Service Identification Approach to SOA Development

Nafise Fareghzadeh

Abstract—Service identification is one of the main activities in the modeling of a service-oriented solution, and therefore errors made during identification can flow down through detailed design and implementation activities that may necessitate multiple iterations, especially in building composite applications. Different strategies exist for how to identify candidate services that each of them has its own benefits and trade offs. The approach presented in this paper proposes a selective identification of services approach, based on in depth business process analysis coupled with use cases and existing assets analysis and goal service modeling. This article clearly emphasizes the key activities need for the analysis and service identification to build a optimized service oriented architecture. In contrast to other approaches this article mentions some best practices and steps, wherever appropriate, to point out the vagueness involved in service identification.

Keywords—SOA, service identification, service taxonomy, service layer.

I. INTRODUCTION

SERVICE-oriented architectures represent an approach to distributed computing that treats software resources as discoverable services available on a network. Each service defines logical interaction between service provider and consumer through one or more interfaces with the service contract. The lifecycle of SOA delivery projects is simply comprise of a series of steps that need to be complete to construct the services for a given service-oriented solution. Developing SOA generally includes some common phases like Service-oriented analysis, Service-oriented design, Service development, Service testing, Service deployment and Service administration [1]-[3].

Although the concept of SOA has been intensively debated in recent years, a unified methodical approach for identifying services has not yet been reached. Instead, a variety of heterogeneous approaches have been proposed. Approaches especially vary in terms of service hierarchies and analysis objectives. Thus, methods are proposed to identify services by utilizing the information systems in place in a bottom-up approach or follow a procedure of analysing business requirements in a top-down approach. Other approaches integrate both perspectives into a hybrid strategy, referred to as 'meet-in-the-middle' approach. The lack of a consolidated

Nafise Fareghzadeh is with Computer Department of Islamic Azad University of Zanjan and Tehran, Payame Noor University of Tehran, Raja Uniersity of Quazvin (e-mail: n_f2840@yahoo.com).

approach also becomes obvious in a deviant regard of service categories. Additionally, approaches vary in forms of documentation as well as in use of IT criteria to support the service identification [1]-[3]-[7].

The goal of this paper is to propose an optimized method for service identification in the service oriented analysis phase of SOA lifecycle. It combines different approaches and advantages and avoid from disadvantages and overheads of these strategies to help attain understanding of what constitutes contemporary SOA along with step-by-step and clear guidance for realizing its successful implementation. In contrast to other approaches attempting to transfer each existing function in a company into a service, the method presented in this paper proposes an identification of candidate service types in service analysis phasec SOA from a business point of view, based on business process models and reusable existing services. In the focus of the development, this is a methodical approach for identifying services. Core to this approach is an optimized integration of existing services and business analysis and an SOA principles point of view. Another great benefit of this approach is that the resulting services have a guaranteed fit with an organization's functional needs. This method is also very intuitive, allowing project teams to use it for proof-of-concepts and pilot projects. A corresponding approach for service identification is considered as a substantial research contribution for the conceptual design of a SOA. This article emphasizes the key activities need for the analysis and service identification to build a optimized service oriented architecture [13]-[24]-[27].

The remainder of the paper is organised as follows: The suggested approach for service identification is introduced and discussed. Afterwards, the applicability of the method is shown by presenting a case study in Section IV. At the end of article related work and conclusion mentioned.

II. SUGGESTED APPROACH

The identification of business functions to be provided as services is a basic precondition for a detailed specification and implementation of services in a Service-oriented architecture (SOA). Summarizing objectives of SOA implementation, issues referring to both business strategy and IT can be identified, e.g. integrating business processes and a broader reuse of implemented functionality. By combining both points of view and combine both top-down and bottom-up analysis an integrated approach for identifying services is constructed.

With respect to the business context in which a company acts in, not only are the relations with customers crucial, but the relations with other relevant stakeholders (e.g. suppliers, subsidiaries, commercial representatives or service providers) are just as important. When developing an approach for service identification, it is therefore essential to extend the perspective of the approach to important stakeholders. Designing an approach for service identification based on stakeholders is a matter of the design science paradigm [8]-[19].

To make the work easier, we can categorize the different services into reasonable groups and types that showing the operational state of services. Enterprise logic can be divided into two primary domains: business logic and application logic. Services can be modeled to represent either or both types of logic, as long as the principles of service-orientation can be applied. So we can build specialized layers of services. Each layer can abstract a specific aspect of our solution, addressing one of the issues we identified. This alleviates us from having to build services that accommodate business, application, and agility considerations all at once. Also to reach the principle of Loosely Coupling, it is necessary for services to be separated physically so that no organizational domains aren't depended on each other. It allows the automatic show of business logic to be developed independent of application logic. The three layers of abstraction we defined for SOA are [2]-[4]-[7]:

- the orchestration service layer ;
- the business service layer ;
- the application service layer ;

The application service layer establishes the ground level foundation that exists to express technology specific functionality. Services that reside within this layer can be referred to simply as application services. Their purpose is to provide reusable functions related to processing data within new or legacy application environments [9]. The business service layer introduces a service concerned solely with representing business logic, called the business service. They are responsible for expressing business logic through service-orientation and bring the representation of corporate business models into the Web services area [1].

