Using Goal Diagram and Business Process Modeling to facilitate the identification of Domain Specific Services

Fernanda Aparecida Rocha da Silva
Federal University of São Carlos
São Carlos/SP
Brazil

ABSTRACT

Background: A promise of SOA is to make business processes be quickly adapted to organizations dynamic through flexible software. For this, it is essential to have a support for services identification in order to meet the business goals. However, many available services found on the web environment are too specific and can hardly be reused in different applications. This happens because there is a lack of systematic approaches to support generic services identification in a systematic way. **Objective**: To present a strategy for identifying generic services that support business processes. The identification is supported by Goal Diagrams and Business Process Models and is composed by a set of guidelines which assist the domain engineer in extracting the services. The identified services are generic enough to be reused in similar applications of a specific domain. **Methodology**: To elaborate the proposed strategy some domain-specific business processes were analyzed, aiming at extracting key tasks and turning them into generic web services. This analysis was supported by an extended version of goal diagrams (GTR) and conventional BPM models. **Results**: As a proof-of-concept we applied the strategy for identifying services in the planning processes domain and we developed a real e-gov web portal based on the identified services. The web portal was used successfully by two different schools for elaborating their planning processes. **Conclusion**: We claim that the strategy is generic and can be applied to other business processes providing software suitability to the organization dynamics. In addition, it can be potentially reused with services in different instances of the same business process.

General Terms

Service-Oriented Architecture, Software Engineer, Business Process Modeling.

Keywords

Service-Oriented Architecture, Business Process Management, Goal-Oriented Requirements Engineering

1. INTRODUCTION

Usually, enterprises are submitted to constant challenges in real scenarios, because of several reasons, such as: market pressures, new user requirements or even the need to evolve. In the environment, SOA (Service Oriented Architecture) is one of the methods to make an enterprise system adaptable. This is because SOA provides a platform for more agile IT environment by enabling IT professionals to build flexible systems that can adapt. SOA is a development style as well as a way to integrate software, in which ‘services’ are defined to be common to a range of systems, as internals as externals of an organization, dealing with them individually and making them available over a network [1].

One of the main discussion issue related to SOA refers to the way of identifying services [2][3][4][5][6][7]. Identifying suitable services having the right level of granularity can have broad influence on the whole system and is critical for achieving agility. For identifying proper services, an organization’ business requirements and business change factors should be analyzed to meet the business objectives and agility, especially in light of potential changes in the business environment [8]. If not, it is difficult to make service-oriented systems flexible enough to cope with future changes in terms of business agility.

This paper presents a set of guidelines for web service identification. The main point is to identify reusable services which can be used in different instances of the same business process. The strategy is conducted from a goal-oriented diagram (AWARE model - Analysis of Web Application Requirements) and BPM (business processes modeling). It is involves 6 stages, including analysis of business process, the identification and design of the services. One of the advantages of the strategy is the identification of fine-grained services. This happens because the services are identified from tasks of the AWARE models. Furthermore, the service reusability is one of the main points considered in the proposal. The AWARE model is used because it aims to capture business goals; the business process modeling was chosen because it shows process dynamics and SOA concepts, and because it allows software development that meets organization needs in an agile and flexible way.

In Section 2 are presented the concepts related to goal-oriented requirements and business process. Section 3 describes the strategy of identification service proposed in this paper, which was applied in the construction of a web portal in Section 4. Section 5 describes related works. Finally, Section 6 presents final considerations.

2. GOAL-ORIENTED REQUIREMENTS AND BUSINESS PROCESSES

The description of organizational behavior through “goals” was proposed by several authors and presented by various methods and techniques that incorporate Goal-Oriented Requirements Engineering [9][10].

The AWARE model [11][12] is considered an extension of the framework i* [13].[14] This model adopts a goal-oriented approach in order to identify and specify requirements. In this model, each stakeholder’s goals are identified. Goals, tasks, and requirements are depicted in a hierarchical diagram, which we call "Goal-Task-Requirement" (GTR) Diagram.

Regarding to goals, Bolchini and Paolini [11] asserts that “a goal represents a high level objective of one or more stakeholders. Goals may represent users’ goals and main stakeholders’ goals”. For each identified goal we must verify if it can be broken into sub-goals. Bolchini and Paolini [11] asserts that “by refining user goals, user tasks may be defined. While a goal is a desired state, a user task is a specialized activity”.

According to Bolchini and Paolini [11], "requirements don't aim to capture all the functionalities, but only those crucial features needed by designers to shape the user experience and by stakeholders to agree on initial specifications". Therefore, a requirement is a short description of the task, what makes its understanding better.

Figure 1 shows an example of GTR diagram of the AWARE model [12]. At the upper part of the figure there is a symbol representing a stakeholder of the system. Stakeholders can have "goals", which are represented by ellipses. Each ellipse can be broken down into tasks or requirements. Requirements are represented by rectangles.

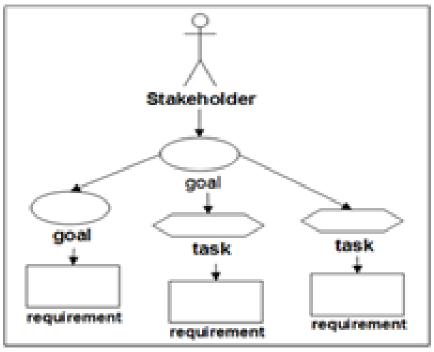

Fig 1. Example of Goal-Task-Requirement Diagram

A Business Process is another concept used in this work. According to Hammer and Champy [15], a business process is a set of activities that are carried out in a logical sequence with the purpose of producing a product or service of value to a specific group of customers.

