Adaptation Services-oriented Systems LifeCycle

I. Elmagrouni[1], A. Kenzi[2], M. Lethrech[1] and A. Kriouile[1]
[1]*SIME Laboratory, ENSIAS, Mohammed V University, Rabat, Morocco*
[2]*Sidi Mohamed Ben Abdellah University, Fes, Morocco*

Keywords: Development Process, SOC, CAC, SOA / Web Services, MDA.

Abstract: This work presents the approach of the Development of adaptable Services-oriented Systems. Not the only adaptability is important for the survival and success of Services-oriented Systems, but it is also important for the rapid changes in technology, organizational structure, human perception and needs. This paper discusses the shortcomings of current solutions for adaptive service-oriented System. To address those shortcomings, some techniques are used to build and evolve proactive Services-oriented Systems. Using those techniques in an integrated way is described along the phases of the service lifecycle.

1 INTRODUCTION

The service-oriented computing (SOC) uses services as basic constructs to support the rapid development of low cost, and easy composition of distributed applications even in heterogeneouscomputing environments.SOCperspectiveis to join services in a network of loosely coupled services, create flexible business processes and agile applications that can span different organizations and computing platforms.Besides, CAC (Context-Aware Computing) has been proposed to adapt applications and software systems to different useful contexts. These cover all the information that characterizes the situation of an entity. An entity can be a person, place, or object that may be relevant to the interaction between the user and the application. One of the challenges oflifecycleservicesis the stage that identifies the services that support the business activities of the organization.There are many objectives of this paper. Firstly,is the definition of a development process to guide the development of adaptable SOS from business requirements. Secondly, is the definition of phases, activities, and artifacts that allow the identification, specification and implementation of adaptive services.The life-cycle models; for Service-Based applications that have been presented in the literature (examples include SLDC, RUP for SOA, SOMA, and SOAD) are mainly focused on the phases that precede the release of software; and even in the cases in which they focus on the operation phases, they usually do not consider the possibility for to adapt dynamically to new situations, contexts, requirement needs, service faults, etc.Specifically, the following aspects have not been yet considered in those life-cycle models: Requirements elicitation, design for adaptation.

The first section briefly presents the work and the approaches that are related to the study. The second section describes the case study to illustrate the approach. The Third section focuses on the process definition which will enable the development of adaptive services. Last but not least, the article will be summed up with a conclusion and outlook.

2 RELATED WORK

(S.Lane et al., 2011).conducted identified adaptation activities that could be used to adapt Service oriented system (SOS). These activities combined with a skeleton life-cycle model; proposed by the S-Cube consortium, formed the basis of reference process model' frame for adapting Service-Based applications (SBA). This frame of reference was used to guide interviews with development practitioners who had experience and could provide expert' opinion Service-Based applications' adaptation. Thecollected data of these interviews was transcribed and analyzed by using qualified content analysis techniques. The result of adaptation activities and tasks were constructed into a detailed

Elmagrouni, I., Kenzi, A., Lethrech, M. and Kriouile, A.
Adaptation Services-oriented Systems LifeCycle.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 223-231
ISBN: 978-989-758-187-8

process model identifying the relevant stakeholders and development artifacts for each stage of the process. The model's transfer and ability were demonstrated during an evaluation process where the model was systematically compared to a component-based application adaptation model and an empirically based SBA development life-cycle. The approach's advantages are over similar approaches because it focused and based on input provided by experts from the field.

A method proposed by Azevedo et al. [14] with activities to guide the designer to identify the most suitable set of services to support the business activities of the organization. The method consists of the following steps: (1) Selection of activities subject to automation - this stage selects process activities TO-BE where can be identified candidate services, (2) Process models are represented using the Event-driven Process Chains (EPC) and Function Allocation Diagram (FAD), (3) Candidate services identification and classification - activities identified in the previous step are analyzed within their contexts in process models according to a set of heuristics, and (4) Consolidation of candidate services supported by the use of heuristics.