The orchestration service layer introduces a parent level of abstraction that alleviates the need for other services to manage interaction details required to ensure that service operations are executed in a specific sequence. Within the orchestration service layer, process services compose other services that provide specific sets of functions, independent of the business rules and scenario specific logic required to execute a process instance [11]-[13].

So the most important service types and taxonomy we have used for services are as follows:

- Application services: A generic category we use to represent services that contain logic derived from a solution or technology platform. Services are generally distinguished as application services when creating service abstraction layers.
- Business services: containing the services having the

business logic. They are divided into two groups based on the operational state:

a) Task Services: A business process-specific variation of the business service that represents an atomic unit of process logic.

b) Entity Services: Business services that represents one or more related business entities. This type of services is created when establishing a business service layer.

- Process Services: containing the services that represent a business process that implemented by an orchestration platform and described by a process definition. Process services reside in the orchestration service layer. All process services are also controller services by their very nature, as they are required to compose other services to execute business process logic [1]-[4]-[27].

The suggested approach integrates both perspectives (Top-Down and Bottom-Up approach) into a hybrid strategy to identify different service types. It has the benefits of two methods without any additional complexities and overhead of either. The proposed approach can be mainly divided into the three phases Initial Analysis; In Depth Analysis; and Make A Service Taxonomy (cf. Fig. 1).
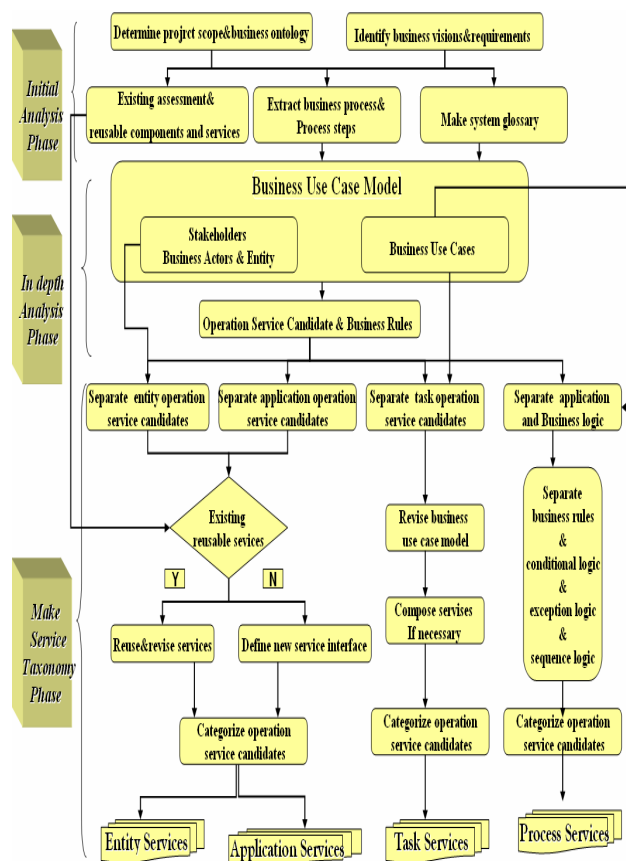

Fig. 1 Suggested Approach

Suggested approach, represented in the last figure, includes the following phases:

A. Initial Analysis

During the initial analysis phase, basic decisions regarding project scope, business requirements, business visions and goals and purpose of the service analysis are to be made. The area of service analysis may be documented by a framework, identifying the most important functions of the domain and showing their interrelationships. This framework can act as a starting point for the business process model based service analysis procedure. Crucial is also a meaningful documentation of existing business processes, assessment ideally in the form of hierarchical process models. This step involves taking the enterprise level application portfolio analysis, assessment for reuse, redundancy, and rationalization effort. Such process models often already exist in companies, e.g. as an output of formerly conducted Business Process Management (BPM) projects. Typically existing assets would be turned into services in any of the following ways:

(i) Wrap an existing function as a service;
(ii) Wrap and replace an existing function with a service;
(iii) Create a service adapter to make the application work with services;
(iv) Integrate the function into a service;

So in the initial Analysis phase, the primary activity is developing the vision for the project. Output from this activity is a vision document that identifies the high-level user or customer view of the SOA system to be built. The vision expresses initial requirements as key features the system must possess in order to solve the most critical problems and meet key stakeholder needs. The system analyst has the primary role in this workflow. Active involvement from various project stakeholders is required and all relevant stakeholder requests should be considered.

