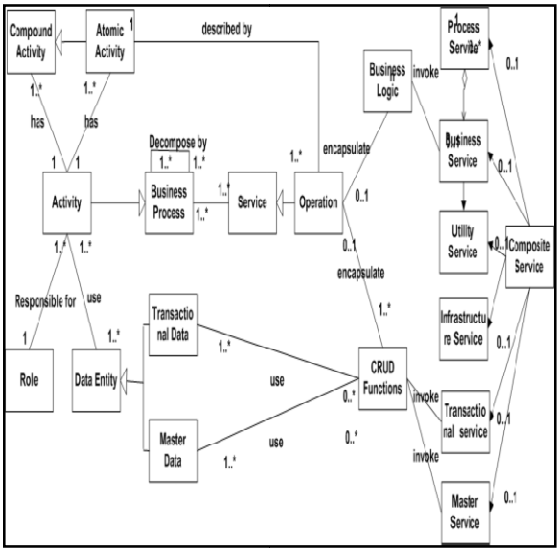

Figure 3. SOEA Meta-Model

Each business process consists of one or more activities and may be also composed of other sub-business processes (activities). Each activity either has one or more atomic activity or is a compound activity (i.e. an atomic activity that is described by several operations). One or more activities belong to a role, which could be a person or department. One or more activities use one or more data entity, which could be a transitional data entity or master data.

The SOEA meta-model provides a comprehensive understanding of two major characteristics: a business process and a service. When we apply our SOEA meta-model (see the example presented later), we generate a new set of services with an 'optimal' level of granularity. To classify the identified services in a process-oriented system, we define the following rules:

- The activities of every process are partitioned into atomic activities (only systematic activities are considered).
- The service type is specified depending on the process functionality (purpose) and interoperability with other processes.
- Every business process can be modeled as one operation or more of a service, or as a service itself.
- A high-level business process can be modeled as a service process type.
- An operation encapsulates either business logic or CRUD functions.
- Business logic can be implemented by a business service or by many business services (as part of service composition).

- One or many business service is orchestrated by a process service and supported by one or many utility services.
- CRUD functions can be implemented by either a transactional service or a master-data service (depending on the type of the data entity) or alternatively as part of similar collaborative services in service composition.
- One or more infrastructure services can be invoked directly or as part of a service composite. Any type of service can be part of one or many composite services.

V. EXAMPLE E-COMMERCE PROJECT

This section describes and discusses a pilot application of our framework to re-engineer a legacy code implementing an e-commerce project [30]. In this example, we concentrate on object-oriented systems. The original objectives of the e-commerce project were to build a servlet-based, object-oriented web store to provide a dynamic, customizable website which would offer online customers and an internal administrator a wide range of functionalities. The web store as originally implemented (i.e. not a SOEA design) consists of six java packages with a total of twenty-eight classes. For brevity, in this example project we have selected the main four java packages (as a sample) to apply our approach to generating a SOEA design from the legacy code.

In the analysis and re-engineering stage, after the legacy code is processed to produce class diagrams, the activity diagrams are created manually as part of the UML model. The activity diagrams are transformed automatically to business processes in BPMN (as described previously). The transformation results in two main business processes with twenty one sub-processes and tasks.

In the service elements identification stage, our approach (as explained in section IV-B) is applied to the atomic processes and business entities. A set of a coarse-grained services (business and process services) are identified implementing cohesive functions with wide ranges of granularity. Table 2 shows the aggregation of relative atomic processes and business entities against candidate services (e.g. coarse-grained services).

In the service evaluation stage, the identified services from the second stage are evaluated using the SOEA meta-model definitions and clustering rules. This stage produces a new set of various types of candidate services with an optimal range of coarse-grained and fine-grained services (as shown in table 3). Table 3 includes two columns: The first column shows processes identified from the processes portfolio after adopting the BPMN standards. The second column lists candidate services with the optimal level of granularity after applying the metric and definitions of SOEA. For example, Pro9 and Pro10 have previously only one service (service_3). After applying the third stage of our approach, we found that (service_3) can be a composite service including a business service (service_3_A) and a transactional-data service (service_3_B). Both of the newly-identified services represent services with optimal level of granularity for this particular set of atomic business processes and business entities.

