Journal of Software Engineering and Applications, 2014, 7, 770-778
Published Online August 2014 in SciRes. http://www.scirp.org/journal/jsea
http://dx.doi.org/10.4236/jsea.2014.79071

Extracting SOA Candidate Software Services from an Organization's Object Oriented Models

Rana Yousef, Omar Adwan, Mohammad A. M. Abushariah

Computer Information Systems Department, King Abdullah II School for Information Technology,
The University of Jordan, Amman, Jordan
Email: rana.yousef@ju.edu.jo, adwanoy@ju.edu.jo, m.abushariah@ju.edu.jo

Received 25 May 2014; revised 20 June 2014; accepted 15 July 2014

Abstract

Class diagrams and use case models are system models that are used to analyze, design and model object oriented systems. In this era of agile computing, service-oriented architecture has become increasingly popular for achieving efficient and agile business solutions that can maintain changes demanded by the business world. This paper proposes a methodology to identify services from a set of class diagrams and use case models in order to generate a service oriented model. An extensive evaluation of the generated services has shown that these services conform to the principles of Service Oriented Architecture (SOA), and provide a straightforward methodology, which can reuse the valuable business logic that resides within legacy applications to migrate to SOA-based systems.

Keywords

Service Oriented Architectures, Service Identification, Candidate Software Service, Class Diagrams, Affinity Analysis

1. Introduction

A Service Oriented Architecture (SOA) is a software architectural style that defines the use of loosely coupled and inter-operable software services to support the requirement of the business processes and software users [1] [2]. Service Identification (SI) is one of the main activities in the process of developing service oriented models, and is the first step required for organization's migration from legacy system to a SOA-based system.

System models represent the actual desired behavior of legacy systems; they represent abstract implementa-

How to cite this paper: Yousef, R., Adwan, O. and Abushariah, M.A.M. (2014) Extracting SOA Candidate Software Services from an Organization's Object Oriented Models. *Journal of Software Engineering and Applications*, **7**, 770-778.
http://dx.doi.org/10.4236/jsea.2014.79071

tions of the actual system. Therefore, almost all organizations urge that these models should be available at any time. Class diagrams and use case models are inevitable models that are required for object oriented design and development.

Class diagrams are used to document the structure of the system; they represent classes and the relationships between them, whereas use case models document the behavior of the system from the users' points of view. An individual use case represents a kind of task which has to be done with support from the system under development. An actor represents a kind of user of the system, who/which is anything external to the system that interacts with it [3].

Since most of the available systems are object oriented based systems, this paper aims at helping organizations to reuse the valuable business logic that resides within legacy applications to migrate to SOA-based system. In addition, this paper specifically presents a new service identification approach to identify candidate services starting from available class diagrams and use case models.

Section 2 provides background knowledge about common service identification approaches as well as related research efforts to our work. Section 3 describes the method of generating services from class diagrams and use case models. The proposed method using a case study from the healthcare domain is demonstrated in Section 4, and the proposed service identification approach is evaluated in Section 5. Finally, Section 6 summarizes and concludes the paper.

2. Background and Related Work

2.1. Common Service Identification Methods

Table 1 represents some service identification methods that are commonly found in literature, however, a service identification approach is actually a combination of some of these methods, so that a certain criterion can distinguish this approach [4]. Many research papers have compared different SI methodologies, some of them can be found in [5]-[8].

2.2. Related Work

Service identification approaches have been extensively investigated in literature, ranging from top-down to bo-

Table 1. Common service identification methods.

Service Identification Methods	Method Description
Business Process Decomposition	Using this method, the business process is divided into sub-processes or decomposed into smaller activities and tasks. The lowest level tasks can consist of small, cohesive "logical units of work" that can be regarded as services.
Business Functions	This method starts from a business function model, where the most detailed business functions in the functional decomposition are translated into services. Each service is determined to employ only one service operation definition to render a message specification of a respective activity that is associated with each service.
Business Entity Objects	By modeling services according to business object models, business entity-based services can be identified commonly requiring CRUD-type functions (create, read, update and delete).
Goal-Driven	With this approach a project team decomposes a company's goals down to the level of services. In this context, a service is regarded as a goal that can be executed through automated support.
Component-Based	Components are self-contained units of functionality. Various methods to identify components have already been introduced. The responsibilities of each component have to be defined as precisely as possible. These responsibilities can be used as a starting point for identifying services.
Existing Supply (Bottom-Up)	Services are defined based on immediate requirements for information and functionality. In this case, the starting point is the functionality provided by existing legacy applications. And then the systems that provide the bulk of the automated support required by the primary business processes are selected.

