A Framework for Goal-Oriented Methods for Services

Omnia Saidani
Ker RIADI, University of Manouba Tunisia
omnia_iag@yahoo.fr

Rim Samia Kaabi
RIADI, University of Manouba Tunisia
kaabi_rim@yahoo.fr

Naoufel Kraiem
RIADI, University of Manouba Tunisia
naoufel.kraiem@ensi.rnu.tn

Yousef Baghdadi
Department of CS, Sultan Qaboos University, Oman
ybaghdadi@squ.edu.om

Abstract: An agile enterprise rapidly adapts to changes in business requirements by identifying, conceiving, deploying, and composing capabilities into solutions in response to normal or disruptive events. Service and Service-Oriented Architecture (SOA) plays a crucial role in promoting agility by rapid modeling and realization of such solutions as compositions of services. However, services that compose solutions are developed from an IT perspective, which yields brittle services, as they are not aligned with business requirements. A business perspective of service would raise the level of abstraction, which would better align IT with business. This work proposes a framework that considers four critical views of service in terms of What (subject), Why (goal), How (method), Who/Which (realization role). Each view presents a set of attributes. The framework questions: (1) to what extent a goal-oriented approach to service would consider services from different views representing intentional aspects, (2) to what extent these views would conform to the abstraction-refinement of services in order to align IT with business requirements, and (3) whether the existing approaches could be extended or consolidated into a new comprehensive approach. It is meant first to compare different types of approaches, then to propose a consolidated goal-oriented approach, and finally to highlight issues that need further research.

Keywords: Service, SOA, Service Abstraction, Service Intentional Aspect, Service View Goal-Oriented Approaches, Comparison Framework, Consolidate Approach

I. INTRODUCTION

Enterprises face changes both locally in their own context and in their environment. The changes that alter the enterprise and its relationships with its customers, partners and suppliers are mostly enabled, if not driven, by Information Technology (IT) [1]. Therefore, enterprises need to adapt promptly to changes in business requirements by identifying, conceiving, deploying, and composing capabilities into solutions, in response to normal or disruptive events. Service Oriented Architecture (SOA) promotes agility by composing existing assets provided as services into solutions [2]. Although, SOA is an architectural style for business promoting agile e-business with lower complexity, lower cost, and more flexibility to changing requirements [2]. It is mostly used as computational architecture [3] [4] [5] [6] as it is a distributed architecture, realized with Web services, that promotes interoperability and loose coupling.

Despite the advantages offered by SOA, the solutions associated with it are dedicated to a certain category of stakeholders, namely IT developers; and remain difficult to translate to the business world. This is due to a lack of methodologies to develop consistent and robust Web services-based SOA that can re-architect enterprise computing systems [7] [8] [9] [10] [11] [12] [13] [14]. Indeed, the existing approaches, based on a case-by-case wrapping of existing applications, yield brittle Web services [15] and not enough reliable, manageable, or reusable Web services [16] [17]. Nowadays, Information Technology platforms such as J2EE and .NET drive web services development processes [18] [19] [20] [21]. Moreover, these IT-oriented approaches are not sufficiently aligned with the business requirements [22] [23] [24] [25], which yields a brittle SOA. For instance, tools do not enable a critical task such as the identification of the services that a business requires their IT to support, which is a determining factor in creating and migrating to a successful SOA [26] [3] [27] [28].

A business perspective of service would raise the level of abstraction, which would better align IT with business. A perspective that describes the services at a higher level of abstraction, in terms of needs that enterprise can fulfil, is required. That is, these services need to be described with an intentional perspective, whereby their description,
publication, matching, and composition are geared towards some goals. This would allow a transition in SOA levels: from a technical level to an intentional level, we refer to as ISOA for Intentional SOA [29]. Many goal-oriented approaches have set this transition objective. Yet, they have not taken into the relevant well-specified views of service from an intentional aspect, which would facilitate a kind of model-driven development of services and solutions built by using these services.

