

Semantic Web Services Adaptation and Composition Method

Hamid Mcheick

Department of Computer Science and Mathematics
 University of Quebec at Chicoutimi (UQAC)
 Chicoutimi, Canada
 hamid_mcheick@uqac.ca

Amel Hannech

Department of Computer Science and Mathematics
 University of Quebec at Chicoutimi (UQAC)
 Chicoutimi, Canada
 amel.hannech1@uqac.ca

Abstract—Web services adoption is a major advance in the development of interoperable information systems. In particular, the composition of services can meet the needs increasingly complex of user, by a combination of web services within a single business process. However, despite this widespread adoption of Web services, many obstacles prevent their reconciliation in the composition, or may occur within a BPEL process in a state change, the context for example. ASWSCC Method (Adaptation of Semantic Web Service Composition to Context) is an implementation of a theoretical model made in our an earlier work. It focuses on composition process adaptation to use context (preferences, user type and its environment as the device used, location, access mode and many others). This context and request service matching should be taken into account while composing new services. Our goal is to develop a model which ensures, on the one hand, web services matching during composition process by using domain ontology as lexical database WordNet, its purpose is to identify, classify and relate in different ways semantic content and lexical language. On the other hand, this model allows management and taking into account the context that makes composition process adaptable to different instances of use context, which may change during the same session. For this reason, we are interested to capture and manage the context and its impact on basic services and composition process at once. Changes can affect the context of web services during their executions and the need to adapt their dynamically becomes increasingly crucial. From here comes the need for a coherent solution to adapt web services context. We exploit the benefits of aspect weaving tool in this approach to inject aspects of web services to adapt them to change of context.

Keywords-Context definition and management; adaptation; web services composition.

I. INTRODUCTION

Internet evolution and competition between firms were factors in web services explosion services. Indeed, web services may constitute speed and efficiency contribution for e-business. This notion of web service essentially means an application made available on Internet by a service provider and accessed by clients through standard internet protocols [1]. Their characteristics compared to other distributed computing technologies lie in the fact that they offer component model in weak coupling using Internet

technology as infrastructure for communication. If application designer target is not achieved by invoking a simple web service elementary, designer must combine the functionality of a set of services.

This process is called web service composition. It specifies which services need to be invoked in what order and how to manage interactions between them, and exception conditions. Service-oriented architecture (SOA) enables integration of applications and resources flexibly, representing every application or resource as a service. In particular, SOA suffers from a number of limitations and weaknesses in the context of composition on demand; hence web services adaptation to context remains essential to better exploit services. We introduced a basic idea of our service composition model in [5]. This paper describes the steps, a case study and comparison with exiting methods details of our service semantic composition method. The case study is a proof-of-concept to illustrate our service composition method (ASWSCC).

This article is organized as follows. Section 2 describes related works. Section 3 shows our process of service composition based on context, in which we describe the process of service composition. The validation of this model, through a case study and a comparison, are given in Section 4. Aspect-Oriented computing is illustrated in Section 5. Section 6 compares service compositions approaches. A conclusion and future work are discussed in Section 7.

II. RELATED WORK

Web services adaptation to contextual changes can occur at several levels. To adapt a process composition with Business Process Execution Language (BPEL), several researchers act on composition and orchestration of services. In their approaches, Keidl et al. [2] present a context framework that facilitates context-aware Web services development and deployment. In their framework, context information is exchanged in the header of Simple Object Access Protocol (SOAP) messages. In their proposal, authors use pre-defined Web services with contextual information clients. This information will be used later to provide a service to custom behavior.

Processing context is provided by web services, context plug-ins, or context services. Messages exchanged before and after this operation are based on contextual information

and are essential for processing and automatic adaptation to changes in web services context without need to adapt manually web services. Once context information processing is made, it is to component Invocation Manager to invoke the web service properly. It has already been pre-set with a context similar to that returned by the Context Manager. In their approach, Keidl et al. includes a new component called Context Manager responsible for handling context information customer.