ttom-up. Service identification approaches use process models in different ways; some provide general guidelines for identifying services from business process models, others use algorithmic and/or sematic methods. For example in [9] the authors emphasized the need to decompose business processes in order to effectively identify services. They accordingly proposed using the separation of concern principle to facilitate the consistent decomposition of a business process and the identification of its atomic activities. Klose *et al.* [10] proposed a method for the identifying services from a business point of view based on process models. In their method, functions are implemented and provided as a service, only if both business potential and technical feasibility have been verified. In [11] service model was created according to a process model that has activities and a process flow. Each service was determined to employ only one service operation definition to render a message specification of a respective activity that is associated with each service. The activities, the process flow, and the message specification were utilized to produce the service component architecture module in executable implementations.

A formal approach for service identification from the business process model is suggested by Kim and Doh [12]. The authors used graph clustering and provided a systematic approach for service identification by defining the cost metric as a measure of the interaction costs. In order to extract service information from the business model, they take activities as the smallest units in service identification and cluster activities with high interaction cost into a task through hierarchical clustering algorithm, so as to reduce the coupling of remote tasks and to increase local task cohesion.

Kim *et al.* [13] pointed out that business goals and business change factors must be analyzed because SOA aims to achieve business goals and business agility in a changing business environment. Accordingly, the authors proposed a service identification method based on goal-scenario modelling and a conceptual framework to elicit possible business changes. Traceability among business goals, business changes and identified services were also constructed in this approach.

Boerner and Goeken [6] have drawn attention to the lack of economic and governance aspects which constitutes a problem in SI. The authors propose a process-oriented method of SI. This approach includes the business point of view, strategic and economic aspects as well as technical feasibility. Taking these aspects into consideration supports SOA governance.

Alahmari and Zaluska [14] have argued that important aspects of service granularity is not addressed in the wide range of current migration techniques for legacy systems in different implementations technologies where these aspect affect service reusability, governance, maintainability and cohesion. In their paper, the authors proposed a framework for the effective identification of the key services in legacy code. The approach focuses on defining the right services based on standardized modeling languages (UML and BPMN). The framework provides guidelines for optimal service granularity for different service types.

Fareghzadeh [15] proposed a method for service identification that combines multiple approaches and advantages and tries to avoid the disadvantages of each. The method is based on business process analysis coupled with use cases and existing assets analysis and goal service modelling.

Jamshidi, *et al.* [16] realized that the key activities that are needed to construct a quality based service-oriented solution is the identification of its architectural elements with the right granularity. Accordingly, they proposed a process to identify and specify enterprise software services along with their architectural elements. They presented a method for identifying services based on affinity clustering, Elementary business process and business Entity Affinity analysis Technique (EEAT).

Compared to the current service identification approaches, our proposed methodology considers an organization’s set of class diagrams and use case models in order to reuse the valuable business logic that resides within legacy applications. Accordingly, our approach does not start with business process models to analyze activities in different levels of complexity, but starts with the models that describe the system structure and system interactions providing the required information to specify the service boundaries.

3. Identifying Candidate Services from Object Oriented Models

Algorithm 1 describes the methodology proposed in this research for identifying candidate services from a set of class diagrams and use case models. This includes the following:

1) Extracting entities from class diagrams
2) Extracting tasks from use case models
3) Creating a CRUD Matrix

4) Performing affinity analysis

Algorithm 1: Identifying Candidate Software Services from object oriented models

```
Input: the set of Class Diagrams, $CD = \{cd_1, cd_2, \ldots, cd_i, \ldots, cd_n\}, 0 \le I \le n$, and the set of use-case diagrams, $UCD = \{ucd_1, ucd_2, \ldots, ucd_x, \ldots, ucd_y\}, 0 \le x \le y$
Output: The set of the identified candidate services $S = \{s_1, s_2, \ldots, s_j, \ldots, s_m\}, 0 \le j \le m$
Begin
For each class diagram $cd_i$ in CD do
1) Add each class name to the set of entities $E = \{e_1, e_2, \ldots, e_l, \ldots, e_y\}, 0 \le l \le y$
2) End for
3) For each use-case diagram $ucd_x$do
4) Add each use case functionality as to the set of tasks $T = \{t_1, t_2, \ldots, t_k, \ldots, t_x\}, 0 \le k \le x$
5) End for
6) Build the CRUD matrix as follows:
7) for each task $t_k$ in T do
8) set $t_k$ as a row in the CRUD matrix
9) for each business entity, $e_l$ in E do:
10) set $e_l$ as a column in the CRUD matrix
11) Set the matrix cell as the relationship between $t_k$ and $e_l$ which is one of the CRUD functions (Create, Read, Update or Delete)
12) Indicate the boundary of each service as a cluster of the service capabilities and entities on which the capabilities act. $C = \{c_1, c_2, \ldots, c_p, \ldots, c_z\}, 0 \le p \le z$
13) Assign each group of capabilities a service name to produce the set of candidate services: $S = \{s_1, s_2, \ldots, s_j, \ldots, s_m\}, 0 \le j \le m$
End
```