To form a sufficient foundation for the identification and definition of services, process models however should correspond to service specific modelling conventions, which are specially designed for the service identification approach presented in this paper:

■ Process models shall describe the degree of IT support for each function in the business process (automatic, semi-automatic/dialog or completely manual).

■ Organizational units involved in the business process shall be annotated, so that external process interfaces are made visible in the model as well.

■ Models can be hierarchically divided into several layers. From a top-level perspective, process models represent core business actions. On a more detailed level, the core business functions are split up into more detailed functions, which can be assigned to a specific organisational unit.

The main outputs of this step:
● Business requirements, visions and goals;
● Business process model;
● Existing assets, reusable components and services;
● glossary of terms to facilitate common vocabulary among team members;

B. In Depth Analysis

Good SOA requires a well defined structure for decision making, where roles and responsibilities for processes and services are clearly allocated and assigned. When identifying services, the party that carries the responsibility to make available the required functionality determines which services will ultimately be offered. After completing the initial analysis phase, the in-depth analysis is conducted. The in-depth analysis is carried out in two steps: At first, should Take the documented business process and break it down into a series of granular and independent process steps then assign these process steps to critical roles to do them. These critical roles are business entities and actors. When the business process is subdivided into sub-processes or decomposed into granular activities and tasks. The lowest level tasks can consist of small, cohesive "logical units of work" that are supported by the functionality offered by distinct services.

Also besides determining the scope of the analysis and analysing business process models, relevant stakeholders are to be identified as external business partners (e.g. customers, suppliers or service providers) or internal business partners (e.g. subsidiaries, other facilities or the company headquarters). Supporting a business-driven approach for service-identification, we provide some central questions as a means to ascertain the potential of functions more reliability during the identification process (see excerpt in Table I).

TABLE I
EXPERT OF CENTRAL QUESTIONS FOR BOTH AREAS OF ANALYSIS

Aspect	Central question
	1. Is the function eligible to be outsourced to a stakeholder (outsource non-core functions only)?
	2. What are potential losses of knowledge after outsourcing the function to a stakeholder?
Takeover	3. Which activities is the customer capable to provide in a timely manner and with a high level of quality? Which information technology configuration is necessary for a successful cooperation?
	4. What problems are related to stakeholders not contributing in a required timely, qualitative and quantitative manner (e.g. exceeded time and cost constraints)?
	5. Can the outsourced function be governed in a sufficient way?
	1. Which aspects of the function are currently made visible to stakeholders (as-is state)?
	2. Which aspects of the function should be made visible to demonstrate effectiveness and efficiency in the companies' own business processes?
Visibility	3. Which business processes does the customer need to track in order to provide his own goods and services (process evidence)?
	4. Access on which processes and data must be constrained (e.g. compliance to legal directives and contracts)?
	5. Which channels (internet, mobile devices etc.) should be used to provide customers with the required information?
	6. Should a push or pull concept be used when transferring data?

Now we should determine interrelate dependencies to make business use case and then extract operation service candidate

and business rules from business use case and process steps and make business use case model. A business use case model consists of actors, use cases, and relations among them. Actors represent everything that must exchange information with the system, including what are typically called users. When an actor uses the system, the system performs a use case. A good use case is a sequence of transactions that yields a measurable result of value for an actor. The collection of use cases is the system's complete functionality and we extract candidate services from this functionality plan [13]-[20].

Secondly, and in the other aspect of in-depth analysis we should refine the system definition and diagnose critical areas that require more analysis. Now we have previous list of use cases, which were separated by actor and possibly duplicated or can integrate by some change in existing assets and services. Typically existing assets would be turned into services in any of the following ways:

(i) Wrap an existing function as a service;
(ii) Wrap and replace an existing function with a service;
(iii) Create a service adapter to make the application work with services;
(iv) Integrate the function into a service;

We should identify these components then refine and assign suitable service interface or merged them into a single list of use cases. Then use cases can be grouped into sets and can identify service groups. After that it is possible to look at these service areas in a more generic way. This allows the extraction of common support services and also the relationship between the services to be identified. The system definition refinement process summarized in Fig. 2:
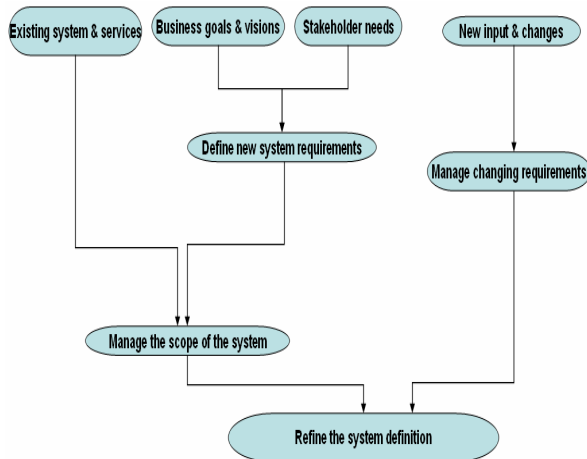