TABLE II. THE CLUSTER MATRIX

Process/Entities		Customers	Orders	Product	Items Ordered
Pro9	select products from item list	--	--	R	Service_3
Pro10	submit selected products	--	--	R	--
Pro11	create a shopping cart	--	BL	R	Service_4
Pro12	view an order	Service_5	R	--	R
Pro13	submit an order	U	C	--	U
Pro14	delete shopping cart	--	--	U	U
Pro15	cancel an Order	--	D	U	U
Pro16:	modify an order	Service_6	U	U	U/BL
Pro17:	generate a bill		U/BL	Service_7	
Pro18:	provide shipping information	--	U/BL	--	--

TABLE III. OPTIMAL CANDIDATE SERVICES

Process Name	Candidate Services
Pro9: select products from a list	Business service (service_3_A) Transactional-data service (service_3_B)
Pro10: submit selected products	Business service (service_3_C) transactional-data service (service_3_D)
Pro11: create a shopping cart	Business service (service_4_A) transactional-data service (service_4_B)
Pro12: view an order	Transactional-data service (service_5_A)
Pro13: submit an order	Business service (service_5_B)
Pro14: delete shopping cart	Transactional-data service (service_5_C)
Pro15: cancel an order	Business service (service_5_D) Transactional-data service (service_5_E)

VI. EXAMPLEEVALUATION

As the example demonstrated, there are several key requirements in order to identify services with optimal granularity effectively from legacy code. In particular, information in the knowledge portfolio and also distinct functionalities need to be captured in the appropriate formal representations (e.g. UML and BPMN). Having defined business processes and business, a metric can now assist the process of service identification. Because there are no agreed definitions and standards for a service, it is necessary to use an *ad hoc* metric. This approach assists in deriving the optimal services by considering the service purpose (e.g. CRUD functions and business logic) together with the metric rules. Because of the different interpretation of a business process and the different level of abstraction, we define an additional different set of rules for the process portfolio.

The SOEA meta-model provides a foundation for defining the optimal granularity by encapsulating a service type classification. Our evaluation process indicates that the purpose of the service can assist in identifying the service for a process or activity. We need also to note that there are other elements that can be considered along with the purpose of the service. Adding those elements (e.g. service contract and service interface) will again provide a better foundation for

defining the service granularity. They are presently excluded in our approach because they cannot be discussed without considering the service design aspects (e.g. flexibility, reusability, complexity), which are out of our scope at this level The technique used shows that having a well-defined meta-model considering all service definition aspects (e.g. service types) and definitions of business processes can enhance the service identification approach. An outcome and a possible limitation of this approach is that it was not possible to generate automatically UML activity diagrams either from class diagrams or the legacy code. This is the subject of further research aimed at extracting a behavior model from legacy code automatically using a bidirectional MDA transformation.

VII. CONCLUSIONS

A major objective in our research has been to develop a usable approach for identifying the right services for migrated legacy enterprise software systems. Our research emphasizes the importance of the service identification phase for defining the right service, because any faults at the service identification phase can result in compromises affecting the entire SOEA project. We conclude that defining measurement standards for service design aspects (e.g. complexity, flexibility, and reusability) is mandatory to evaluate the service granularity accurately. The main contributions of this paper are a methodology and effective guidelines for the efficient identification of specific services from legacy code, together with the introduction of a meta-model that defines uniquely the characteristics of business processes and service types in the way that definitions are mapped. The paper also emphasizes the importance of the classification of service types to define service properties correctly.

We found that using UML activity diagrams identifies coarse-grained services, whereas using BPMN business process diagrams is suitable to identify both coarse-grained services (as composite services) and fine-grained services, depending on the level of process granularity. Applying our proposed service granularity framework will always require tradeoffs to be made between the different non-functional requirements of the system. These tradeoffs will change from one system to another because they are inherently system dependent. However, our proposed approach represents a significant potential contribution to the field because it defines a novel comprehensive methodology for enterprise engineering based on existing legacy software

Our future work will be to expand this approach to automate the service identification process and hence generate service definitions with the optimal level of granularity automatically. In addition, we will develop the metrics further to help decide the granularity at an early stage of the service model design process. The SOEA meta-model can also be extended to include other essential enterprise resources at the service operational level to support service delivery. We also plan to enhance the transformation technique between UML and BPMN to enable the generation of a schema in an XMI format which will allow us to map major constructs efficiently between the different business models.

ACKNOWLEDGMENT

We acknowledge helpful discussions with Rob Phippen and Kim Clark (IBM Hursley Park, UK).

REFERENCES

[1] A Arsanjani, , “Service-oriented modeling and architecture: How to identify, specify, and realize services for your SOA”, online,Available at: http://www.128.ibm.com/developworks/webservices/library/ws-soa-design1/ [Accessed 30/12/2009]

[2] OMG, “OMG and Service-Oriented Architecture", The OMG Standard, Vol. 2, Issue 1, pp. 3-4, 2006.