This work builds on top of the framework, proposed in [30] to consider four critical views of service in terms of What (subject), Why (goal), How (method), Who/Which (realization role). Each view presents a set of attributes. The framework questions are:

Q1  To what extent a goal-oriented would consider service from different views representing intentional aspects
Q2  To what extent these views would conform to the abstraction-refinement aspects of services in order to align IT with business requirements
Q3  Could the existing approaches be extended or consolidated into a new comprehensive approach

Then, it proposes a comparison framework to [30]:
1. Provide answer to those questions
2. Highlight issues that need further research to provide the required, comprehensive goal-oriented that would align business with IT in order to respond to the rapid business changes and enterprise transformations.

The remainder of the paper is organized as follows: Section 2 shows how to use SOA to align IT on business. Section 3 presents the multidimensional framework. Section 4 compares and classifies the existing goal-oriented approaches for services within the framework. Section 5 presents some related work. Finally, a conclusion presents some perspectives and further development.

II. SERVICE, SOA, AND ALIGNMENT BUSINESS/IT

II.1 Service

From a general perspective, Hill in [31] defines service as “a change in the condition of a person or a good belonging to some economic activity, brought about as the result of the activity of some other economic activity, with the approval of the first person or economic activity”.

From a business perspective, service is considered as “the application of a capability by a business entity to provide a business value that addresses the needs of a community of service users” [2].

From IT perspective, a service is either a Web-enabled service such as e-service, Web service or data service. Specifically, from the software perspective, a service is defined by [32] [6] as “the basic building block for developing a software system in a specific domain”, and by [3] as “some set of application functionality”. From [33], we can compile a set of properties to define a Web-enabled service as a computational resource that is identifiable by a unique identifier. It performs one or more tasks through an agent (e.g., a provider agent that is a software unit); and is used by a requester agent. It has a description and an interface. The description is a machine-processable description of the service’s interface that shows the messages that are exchanged by the service. The interface defines the messages relevant to the service, that is, the different types of messages that the service sends and receives, along with the message exchange patterns that may be used. In addition, a service description may include one or more policies applied to it. Finally, a description is expressed in a service description language. In addition, services should (i) define the semantics of their functional (e.g. capabilities), non-functional (e.g. quality of service), and policy requirements in a machine-readable standard, and (ii) communicate through messaging protocols built on top of the Internet protocols [33].

Service orientation shapes service with a set of design principles, whereby a design principle is an accepted industry practice [5]. We build on Erl’s principles to provide a set of desirable properties (DP):

**DP1. Standard Contract:** Service contract, namely the request messages and the response messages (the value-added by the services) should be understandable by the requester services. Services adhere to a communications agreement defined in the service descriptions or any attached documents.

**DP2. Semantics:** what value services provide and what is the meaning of messages and the data elements they manipulate and communicate is a critical issue for using the service.
DP3. **Sharing and Reuse:** service value added is conceived with the intention of promoting reuse.

DP4. **Discoverability:** services are designed to be semantically descriptive so that they can be found, matched, and decided via discovery and composition mechanisms.

DP5. **Abstraction:** except from what is described in the service contract, services hide their logic from the outside world. Abstraction should be cohesive, complete and consistent. In other words, service contract, namely the request messages and the response messages (the valued added by the services) should be understandable by the requester services.

DP6. **Autonomy:** services have the entire control over the logic they abstract through contracts.

DP7. **Stateless:** services minimize keeping track of information specific to the requests. Services should not be dependent on the context or state of any requesters. That is services should be independent and self-contained.

DP8. **Messaging:** messaging plays a critical role in supporting the interoperability role. A message with an agreed-upon format allows services to interact with an interface abstraction that insulate consumers and providers from the physical characteristics of their specific technology.

DP9. **Self-contained:** service can run independently of the state of any other service or any other software.

DP10. **Granularity:** defining the granularity depends on how the service is (re-)use of the service.

These principles enable SOA, which itself promotes solutions as composition of principled services.