The modeling of business processes consists in facilitating the understanding of organizational structure, business rules, objectives, activities and responsibilities of those involved as well as the data handled using it for business models represented in diagrams such as BPM [16].

In order to make GTR diagram support service identification and, consequently, support the strategy definition, its notation was extended as presented in [17]. The proposed strategy is presented in the next section.

3. STRATEGY FOR IDENTIFYING REUSABLE DOMAIN-SPECIFIC SERVICES

The strategy aims at identifying and designing generic web services, so a system that takes advantage of these services should be flexible enough to support different instances of the same business process. Goal diagrams (GTR) of the AWARE model and BPM diagrams are used as the main artifacts during service identification. Figure 2 outlines the stages that comprise the strategy.

The rounded corner rectangles are stages of the process and the white sheets are the generated artifacts. In Stage 1 (Recovering Business Processes Instances) we must obtain different instances of the same business process to characterize the domain. In Stage 2 (Elaborating Instance GTRs), a Goal Diagram is elaborated for every business process instance recovered from Stage 1. In Stage 3 (Formulating the Domain BPM), the BPM diagram is created to represent the business process dynamics. This diagram specifies the interactions among the tasks needed to reach the aims. In Stage 4 (Elaborating the Domain GTR) a GTR diagram is built representing the domain. In Stage 5 (Identifying Services) we must analyze every task of every business process instance to identify the services. Therefore, a list of services must be built based on the GTR and BPM. Here, we must also take note of the tasks identified as service. In Stage 6 (Design Services), WSDL (Web Service Description Language) drafts.

From here, every stage of the proposed strategy is explained in more details. The guidelines were structured in three subsections to make the comprehension better: Aim, Implementation Steps and Resulting Artifacts.

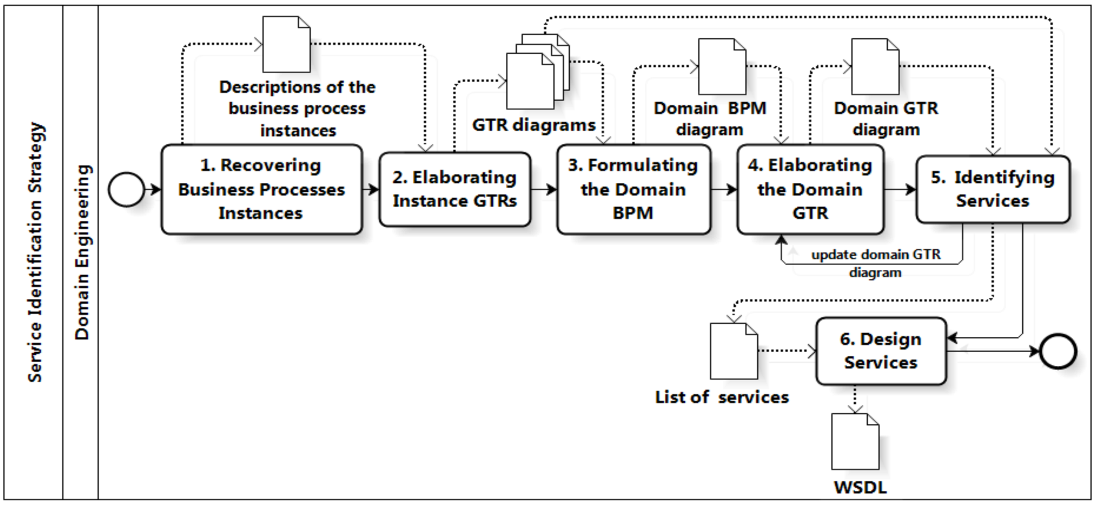
Service Identification Strategy
Domain Engineering
Descriptions of the business process instances
GTR diagrams
Domain BPM diagram
Domain GTR diagram
1. Recovering Business Processes Instances
2. Elaborating Instance GTRs
3. Formulating the Domain BPM
4. Elaborating the Domain GTR
5. Identifying Services
update domain GTR diagram
6. Design Services
List of services
WSDL

Fig 2. Domain Engineering Phase of the Service Identification Strategy

Stage 1: Recovering Business Process Instances

***Aim**:* The aim here is to characterize the domain by searching for instances of a business process. An instance is a practical example of the carrying out of a business process.

Implementation Steps: The first step is to define the target business processes: planning process, managing process, manufacturing process, process of tracking changes, etc. Next, it is necessary to find instances of the chosen business process in the literature or in industry case studies. These instances can be recovered in any format: diagrams, textual documents, UML models, etc. The format is not relevant; what matters is to recover as much information as possible about the instances.

For example, suppose that the strategy is applied in the context of the trade process. We need to conduct a literature review and search for case studies which describe instances of this business process. When analyzing the papers, we needs to identify several kinds of sales; it is not good enough to concentrate just on one type, because this can constrain the number of variabilities found for the domain, impacting the reusability of the target web services.

Resulting Artifact: The format of the outcome artifact is not important here; the most important is to group information about the business process instances. These artifacts can be structured in diagrams, artifacts, textual documents, source codes or any other useful format.