Lines 1 - 3 of algorithm 1 extract entities from each class diagram by considering each class name as an entity. This ensures that we can identify each essential entity of the organization. This is because a class is originally identified by picking all the nouns and noun phrases out of a requirements specification of the system, and discarding inappropriate candidates such as redundant, vague, events, operations, metalanguage and attributes [1].

Lines 4 - 6 of algorithm 1 extract tasks from each use case model by considering each use case as well as any included use case as a task. This ensures identifying all functionalities of the organization.

The SI criterion used in [16] is based on affinity clustering where the aim is to deduce groups of functions and entities that share create and update operations. This aim is accomplished by using the Elementary business process and business Entity Affinity analysis Technique (EEAT) [16]. So, grouping together all functions that create and update the same entities defines non-redundant building blocks or candidate software services.

Accordingly, a CRUD matrix was built after identifying the tasks which form the matrix rows and the entities which form the matrix columns (lines 7 - 12), then using the affinity clustering technique, candidate services were identified as the groups of tasks that create and update the same entities (Lines 13 - 14).

4. Demonstrating the Proposed Method Using a Case Study from the Healthcare Domain

4.1. The Cancer Care and Registration Case Study [1]

The Jordan's Cancer Care and Registration (CCR) case study [17] is a real case study that has been validated and improved. It provides a number of business process models (BPM) which we have used to demonstrate the process of deriving the candidate software services. In this regard, the CCR case study is rich in entities which can be identified from the set of class diagrams. Also, the boundaries of each service can be identified from the comprehensive representation of the activities and their relationships in the CCR class diagrams.

Information in the CCR process was derived by Abu Rub [17] from interviews with process actors, and observation of the workflow of processes. Many interviews were conducted with many stakeholders both in (1) King Al-Hussein Cancer Centre (KHCC), which is the only centre in Jordan specialising in cancer care, and (2) the Jordan Cancer Registry (JCR) (Amman-Jordan).

The main participants and their roles within the CCR process were identified as follows [17] [18]:

1) Patient: aims to get suitable treatment and follow-up by interacting with the other roles;

2) Receptionist (outpatient clinic): responsible for arranging appointments with specialists for diagnosed patients;

3) Receptionist (detection unit): responsible for arranging appointments for non-diagnosed patients with doctors and for registering patients' details;

4) Medical records clerk: responsible for essential administrative tasks in the CCR process, such as managing patients files, registering patient's details, generating main statistical reports for the hospital, performing hospital cancer registration and sending required information to the Jordan Cancer Registry (JCR);

5) Physician (Diagnostician): responsible for diagnosing the non-diagnosed patients and identify cancer type;

6) Specialist: responsible for providing suitable treatment for the diagnosed patients as well as follow up;

7) Laboratory: performs required tests on patients and provide results;

8) Imaging department: responsible for performing required investigations on patients and providing the results;

9) Admission clerk: responsible for arranging patients' admission to the hospital; and

10) Jordan Cancer Registry (JCR): responsible for collecting data about cancer cases in Jordan from different laboratories and hospitals, classifying, analysing the data and generating statistical reports about cancer incidents in Jordan.

Figure 1 shows part of the CCR class diagram, and **Figure 2** presents part of the CCR use case model.

4.2. Identifying Candidate Services for the CCR Case Study

Figure 3 shows part of the CRUD matrix which results after executing lines 7 - 12 of Algorithm 1. A CRUD matrix is developed using the information provided in the previous step. As can be seen from **Figure 3**, services are bounded in column A. Also, for each service, the cluster containing the capabilities and entities are bounded with a dashed line.