The assignment of this operation to a full-fledged component seems to us very useful, because it allows distributing adaptation operation load, and on the other hand makes the approach more flexible especially when web services are defined as activities of BPEL process. In other words and in our case we can place this component in the form of activities in the BPEL process concerned.

Other work that is akin to our research problem is that of Chaari et al. [3] who defined three techniques of adaptation to web services context: Web service internal adaptation, Web service external adaptation, and Web services polymorphic adaptation. We can say from these three adaptation techniques that the first is a web service manual adaptation; the second is a dynamic adaptation, while the third is an adaptation in services composition. Among the works that have addressed such as adaptation, we cite the work of Vukovic and Robinson [4] who presented a system architecture for building context-aware applications based on the notion of Web services dynamic composition. In their approach, any contextual change can lead to new reconstruction during the execution of services, resulting thus dynamic evolution of the application. They developed a Framework which uses planning shop2 [13], BPEL4WS [4][21] and BPWS4J [22] for composition, design and execution of the composite service, respectively. Otherwise, the context changes will be doing manually from a client interface. Using context values as input, their service composition engine generates the query composition services suitable to adapt their application to these changes in context.

We learned from this approach using a client interface for entering information on the context. Thus, we can be inspired to capture the context information for client and allow him to change its environment even during the execution of the composition process.

Recent work has focused on semantic description web services and ontologies are mainly used to model the semantic service representation. It helps to establish semantic relations between concepts of the domain under consideration. We also have to mention that the OWL-S [20] approach that uses the ontology OWL-S to extend UDDI with semantic description of Web services.

III. PROCESS OF ADAPTATION OF SEMANTIC WEB SERVICES COMPOSITION TO CONTEXT

In a previous work, we have built a prototype system for Semantic Web services composition, we have shown through this system for use benefit of behavioral descriptions of Web services (class, function, operations, input, output, context, etc.), the description of use context and their contributions to semantic Web services composition, as well as the interest of

taking into account the alignment of ontologies and similarity measures between web services candidates concepts during composition process [5]. In this paper, we present our case study and experiment validates the proposed prototype.

We start first by giving definition and context modeling approach that will be taken into account during web services selection process and composition due to a user request.

A. Contexte Definition

Many works from computer field sensitive to context try to give a definition of context to establish a basis for the adaptation process, but they have not yet resulted in a definition that is both generic and pragmatic context, and more precisely parameters constituting context. The most common definition in the literature and accepted by most researchers is that proposed by Dey et al. [6]. The latter defines context as «any information that characterizes the situation of an entity. An entity is a person, place or object considered relevant with respect to the interaction between a user and an application, including the user and application themselves». The context describes user situation in terms of user profile, environment, terminal used, location, time, etc. In addition, contextual data are completely independent of applications and are not retrieved from a storage medium connected to the field of application. From a practical point of view, a context can be given as it cannot be supplied by the user when invoking web services. More concretely, we define context as the set of external parameters that can affect the behavior of web services involved in the process of basic composition and composition process itself by defining new web service candidates constituent other dial plans. These parameters have dynamic appearance allowing them to change during the runtime. For example, during an operation «buy online book» by a student, a contextual situation can be defined with the following parameters: (access type = «Student», type = device «Smartphone», location = «UQAC»). A new value of these parameters presents a new contextual situation that can change all selected web services, and thus, change the execution plan process composition of these services.

To define a contextual situation more precisely, and based on a previously defined pattern in a patient records management application [23]; we define a three-dimensional space where each dimension represents an axis of context that covers: user type «access type», device type, and location. A change in the value of these parameters defines a new contextual situation in which the composition process must adapt. To better explain our definition, a space $E(x, y, z)$ is explained in Figure 1. The state E1 represents a contextual situation in which the user is a student located in the library of the university using his mobile phone to buy a book for his class PHP development. The state E2 represents a contextual situation in which the user is a student in a building still inside the university but outside the library using a desktop PC to buy a book for his course PHP. The state E3 represents a contextual situation where the user is a student not located within the university using a desktop PC trying to buy a book for a class for PHP development.