A guideline is proposed by Shirazi et al. [15] for the service identification using two approaches: top-down and bottom-up. The bottom-up approach is used to identify applications and entities services; while top-down approach's goal is to recognize the business services and services oriented tasks. The method consists of the following steps: identifying business processes, making business use-case model, identifying entity-centric services, recognizing application services, identifying task-centric services and recognizing process centric services. Marks and Bell proposed a Service Lifecycle; this cycle includes the service evolution from conception to maturity along its execution. The identification of business services is performed using a top-down and bottom-up approach in iterative cycles. In order to identify new candidate business services, the author proposes an analysis of the following sources: business process, entities (interest and principal), budgeted projects, business experience, preexisting services and existing business applications. Arsanjani et al. [8] presented a method for service-oriented solutions developing called Service-Oriented Modeling and Architecture (SOMA). Specifically for services identification phase, the paper points out that a good practice uses a set of complementary techniques to identify services and cites three service identification techniques: (1) Goal-Service Modelling (GSM), treats the services aligned to the business goals, (2) Domain decomposition is performed through a top-down analysis of business domains and business processes modelling that are identified services, components and flows. The aim is to consider the static and dynamic view of the business including information rules and variations.

(3) Analysis of the existing asset is performed by bottom-upanalysis of the existing application portfolio and other assets and patterns that can be used to identify candidate services. After the application of the described techniques, the method also comprises the following step: refactoring and rationalization of the service whose service granularity is determined. Finally, a series of criteria are applied to determine which service is appropriate for candidates' publication.

3 E-TOURIST MOTIVATING SCENARIOS

Let us consider the following scenarios. Assume that a user would like to use a mobile device (e.g., PDA, iPhone, Smartphone, BlackBerry, etc.) with thedifferent operating system (IOS, Android, etc.) which is equipped with a GPS (Global Positioning System) to find relevant restaurants with/without open gardens. Figure 1 shows a model of application mobile "Restaurant Finder". Customers want to browse available restaurants to view offered food items and their cost. To this aim they must search a set of potential restaurants and select one among them. This application also aims to collect the customers' feedbacks regarding selected restaurants. Finally, the average satisfaction of all customers must be high. It can utilize various other services, such as the Reverse Geocoding for mapping GPS information to addresses, the Google Map for finding businesses close to an address, the Restaurant Data Service for searching restaurants based on the user preferences, and the Weather Information Service for obtaining weather information. The service also offers graphical information to the user with a reasonable delay (Wifi, 3G, and text/image).

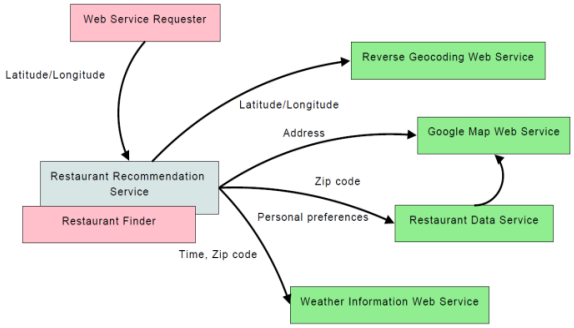

Figure 1: The Restaurant Finder.

4 ADAPTABLE SOS DEVELOPMENT PROCESS

The proposed method described in Figure 2 is based on MDA (Model driven architecture). It is composed of three abstraction layers: the CIM (Computation Independent model) which describes the system's requirements, the PIM (Platform independent Model) which specifies the system independently of any platform and the PSM (platform specific model) which contains the service models related to a specific platform.

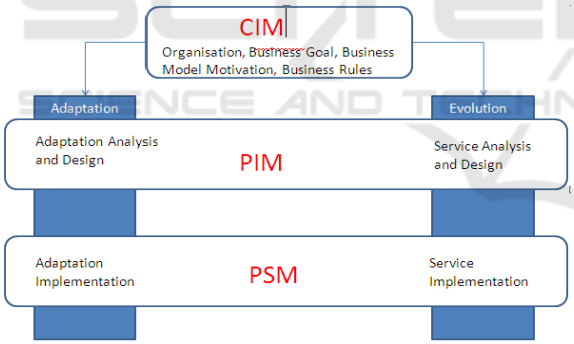

Figure 2: Development Process of adaptable Service Oriented System.