Stage 2: Elaborating Instance GTRs

Aim: The aim here is to understand and make explicit the main goals of the business process by building a GTR diagram for each business process instance that was identified in the first stage. We call these diagrams "Instance GTRs" because they represent specific instances of the business process.

Implementation Steps: Before beginning this stage, we must standardize the name of the elements used in the instances recovered in Stage 1. These elements can be tasks, activities, sub-processes, etc. This must be done because usually there are different terms representing the same action, and the same term representing distinct actions.

The instances must be compared in pairs. Elements which represent the same action/function in the different instances must be renamed using the same name (one specific term). This term must represent the function and purpose of this element. Next, we need to start the building of the instance GTRs. So, the first step is to identify stakeholders. Next, for every instance of the business process we need to identify the goals of the identified stakeholders. The goals are represented by every stage of the business processes. For each identified goal we must verify if it can be broken into sub-goals. Each activity of each business process instance is a specialization of its stages and can therefore be considered a sub-goal. In this way each activity must be modeled as a sub-goal. Next, we need to break goals and sub-goals into tasks. As a final step, we need to break tasks into requirements.

Resulting Artifact: The outcome of this stage is a set of GTR diagrams, one for each business process instance.

Stage 3: Formulating the Domain BPM

Aim: A BPM diagram of the domain must be built to improve the understanding of the whole business process.

Implementation Steps: The notation used here is BPMN notation (Business Process Modeling Notation) [16]. The elements of this notation are grouped into four basic categories: flow objects, connecting objects, swimlanes and artifacts. The flow objects are: activities, events and gateways. Activities can be decomposed into tasks and subprocesses. The connecting objects correspond to sequence flow, message flow and association. The swimlanes are pools and lanes. Finally, the artifacts are objects of data, groups and annotations. The business process instances recovered in Stage 1 and Instance GTRs produced in Stage 2 must be used for construction of Domain BPM.

The first step is to analyze each element of the business process instances in order to understand the role of this element and identify which is the adequate BPMN notation to represent it [16]. The second step is to map every goal or sub-goal of the Instances GTR (Stage 2) as a process in the BPM diagram. The third step is to map each GTR task as a task in the BPM diagram because a task is an atomic activity that is included within a process. This can be applied to GTR and BPM diagrams [11, 16].

The fourth step is to identify the relationships among tasks for each business process instance to represent sequence flows, message flows and associations, according to Wen et al [17, 18]. The fifth step is to identify and represent the events that occur during the execution of each business process instance, as [16]. The last step is to identify and represent the decision points (gateways) as [16].

Resulting Artifact: The outcome of this stage is the Domain BPM diagram.

Stage 4: Elaborating the Domain GTR

Aim: In this stage the aim is to identify the commonalities and variabilities of the domain and to build a GTR diagram of the domain which we call here “Domain GTR”.

Implementation Steps: The first step is to identify the variabilities and commonalities of the domain. This can be done by comparing each element in every instance GTR. By “element” we mean goals, sub-goals and tasks identified according to the artifacts recovered in Stage 1. To do this the instance GTRs must be compared in pairs in order to identify the alternative, optional and mandatory elements. To conduct this analysis, any domain analysis method can be employed, such as the FODA (Feature-Oriented Domain Analysis) [20]. Mandatory elements are those that are required in all instances of the business process. Alternative elements are those that are mutually exclusive, that is, they are not present in all business process instances. Optional elements are those that may or may not be present in the business process instances.

In order to build a GTR which represents a domain, instead of an instance, it is important to use a new notation for representing the mandatory, optional and alternative elements. So we performed an extension on GTR diagrams in order to allow for a representation of these peculiarities. When an element (task, goal and subgoal) is mandatory in the domain, the label "man" is used; when the element is optional, the label "opt" is used; when the element is alternative we use "alt". Additionaly, when the element is a service, the label "ws" is used [16].

The next step is to identify dependencies. The dependencies among tasks are associated with the control flow of the business process, i.e. the execution sequence [18][19]. Consequently, we can infer that the data flow is contained inside the control flow. When the data transfer occurs between two tasks, the successor task uses the data modified or produced by predecessor task to perform its function (purpose), characterizing a dependency. Goethals et al [18] e Wen et al [19] shows how to identify explicit and implicit dependencies among tasks in the context of business process. In the strategy the dependency among tasks is related to business rules, which are identified in the artifacts recovered in Stage 1. The dependency among tasks is indicated in Domain GTR by a message (data) exchanged among them [16].

Resulting Artifact: The outcome of this stage is the Domain GTR.

Stage 5: Identifying Services

Aim: The aim here is to identify services based on the GTR and BPM of the Domain developed in the earlier stages. The granularity we have adopted for services identification is the "task". We took this decision because we believe that tasks contain well defined functions, giving to the generated services a greater reuse potential. So, for us, each task is a service.

Implementation Steps: The procedure here can be divided into two main actions; the first one is to identify services and the second one is to identify composition of services. The services are identified taking into account all the mandatory, optional and alternative tasks of Domain GTR and the compositions of services are identified through dependency relationships among tasks of the business process instances.

Tasks which are considered "services" are those that abstract the logic of functionality, are autonomous, and have a greater reuse potential. Tasks not directly associated to the main concepts of the business process are not considered services. In the same way, the tasks representing only create, read, update and delete operations are not considered services. During this stage Domain GTR must be updated, indicating which tasks are considered services, using the notation "ws" to denote each service.