Table 2 represents the final output of Algorithm 1 when applied to the CCR case study. In this table, the set

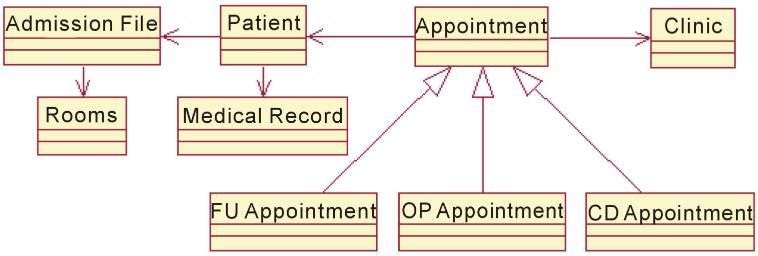


Figure 1. Part of the CCR class diagram.

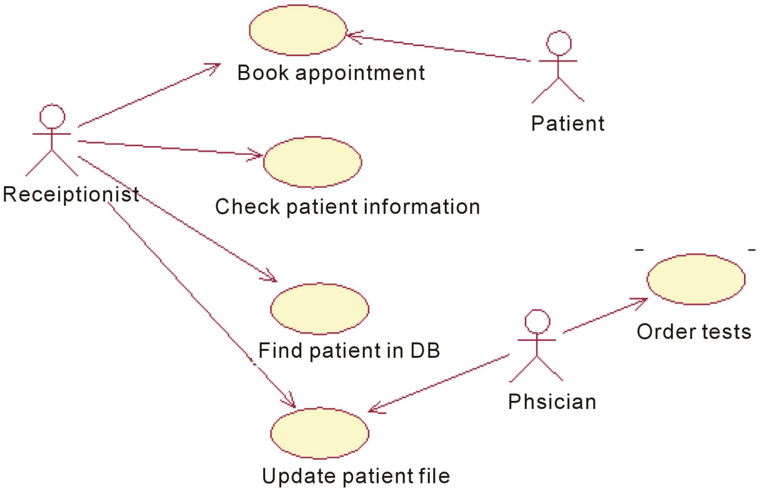
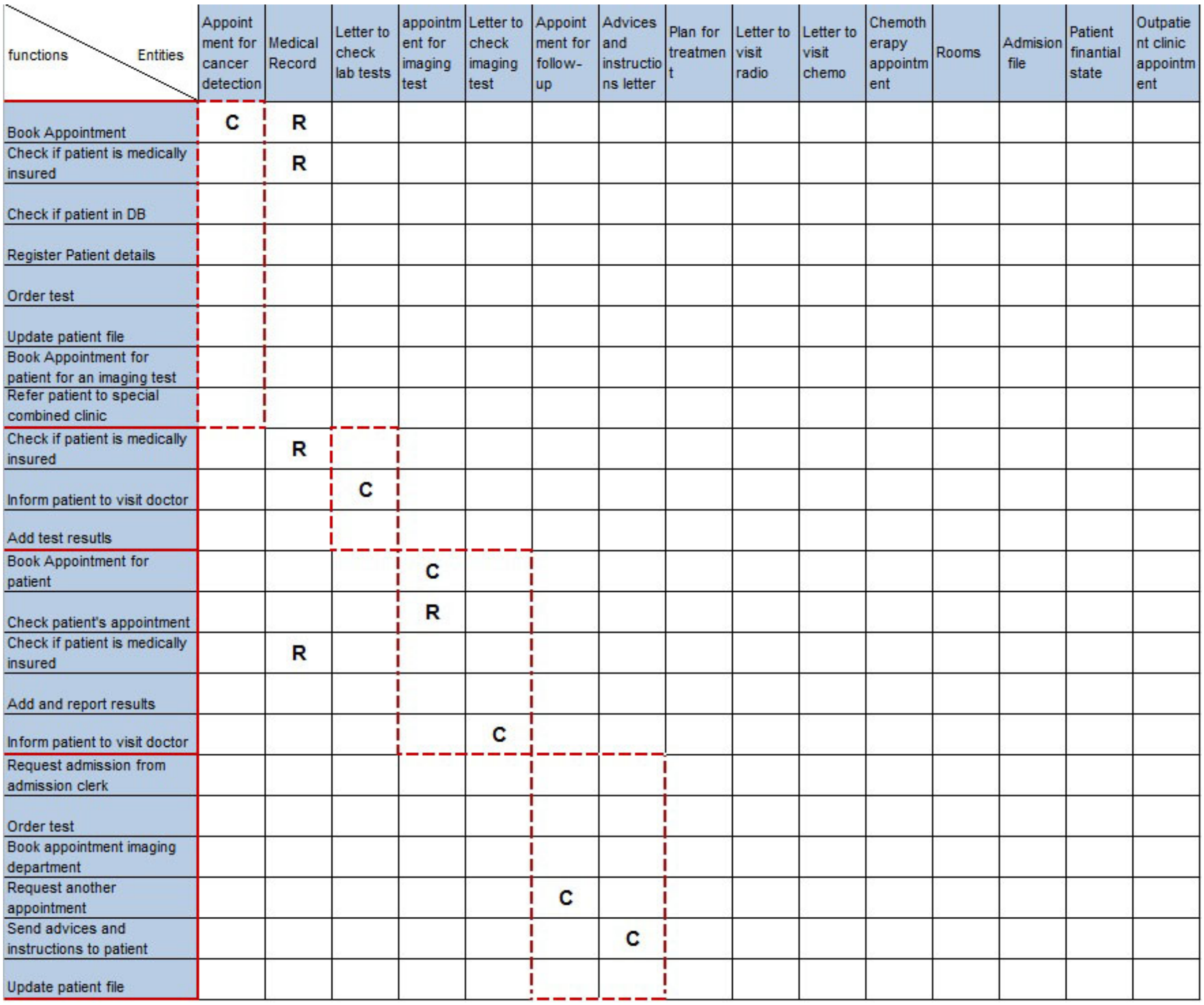