The first phase "Global analysis" is interested in the feasibility study.

Furthermore, the method is composed of two sub-processes:

An Evolution Process composed of the following activities: analysis and design service identify business requirement and implementation service.

An Adaptability Process which the main activities is: analysis and design Adaptation concerns the adaptation where system processes should adapt to certain conditions and implementation Adaptation.

4.1 Preparation

This phase is the preliminary study of the complicated organization that identifies two points: (1) business motivation model (2), Legacy system analysis.

4.1.1 Business Motivation Model

We define business requirements for business processes as the overall set ofrequirements that relate to business processes as given by the Business Motivation Model (BMM) of OMG, such as vision, mission, goal, strategy, objective andtactic. More specifically, a vision describes the future state of the application, withoutregard to how it is to be achieved, and mission indicates the ongoing activity thatmakes the vision a reality. A goal indicates what must be satisfied on a continuingbasis to effectively attain the vision, and a strategy is a long-term activity designed toachieve a goal. An objective is a specific and measurable statement of intent whoseachievement supports a goal, and a tactic is a short-term action designed to achieve anobjective.

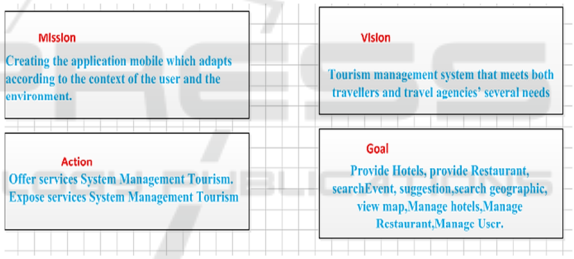

Figure 3: Business Model Motivation of Restaurant provider.

4.1.2 Legacy System Analysis

In this step, the decomposition of existing systems in the form of application modules is used to provide an implementation for business services that were previously identified. Then, we apply a bottom-up approach, i.e. starting from the existing system to the business services and business processes.

4.2 Services Analysis and Design

Business service identification is critical. Identifying appropriate services with the right level of granularity can have a major influence on the whole system. It is agreed that all business services should be identified to meet a business goal. Goals can be formulated at different levels of abstraction; ranging

from ahigh level and strategic to low level and operational.

4.2.1 Business Process Design with Goal Models

A goal model is used to capture why a business process is needed – its purpose or goal – and the different ways from which a goal can be attained. The goal model has been designed based on the design principle of Object-oriented Design, namely Decomposition and Abstraction. Decomposition principle is used to decompose a large problem into subproblems. Each sub problem is at the same level of detail, can be solved independently and can be combined to solve the original problem. Based on the above definition the goal model is designed to specify a high-level goal which is decomposed into subgoals and a hierarchical ordering of the subgoals is done. Goal tree structure is used to represent the model. In this, the high-level goal (problem) is decomposed into one or more subgoals (subproblems) and each subgoal is decomposed further into one or lower level subgoals. This goal becomes the root of the goal model. It is refined using AND/OR decompositions until the resultant subgoals can be delegated to either human actors or software services.

Figure 4 shows the goal of Restaurant Provider by tool OpenOME is an Eclipse-based tool designed to support goal-oriented, agent-oriented and aspect-orientedmodeling and analysis.

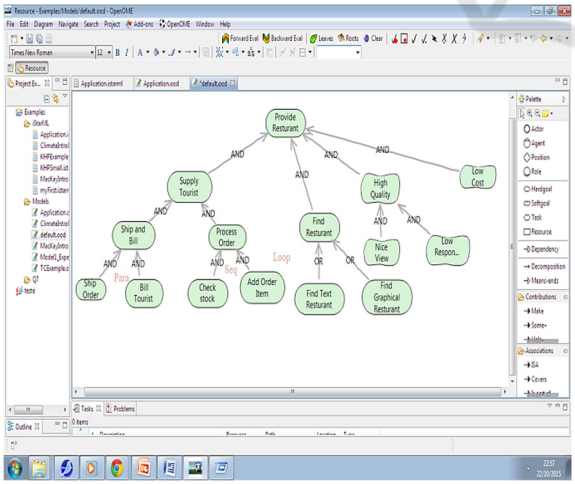

Figure 4: A goal model for the "Tourist System Management" business process.