Next, the composite services must be identified. For this, the dependencies shown in Domain GTR and also in the Domain BPM must be analyzed. The identification of composite services got easier as the dependency relations become clearer. So, dependent tasks are considered composite services, because they can act together for reaching their goal. When all the tasks of a Goal are indicated as a service, the goal must also be considered a composite service. So, the strategy must be used in an incremental form.

Finally, we need to build an artifact called "List of services" as can be seen in Table 1. This list summarizes the results produced in the previous stages, showing main elements that compose the business process. The purpose of this is to assist in building WSDL drafts and afterwards in the service implementation. The list must contain the following fields: 1. Task, 2. Dependent Task (if any), 3. Task Type, 4. Service, 5. Component Service - (corresponds to the dependencies between services), 6. Operation (signature of the methods) and 7. Message (to be filled in stage 6).

Table 1. List of Services

1	*2*	*3*	*4*	*5*	*6*	*7*
Task	***Dependent Task***	***Task Type***	***Service***	***Component Service***	***Operation***	***Message***
Task A		Mandatory	Service A		OperationA()	MessageA
Task B	Task A	Optional	Service B	Service A	OperationB()	MessageB
Task C		Alternative	Service C		OperationC()	MessageC
Task D	Task A, Task B, Task C	Mandatory	Service D	Service A, Service B, Service C	OperationD()	MessageD

Note that in Table 1, the field "1.- Task" is populated with the name of each task considered as service. The field "2.- Dependent Task" is populated with the name of the task considered dependent. When there are two or more levels of dependence associated with a task, each dependent task is annotated in field 2 without repetition and separated by a comma. In field "3.- Task type" is annotated the indication given the task among task among mandatory, optional and alternative. In field "4. - Service" is populated with the name given to a service equivalent task annotated in field 1. The field "5.- Service Component" is populated with the service name created from the dependent tasks (column 2). The fields "6.- Operation" and "7.- Message" are populated in the next stage, to support the construction of the WSDL.

Resulting Artifact: The outcome of this stage is the List of Services.

Stage 6: Design Services

Aim: The aim here is to design the services identified in stage 5.

Implementation Steps: The first step is to fill the fields "operation" and "message" in the List of Services, as follows:

1. The operations correspond to functions of each task identified as "service" in stage 5. This information is extracted from the descriptions of business process instances recovered in step 1.

2. The messages correspond to data that are exchanged among tasks. If the task is not associated with a composite service, then the messages are extracted only from the requirements shown in the Domain GTR. If the task is associated with a composite service, then this task has a dependency relation with another task and, therefore, the messages are extracted from the relationship that connect them and from the Domain GTR requirements.

The second step is to draft the operations and messages of the WSDL, based on the List of Service. Therefore, we need to know which are the possible actions associated to a task identified as service; that is, to know the behavior or what is executed in the task. The guideline for construction of the WSDL document is the standard available by W3C.

Resulting Artifact: The outcome of this stage are drafts of the WSDL and a class diagram.

4. APPLYING THE SERVICE IDENTIFICATION STRATEGY

The motivation of the case study was a request from the Brazilian Government for a Web platform called Web-PIDE which could support schools in elaborating their planning processes. One of the main requirements is to make the solution generic enough to be applicable to a number of schools containing slightly different instances of planning processes. Therefore, we developed and applied the strategy in the identification of generic services for "planning processes" so that we could come up with a set of generic services which support different instances of this business process.

Stage 1: Recovering Business Process Instances

The first stage of the strategy aims at retrieving information on the business process under analysis. "Planning Process" is the development of a project for performing organizational goals involving the choice of an action course, the foreseen decision about what must be done and the definition of when and how the action must be carried out [21]. We found in the literature four sources about planning processes:

Steiner [21], Fischmman [22], Tavares [23] and Oliveira [24]. We also count on the support of two schools that allowed us to perform interviews with the staff, describing their daily tasks regarding the planning process. So, based on the literature and also on the interviews, we were able to identify the stages, tasks of each stage and the main flows among the stages.

Stage 2: Elaborating Instance GTRs

In this stage we need to build a GTR for every planning process instance recovered in Stage 1, as can be seen in Figures 3(a), 3(b), 3(c) and 3(d). We opt for showing only the Tavares's instance in a bigger size because of space limitations. The others are only for illustrative proposals. Initially, we standardized the names of every element of the planning process instances recovered. This standardization avoids elements with the same function from being represented differently. In the case, the names of tasks already standardized, for example, are: "Define Mission", "Define Threats and Opportunities", "Define Strong Aspects and Weak Aspects", "Define Objective", "Define Strategy", "Define Action Plan", "Define Activity", "Define Indicator", "Define Criterion", "Define Goal" and "Define Cost." Next, we identified stakeholders. In the context of school management the following stakeholders were identified: pedagogical and administrative coordinators and school manager.

Due to space limitations only the stakeholder "Manager" and the "Formulate the Strategic Plan" stage of the planning process are shown from here, but the other stages, such as Implementation, Evaluation and Control, and Corrective Action were also analyzed and modeled.

Regarding the building of the instance GTRs, we decided to consider the stage "Formulate the Strategic Plan" as a starting point for all of them. We did that because this stage is one of the main goals of this process. Moreover, we also adopt the "Register Planning" as a default sub-goal for all instances.