[3] F. Kamoun, “ A Roadmap Towards the Convergence of Business Process Management and Service Oriented Architecture”. Ubiquity, 8(14), ACM Press, 2007.

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

[5] K. Suntae, M. Kim, S. Park, “Service identification using goal and scenario in service oriented architecture”, Proc. 15th Asia-Pacific Software Engineering Conference, IEEE Computer Soc., 2008.

[6] 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., Institute of Technology Linkoping University, 2007.

[7] H.M. Sneed, “Integrating legacy software into a service oriented architecture”, Proc. 10th European Conference on Software Maintenance and Reengineering, IEEE Computer Soc., 2006.

[8] N. Fareghzadeh, “Service identification approach to SOA development”, Proc. . World Academy of Science, Engineering and Technology, 35, .2008.

[9] Y. Zou, K. Kontogiannis, K.,”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, 2001.

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

[11] 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.

[12] Z. Zhang, L. Ruimin, H. Yang, “Service identification and packaging in service-oriented reengineering”, Proc. of the 17th International Conference on Software Engineering and Knowledge Engineering, IEEE Computer Soc., 2005.

[13] Z. Zhang, H. Yang, “Incubating services in legacy systems for architectural migration”, Proc. 11th Asia-Pacific Software Engineering Conference, IEEE Computer Soc., 2004.

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

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

[16] J. Lawson,“Data services in SOA: maximizing the Benefits in enterprise architecture”,Oracle, [Online] April, 2009. Avilable at : http://www.oracle.com/techonlogy/pub/articles/jj_lawson_soa_data.html. [Accessed 30/12/2009].

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

[18] A. Erradi, N. Naveen Kulkarni, P. Maheshwari, “Service design process for reusable services: financial services case study”, ICSOC, pp.606-617,2007

[19] H.M Shirazi, N. Fareghzadeh, A.Seyyedi, “A combinational approach to service identification in SOA”, Journal of Applied Sciences Research, INSInet Publication, 5(10), pp. 1390-1397, 2009.

[20] V. Dwivedi, N. Kulkarni, “A model driven service identification approach for process centric systems”, Proc. 2008 IEEE Congress on Services Part II (SERVICES-2), pp. 65-72, IEEE Computer Soc., 2008.

[21] N. Kulkarni, V. Dwivedi, “The role of service granularity in a successful SOA realization - A Case Study”, IEEE Congress on Services. Honolulu, Hawaii, IEEE Computer Soc., 2008.

[22] R. Haesen, M. Snoeck, W. Lemahieu, S. Poelmanset, “On the definition of service granularity and its architectural impact”, Proc. Advanced Information Systems Engineering. 20th International Conference, CAiSE 2008, pp. 375-389, Springer-Verrlag, 2008

[23] C. Steghuis, “Service granularity in SOA projects: a trade-off Analysis”, Master’s thesis, University of Twente, 2006.

[24] H. L-K., Kyu, B-W Kang, “Meta-Validation of UML structural diagrams and behavioral diagrams with consistency rules”, Proc. of IEEE Pacific Rim Conf on Communications, Computers and Signal Processing, PACRIM, 2, pp. 28-30, 2003.

[25] A. Egyed, “UML/analyzer: a tool for the instant consistency checking of UML models”, Proc. 29th International Conference on Software Engineering (ICSE'07), pp.787-790, IEEE Computer Soc., 2007.

[26] N. Russell, Van der Aalst, W.M.P., A.H.M. Ter Hofstede, P. Wohed, “ On the suitability of UML 2.0 activity diagrams for business process modelling”, BPM Centre Report BPM-06-03, BPMcenter.org, 2006.

[27] J. Recker, M. Muehlen, K..Siau, J. John, M. Indulska, “Measuring method complexity: UML versus BPMN”, 15th Americas Conference on Information Systems, San Francisco, California, 2009.

[28] C. Griffen, R. Huang, Z. Sen, M. Fiammante, “Transforming UML «Activity» diagrams to WebSphere Business Modeler processes”. IBM WebSphere Developer Technical Journal,10.6, July, 2007.

[29] P. Jamshidi, M. Sharifi, S. Mansour, “To establish enterprise service model from enterprise business model”, Proc. IEEE International Conference on Services Computing, pp. 93-100,,IEEE Computer Soc., 2008.

[30] S. Alahmari, “ Web Store Front ” , Master ’ s thesis.,Department of Computing Sciences, University of Scranton, Pennsylvania, 2002.