The general goal is to provide Restaurants that AND-refined into the following sub-goals: Find Restaurant withhigh quality service and which provide both textual and graphical mode.

4.2.2 Enriching Goal Models for BP Modeling

Since we are interested in the automated execution of business processes, we needto capture more information about BPs than the basic goal models allow. A few annotationsare introduced for this purpose. Note that the annotations presented here arenot required to be formal. We use the following control flow annotations when employinggoal models to represent business processes:

• Parallel ("Para") and sequence ("Seq") annotations can be used with ANDdecomposedgoals to specify whether or not their subgoals are to be achievedin a temporal order. For example, billing customers and shipping goods are doneconcurrently in the process.

• Sel ("if(condition)") indicate the necessary conditions for achievingsubgoals. For example, in Fig. 2 the goal Order Out Of Stock Product isachieved only if the item is not already in stock.

•Loops ("while(condition)" or "for(setOfItems)"). For instance, the goal Add OrderItem must be achieved for all items in the order.

Table 1: Annotation for Business Processes.

Annotation	Meaning
Seq	Sequential execution of activities
Loop	Repeated execution of activities in a loop
Sel	Conditional selection of activities
Para	Parallel execution of activities with complete synchronization

4.2.3 Modeling of Input/Output

Modeling of input/output parameters of goals is also important for BP modeling. Identifying inputs and outputs during the analysis of a business domain help in determining resource requirements for achieving goals as well as for the sequencing of the goals. The types of inputs and outputs can also be specified. While optional, the input/output types can be used to generate detailed specifications for messages and service interfaces in a BP implementation.

Name: Collect Requests
Input/Output:r: ReceiveRequest; rc :RestaurantsCollection
DomainPrecondition:rc:state = default
DomainPostcondition:rc:state = req initialized
RequiredPrecondition:rc:keyword = "" ^ rc:date = null ^rc:distance = "" (r:keyword ≠ ""∨ r:date ≠ null∨ r:distance≠ "")

RequiredPostcondition:rc:keyword = r:keyword ^ rc:date = r:date ^ rc.distance=r:distance^Collect(rc:keyword;rc:date)
User: Tourist
Context: Device, Location, Time, Profile, Resturants Preference, Weather

Name: Find Graphical Content
Input/Output:rc :RestaurantsCollection
DomainPrecondition:rc:state = req_initialized
DomainPostcondition:rc:state = restaurants received
RequiredPrecondition:rc:keyword = "" ^ rc:date = null ^ rc:distance="" ^ (r:keyword≠ ""∨ r:date ≠ null∨r.distance≠ "")
RequiredPostcondition: ∀ *n* ∈ rc:restaurants: n:keyword = rc:keyword∨rc:date n:date∨ rc:distance= n:distance ^n:text≠ null ^ n:images ≠ null
User: Tourist
Context: Device, Location, Time, Profile, Resturants Preference, Weather

An operation is defined through name, input and output values and pre- and post-conditions. Required preconditions (ReqPre) define when the operationcan be executed.Required post-conditions (ReqPost) define additional conditions that mustbe true after execution. Domain pre- (DomPre) and post-conditions (DomPost) define the effects of the operation on the domain.The definition of operation Find Text Content is similar to operation Find Graphical Content except for the required post condition that is specified as follows:

ReqPost: ∀ *n* ∈ rc:restaurants: n:keyword = rc:keyword ∨ rc:date n:date∨ rc:distance= n:distance ^n:text ≠ null ^ n:images = null

4.3 Adaptation Analysis and Design

It is interested in capturingspecific adaptation requirement for SOS. To control service Adaptation, a designer needs to knowwhy a change was made, what are the implications, and whether the change is consistent or not. Eliminating spuriousresults and inconsistencies; that occur due to uncontrolled changes, is a necessary condition for services toevolve gracefully, ensure stability and handle variety on their behavior.