Figure 3(c) shows part of the Instance GTR that was built based on Tavares's planning process [22]. This diagram concentrates only on the "Formulate the Strategic Plan" stage and shows the tasks we have identified as the most significant. As can be seen, for this author, the elaboration of the strategic plan must have five tasks: "Define Objective", "Define Strategy", "Define Indicator", "Define Criterion" and "Define Goal". The "Define Objective" task corresponds to expectations of an organization for medium and long term.

The "Define Strategy" task corresponds to ways the objectives can be met. The "Define Indicator" task corresponds to the instrument used to measure the extent of the efficacy and efficiency of the strategy. The "Define Criterion" task is used for choosing the source of the indicator. The "Define Goal" task corresponds to quantitative values for every objective.

Stage 3: Formulating the Domain BPM

In this stage we analyzed the planning process flow. The BPM domain model describe refers to the sub-goal "Register Planning" of the Instance GTRs diagram of Figure 3.

The sub-objective "Register Planning" was mapped as a sub-process in the BPM domain model. Every task related to it was mapped as a task in the BPM domain model. Analyzing the planning process instances of stage 1, we note that the starting point of the sub-process, "Register Planning", is to define the objective. Strategies are developed from the objectives. Every strategy is composed of an action plan, an indicator, and a set of goals. An indicator is composed of several criteria that guide the performance measurement. A list of goals is produced to quantify the expected results of the strategies. The action plan details the procedures for the execution of the strategy, which in turn is decomposed into a set of activities, producing a chronogram. At the end of this process we obtain documentation of planning: operational plan, list of goals, list of performance indicators and chronogram of execution.

Stage 4: Elaborating the Domain GTR

In this stage we built the Domain GTR, which can be partially seen in Figure 4. Initially, we analyzed every instance GTR in order to identify which elements are mandatory, optional and alternatives. Looking at the instances of Steiner [21] (Figure 3 a), Fischmman 22] (Figure 3 b), Tavares [23] (Figure 3 c) and Oliveira [24] (Figure 3 d) we can notice that "Define Objective", "Define Strategy", "Define Indicator", "Define Goal" and "Define Criterion" tasks can be considered mandatory because they are common to all instances. These tasks are labeled "man".

The "Define Action Plan" and "Define Activity" tasks are considered optional because they appear just in three of the four instances. These tasks are labeled "opt". In the context of sub-goals "Register Planning" was not identified as alternative tasks, as can be seen in Figure 4. This is because, none of the tasks analyzed here are mutually exclusive.

Analyzing the instances of Steiner [21] and Tavares [23] we noticed that in order to build indicators it is necessary to define formulas that quantify the performance. Thus, we need to establish what the appropriate criteria for this measurement are. Therefore, we noticed that the "Define Criteria" task complements the "Define Indicator" task, and so we identified a dependency between these tasks.

