Assessing the Quality of Candidate Software Services Generated Using Service Identification Approaches

Rana Yousef

Computer Information Systems Department, The University of Jordan
Amman, Jordan

Abstract
Service identification (SI) is one of the challenging activities in developing service oriented models. Many service identification approaches have been proposed in literature. However, the literature lacks enough research in the field of assessing the quality of the candidate services identified using such approaches. Most of the available techniques to validate SI approaches are theoretical in nature and uses case studies and/or examples to explain the approach's features. In this paper, we present a framework to assess the quality of services identified using SI approaches. This framework is used to assess the granularity level, reusability, statelessness, composability, cohesion and low coupling of the identified services. The assessment framework is demonstrated using a real case study from the healthcare domain.
Keywords: *Service oriented architecture, service quality assessment, service identification, quality of service, SOA principles.*

1. Introduction

Service-oriented modeling and design consists of three main steps: service identification (SI), specification and realization of services [1]. The identification of services is concerned with determining the appropriate services to be implemented in a service-oriented architecture and defining which functions should be part of each service, most of the current approaches relay on business process descriptions to identify services [2, 3].

In literature, most of the service identification approaches are evaluated by deploying them into practice. Some approaches are validated by showing their effectiveness in real life projects or by experimenting them in case studies. Other approaches only provide some examples to explain the proposed service identification method [3, 4].

In this paper, we use a systematic framework to assess the quality of the services identified using SI approaches as part of the SI evaluation process. This will be accomplished through assessing the service's conformance with the SOA principles described by Erl [5].

The rest of this paper is organized as follows. Section 2 reviews the key related work in the field of service quality assessment. In Section 3, a brief description of SOA principles is presented. Section 4 describes one of the SI approaches which we've used to demonstrate the proposed quality assessment framework. In section 5 the framework for assessing the quality of software services identified using SI approaches is described and presented using a case study. Finally, Section 6 concludes the paper, in addition to identifying directions for future work.

2. Related Work

The literature lacks a comprehensive framework to assess all service quality attributes, where most of the research available in literature regarding service quality assessment addresses only one of the quality-attributes of services. For example, the authors in [6] defined a new metric for measuring the cohesiveness of a service and the whole design based on design level information, especially the information embedded in entity model.

In [7] the authors proposed some metrics for measuring service granularity through measuring its composite level, functional richness and its interface granularity. The authors in [8] have proposed a framework to semantically identify services from business process architectures. They have suggested the use of QoSOnt as part of the overall ontology used to identify services to ensure the conformance of identified services to service quality measures.

The literature provides many approaches for improving quality of service in service oriented architectures [9, 10, 11]. However, most of these approaches are applicable in service composition phases and are not concerned with assessing the quality attributes for the candidate services which are identified using service identification approaches. In this research we provide a comprehensive framework that helps in assessing the quality of the identified services at an early stage of developing SOA models, paving the way to provide a way to measure each of the service quality attributes that makes this service "SOA-able" , i.e. suitable for a SOA-based environment.

3. Main Principles of SOA

Erl [5] identified and described a set of common principles that are supported by all major SOA platforms which has been increasingly accepted in the SOA industry. These principles are:

3.1 Services share a formal contract

Services are formally defined using service description documents such as WSDL description document for web services. Another important document is the policy. Service description documents can be collectively viewed as establishing a service contract – a set of terms and conditions that must be met and accepted by a potential service requestor in order to enable successful communication and interaction. Service contract can also refer to additional non-technical documents or legal agreements, such as service level agreements (SLAs). Within SOA, service contracts provide a formal definition of the service endpoint, each service operation, every input and output message supported by each operation, the data representation model of each message's contents and rules and characteristics of the service and its operations.

3.2. Services are loosely coupled:

Coupling between software programs can be viewed as representing a measure of dependency. The higher the dependency is, the tighter the coupling. It is important that services within SOA minimize their respective dependencies as much as possible. This specific relationship has been termed "loosely coupled" and it is accomplished by limiting the dependencies between a service and its requestors to the information expressed in the service contract and designing the service contract in such a way that it is not necessarily specific to any one service requestor.

3.3. Services abstract underlying logic:

This principle encourages the establishment of services as black boxes. There are no restrictions for the amount of logic a service can represent. A service may be designed to perform a simple task or it may be positioned as a gateway to an entire automation solution.