4.3.1 Typology of Adaptation

The nature of service Adaptations can be classifieddepending on their causal effects as follows:

- Minor Adaptations: these are small-scale incremental changes that are localized to a service or arerestricted to the clients of that service.
- Major Adaptation: these are large-scale transformational changes cascading beyond the clients of aservice, possibly to entire value chains.

Typical Minor Adaptation focuses on structural level changes (service types, messages, interfaces,andoperations) and business protocol changes (the conversations in which the service participates). TypicalMajor adaptation includespolicy-induced (pertaining to business agreements between service providers andconsumers.

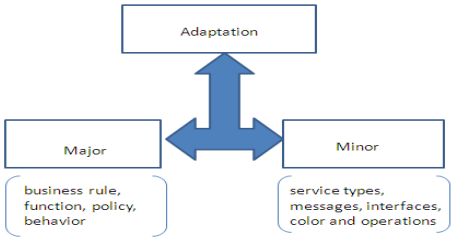

Figure 5: Minor and Major Adaptation.

4.3.2 Dealing with Adaptations

Service Adaptation requires an adaptation-oriented service lifecycle methodology to provide a sound foundation for spreading changes in an organized fashion that impacted services in a servicechain are appropriately configured, aligned and controlled as thechanges occur.

The purpose of the adaptation-oriented service life cycle is to ensure that standardized methods andprocedures are used for efficient and prompt handling of all servicechanges in order to minimize theimpact of change-related incidents upon service operation and quality. This means that in addition tofunctional (structural and behavioral) changes, adaptation-oriented service life cycle must deal with policy-induced operational behavior and non-functional changes. Figure 6 illustrates an adaptation-oriented service life cycle that comprises a set of inter-related phases, activities, and tasks that define the change process from the start through to completion. Each phaseproduces a deliveredmajor that contributes towards achieving change objectives. Logical breaks arealso provided in the change process and are associated with key decision points. The phases of the lifecycle are discussed in the next part.

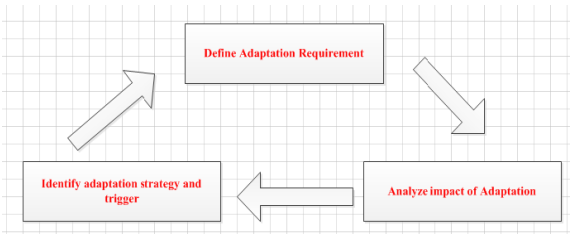

Figure 6: adaptation-oriented service life cycle.

The initial phase Figure 6 “Define Adaptation Requirement” focuses on identifying the need for adaptation and scoping its extent. One of the major elements of this phase is understanding the causes of the need for adaptation and their potential implications.

The secondin Figure 6 (“**Analyze impact of Adaptation**”) focuses on the actual analysis, re-designor improvement of existing services. The ultimate objective of service adaptation analysis is to provide an in-depth understanding of function, scope, reuse, and granularity of services that are identified for adaptation. The problem lies in determining the difference between existing and future service function.To analyze and assess the impact of changes, organizations rely on the existence of an “as-is” and a “tobe”service model rather than applying the changes directly to operational services. Analysts rely on an“as-is” service model tounderstandthe portfolio of available services. This model is used as thebasis forconducting a thorough re-engineering analysis of the current portfolio of available services that need toevolve. The “to-be” service model is used as thebasis for describing the target service functionandperformance levels after applying the requiredadaptation.To determine the differences between these two models, a gap analysis model is used to help prioritize, improve and measure the impact of service adaptation.

During the third and final phase **“Identify adaptation strategy and trigger”** in Figure 6, In order to select the adaptation strategy which should be applied, it is necessary to consider that adaptation maybe associated with a set of conditions and a trigger that are important for designing and performing adaptation. The trigger states when the adaptation Service must be activated. Each adaptation Service is operated through adaptation actions as explained in Table 2.

Table 2: Description of the two adaptation action.