**II.2 SOA: Service Oriented Architecture**

OASIS [34] defines SOA as “a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations”.

Originally, SOA involves three actors and three operations. The actors are: service provider, service consumer, and registry, whereas the operations are: publish, find, and bind. It enables sharing of capabilities provided as services, whereby:

1. Consumers, providers, and registry are all services, where:
   - Services have separated concerns
   - Services have hidden implementation
   - Services are loosely coupled
   - Services are interoperable

2. Interaction and communication mechanisms are based on messages to facilitate operations such as publish, find, match, and bind for efficient interoperability

3. Service management artifacts, including services inventory (e.g., adapted registry)

4. Composition and service organization and management, including inventory and discoverability mechanisms, whereby a composition defines:
   - The components involved as services
   - The service relationships in terms of (co-)values creation
   - The orderly execution of these services
   - The controlling of their execution
   - Their interactions in terms of messages:
     - Publishing of the services
     - Matching of the services
     - Finding of the services
     - Binding to the services
     - Substituting to the services
   - The dependability of the services
   - The exceptions

SOA mainly aims at:

1. Reducing redundancy of functionality, which results in an increased reuse of services, which implies a reduction of the development costs, and a rapid marketing of applications that support new BPs

2. Decoupling business logic from the service, which enables more flexibility, as this logic can be easily changed, updated, or even replaced to address new business needs.
Composing the business logic provided as services in new ways without having to change or rewrite them. This leads to business agility as well as lower cost. Any application can reuse the existing business functions without having to re-write their logic.

Organizing and managing the service and solutions built on top of them such as EAI and B2B. Without a set of management and monitoring capabilities, SOA performance and stability will degrade [3] [35] [17] [11] [16].

II. Alignment business/IT

Although SOA is well specified and its benefits are clear, we still have to answer questions such as ‘Why should we use it?’, ‘When to start it?’ and ‘How to start it?’ [36] [28] [12] in order to avoid any effects of IT fashion-kind on organizations [25]. Therefore, to use fully Web services as Internet instance content of SOA, Web services need to be systematically identified, specified, designed, deployed, organized and managed. This requires taking into account not only the properties of the Web service itself as fundamental building block, but also many other perspectives such as business and IT [37] [24].

Therefore, an architecture that decouples business and IT is necessary [22] [24] [37]. Indeed, the Web services development concerns with multiple perspectives from both business and IT. For instance, deciding ‘Why should we move to Web services?’, ‘What are the intentions of Web services?’, or ‘What should be or should not be Web service?’ is independent of the technology that develops the purposeful Web services. Only such architecture should provide guidance for a method to develop Web services. That is, a method to first identify, design, and organize Web services, then deploy and manage with a well-suited technological environment.

III. THE FOUR VIEWS FRAMEWORK

III.1 Framework definition and role

The four views framework has proved its efficiency in enhancing the understanding of various engineering disciplines, information systems engineering [39], requirements engineering [40] and method engineering [41]. It can also help in understanding the field of service engineering which consists of applying engineering approaches, techniques and tools the understanding of services approaches.

The subject view contains knowledge of the domain about which the proposed information system has to provide. It contains real objects which become the subject matter for system modeling.

The goal view describes the organizational environment of the information system. It deals with the intentional aspects of the information system to be built.

The method view includes specifications at different levels of details of what the system does. It holds the modeled entities, processes, etc. of the subject view.

The tool view concerns tools and means used for the implementation of the approach.

III.2 The proposed Framework

The framework adopted consists of four views, as shown in Figure 1. Each of view provides a specific knowledge and analysis of service. Each view tries to answer fundamental perspectives of services, in terms of What (subject), Why (goal), How (method), and Which (tools), in order to identify:

- The subject service is about, in terms of the properties describing the service, specifically at higher level of abstraction
- The goal of the service
- The method to achieve the goal
- The mechanisms and techniques used to realize the goal

Each viewpoint is measured by a set of relevant attributes. These attributes have values that are defined within a domain, whereby a domain may be a predefined type such as integral or boolean, an enumerated type ({x, y, z}), or a structured type (Set {x, y}).
III.2 The subject view

The subject view concerns with the goal of the service. In this view, at least two perspectives of services must be taken into consideration account: technological perspective and business perspective [42].