3.4. Services are reusable:

Service-orientation encourages reuse. This fundamental principle forces us to pay attention to the granularity level of each service. As the amount of reusable assets accumulate, the chances increase to develop applications by building less and using more of the available services.

3.5. Services are composable:

Composability is an important aspect of building service-oriented solution and can be seen as another form of reuse; therefore, operations need to be designed in a standardised manner and with an appropriate granularity level to maximize composition opportunities. A common SOA extension that underlines the relevance of composability is orchestration. Here, a service-oriented business process can be expressed through a composition language, such as BPEL, essentially classifying the process itself as a service composition represented by a parent process service.

3.6. Services are autonomous:

For services to provide reliable and predictable performance they must exercise a significant degree of control over their underlying resources. Autonomy represents this measure and this principle emphasizes the need for individual services to possess high levels of individual autonomy. In a service level autonomy, Service boundaries are distinct from each other, but the service may still share underlying resources. But in a pure autonomy, the underlying logic is under complete control and ownership of the service.

3.7. Services are stateless:

State refers to a particular condition of something. Stateless services means that services should minimize the amount of state information they manage, as well as the duration for which they remain stateful. In a service-oriented solution, state information usually represents data specific to a current service activity.

3.8. Services are discoverable:

This principle refers to the design of a service so that it becomes as discoverable as possible to help avoid the accidental creation of redundant services or services that implement redundant logic. This principle is related to discoverability on an architectural level, in which case service discoverability refers to the technology architecture's ability to provide a discovery mechanism, such as a service registry or directory. These extensions effectively become part of the overall infrastructure in support of SOA implementations.

4. The Selected SI approach to be used for demonstration: Riva-Based SI approach [12]

In order to demonstrate the proposed framework, we use the candidate software services generated using the BPA-based services identification approach described in [12]. This SI approach identifies services from an organization's Riva business process architecture, and is deployed here using a real case study from the healthcare domain, the Cancer Care and Registration in Jordan. First, we provide a brief explanation of the selected SI approach to be evaluated, then in the following section we explain our framework demonstrated using the identified services.

The Riva method is used to identify an organization's business process architecture [13]. The Riva-based architecture is derived from an understanding of what business the organization is in, rather than its current structure or culture. So, once the architecture is understood, it becomes apparent what is required from the IT systems supporting these processes. Riva-based BPA was used in [12] as a starting point to generate software services for a SOA-based system.

In order to identify an organization's process architecture in Riva, the following steps should be taken [14]:

1. *Agree the boundary of the organization.*
2. *Brainstorm the organizations' subject matter to identify Essential Business Entities (EBEs)*
3. *Classify these EBEs that have a lifetime which is handled by, or are the responsibly of, members of the organization as Units of Work (UoWs)*
4. *Draw a UoW diagram that depicts the dynamic relationships between UoWs.*
5. *Assume that for each UoW, there is:*
   a) *a case process (CP) that handles single instances of the UoW; and*
   b) *a case management process (CMP) for dealing with the flow of instances.*
6. *Transform the UoW diagram into a first-cut process architecture; then, use the provided heuristics, to generate a second-cut process architecture.*

The Riva-based SI approach, uses the second-cut process architecture diagram to identify Riva Process Architecture (RPA) Clusters. These RPA clusters are identified from the BPA diagram as the set of standalone CPs (have no Start, Request or Deliver relations) as well as the set of CPs and CMPs related together through the Start, Request and Deliver relations, but not with other clusters. The RPA clusters were proven to be suitable candidate services that satisfy SOA principles.

The method was deployed in [15] using the Cancer Care and Registration in Jordan case study [15, 16].
Figure 1 shows the Riva 2nd cut process architecture for the CCR case study, and figure 2 shows the candidate services identified using the Riva based approach.

The capabilities of each service can be derived from the process models associated with each case process and case management process. The entities are also tractable from the identified candidate services through the business process models [17]. The business process model for each case process and case management process can be found in [15].

5. The Proposed Candidate Service Quality Assessment Framework

In this section, we describe a framework to assess the quality of candidate services identified using a certain SI approach. This will be accomplished by examining the extent to which each of the identified candidate services conforms to SOA principles and hence is "SOA-able", i.e. suitable for a SOA-based environment.

The following questions, which are based on Erl's SOA principles [5], constitute the main building blocks of our proposed service quality assessment framework. For each question, a method or technique is described to enable answering that question. The more number of questions answered positively the higher the quality of the identified software services.