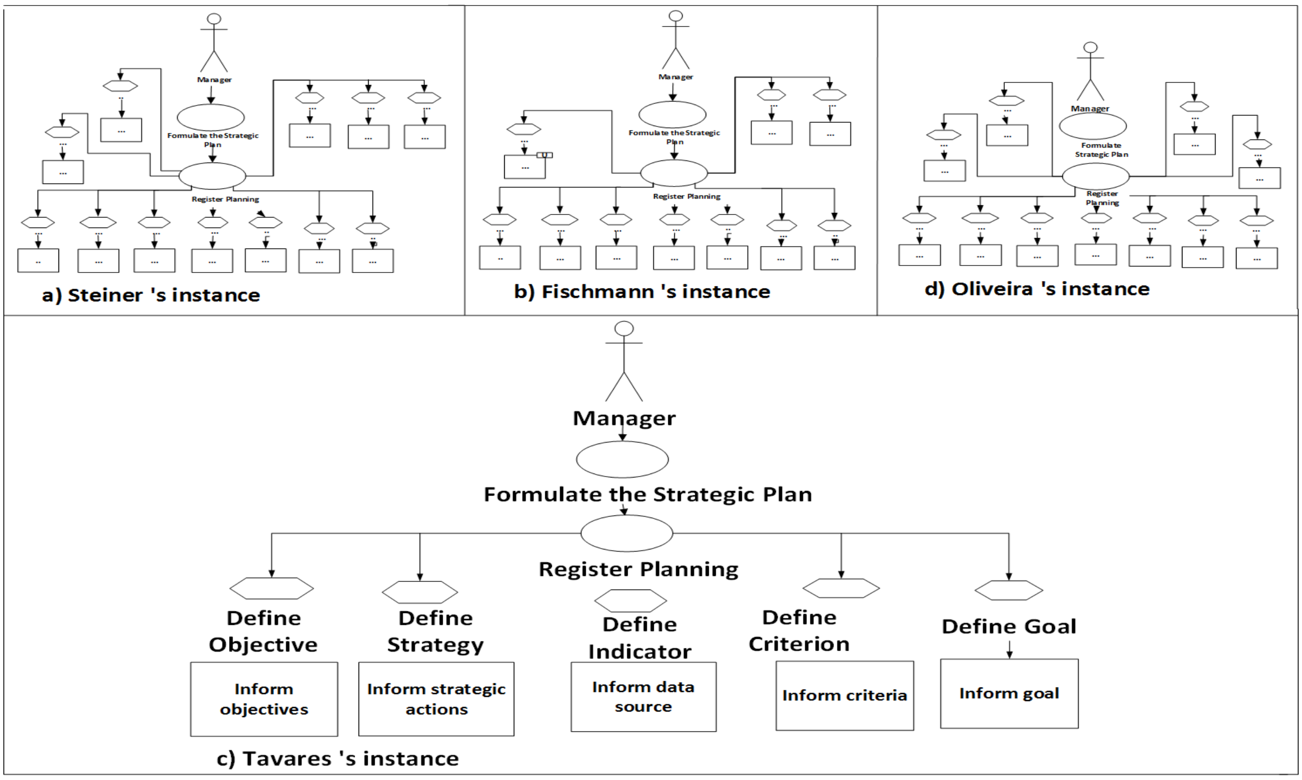

Fig 3. GTR Diagrams of the planning process instances

According to the descriptions of instances of Steiner [21] and Tavares [23], a strategy is detailed at two levels: action plan and activities, so that the objectives in planning are achieved. The action plan describes how each strategic action will be implemented. However, the activities detail what will be executed by each collaborator for every strategic action, generating a schedule. Therefore, to realize their functions the "Define Activity" and "Define Action Plan" tasks depend of the data processed by the "Define Strategy" task. In this way, the “Define Strategy”, “Define Action Plan” and “Define Activity” tasks are dependents.

Stage 5: Identifying Services

In this stage we identified the services, shown in Table 2. As can be seen in the Domain GTR of Figure 4, we identified mandatory and optional tasks. All these tasks contain well defined functions and represent concepts directly associated with planning. Besides, they are autonomous and have greater potential for reuse, so they are considered services. Therefore, the services are: “Define Objective”, “Define Strategy”, “Define Indicator”, “Define Criterion”, “Define Goal”, “Define Action Plan” and “Define Activity”. These tasks are labeled "ws" in the Domain GTR diagram.

The dependencies among tasks and the business process flow were analyzed to identify composite services. We note that in Domain GTR diagram show in Figure 4, the “Define Strategy”, “Define Action Plan” and “Define Activity” tasks are dependents; therefore they are considered a composite service. The “Define Indicator” and “Define Criterion” tasks are dependents; therefore together they form a composite service.

The Domain BPM model show interactions among tasks related to the sub-goal "Register Planning". We note that the information flow and sequence in which these tasks are organized shows the composition of a functionality, which carries out the purpose of one or several stakeholders. We note that all these tasks are considered as services. Therefore, the sub-goal "Register Planning" can be considered a composite service. This sub-goal is annotated with a label "ws" in the Domain GTR diagram.

Next, a List of Services is created. The reason for this is to facilitate the construction of WSDL drafts and the implementation of services. Table 2 presents the List of Services filled out. Column "1 - Task" refers to each task identified as a service. Column "2 - Task Dependent" shows dependent tasks. As can be seen, we show only one level of dependency to facilitate the understanding of the strategy of identifying service. However, the representation of the