From a technological point of view, a service is defined as a set of modular applications self-contained and self-describing, which can be published, located and invoked from the web [43].

From a business perspective, the focus is on the specification of the user requirements that need to be satisfied by services independently of the technical details underlying the running execution platform. From this perspective, a service is a logical grouping of required components to satisfy a particular business request [44]. Moreover, the intentional perspective can be used. It reflects the fact that a service presents an intentionality made by a goal that allows its customers to achieve [45]. Thus, the subject view can be characterized by the attribute Entity defined as:

\[ \text{Entity: set \{technological, business, intentional\}} \]

Service can be characterized by its adaptation to the customer profile that requests it. That is, the adaptation of the service to the user context. According to [46], context represents all information that can be used to characterize the situation of an entity (person, physical or Data object). More generally, any element may influence the behavior of an application. Thus, the adaptability is another relevant attribute. It is defined as:

\[ \text{Adaptability: Boolean} \]

Decision-making means the guidance process to the user towards the realization of its goal by the execution of the most appropriate service to its usage context [47][48]. In [47], the authors proposed such a kind of user’s guidance process, which allows the execution of the most suitable service to their context. Service can be characterized by the attribute called Decision-making defined as:

\[ \text{Decision-making: Boolean} \]

III.2.2 The goal view

The goal view explains the WHY of the approach. According to [49], the publication of services is the process that allows the service provider to introduce the service descriptor in the directory and associate a number of information elements to make the service visible and discoverable, which allow users to find it later.

A web service is called compound or composite when its execution involves interactions with other web services to use their features. The web service composition specifies which services need to be invoked, in what order it is invoked, and how to handle exception conditions if any [50].

Validation aims to ensure that compounds services meet the functional and non-functional user requirements [51]. The foregoing suggests only one attribute: Goal. It defined as:

\[ \text{Goal: set \{publication, discovery, composition, execution, validation\}} \]
III.2.3 The method view

This view describes the method used to fulfill user’s goal. This view can be described by five attributes namely: publication method, discovery method, composition method, execution method, and validation method.

The attributes values are:

- **Publication method**: set {Map, ontology}
- **Discovery method**: set {Map, ontology}
- **Composition and execution method**: set {static, dynamic}
- **Validation method**: set {language (), formalism ()}

“Ontology” is an explicit formal specification of a shared conceptualization [52]. The term “conceptualization” refers to an abstract model of some phenomenon of the reality; and who can identify relevant concepts of this phenomenon. It consists of a finite set of concepts and relationships between these concepts where a concept is a class of domains [53].

A Map is a labeled directed graph with intentions as nodes and strategies as edges between intentions [54][55]. An edge enters a node if its strategy can be used to achieve the corresponding intention. Since, multiple edges can enter a node, the map is capable of representing the many strategies that can be used for achieving an intention.

An intention is a goal to be achieved by the performance of the process. Each map has two special intentions. A strategy is an approach, or a manner to achieve an intention.

In [49], the author proposes a methodological guide to (i) publish the intentional service descriptor, and (ii) the service publication process. He uses ontology to capture, from the provider side, the semantics of intentional service descriptor and, from the user side, the need to satisfy it. In addition, he uses the conceptual model ISM [29] to help in describing the nearest service logic to the business side.

SATIS approach [56] proposes to search web services based on the needs of end users. On the one hand, this approach relies on a process of intentional representation. On the other hand, it relies on the semantic description in OWL-S (Ontology Web Language for Services), such as in [57], where web services are augmented with semantic descriptions. It is designed to provide greater expressiveness by allowing the description of the services in order to discover, invoke, compose and manage Web services in automated way as much as possible.