Adaptation Action	Description
Substitution	The Possibility of configuration with a dynamic substitution of the service with another one
Performance	The possibility of going back in the process for performing an alternative path or redoing the same set of tasks

4.4 Service Implementation

It is based on the following step: Generating Business Processes

4.4.1 Generating Executive Business Processes

A method has been devised for using goal models to assist with the development and configuration of high- adaptability (flexible) BPEL processes. This makes it possible to generate BPEL processes that are easily readable by humans and are structured after the respective goal models. The BPEL code generation is semi-automatic. The generated code is not immediately executive so it needs to be completed. Nevertheless, provides valuable help in producing an executable BP based on the source goal model, the code is to be further developed by integration developers, who will also be selecting/designing Web services to be used by the process.

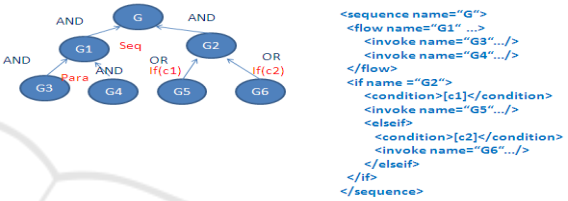

Figure 7: Example of WS-BPEL 2.0 code generation.

We start BPEL generation from the root goal and recursively through the goal tree until we reach leaf goals. Some of the goal models are presented to BPEL mappings through the example in Fig. 5, which shows a generic annotated goal model fragment. The root goal G has a sequential AND refinement, so it corresponds to the sequence operator in BPEL. G1 has a similar AND refinement, so it maps to the flow construct. G2 has a data-driven OR refinement, so it generates the if-elseif(BPEL 2.0) or the switch (BPEL 1.1) operator. Note that the conditions c1 and c2, which areinformal descriptions in the goal model, will be replaced with the appropriate conditions by a BPEL developer. While we abstract from some of the low-level BPEL details such as correlations, with the information captured in the annotated goal models, we also generate the following aspects of BPEL/WSDL specifications.

• We do an initial setup by defining the appropriate interface (portType), etc. for the process. A special portType for invoking the process and providing it with the (initial) configuration is also defined.

• A conditional/loop annotation for a goal G is mapped to the appropriate BPEL construct (e.g., if-elseif or switch, while, etc.) with the activity to be executed being the result of mapping the goal model subtree rooted at G into BPEL. The formal

conditions currently have to be specified manually.
• Leaf-level goals map into Web service invocations. The information in the goal model helps in defining the interface for the Web services invoked by the BP. We define appropriate WSDL messages based on input/output parameters of these goals. If data types are omitted from the goal model, they have to be supplied by a developer.
• Softgoals are used as the evaluation criteria in the configuration process and thus do not map into the resulting BPEL specification.

4.5 Adaptation Implementation

This activity involves in the implementation of the adaptation Mechanisms that were described in the phase analyze and design adaptation. Each conventional goal, which represents a functional requirement (i.e., it is operationalized),is mapped onto the corresponding sequence activity in the BPEL process. If the goal represents a non-functional requirement, but its nearest ancestor goal is operationalized, it is associated with the same sequence of its parent goal. This activity must represent an interaction of the process with its partner services (e.g., invoke, pick, and receive).Each adaptation goal is associated with a set of actions that must be performed at theprocess level.A triggering rule activates the evaluation of the trigger associated with the goal. A condition rule evaluates the conditions linked to the goal. If the two previous rules provide positive feedback, an activation rule is in charge of the actual execution of the adaptation actions. Performed when an adaptation goal can potentially fire (i.e., the corresponding Activation fact is available in the working memory) and is selected by the rule engine to be performed, among the other adaptation goals that can be performed as well. It executes the actions associated with that adaptation goal. For example, the triggering rule associated with ABG1 is the following:

```
when
Goal(id=="FGR", satisfaction < 1)
then
workmemory.insert(new
Trigger("TriggerABG1"));
```

5 ADAPTATION ASPECTS WEAVER

The adaptation Implementation follows a three-step process (see Figure 10):
1. Context detection consists of checking the runtime context information, in order to detect possible context changes. These tasks are performed by the Context Manager Service which is developed as a Web service in the BPEL process.
2. Aspect Activation is responsible for the plug-in and the removal of pre-defined Aspects into the BPEL process using the Aspect Activator Module.The Aspect Activator Module is conceived as an extension to the BPEL engine when running a process instance; the Aspect Activator receives the context change information from the Context Manager Service.
3. Updating original BPEL Process by activating the right Aspect which is executed in the BPEL process.