While answering the questions, we assume that the capabilities of each service are represented in a process model.

IJCSI International Journal of Computer Science Issues, Volume 13, Issue 2, March 2016
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org http://dx.doi.org/10.20943/01201602.6471

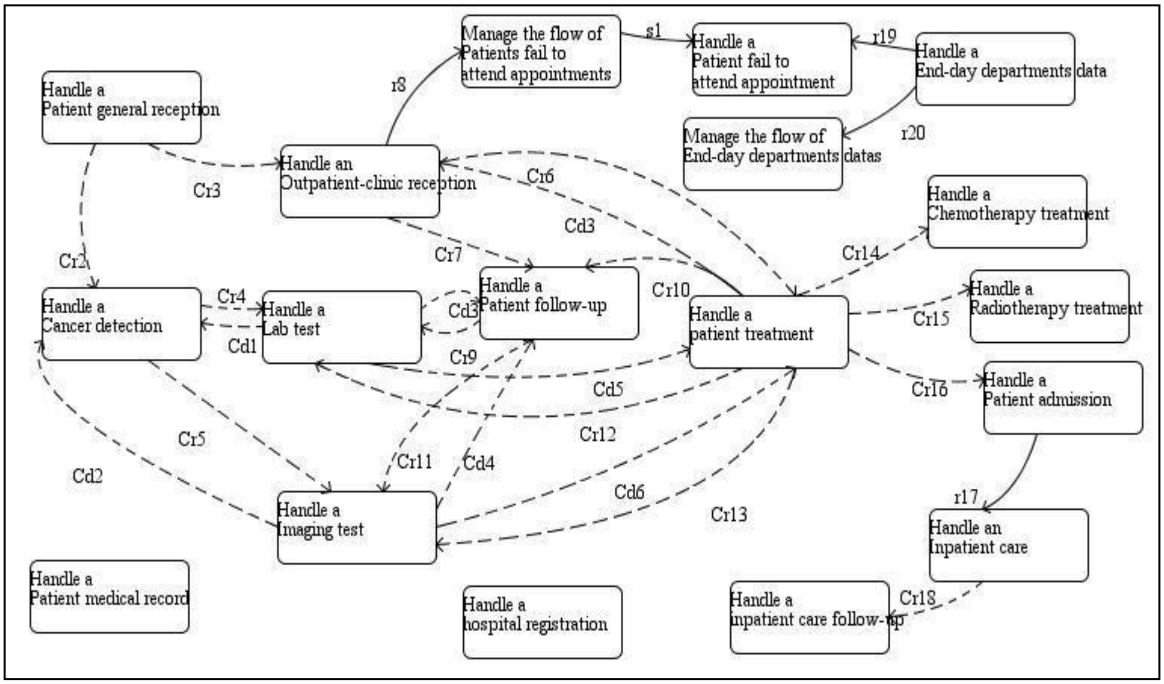

Fig. 1 2nd Cut BPA Diagram for the CCR Processes

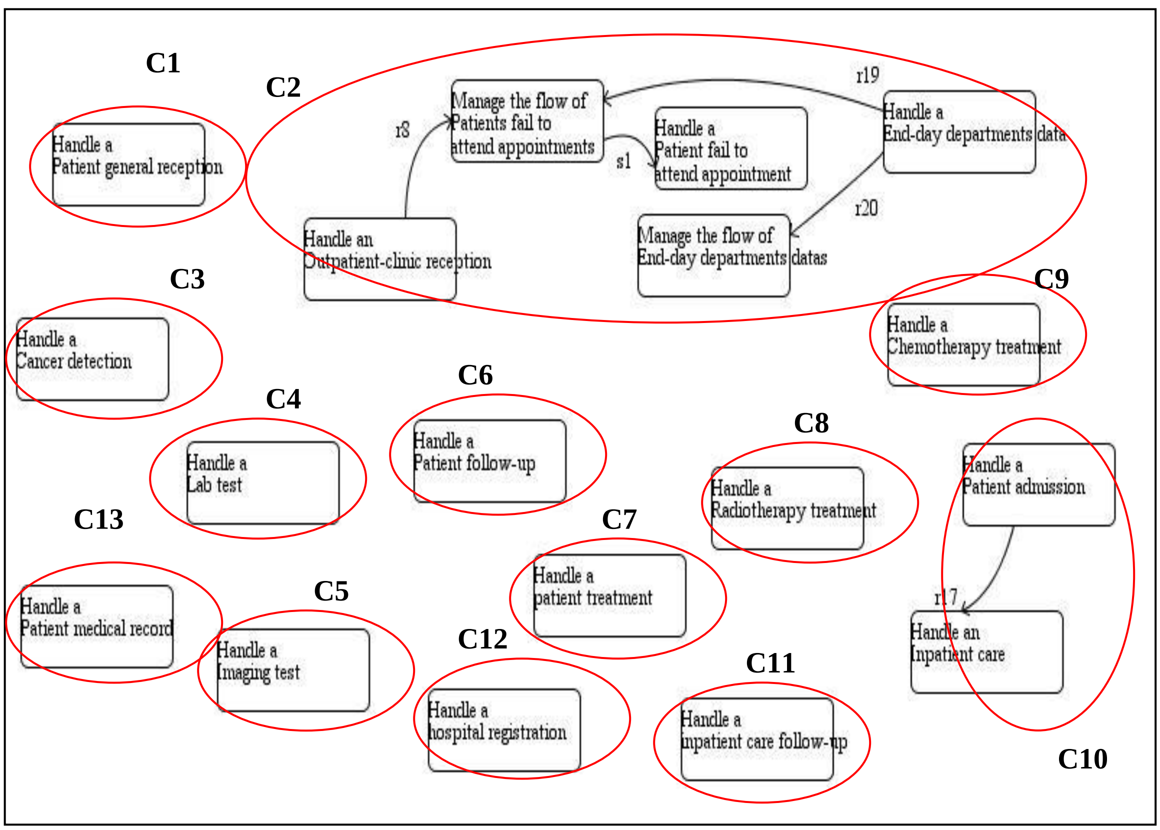

Fig. 2. Candidate Services Identified using the Riva method

5.1. Are Candidate Services Loosely Coupled?

It is important that software services minimize their respective dependencies as much as possible. In order to assess coupling between services, identify points of interactions between services, interaction relationships between services are usually request, start and/or deliver relationships. The more relationships present between services, the more dependencies, and hence unsuitability to be loosely coupled services.

For example, the services identified in the case study, Figure 2, are either stand-alone CPs, as they do not have require, start or deliver relationships with other CPs or CMPs, or they are sets of CPs and CMPs that are related together through request, start and/or deliver relationships, where no members of these groups request, start or deliver to any member in another group. Accordingly, this set of identified services is loosely coupled to a great extent.

5.2. Do Candidate Services Abstract Underlying Logic?

This principle is directly related to the previous principle of loose coupling, where if no or at least a limited number of relations between services are present then each service handles a set of related capabilities, and these detailed capabilities are not seen outside the box. For example, C4 (Handling a lab test) is one of the candidate services identified in figure 2. By referring to the process model representing this service, which is shown in figure 3, the candidate software service encapsulates the functionality of the service represented by the capabilities of each service. For example, a “Lab test” service performs a set of functionalities, such as “add lab’s test results” and “check if patient is medically insured”, hidden from other service that may request it.