In [51], the author proposes a validation of functional and non-functional requirements. Moreover, the used precision technology measures the number of real services among the relevant services and selected components, and the recall technology assesses the number of selected and component services from the relevant actual services and high QoS that exist. Concerning non-functional requirements, [51] proposes to use simulation technology to automatically check that the services selected and compounds that best meet these requirements.

III.2.4 The tool view

The tool view deals with the way in which services can be discovered, composed and executed. That is, the tool environment needed to support service discovery, composition and execution. The attributes in this view are: Discovery tool, Composition and execution tool.

- **Discovery tool**: set {language () platform ()}.
- **Composition and execution tool**: set {language () platform ()}.

[49] is a platform for services discovery and selection that promotes intelligent interaction between the natural user intentions and the execution environments of existing semantic web services. [48] provides a platform for discovery and dynamic composition of services. It includes a service ontology-based goals and an algorithm for dynamic composition. In the next section, we present eight goal-oriented approaches for services with respect to the presented framework.
IV. CLASSIFICATION OF GOAL-ORIENTED APPROACHES FOR SERVICES WITHIN THE FRAMEWORK

In this section, we use the multidimensional framework to present the state of the art in goal-oriented approaches for services; and to open some research issues in the field.

IV. 1 GODO: Goal-oriented discovery for semantic web services

The GODO approach [58] provides a platform for services discovery and selection that enables interactions between the user's intentions and execution environments of existing semantic web services. The main objective of this platform is to help users to express their wishes and realize it using semantic web services.

On the basis of the proposed framework, we can situate the GODO approach as follows:

**Subject view**
- Entity: technological
- Adaptability: no
- Decision-making: no

**Goal view**
- Goal: discovery

**Method view**
- Discovery method: ontology

**Tool view**
- Discovery tool: platform (GODO)

IV. 2 A methodological approach for intentional service modeling

In [29], the authors introduce the notion of intentional service, i.e. a service spelt out in terms of the goal it allows to fulfill with the aim to provide the business executives a description of services available in a service portfolio that is adapted to their own perceived needs for services. Further, the authors explained how modeling intentionality of a given Business, through the Map model, provides the methodological support for intentional service discovery and recursive service composition definition.

Business centric organizations supported by e-business services might decide to describe them in an intentional manner, using the ISM (Intentional Service Model).

This approach can be situated within the presented framework as follows:

**Subject view**
- Entity: intentional
- Adaptability: yes
- Decision making: no

**Goal view**
- Goal: {discovery, composition, execution}

**Method view**
- Discovery method: Map
- Composition and execution method: dynamic

**Tool view**
- Agent based architecture

IV. 3 A refined goal model for semantic web services

This model provides a goal description defined by the extension of WSMO (Web Service Modeling Ontology) for the service discovery and selection based on ontologies. WSMO (Web Service Modeling Ontology) [60] is a service ontology that defines four high-level elements to describe the following semantic web services: ontologies, services, mediators and objectives.
The original WSMO goal model is extended by:

- The differentiation of goal templates and goal instances, which allows performing expensive discovery and composition operations at design time; and supporting goal formulation by customers through graphical user interfaces,
- The use of WG mediator, that links goal templates with usable web services, makes all information for automated invocation and consumption in form of client interfaces, and
- The composite goals for specifying restrictions on the workflow for solving a goal.

We can situate this approach to the presented framework in the following way:

**Subject view**
Entity: technological
Adaptability: no
Decision making: no

**Goal view**
Goal: discovery

**Method view**
Discovery method: ontology

**Tool view**

---

**IV. 4 A semantic approach guided by intentions**

The approach proposed by [56] focuses on web service search based on the needs of end users. It is based on both intentional and semantic representation of services. This approach introduces on a modular representation of the search process using reusable and combined process fragments to keep and share knowledge in web services search within the semantic memory of a neuroscientific community.