Figure 2. Part of the CCR Use case diagram.



functions Entities	Appointment for cancer detection	Medical Record	Letter to check lab tests	appointment for imaging test	Letter to check imaging test	Appointment for follow-up	Advices and instructions letter	Plan for treatment	Letter to visit radio	Letter to visit chemo	Chemotherapy appointment	Rooms	Admision file	Patient finantial state	Outpatient clinic appointment
Book Appointment	C	R													
Check if patient is medically insured		R													
Check if patient in DB															
Register Patient details															
Order test															
Update patient file															
Book Appointment for patient for an imaging test															
Refer patient to special combined clinic															
Check if patient is medically insured		R													
Inform patient to visit doctor			C												
Add test resutls															
Book Appointment for patient				C											
Check patient's appointment				R											
Check if patient is medically insured		R													
Add and report results															
Inform patient to visit doctor					C										
Request admission from admission clerk															
Order test															
Book appointment imaging department															
Request another appointment						C									
Send advices and instructions to patient							C								
Update patient file															

Figure 3. Part of the CRUD Matrix for the CCR Case Study.

of identified candidate software services are presented, where services were named to represent the functional boundary encapsulated within each defined service. **Table 2** also lists the set of capabilities associated with each service.

All of the information required to derive the candidate software services for the CCR case study is available from the set of class diagrams and use case models, where we have extracted the entities, the tasks and the required relations between them.

As can be seen from **Table 2**, thirteen candidate software services were identified using Algorithm 1. Each of these services can be implemented as an abstract, reusable and composable software component. And the resulting set of services will be loosely coupled and interoperable.

Analyzing the capabilities of each service, we can figure out the loose coupling of the identified services. Each identified service is a stand-alone cluster which has low dependability on other clusters. These clusters act as black boxes, where they abstract the underlying functionalities. The identified services are also highly reusable and are composable. The granularity level is not too coarse-grained nor too fine-grained.

It is accordingly obvious that a service oriented architectural model is suitable for such processes, where the identified services act as a set of loosely coupled and interoperable components that could be reused not only in the same organization (KHCC) but within the whole enterprise or the healthcare domain.

5. Evaluating the New Service Identification Approach

In this section we evaluate the introduced service identification approach in terms of its conformance with SOA

Table 2. Identified candidate software services for the CCR case study.