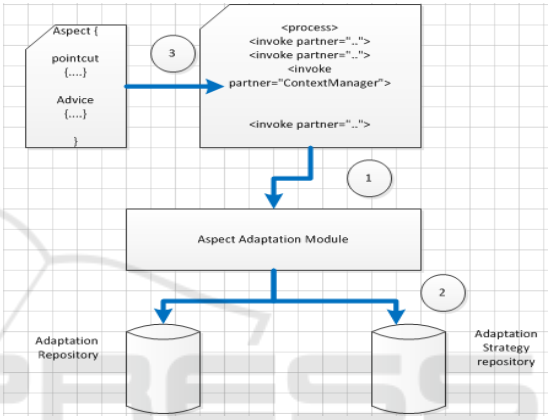

Figure 8: The adaptation process.

5.1 Adaptation Tools

We propose an evolved adaptationtool, based on the context manager and aspect, this tool allows selectingwhich services to invoke, and adapt them. These toolsare divided into three distinct layers, namely (see Figure 9): Application layer, Adaptation layer, and a Resource layer. These different layers are described as follows in the following.

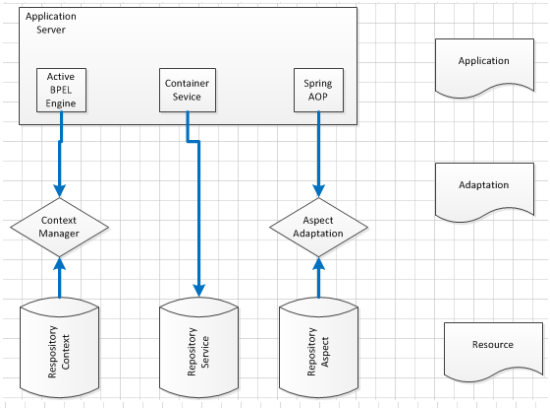

Figure 9: Technical architecture of adaptation.

Application Layer: It is the top layer of the technical architecture, including application platform which we implement our approach to adaptation. The central element of this layer application server. Typically, an application server is a server that Are installed applications used by customers.

Adaptation Layer: This is the second layer of this architecture. Placed between the application layer and resource, it contains components which will ensure the processing context information and any other operations required to carry out the adaptation of invoking services. Essentially, this layer contains two modules:

The context manager is charged to collect the information in context and to detect the possible changes of this information. The context manager will be called upon by process BPEL like any other web service.

The second module is the activation of aspect. Just like the context manager, the activation of aspect is implemented under the form of a Web service to interact better with the other elements of the architecture. This module is always placed after the context Manager in the process BPEL.

```
<?xml version="1.0" encoding="UTF-8"?>
<AspectRepository>

<aspect>
<aspect-identifier> Asp12</aspect-identifier>

<aspect-name> ExtraFees </aspect-name>
<condition> Delivering Place Casablanca</condition>

<joinpoint>ComputePrice(....) </joinpoint>jointpoint>
<advicetype> around </advicetype>

 <id-ws> 30 </id-ws>
</aspect>

</AspectRepository>
```

Figure 10: Aspect ExtraFes.

Resource Layer: It represents the third layer of this technical architecture. The resource layer includes all the resources needed for the other two layers. It interacts essentially with the matching layer at the two matching modules: the context manager and the aspect activation module. The resource layer includes the following:

- Repository Process: it lists all processes deployed to the BPEL engine.

-Repository Service: this case represents the functional service implemented as Web services.

-Repository Aspect: contains all aspects represented as an XML tree. Every aspect is a node in the tree characterized by an identifier and condition that represents a value of a context parameter.

-Repository Context: is supplied and updated by the context manager, the repository stores the context parameters related to the client and to the opportunity of cooperation.