On the basis of the proposed framework, we can situate this approach as follows:

**Subject view**
Entity: business
Adaptability: no
Decision making: no

**Goal view**
Goal: discovery

**Method view**
Discovery method: {Map, ontology}

**Tool view**
Discovery tool: {language (SPARQL, OWL-S), platform (CORESE)}

---

**IV. 5 Towards a goal-based service framework for dynamic service discovery and composition**

This approach [48] provides a goal-based framework for dynamic service discovery and composition. This framework consists of a set of design principles and guidelines for service platforms to realize dynamic service discovery and composition. It includes a precise definition of goal, techniques for goal assessment and modeling, and the rationale for dynamic service discovery and composition.

This approach can be situated within the presented framework as follows:

**Subject view**
Entity: technological
Adaptability: yes
Decision making: yes

**Goal view**
Goal: {discovery, composition}

**Method view**
Discovery method: ontology
Composition and execution method: dynamic

**Tool view**
Discovery tool: {platform (framework for discovery and dynamic composition of services)}
Composition and execution tool: {platform (framework for discovery and dynamic composition of services)}

**IV. 6 An approach for dynamic services adaptation according to the user context by adopting an intentional approach**

This approach [47] provides a semantic description of intentional services and an intentional service platform, which are sensitive to context, for the dynamic service adaptation according to a given context. In fact, this platform includes the intentional service discovery and composition depending on the context, by the selection of the most appropriate service variant to the user needs and intentions, the predictive selection of services from the user model analysis, and the archiving of its context; and finally, the guidance of decision-making process of the user to its realization by the execution of the service most appropriate to its context of use.

We can situate this approach to the presented framework in the following way:

**Subject view**
Entity: intentional
Adaptability: yes
Decision: yes

**Goal view**
Goal: {discovery, composition, execution}

**Method view**
Discovery method: {Map, ontology}
Composition and execution method: dynamic

**Tool view**

**IV. 7 Multi-perspective approach centered web services composition requirements**

In this approach [51], the authors aim at providing a proposal of a QoS layer. Besides, they propose to extend a tool for automatic discovery of relevant software services. In addition, they present the definition and the implementation of a guidance method for automatic selection of relevant software services and high QoS. Validation of the composition is taken into account in this approach.

This approach can be situated to the proposed framework in the following way:

**Subject view**
Entity: business, intentional
Adaptability: no
Decision making: no

**Goal view**
Goal: {discovery, composition, execution, validation}

**Method view**
Discovery method: Map
Composition and execution method: dynamic
Validation method: Formalism (precision, recall, simulation)

**Tool view**
Composition and execution tool: {language (BPEL)}

**IV.8 PASiS**

PASiS approach [49] proposes an intentional modeling and semantic annotation to reuse business services. It provides a publication guide process that helps the designer of service in this task. This process uses ontologies to capture the semantics of intentional service descriptor. This approach is also devoted to the specification and the implementation of intentional service descriptor, where the annotation is used as a technique of semantic extension to the three ontologies iSOnto, vOnto and pOnto. In addition, the author proposes to extend SAWSDL language to describe intentional services.
PASiS approach may be situated within the presented framework as follows