Candidate Software Service	Capabilities	
S1: General Reception	Appointment booking	Check if patient in DB
	Transfer patient to emergency	Check if patient has appointment
	Inform patient to visit cancer detection unit	Inform patient to visit specialist
	Check if patient diagnosed	Register patient details
S2: Specific Reception and Clerk Work	Check patient appointment	Find patient's address
	Check if patient have medical insurance	Contact patient
	Make list of patients have not come to their appointments	Inform patient's specialist to update files
	Send list to registrar	Update patient file
	Collect data	Collect data from different departments
	Add the collected data to DB	Analyse collected data
	Collect patients' files that have been discharged	Generate main statistical reports
S3: Cancer Detection	Book appointment	Order test
	Check if patient is medically insured	Update patients file
	Check if patient in DB	Book appointment for patient for an imaging test
	Register patient details	Refer patient to special combined clinic
S4: Lab Test	Check if patient medically insured	Add test results
	Inform patient to visit doctor	
S5: Imaging Test	Book appointment for patient	Add and report results
	Check patient's appointment	Inform patient to visit doctor
	Check if patient is medically insured	
S6: Patient Follow-Up	Request admission from admission clerk	Request another appointment
	Order test	Send advises and instructions to patient
	Book appointment imaging department	Update patient file
S7: Patient Treatment	Review patient's history and investigations	Inform patient to visit radio
	Request admission	Book appointment for radiotherapy treatment
	Order tests	Inform patient to visit chemotherapy department
	Book appointment imaging department	Book appointment for chemotherapy treatment
	Device plan for treatment	
S8: Radiotherapy Treatment	Check if patient has appointment	Transfer patient
	Check if patient is medically insured	Add results
S9: Chemotherapy Treatment	Receive request for appointment booking	Check if patient is medically insured
	Check if patient has appointment	Add results
S10: Patient Admission and Care	Check room availability	Open admission file
	Add patient to waiting list	Add notes to file
	Complete paper work	Update patient file
S11: Inpatient Care Follow-Up	Specialists review resident doctors' orders, diagnose patients and review old tests	Check patient financial state
	Check patient financial state	Approve patient financial state
	Specialist makes appointment in outpatient clinic with patient	
S12: Hospital Registration	Extract main details about cancer patients	Add primary tumor
	Check if there is any contradictable data	Add any additional information
	Check if patient exist in DB	Generate reports about cancer incidents in the hospital
	Add patient's details to DB	Add required details in JCR form
	Check if primary tumor exist in DB	Make copy of pathology reports and death certificate
S13: Patient Medical Record	Find patient's file	Open file
	Register file's details	Save patient's file in library
	Send patient's file	

principles.

By conducting a thorough analysis of the set of candidate services that were generated using the CCR case study, we provide a discussion in order to assess the extent to which service definitions and principles which were articulated by Erlin [1] [2] are satisfied by the suggested identified services. This is summarized in **Table 3**.

An entity service is a service whose functional boundary and context is based on one or more related business entities [19]. This makes it highly reusable, where a single entity service can be leveraged to automate multiple parent business processes.

We can deduce that our identified services conform to the above definition, where each was defined as entities. Hence, each class's boundary is based upon one business entity.

The set of tasks that were grouped together depend on each other, but not on other groups, because they create or update the same set of entities, however, none of the tasks in a group can do the same for another one, *i.e.* services are loosely coupled and hence satisfy SOA loose coupling principle.

We can detect the direct dependencies between the identified group members in the case study, and the lower dependencies between the services. Service S1 of the CCR case study, for example, is concerned with general reception functions, which in turn leads to booking appointments, transferring patients to emergency, informing him to visit cancer detection unit and check if a patient is diagnosed. All of these functions depend on each other but not on functions from other services.

As each identified service corresponds to a set of related functionalities presented with a high abstraction level, they act as black boxes abstracting underlying functionalities that are considered as the associated service capabilities. This also hides interaction details leaving services stateless and hence satisfying the SOA principle of Statelessness.

Reusability is an important principle of services to be "SOA-able", and it is related to other service principles, such as loose coupling, composability and statelessness. From both the class definition and use case definition, we can infer that the identified services are reusable. In addition, we note that these services have granularity levels that are not too fine-grained nor too coarse-grained.

6. Conclusions

In this paper, we have proposed a novel approach for identifying services starting from an organization's set of class diagrams and use case models. These models provide the required information to build a CRUD matrix

Table 3. Mapping the characteristics of classes and use cases to service definitions and principles.