Fig. 2 System Definition Refinement Process

Moving forward to refine the System Definition, this workflow detail assumes that system-level use cases have been outlined and actors have been described, at least briefly. Through managing the project scope, the features in the vision have been reprioritized and are now believed to be achievable by fairly firm budgets and dates. The output of this workflow is a more in-depth understanding of system functionality expressed in detailed use cases, revised supplementary specifications, and early iterations of the system itself. As changes occur and they inevitably will the Manage Changing Requirements workflow detail needs to be applied continuously throughout the life of the project, as discussed for the Manage the Scope of the System workflow detail. The output of this workflow can cause modification to every artifact, which in turn requires effective communication among all project team members and stakeholders.

At the end of this phase to make our service candidates truly worthy of an SOA, we must take a closer look at the underlying logic of each proposed service operation candidate. This step gives us a chance to make adjustments and apply key service-orientation principles. Mostly, the following key principles are important to consider:

■ interoperability;
■ modularity;
■ autonomy;
■ reusability;

To reach interoperability, components should be standardised from a technical (e.g. transfer protocols and data formats) and conceptual (e.g. clearly and uniformly specified terms and standardized data models) point of view. These standards are most useful when open, platform independent and widely diffused, e.g. as industry standards. Design requirements regarding modularity and autonomy require the grouping of functionality and resources, according to the principles of high cohesion (strong similarity within the same category) and loose coupling (weak dependencies between different categories) [3]-[18]-[22].

A great benefit of this phase is that the resulting services have a guaranteed fit with an organization's functional needs. This approach is also very intuitive, allowing project teams to use it for proof-of-concepts and pilot projects. The main outputs of in-depth analysis phase are:

● Stakeholders, Business Actors & Entity;
● Business Use Cases & Business Use Case Model;
● Operation Service Candidate & Business Rules;

C. Make a Service Taxonomy

Services can be categorized based on the nature of the logic they encapsulate and the manner in which they are typically utilized within SOA. Make a service taxonomy phase includes following steps:

■ Identify Entity Services

Ideally, when creating entity services, the modeling effort resulting in the service candidates will have taken any existing services into account. Therefore, the first step is to confirm whether it is actually even necessary. If other services exist, they may already be providing some or all of the functionality identified in the operation candidatesorthey may have already established a suitable context in which these new operation candidates can be implemented (as new operations to the existing service). After concider these rules in this step (by using the last outputs) the operation service candidate that related to one or more dependent entities are to be separated

and the operation candidates are scaled and categorized to the logical groups. Then derive an abstract service interface and Apply principles of service-orientation [2]-[17].

■ Identify Application Services

Application services need to be identified and designed in a solution-agnostic manner, by implementing the utility service model so that reuse can be maximized. So it is important to ensure that the functionality required, as per the application service candidate, does not, in some way, shape, or form, already exist. So it is very necessary to review existing inventory of application services in search of anything resembling what you are about to identify and design. Then separates the process steps depended on the technology; platform and application program. After that, should categorize the remaining operation candidates as the logical groups each of them shows the logic of special application service. Now we can Standardize and refine the service interfaces.

■ Identify Task Services

Inputs:

● Business Requirements, visions, goals;

● Business Use Case Model;

● Operation Service Candidate & Business Rules;

Task-centric services typically will contain embedded workflow logic used to coordinate an underlying service composition. Our first step, therefore, is to revision this logic for every possible interaction scenario we can imagine. So first, the work logic extract from business use case model and relations are reviewed and, if required, the new business goals and steps are defined and added to system for supporting the long term goals and organization strategy. Task services can compose application, entity or both and additional task services. Therefore, the implementation of a task service interface requires that any needed underlying service layers are in place to support the processing requirements of its operations. So if it needs should combine the operation candidates of business services and application services for supporting business tasks, the related activities are performed and then the operation candidates related to business tasks are categorized as the logical groups. Now we can Standardize and refine the service interfaces [19]-[24].