**Subject view**
- **Entity**: business, intentional
- **Adaptability**: no
- **Decision making**: no

**Goal view**
- **Goal**: {publication, discovery}

**Method view**
- **Publication method**: Map
- **Discovery method**: Map.

**Tool view**

---

**IV.9 Concluding remark**

The proposed framework allows us to study various aspects of services. It consists of four views, each of which provides an analysis of a particular aspect of the service concept by asking a fundamental question: What, Why, How and Which. In each of the used views, are defined the attributes that specify the characteristics of a given service. We illustrated the framework by studying and positioning eight approaches representing the state of the art. Table 1 summarizes the results of this analysis.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>Entity</td>
<td>Entity</td>
<td>Entity</td>
<td>Entity</td>
<td>Entity</td>
<td>Entity</td>
<td>Entity</td>
<td>Entity</td>
</tr>
<tr>
<td></td>
<td>Technological</td>
<td>Intentional</td>
<td>Technological</td>
<td>Business</td>
<td>Technological</td>
<td>Intentional</td>
<td>Business, intentional</td>
<td>Business, intentional</td>
</tr>
<tr>
<td></td>
<td>Adaptability</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Decision making</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Publication</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Discovery</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Composition</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Execution</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Validation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Goal</td>
<td>Publication</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Map</td>
</tr>
<tr>
<td></td>
<td>Discovery</td>
<td>Ontology</td>
<td>Ontology</td>
<td>Map, ontology</td>
<td>Ontology</td>
<td>Map</td>
<td>Map</td>
<td>Map</td>
</tr>
<tr>
<td></td>
<td>Composition</td>
<td>Dynamic</td>
<td>Dynamic</td>
<td>Dynamic</td>
<td>Dynamic</td>
<td>Dynamic</td>
<td>Dynamic</td>
<td>Dynamic</td>
</tr>
<tr>
<td></td>
<td>execution</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Precision</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Validation</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Recall</td>
<td>Simulation</td>
</tr>
<tr>
<td></td>
<td>Tool</td>
<td>GODO</td>
<td>-</td>
<td>SPARQL</td>
<td>Framework for</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Discovery</td>
<td>GODO</td>
<td>-</td>
<td>SPARQL</td>
<td>discovery and</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Composition</td>
<td>Agent</td>
<td>Agent</td>
<td>Agent</td>
<td>dynamic composition of</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>and execution</td>
<td>based</td>
<td>architecture</td>
<td>architecture</td>
<td>services</td>
<td>BPEL</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Tool</td>
<td>-</td>
<td>-</td>
<td>Framework for</td>
<td>discovery and</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Discovery</td>
<td>-</td>
<td>-</td>
<td>dynamic composition of</td>
<td>services</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table 1. Classification of goal-oriented approaches for services**

V. **Related Work**

Despite the amount of research devoted to service, very little attention has been paid to the understanding of existing goal oriented method for services. We believe that comprehensive comparison and analysis of existing goal oriented approaches will result in a better understanding of the current state of the art.
For instance, Aljoumaa presents in [49] a state of the art of the proposed work in the literature relating to engineering information system based on services. It addresses in particular, to the different models, languages and systems for description, publication, query formulation and services research, with a particular emphasis on goal-oriented approaches.

Approaches that have been selected by Aljoumaa propose on the one hand, an original process for query formulation (intelligent or guided) based on natural language or semantic goal-oriented models and on the other hand, to significantly improve the service matching algorithms, and the semantic extension exploited in the service directories.

In his work, that interested to the insurance of alignment between web services based applications and the business requirements, Driss [51] exhibited various approaches that have adopted for a centered requirements vision for the service composition and classified them according to three categories of approaches: formal, semi-formal and informal approaches. However, none of the presented approaches perform all phases of life cycle of the service-based applications.

In general, several efforts have been made to survey service modeling, composition, execution, each of which has several advantages. However, to the best of our knowledge, none of these efforts share our literature review to cover various goal oriented approaches for services. We present a novel classification of goal oriented approaches for services based on a multidimensional framework in order cover the most important aspects.

VI. CONCLUSION

We have developed a multidimensional framework to understand and classify goal-oriented approaches for services. The different viewpoints of the framework allow identifying and clarifying a particular service aspect. The Subject and the Goal views constitute the universe of discourse within which the Method and Tool viewpoint of services lie. The entity, the adaptability, and the decision making, imposed by the Subject viewpoint, determine the type of contents. The Method view expresses where the goals of the service uses are established. In fact, we can use services for publication, discovery, composition, execution or discovery. In a similar manner, we believe that the technical solution in the tool viewpoint has to be chosen according to the method decided in the method viewpoint.

This framework can further be developed to guide a comprehensive goal-oriented method for service development.

REFERENCES

[18] B. V.Kumar, P.Narayan, and T. Ng, ‘Implementing SOA Using Java™ EE’, Addison Wesley, 2010