Characteristics of a Class and a Use case	Service Principles and/or Definitions	Mapped service characteristic to Service Definitions and/or Principles
Each class represents a tangible or real-word things, roles, events, interactions. They accordingly describe a set of objects with an equivalent role or roles in a system	Entity Service definition	The functional boundary of service is based upon one or more business entities.
Each group of classes are dependent on each other where they are immediately related through association or generalization relationships. However, each identified group have relatively lower dependency on other groups where relations between them don't exceed read relationship.	Principle of Loose Coupling	Each service has low dependability on other serice.
Each class defines an abstract description of a group of concepts that encapsulates functionalities and behaviours	Principle of Abstracting Underlying Logic	Each service acts as black box, where they abstract the underlying functionalities that are considered service capabilities.
Classes are concerned with one or more related entities, allowing classes to be reused, through the object orientation cocncepts such as inheritance and polymorphism	Principle of Reusability/ Principle of Composability	Identified services are highly reusable and are composable. The granularity level is not too coarse-grained nor too fine-grained.
Classes are related through association, generalization and composition relations (*i.e.* relations between class are not conversational).	Principle of Statelessness	Services minimize the amount of state information they manage.

which, after performing affinity analysis, can group together functions and entities that share create and update operations. This conforms to SOA principles such as loose coupling and interoperability. In addition, the use of classes as entities ensures reusability and abstracting underlying logic.

We have demonstrated the process of identifying software services for a SOA-based application using a real case study from the healthcare domain, namely the Cancer Care and Registration (CCR) in Jordan [17].

Choosing the CCR processes case study contributes as part of the research conducted in the "SOA and Healthcare" domain. SOA provides a solution to many challenges faced by healthcare organizations which try to improve their operations, efficiency and operational capabilities as well as managing costs in a more effective way.

The Healthcare domain includes many business lines such as pharmacy, laboratory, nursing, patient billing, accident and emergency, scheduling referral management, accounts, etc. One of the key benefits of SOA in healthcare is that it provides a means to extract common functions from across multiple business lines.

Therefore, enacting the service identification approach using the CCR processes in Jordan case study may provide the initiatives to deploy the SOA industry in the healthcare domain in Jordan.

References

[1] Erl, T. (2005) Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall, Upper Saddle River.

[2] Erl, T. (2007) SOA: Principles of Service Design. Prentice Hall, Upper Saddle River.

[3] Stevens, P. and Pooley, R. (2006) Using UML, Software Engineering with Objects and Components. 2nd Edition, Addison-Wesley, New York.

[4] Hubbers, J., Ligthart, A. and Terlouw, L. (2012) Ten Ways To Identify Services. *The SOA Magazine*, **13**, 1-7.

[5] Terlouw, L.I. and Albani, A. (2009) Identifying Services in SOA. ICRIS White Paper Series, ICRIS B.V.

[6] Boerner, R. and Goeken, M. (2009) Process-Oriented Service Identification, a Method for Business-Driven Service Modeling.

[7] Kohlborn, T., Korthaus, A., Chan, T. and Rosemann, M. (2009) Identification and Analysis of Business and Software Services—A Consolidated Approach. *IEEE Transactions on Services Computing*, **2**, 50-64. http://dx.doi.org/10.1109/TSC.2009.6

[8] Baghdadi, Y. (2013) A Comparison Framework for Service-Oriented Software Engineering Approaches: Issues and Solutions. *International Journal of Web Information Systems*, **9**, 279-316. http://dx.doi.org/10.1108/IJWIS-03-2013-0006

[9] Caetano, A., Silva, A.R. and Tribolet, J. (2010) Identification of Services through Functional Decomposition of Business Processes. In: *Business Information Systems*, Springer, Berlin, Heidelberg, 144-157.

[10] Klose, K., Knackstedt, R. and Beverungen, D. (2007) Identification of Services—A Stakeholder-based Approach to SOA Development and its Application in the Area of Production Planning. *ECIS*'07, 1802-1814.

[11] Kejriwal, M.K. and Raghuraman, L. (2014) Generating a Service Component Architecture (SCA) Module with Service Oriented Architecture (SOA) Model Elements. US Patent No. 8640085.

[12] Kim, Y. and Doh, K. (2009) Formal Identification of Right-Grained Services for Service-Oriented Modeling. In: *Web Information Systems Engineering*, Springer, Berlin, Heidelberg, 261-273.

[13] Kim, S., Kim, M. and Park, S. (2008) Service Identification Using Goal and Scenario in Service Oriented Architecture. 15*th Asia-Pacific Software Engineering Conference*, Beijing, 3-5 December 2008, 419-426.

[14] Alahmari, S. and Zaluska, E. (2009) Optimal Granularity for Service-Oriented Systems. *The* 3*rd Saudi International Conference*, Guildford, 5-6 June 2009.

[15] Fareghzadeh, N. (2008) Service Identification Approach to SOA Development. *World Academy of Science*, *Engineering and Technology*, **45**, 258-266.