■ Identify Process Services

Inputs:

● Business Process Steps;

● Business Use Case Model;

● Operation Service Candidate & Business Rules;

Since the process services play the critical roles of controlling services, it is necessary to first separate the business and application logics. If you have decided to Identify process services as part of your SOA, you should identify the parts of the processing logic that orchestration layer would potentially abstract. Potential types of logic suitable for this layer include:

business rules;

conditional logic;

exception logic;

sequence logic;

So it is necessary to identify and separate all business rules, condition rules, exception logic and sequence logic, then assign the operation candidates of all controlling; conditions and exceptions operations. Finally, it is necessary to categorize the related operation candidates as the logical groups with independent service interface, each of them shows the logic of special process service [1]-[23]-[26].

III. EXAMPLE

In continue, we describe the proposed service identification approach with a case study:

There is a distributed organization, the business goal is to prepare an distributed automatic processing system for record and confirm some data related to the working hours of supervising employees in distributed working units with proposed approach in the service oriented analysis phase of SOA lifecycle. The organization includes a collection of supervising employees, performing all surveillant activities of different distributed units on the organization. When the employees fill out their weekly timetable, it is necessary to determine how many hours have been consumed for each unit. In the other way, these hours are recorded separately in any unit. If both of two documents are the same, the form and timetable of employees will be confirmed. However, it is necessary for employee has allowed for recording the hours and the consumed time in each working unit must not go beyond the defined presumed time for each working unit, also the total working hours of the employee should not exceed the maximum presumed working hours. If the working hours timesheet of an employee isn't confirmed, it should be recorded in his profile and then failure alarm messages should be sent to that employee and his manager. To identify the required services, with suppose basic rules mentioned in this paper and proposed approach; and after predicate and make a documentation of Business requirements and goals; Business process; reusable components and services and make a glossary of common vocabulary among team members; first it's necessary to divide the above working business process into a collection of process steps as follow [1]-[3]-[25]:

1- Receive timesheet.

2- Compare hours recorded on timesheet to hours billed to clients.

3- Confirm that authorization was given for any recorded overtime hours.

4- Confirm that hours recorded for any particular project don't exceed a predefined limit for that project.

5- Confirm that total hours recorded for one week don't exceed a predefined maximum for that   worker.

6- If timesheet is verified, accept timesheet submission and proceed to step 11.

7- Reject timesheet submission.

8- Generate a message explaining the reasons for the rejection.

9- Issue a timesheet rejection notification message to the

worker.

10- Issue a notification to the worker's manager.

11- Terminate the process.
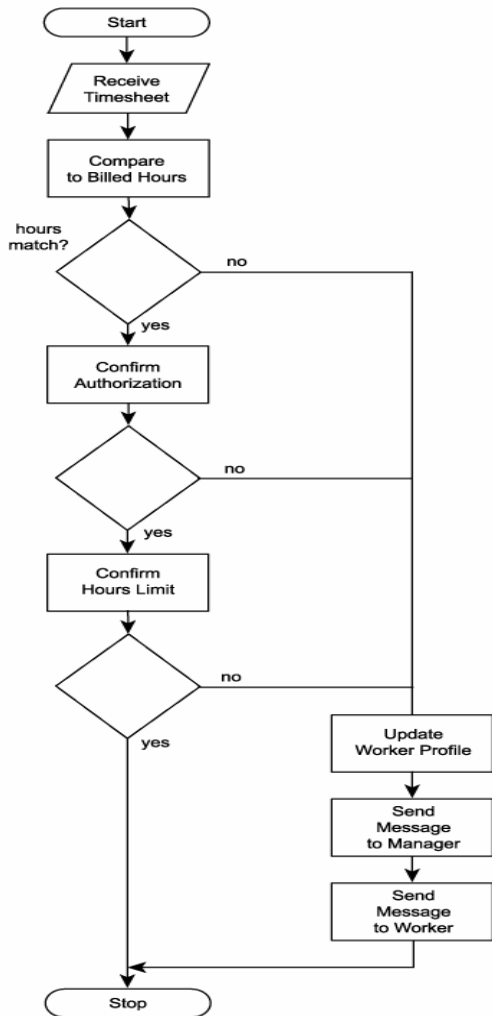
The workflow diagram is shown here:

Fig. 3 Timesheet Submission workflow diagram

In the first step, we assign the entities and roles related to the business process. The following figure represents the different entities and their interrelations:
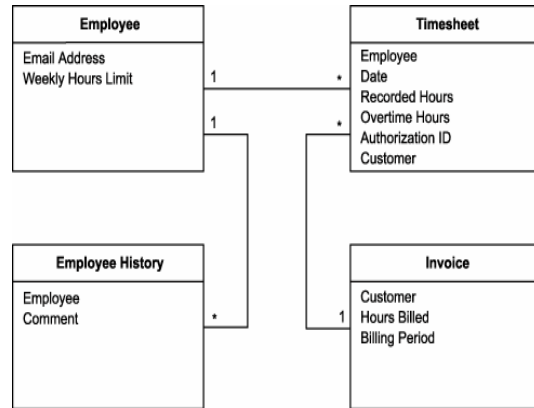