6 CONCLUSIONS

In the previous sections, an approach is proposed to develop service-oriented systems that can be adapted to different contexts that deal with the effects of both major and minor. An adaptable-oriented service life cycle methodology is used to address and describe its phases. In particular, it discussed when a change in a service is triggered, how to analyze its impact and what are the possible implications of the implementation of the change for the service provider and consumers. A formal model for minor and majoradaptation,on the basis of the one, is the main goal of our future work.

REFERENCES

M.lethrech, I.Elmagrouni, A.Kenzi, M.Nassar and A.Kriouile"DSL and SOA, Exploratory Study" *JDTIC, 2012.*

H. Hafiddi, H. Baidouri, M. Nassar and A. Kriouile"A Model Driven Approach for Context-Aware Services Development", in the *2nd International Conference on Multimedia Computing and Systems (ICMCSmodel'11),* Ouarzazate, Morocco, April 2011.

S.Lane,I.Richardson, A Process reference model developing adaptable service-based applications: *information and software Technology (2011).*

M. Shahrbanoo, M. Ali, and M. Mehran, "An Approach for Agile SOA Development Using Agile Principals," *arXiv preprint arXiv:1204.0368, 2012.*

S.consortium,S-Cube knowledge model, 2011, http://www.s-cube-network.eu/km>.URL<http:/ /http:/ /www.s-cube-network.eu/km>

K. Boukadi, L. Vincent, P. Burlat, Modeling adaptable business service for enterprise collaboration, *Springer, Thessaloniki, Greece, 2009*, pp. 51–60.

A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy, K.Holley, SOMA: a method for developing service-oriented solutions, *IBM Systems Journal, 47 (2008) 377–396.*

M.P. Papazoglou,W.V.D. Heuvel, Service-oriented design and development methodology, *International Journal of Web Engineering and Technology, 2 (2006) 412–442.*

A. Kenzi, B. El Asri, M. Nassar, A. Kriouile, A model driven framework for multiview service oriented system development, *IEEE Computer Society, 2009, pp. 404–411.*

S.K. Johnson, A.W. Brown, A model-driven development approach to creating service-oriented solutions, *Springer, 2006, pp. 624–636.*

C. Canal, J.M. Murillo Software adaptation 12 (1) (2006).

I. Christou, S. Ponis, E. Palaiologou, Experiences of Using the Agile Unified Process in the Banking Sector, Software, *IEEE PP (2009) 1–1*.

A. Kenzi, B. El Asri, M. Nassar, A. Kriouile, A mdel driven framework for multiview service oriented system development, in: *IEEE Computer Society, 2009.*

C.Pahl, Semantic model-driven architecting of service-based software systems, *Information and Software Technology, 49(2007) 838-850.*

S. Mittal, Devs unified process for integrated development and testing of soa, University of Arizona 2007.

C. SooHo, A systematic analysis and design approach to. develop adaptable services in service oriented computing, in: *Congress on Services (Services 2007), pp. 375–378.*

Her, J., La, H., Kim, S.: A Formal Approach to Devising a Practical Method for ModelingReusable Services. In: Proc. *of 2008 IEEE Int'l. Conf. on e-Business Engineering.*

(ICEBE 2008), pp. 221–228 (2008).

S. Consortium, State of the art report on software engineering design knowledge and survey of HCI and contextual knowledge, Tech. Rep. PO-JRA- 1.1.1, July 2008.

Erl, T.: Service-Oriented Architecture: Concepts. Prentice-Hall, Englewood Cliffs (2005).

Cappiello C, Pernici B (2009) Design of repairable processes. In: *Cardoso J, van der Aalst W (eds) Handbook of Research on Business Process, Information Science Publishing.*

Bucchiarone A, Cappiello C, Di Nitto E, Kazhamiakin R, Mazza V, Pistore M (2009) Design for Adaptation of Service-Based Applications: Main Issues and Requirements. In: *Proc. of Fifth International Workshop on Engineering Service-Oriented Applications: Supporting Software Service Development Lifecycles (WESOA).*