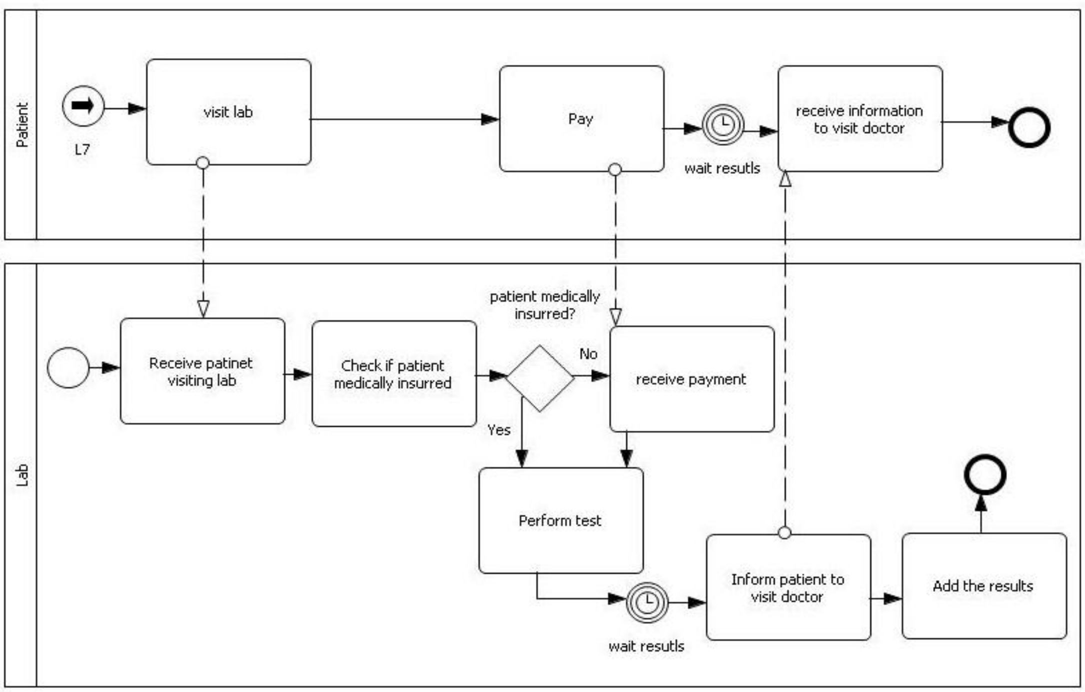

Fig. 3: C4: Handel a lab test

5.3. Are Candidate Services Reusable?

Services are preferred to embody as large functionality as possible. In [18] the authors recommended identifying coarse-grained services, so as to maximize hiding interaction details. Systems’ performance is improved since some of the associated data flows are within a system service, and hence minimizing the overhead of communication in a distributed environment. However, with the increase of granularity, the reusability of a service decreases [5].

In order to determine the granularity level of the identified service, a hierarchy of different levels of granularities can be specified for all components realized while identifying the service; this hierarchy can be represented using a pyramid with small granularity components at the top. The position of the identified services would provide an indication of the granularity level, and hence it's possibility for reuse.

Figure 4 shows the position of the services identified using the selected BPA based SI approach. As can be seen from the figure, the identified services middles the hierarchy, they are not as course-grained as the main BPA, nor too fine-grained like CPs, CMPs or tasks. Therefore, we can conclude that such identification approach generates services with a suitable granularity level, satisfying both reusability and the possibility to hide interaction details.

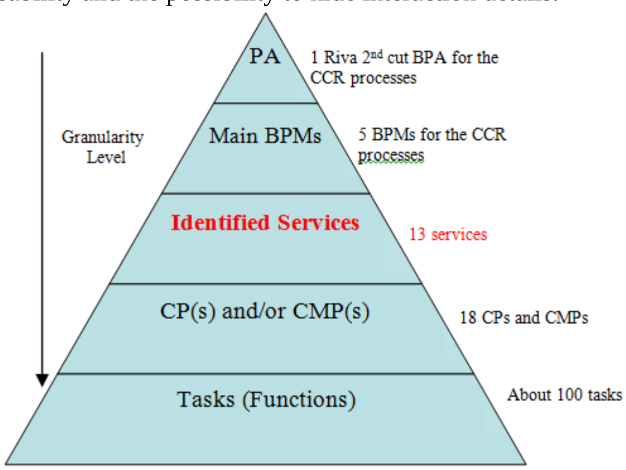

Fig. 4 Identified Services' Position in the Granularity Hierarchy for the selected BPA-based SI approach

5.4. Are Candidate Services Composable?

Composability can be seen as another form of reuse where operations need to be designed in a standardised manner and with an appropriate granularity level to maximise composition opportunities.

Composability can then be proven by inspecting the points of interaction between services as well as the capabilities included in each service. In our example of the BPA-based SI approach, the composability of the identified services can be detected through inspecting the points of interaction between services which represents the BPA for the CCR processes. Also, the capabilities encapsulated within each service represent the corresponding functionality within the BPM. Therefore, the identified services are composable as they provide the same functionalities aimed in the corresponding BPMs.

5.5. Are Candidate Services Stateless?

As was explained earlier, stateless services means that services should minimize the amount of state information they manage, as well as the duration for which they remain stateful. In a service-oriented solution, state information usually represents data specific to a current service activity. This principle can be assessed through inspecting the amount of state information managed by the identified services. This can be achieved by referring to the definitions of service components that constituted them.