Fig. 4 Entity model & Relations

Then, it is necessary to determine the operation candidate by the presumed services. The following Figure explains the candidate operations required for business process:
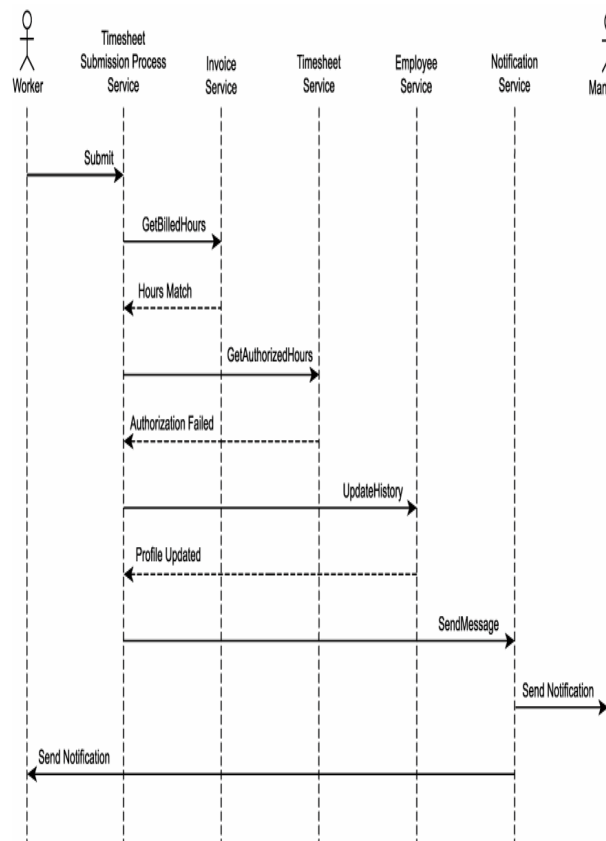
Fig. 5 Business Use Case Model

Scaling and categorizing the operation candidate related to the business entities and related activities from Business process, the entity services candidates are identified as following:
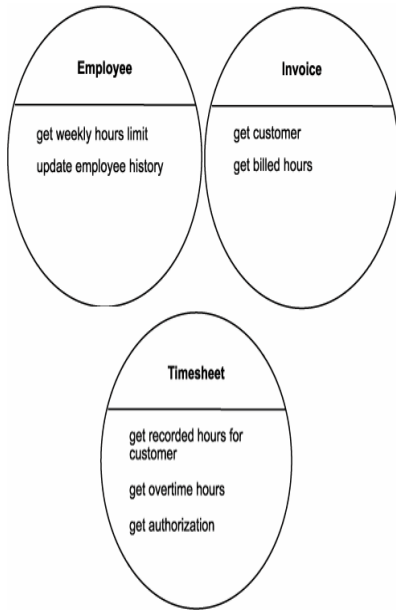
Fig. 6 Entity Centric Services

Since the entities of Employee History and Employee are related and dependent to each other, we can describe the operation candidates related to these entities as a one composite service, in order to prepare the more flexibility and efficiency with service orientation principles.

Separating the operation candidate related to the application business area, we understand that we can categorize the operation related to the notification to the user and related management, if disconfirming the form in the group of Application Services:
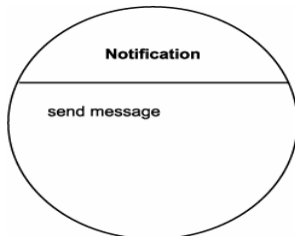

Fig. 7 Application Services

In the next step, it is necessary using process steps and combining the business service candidates to identify the task services. Receiving two entities of working unit invoice and timesheet form employee and combining the related services and comparing the recorded rates, the operation service candidate related to the form confirmation business tasks identified as the verify timesheet service as follows:
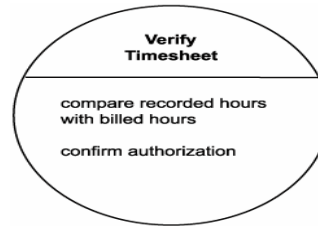

Fig. 8 Task Centric Services

Finally, we can categorize the related controlling operation in the form of process services through comparing the timesheet and presumed rates and accessing data, which are the components of condition and control operation of working process:
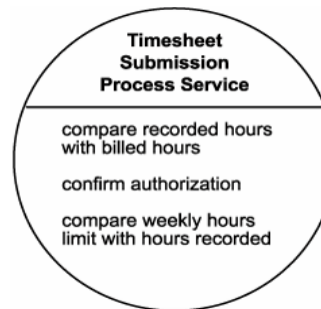