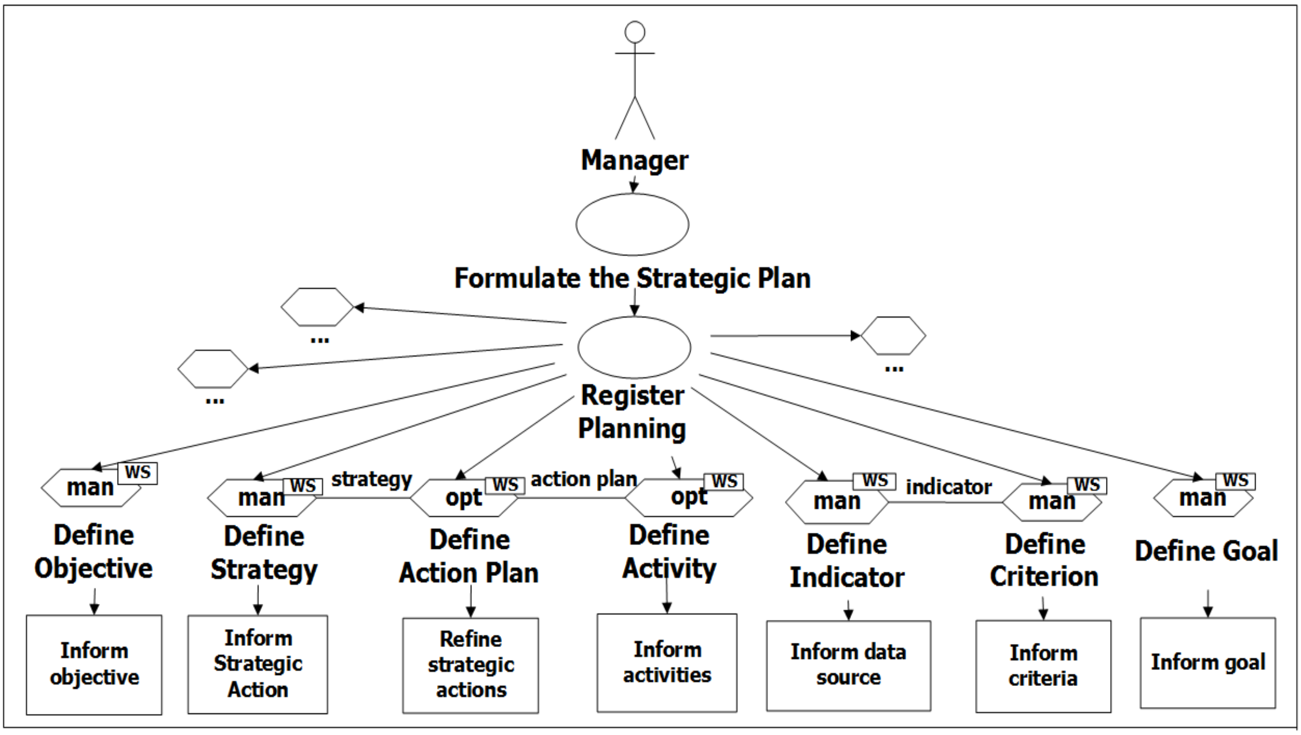

Fig 4. Domain GTR diagram

dependencies is not limited to only one level, as shown the section 3, in stage 5 of the strategy. Column "3 - Task Type" shows the type of each task of column 1. Column "4 - Service" refers to the name given to the identified service. Column "5 - Service Component" shows the formation of the composite service. Column "6 - Operation" and column "7 - Message" are populated with a brief description of the operations and messages to support the construction of WSDL drafts.

Table 2. List of Services

1 *Task*	*2* *Dependent Task*	*3* *Task Type*	*4* *Service*	*5* *Component Service*	*6* *Operation*	*7* *Message*
Define Objective		Mandatory	DefineObjective		defineObjective (objective)	objective
Define Strategy		Mandatory	DefineStrategy		defineStrategy (strategy)	strategy
Define ActionPlan	Define Strategy	Optional	DefineActionPlan	DefineStrategy	defineActionPlan(plan, strategy)	action Plan
Define Activity	Define Action Plan	Optional	DefineActivity	DefineActionPlan	defineActivity (activity, strategy)	activity
Define Indicator		Mandatory	DefineIndicator		defineIndicator (indicator, criteria)	indicator
Define Criterion	Define Indicator	Mandatory	DefineCriterion	DefineIndicator	defineCriterion (criteria)	criterion
Define Goal		Mandatory	DefineGoal		defineGoal (goal)	goal

Stage 6: Design Services

In this stage we designed the service identified in Stage 5. After the identification of services, the following columns of Table 2 are filled out: "6 - Operation" and "7 - Message". Analyzing the Domain GTR diagram of Figure 4 and Table 2, we note that the "Define Strategy" task is a mandatory task. It is present in all instances of the planning process; hence it is considered a service.

According to the instances recovered in Stage 1, the function of a strategy is to define and describe which actions need to be taken to achieve objectives in a certain time period. Based on the comprehension of the purpose and function of the “Define Strategy” task, we identified the operation "defineStrategy". Thus, we can start a WSDL draft and the implementation of the "DefineStrategy" service, which corresponds to this task. A drafting of the WSDL file of “DefineStrategy” service is shown in Figure 5. In this draft the message "strategy" was extracted from the dependency relationship shown in the Domain GTR of Figure 4.

```
<message name="strategy">
<part name="strategy"
type="xs:string"/>
</message>
<portType name="defineStrategy">
<operation name="defineStrategy">
<input message="strategy"/>
</operation>
</portType>
```

Fig. 5. WSDL draft of the DefineStrategy service

5. RELATED WORKS

Service identification is a critical phase in Service Engineering. A variety of SOA design frameworks and methodologies have been introduced by researchers ([3, 4, 5, 6] etc.). Cai et al [7], presents a survey on 153 existing service identification methods, in which the inputs and outputs are compared for different methods.

SOA analysis and design follows three approaches: top-down, bottom-up, and meet-in-the-middle. The top-down approach starts from business requirements. The bottom-up approach starts from analyzing the existing software system. Meet-in-the-middle is the combination of top-down and bottom-up.

According to Cai et al [7] many methods take advantage of Model Driven Approaches and Decomposition for top-down identification, while Reusable Assets Identification is the most widely used activity for bottom-up identification.

Cai et al [7] asserts that decomposition is almost used in every identification method. The advantages of decomposition is obtained granularity of logic units and mapping these into candidate services. The major difficulty of decomposition is to decide when it is pertinent to stop decomposing and when it is pertinent to decompose further. The starting point of decomposition is business requirements or high-level descriptions of requirements, such as various business models, flow charts or structure graphs. The objects to be decomposed can be functions, goals, business processes, business activities, domains and features.

Comparing the proposed strategy with the methods presented in the survey (Cai et al [7]), the strategy presented in this paper follow the top-down approach and can classified as decomposition method.

The advantages of the strategy are: i) the identification of services is totally based on the tasks of GTR Diagrams, this is a novelty and a originality of the work when comparing to the others. As many systems are designed with this type of diagram, we argue that identify services with the proposed strategy is quite straightforward; ii) The reusability of the identified services are suitable. This was verified because the identified services were employed in a real case study; iii) the guidelines provided here were not found in any work. The detail level of the guidelines shown is very good and they provide suitable support for developers identify services.

6. CONCLUSIONS

In this paper we present a strategy for identifying generic services in specific domains. For this, we analyze business goals. This strategy represents a systematic method for understanding business processes with the purpose of extracting key tasks from this process and transforming them into services.

In order to demonstrate the application of the strategy we use the planning process, which was implemented in a web portal called Web-PIDE. We have used information about various instances of the planning process and also we count on the support of two schools.