For example, the services identified using the BPA_based SI approach [12] are stateless because the RPA clusters minimize the amount of state information they manage by definition, where CPs and CMPs are related through require, start and deliver relationships (i.e. relations between CPs and CMPs are request/response relation, not conversational). The conversational relations between roles are included within each CP or CMP. Tracing activities and activity flows encapsulated in the "General Reception" service shows that conversational relations exist between roles, such as receptionist and patient, but relations to other clusters such as the "Cancer Detection" are limited to request/deliver relationships, this is shown in Figure 5.

The remaining Service principles, i.e. "Services share a formal contract", "Services are autonomous" and "Services are discoverable" are more concerned with designing the services as part of the overall infrastructure to support SOA implementation, and hence, are out of the scope of this paper.

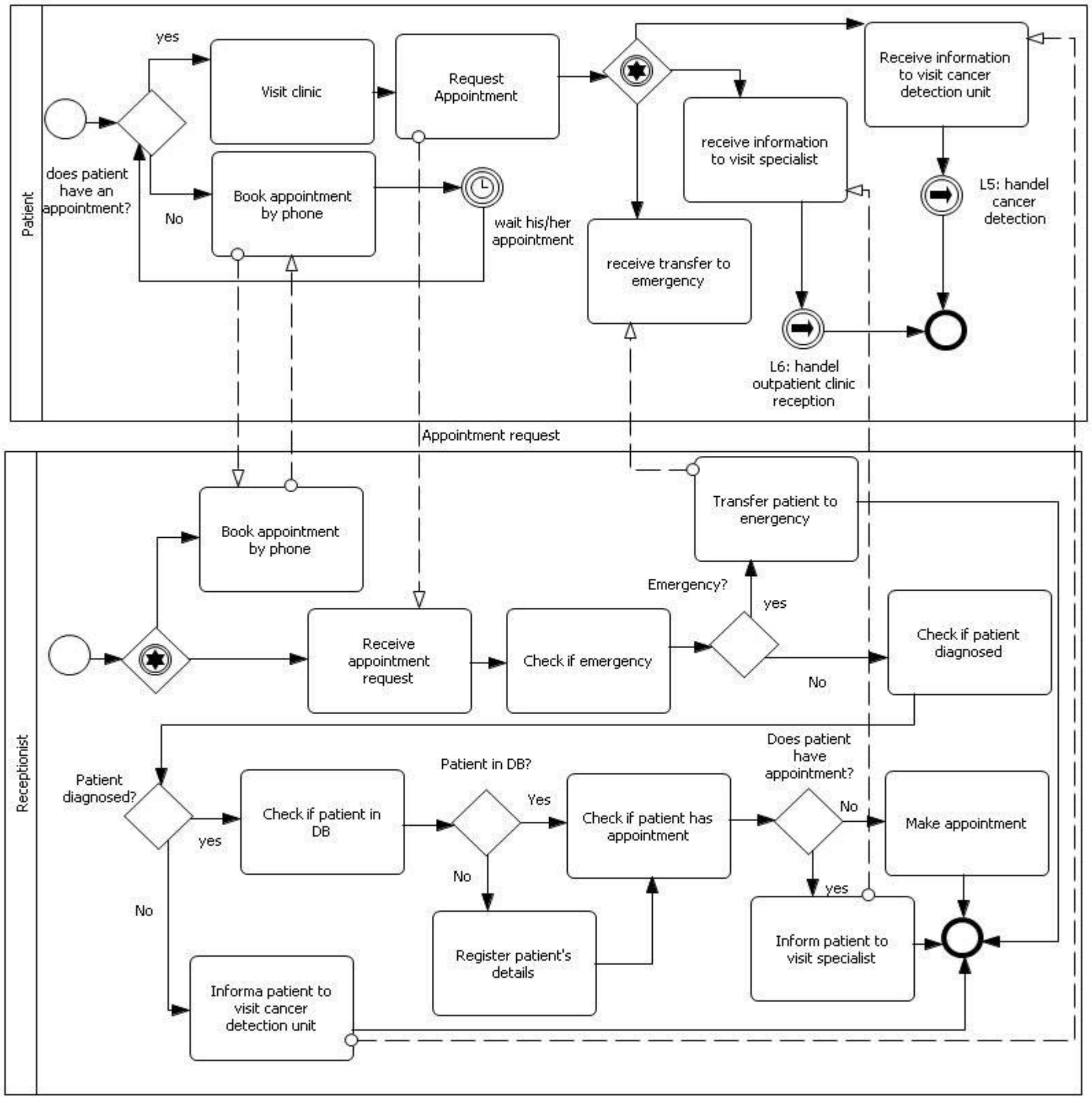

Fig. 5 Business Process Model Representing the Service "Handle a General Reception" with the Conversational relationships and Request to another Service

4. Conclusions

In this paper we presented a framework to assess the quality of services identified using SI approaches. The framework is used to assess the granularity level, reusability, statelessness, composability, cohesion and low coupling of the identified services.