Fig. 9 Process Services

The following figure shows the identified services and layers related to the services:
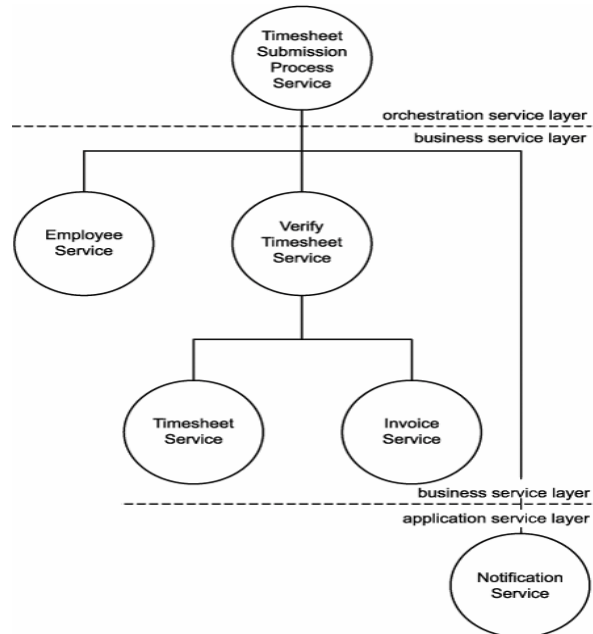

Fig. 10 Identified Servicces with Service Layers

As the Fig. 11 represented, by using the proposed approach, the different service types, categorizaed to the independent

layers of architecture, are identified easily and clearly without any overhead and additional activities. After identification, the services can be designed and tested, and implemented and developed in a service-oriented area.

IV. RELATED WORK

Although the concept of SOA has been intensively debated in recent years, a unified methodical approach for identifying services has not yet been reached. Instead, a variety of heterogeneous approaches have been proposed. Approaches especially vary in terms of service hierarchies and analysis objectives. Thus, methods are proposed to identify services by utilizing the information systems in place in a bottom-up approach (e.g. Nadhan 2004) or follow a procedure of analysing business requirements in a top-down approach (e.g. Erl 2005, Quartel & Dijkman & Sinderen 2004). Other approaches integrate both perspectives into a hybrid strategy, referred to as 'meet-in-the-middle' approach (e.g. Zacharias 2005, Ivanov & Stähler 2005). The lack of a consolidated approach also becomes obvious in a deviant regard of service categories. Furthermore, methods are based on different SOA philosophies, such as SOA as a comprehensive middleware approach (e.g. Gold-Bernstein, Ruh 2005), or SOA as a concept for a flexible configuration of information systems (e.g. SAP Enterprise Services Design Guide 2006). Additionally, approaches vary in forms of documentation as well as in use of IT criteria to support the service identification. Approaches make different use of process models: Instead of deducting services from process models systematically as a meaningful representation of business processes, some approaches restrict themselves to formulating general guidelines for identifying services. A consolidation of these approaches seems beneficial to guide the procedure of service identification more thoroughly $[1]$-[12]-[20]-[27].

Compared to existing approaches, the procedure presented in this paper introduces a strong business perspective into the derivation of service candidates. This is done by integrating business use case model and stakeholders as important part when deriving services from business process models in a both top-down and bottom-up approach. On the other hand, service categories are consolidated to form more complex services in a complimentary bottom-up approach. Therefore, the procedure integrates both perspectives. After assessing their business potential, suitable service candidates are evaluated due to their technical service feasibility by applying SOA design principals. In this way, the proposed approach is a valuable asset to identify suitable services from a business point of view in an integrated perspective.

The goal of this paper is to propose an optimized method for service identification in the service oriented analysis phase of SOA lifecycle. It combines different approaches and advantages and avoid from disadvantages and overheads of these strategies to help attain understanding of what constitutes contemporary SOA along with step-by-step and clear guidance for realizing its successful implementation. In contrast to other approaches attempting to transfer each existing function in a company into a service, the method presented in this paper proposes an identification of candidate service types in service analysis phasec SOA from a business point of view, based on business process models and reusable existing services.

In the focus of the development, this is a methodical approach for identifying services. Core to this approach is an integration of a existing services and business analysis and an SOA principles point of view. Another great benefit of this approach is that the resulting services have a guaranteed fit with an organization's functional needs. This method is also very intuitive, allowing project teams to use it for proof-of-concepts and pilot projects. A corresponding method for service identification is considered as a substantial research contribution for the conceptual design of a SOA.