The identified services enabled the adaptation of the portal to the business rules of every school. This was because it was possible to build different functionalities using the composition of services, once services could be combined in different ways.

When a specific task of the schools was not represented in the Domain GTR, this task was implemented separately, not as a service, because its potential for reuse is smaller. Thus, implementation of the portal obtained flexibility and agility. The web portal was used successfully by two different schools.

The proposed strategy is generic and can be applied to other business processes providing software suitability to organization dynamics, as well as to different instances of the same business process.

7. REFERENCES

[1] H. Estrada, I. Morales, A. Martinez, O. Pastor, Transforming Service-Oriented Business Models into Web Service Specifications. In: International Conference on Software Engineering & Knowledge Engineering. July 2010. pp. 225--230.

[2] E. Ramollari, D. Dranidis, A. Simons, A Survey of Service Oriented Development Methodologies. In Proceedings the 2nd European Young Researchers - Workshop on Service Oriented Computing. Leicester, 2007, pp 75--80.

[3] L. G. Azevedo, F. Santoro, F. Baião, J. Souza, K. Revoredo, V. Pereira and I. Herlain, A Method for Service Identification from Business Process Models in a SOA Approach. In 10th International Workshop on Business Process Modeling, Development, and Support. Springer. North-Holland, Amsterdam. 2009. pp. 99-112.

[4] S. Kim, M. Kim, S. Park, Service Identification Using Goal and Scenario in Service Oriented Architecture, IEEE Computer Society, 2008, pp. 419-426.

[5] N. Fareghzadeh, Service Identification Approach to SOA Development. In: Proceedings of World Academy of Science, Engineering and Technology, vol. 35, 2008, pp. 258–266.

[6] K. Klose, R. Knackstedt, D. Beverungen, Identification of Services - A Stakeholder-based Approach to SOA Development and its Application in the Area of Production Planning. In: ECIS 2007, 2007, pp. 1802–1814.

[7] Cai, S., Liu, Y., Wang, X.: A Survey of Service Identification Strategies. In IEEE Asia -Pacific Services Computing Conference. IEEE Computer Society, 2011, pp. 464--470.

[8] C. Ling, L. Xin, Achieving Business Agility by Integrating SOA and BPM Technology. In: International Forum on Information Technology and Applications – IFITA, IEEE Computer Society, May 2009, pp. 334-337.

[9] A. I. Antón, C. Potts, The use of goals to surface requirements for evolving systems. In 20th International Conference on Software Engineering. vol. 2, IEEE Computer Society, 1998, pp.157--166.

[10] S. L. Tan, L. Liu, Performance Analysis of Reusable Components with Hybrid Modelling of Strategies and Processes: A Real World Case Study. In IEEE 36th Annual Computer Software and Applications Conference Workshops (COMPSACW). 2012. pp 302-309.

[11] D. Bolchini, P. Paolini, Goal-driven requirements analysis for hypermedia-intensive Web applications. In 12th IEEE international Conference on Requirements, IEEE Computer Society, 2004, pp.85--103.

[12] V. Perrone, D. Bolchini, Designing Communication Intensive web Applications: Experience and Lessons from a Real Case. In: Proceedings of WER 2004. Tandil, Argentina.

[13] E. Yu, Modeling Organizations for Information Systems Requirements Engineering. In 11th International symposium on requirements engineering, IEEE Computer Society, 2003, pp. 34--41.

[14] E. Yu, P. Giorgini, N. Maiden, J. Mylopoulos, Social Modeling for Requirements Engineering, 2011, Cambridge: MIT Press.

[15] M. Hammer, J. Champy, Reengenharia: repensando a empresa em função dos clientes, da concorrência e das grandes mudanças da gerência. 1 ed. 1994. Rio de Janeiro: Campus.

[16] BPMN. Specification of Business Process Modeling Notation.http://www.omg.org/bpmn/Documents/BPMN_1-1_Specification.pdf

[17] Notation extended of AWARE model. http://lapes.dc.ufscar.br/aware.html

[18] F. Goethals, M. De Backer, W. Kemahieu, M. Snoeck, J. Vandenbulcke, Identifying Dependencies in Business Processes. Proceedings of the LAP-CCBP (Communication and Coordination in Business Processes). Research Center for Management Informatics - LIRIS, 2005, pp.17-31.

[19] L. Wen, J. Wang, J. Sun, Detecting Implicit Dependencies Between Tasks from Event Logs. Frontiers of WWW Research and Development - APWeb. Springer Berlin / Heidelberg, 2006, pp. 591-603.

[20] K. Kang, S. Cohen, J. Hess, W. Nowak, S. Peterson, Feature-Oriented Domain Analysis (FODA) Feasibility Study (Report). Software Engineering Institute, Carnegie Mellon University, 1990, CMU/SEI-90-TR-21.

[21] G. A. Steiner, What Every Manager Must Know, 1979, New York: Free Press.

[22] A. A. Fischmann, M. I. R. Almeida, Planejamento estratégico na prática, 1991, São Paulo: Atlas.

[23] M. C. Tavares, Planejamento Estratégico: A Diferença entre Sucesso e Fracasso Empresarial, 1991, São Paulo: Harbra.

[24] D. P. R. Oliveira, Estratégia empresarial & vantagem competitiva: como estabelecer, implementar e avaliar, 2001, São Paulo: Atlas.