A number of questions constituted the main building blocks of the presented framework, where an appropriate method is described to answer each of these questions, the more positively answered questions, the higher the quality of the identified services and hence the service identification approach used to derive them.

The assessment framework was demonstrated using a real case study from the healthcare domain and a BPA-driven service identification approach was used to evaluate the quality of its identified services.

As a future work, our framework will be deployed on other types of service identification methods to include both top-down and bottom-up approaches.

References

[1] A. Arsanjani. "Service-oriented modeling and architecture: How to identify, specify and realize services for your soa". Technical report, IBM, Software Group, 2004.

[2] R, Huergo , P. Pires and F, Delicato. "MDCSIM: A method and a tool to identify services", IT Convergence Practice (INPRA), Vol. 2, no. 4, 2015, pp. 1-27.

[3] D. Bianchini, C. Cappiello, V. De Antonellis and B. Pernic. “Service Identification in Interorganizational Process Design”, IEEE Transactions on Services Computing, Vol. 7, no. 2, 2014, pp. 265-278.

[4] Q. Gu and P. Lago, “Service Identification Methods: A Systematic Literature Review”. Proc. ServiceWave, 2010, pp. 37-50.

[5] T. Erl, “SOA principles of Service Design”, Prentice Hall, 2009.

[6] M. Daghaghzadeh, A. Dastjerdi, and H. Daghaghzadeh. "A Metric for Measuring Degree of Service Cohesion in Service Oriented Designs." International Journal of Computer Science Issues (IJCSI) vol. 8, issue 5, no. 1, 2011.

[7] T. Karthikeyan and J. Geetha. "A Quantitative Measurement and Validation of Granularity in Service Oriented Architecture." International Journal of Computer Science Issues (IJCSI) vol. 9, issue 2, no. 1, 2012.

[8] R. Yousef, M. Odeh, D. Coward, D and A. Sharieh, “A. BPAOntoSOA: A generic framework to derive software service oriented models from business process architectures”. In: ICADIWT, 2009, pp. 50–55.

[9] Potena, Pasqualina, et al. "Software Architecture Quality of Service Analysis Based on Optimization Models." Intelligent Decision Making in Quality Management. Springer International Publishing, 2016. 421-446.

[10] Nayrolles, Mathieu, et al. "Towards Quality-Driven SOA Systems Refactoring Through Planning." E-Technologies. Springer International Publishing, 2015. 269-284.

[11] Niklas, Kai, Joel Greenyer, and Kurt Schneider. "Towards application and evolution of model-based heuristics for improving SOA service design."Proceedings of the Seventh International Workshop on Modeling in Software Engineering. IEEE Press, 2015.

[12] R. Yousef and M. Odeh. “A BPA-Based Approach for the Identification of SOA Candidate Software Services”. The 5th International Conference on Information Technology, 2011, Jordan.

[13] M. Ould. “Business Process Management: A rigorous Approach”, BCS, UK, 2005.

[14] S. Green, and M. Ould. “The Primacy of Process Architecture”. 5th Workshop on Business Process Modelling, Development, and Support, in conjunction with the 16th Conference on Advanced Information Systems Engineering (CAiSE04), Riga. 2004.

[15] R. Yousef. “BPAOntoSOA: A Semantically Enriched Framework for Deriving SOA Candidate Software Services from Riva-based Business Process Architecture”. PhD thesis, The University of the West of England, 2010.

[16] F. AbuRub. “Process Modelling for Business Process Improvement, with particular reference to healthcare processes”. PhD thesis, The University of the West of England, 2006.

[17] R Yousef “Extracting Essential Business Entities from BPMN Models”. International Journal of Computer Science Issues. 2014, Vol 11, Issue 4, No 1. 131-134.

[18] Ma, Q., Zhou, N., Zhu, Y., Wang, H. “Evaluating Service Identification with Design Metrics on Business Process Decomposition”. IEEE International Conference on Services Computing, 2009, pp.160-167