V. CONCLUSION

In this paper we outlined an approach for service identification in SOA analysis phase. The method defines how a business process should be transformed into services and how these services should collaborate to full business goals. Furthermore, we have proposed how to consider fundamental SOA principles, such as service autonomy and reusability. Future work is mainly related to integration and combine of the presented approach with formal models and patterns to improve service identification performance because service identification is one of the main activities in the modeling of a service-oriented solution, and therefore errors made during identification can flow down through detailed design and implementation activities that may necessitate multiple iterations, especially in building composite applications.

REFERENCES

[1] Thomas Erl, "Service-Oriented Architecture: Concepts, Technology, and Design". August 04, 2005.
[2] Baskerville, R. et al, "Extensible architectures: The strategic value of service-oriented architecture in banking", In: Proceedings of the European Conference on Information Systems, may 2005.
[3] Endrei M. et al, "Patterns: Service-oriented Architecture and Web Services", Redbook, SG24-6303- 00, April 2004.
[4] Dostal, W. et al, "Service-orientierte Architecture meet Web Services", Konzepte - Standards - Praxis. München, 2005.
[5] "Patterns: Implementing Self-Service in an SOA Environmen", Redbooks Paper © Copyright IBM Corp, 2005.
[6] L. Abeti, "Service Oriented Software Engineering in Grid systems", Master's thesis, University of Bologna, 2005.
[7] S. Jones, "Toward an acceptable definition of service", IEEE Software May/June 2005.
[8] Alan W. Brown, Simon Johnston, Kevin Kelly, "Using Service-Oriented Architecture and Component-Based Development to Build Web Service Applications, Rational Software Corporation", April 2002.
[9] D. K. Barry, Morgan Kaufman, "Web Services and Service Oriented Architectures", San Francisco, CA, 2003.

[10] Alan W. Brown, "Designing and building service-oriented solutions with the IBM rational software development platform", In IBM IAA User Group and Insurance Solutions Conference. IBM, April 2005.

[11] Karin Duermeyer, "Bridging business value to SOA: SOA best practices", January 2004.

[12] Dirk Krafzig, Karl Banke, and Dirk Slama, "Enterprise SOA: Service-Oriented Architecture Best Practices", Prentice Hall PTR, 2004.

[13] Boris Lublinsky and Dimitry Tyomkin, "Dissecting serviceoriented architectures", Business Integration Journal, October 2003.

[14] Boris Lublinsky, "SOA design: Meet in the middle", Java Pro, August 2004.

[15] Ali Arsanjani, "Toward a pattern language for service-oriented architecture and integration, part 1: Build a service eco-system ", July 2005.

[16] Steve Jones, "A methodology for Service Architectures", Cap Gemini's contribution to OASIS Adoption Blueprints technical Committee.2006.

[17] Jason Bloomberg, "The role of the service-oriented architect, Senior Analyst ZapThink LLC. January 2003.

[18] Ali Arsanjani, "Principles of advanced software engineering: Variation-oriented analysis, design and implementation", January 2000.

[19] Boris Lublinsky , "Unifying data, documents and processes", Enterprise Architect, 2(2):6–11, Summer 2004.

[20] Addison Wesley, Wokingham. Jacabson I, Ercisson M. and Jacobson A, "The Object Advantage: Business Process Reengineering with Object Technology", 1995.

[21] Dong M. and Chen, "Process Modelling and Analysis of Manufacturing Supply Chain Networks Using Object-Oriented Petri Nets", F. 2001

[22] Giaglis G.M, "A Taxonomy of Business Process Modelling and Information Systems Modelling Techniques" International Journal of Flexible Manufacturing Systems, 2001.

[23] Dong M, "Service Oriented Architecture Reference Model", Working Draft 07, 12 May 2005.

[24] Srikanth Inaganti, Gopala Krishna Behara, "Service Identification: BPM and SOA Handshake" March 2007.07.1

[25] Ali Arsanjani , "Service oriented modeling and architecture - How to identify, specify, and realize services for your SOA", 09 Nov 2004.

[26] S. Chen, "Service Oriented Architecture Adds Flexibility to Business Processes", Research Paper. 16 February, 2005.

[27] Ali Arsanjani, "Service Oriented Architecture and Process", Research paper. 10 February, 2006.

Nafise Fareghzadeh was born in Tehran in Iran, on July 9, 1983. She graduated from the South Tehran Branch Islamic Azad University with the degree of M.Sc of software computer engineering. She is a is the associate professor at Computer Department of Islamic Azad University of Zanjan and Tehran, Payame Noor University of Tehran, Raja Uniersity of Quazvin. She has been a computer software engineer with several software projects in Iran Railway and other Organization since 2003. She is author of two books. Furthermore, she has published several technical papers in conferences and journals.