[16] Jamshidi, P., Sharifi, M. and Mansour, S. (2008) To Establish Enterprise Service Model from Enterprise Business Model. *IEEE International Conference on Services Computing*, **1**, 93-100.

[17] Rub, F.A., (2006) Process Modelling for Business Process Improvement, with Particular Reference to Healthcare Processes. Ph.D. Thesis, The University of the West of England, Bristol.

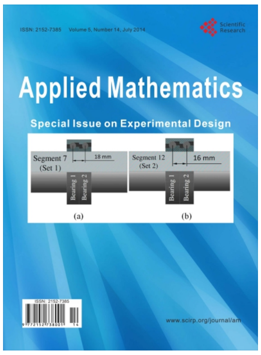
[18] Rub, F.A., Odeh, M., Beeson, I., Pheby, D. and Codling, B. (2008) Modelling Healthcare Processes Using Role Activity Diagramming. *International Journal of Modelling and Simulation*, **28**, 147-155.

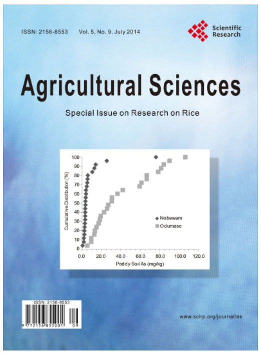
[19] Arcitura Education Inc. (2010) SOA Glossary. http://serviceorientation.com/soaglossary/index

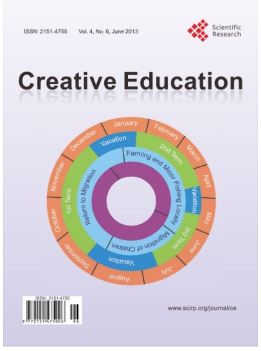
Advances in
Bioscience and Biotechnology

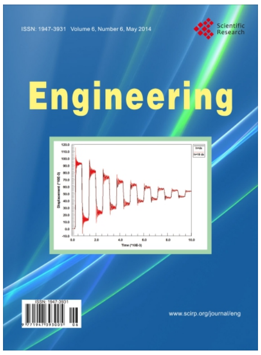
American Journal of
Analytical Chemistry

American Journal of
Plant Sciences
Special Issue on Photosynthesis

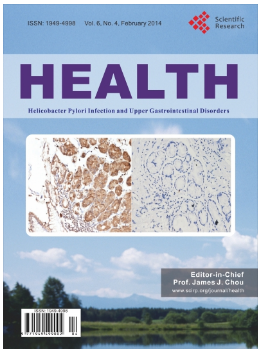

Applied Mathematics
Special Issue on Experimental Design


Agricultural Sciences
Special Issue on Research on Rice


Creative Education

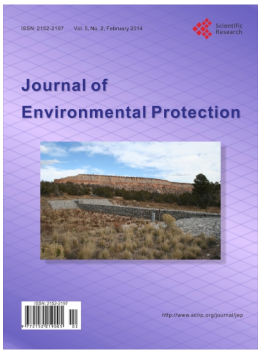

Engineering

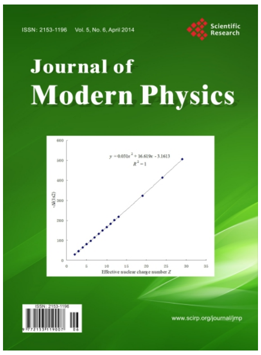
Food and Nutrition Sciences


HEALTH
Helicobacter Pylori Infection and Upper Gastrointestinal Disorders
Editor-in-Chief
Prof. James J. Chou

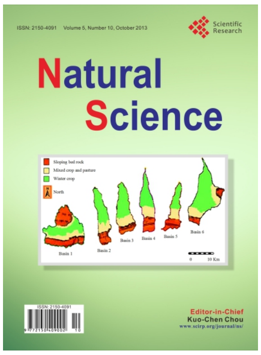
Journal of Computer
and Communications

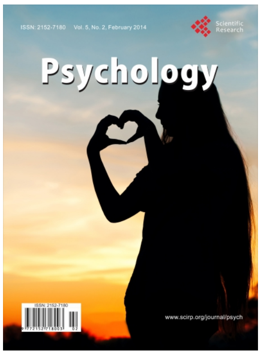
Journal of Cancer Therapy
Special Issue on Bone Tumor Research


Journal of
Environmental Protection


Journal of
Modern Physics

Modern Economy


Natural
Science
Editor-in-Chief
Kuo-Chen Chou


Psychology