Each of these contextual parameters causes a change in the search result. Despite using the same query "buy book PHP development" the answer is not the same for these three users, and the difference in contextual situation. Some services are only available to students «access type = Student» and only using the university's computer network «location = UQAC Network». In addition the university offers online books that are only accessible for students.

The device modifies the display mode and interaction mode with user (cutting all the information into subgroups to the small terminal, a single window with the entire information on the desktop screen, a graphical display of multidimensional arrays on a PC, texts speech synthesis on a mobile phone and smartphones). This implies that services that correspond to the parameters of location and type of access may not be compatible with mobile phones for example, and then they will remain unavailable «hidden» to user.

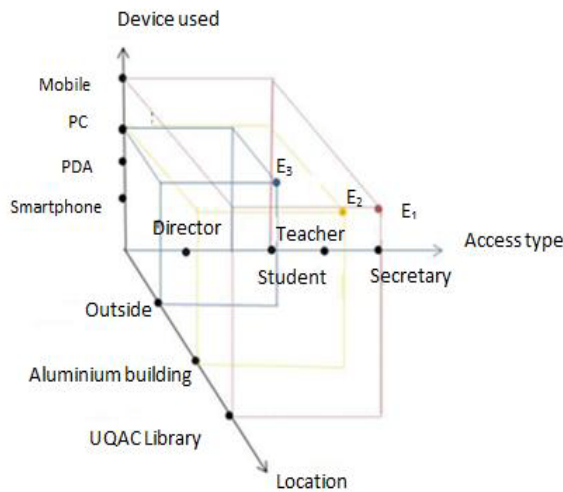


Figure 1. Representation of Context Categories

The device modifies the display mode and interaction mode with user (cutting all the information into subgroups to the small terminal, a single screen with the entire information on the desktop screen, a graphical display of multidimensional arrays on a PC, texts speech synthesis on a mobile phone and smartphones). This implies that services that correspond to the parameters of location and type of access may not be compatible with mobile phones for example, and then they will remain unavailable «hidden» to user.

B. Management and Modeling Context

We store context using a set of pairs (attribute, value). For example, ContexteUtilisation (id_ContexteUtilisation = «context1», username = «Amel», attribute = «device type», value = «PDA»). (id_ContexteUtilisation = «context2», username = «Amel», attribute = «location», value = «UQAC»).

We define three aspects to model context (access type, device type, location). We also propose to store the context before its release in the selection process and web services

composition, to keep track of historical values captured. This allows us to have a rich and reliable representation of data captured in form of ontology that represents the context as a set of entities describing their aspects.

C. Composition Process

Composition process starts by specifying the set of tasks to be performed that allows expressing composition purpose. This activity is triggered by a user request, it determines and organizes tasks to be performed by web services, each of which can perform one or more tasks in a dialing plan. This is done by decomposing user request represents a complex task into simpler and functional tasks, based on a knowledge base, this step is explained in out precedent word [5].

Planning is an essential step. Indeed, it determines web services that will participate in the composition. In addition, it determines web services ordering present in composition. Each dial plan is based on all sub queries such that each one contains its own semantic web services returned after discovery stage, and are used with others to build a plan or plans of dynamic composition, from the first sub query to the last.

Composition plan starts from the first sub query to the last, as we apply semantic matching algorithm [19] between the Output parameter of all semantic web services candidates from the first sub query with input parameter of all web services from the second sub query, and so, with the second sub query to the last.

Web services discovery that meet each sub task is based on semantic matching, it can search the descriptions of web services that have a semantic correspondence between the functional parameters defined in these descriptions (category, function, output, input, etc.) and those introduced into sub tasks. This is based on global domain ontology «WordNet» by comparing terms and gives an approximation between concepts [7]. The approximation is a degree of semantic similarity [19] to determine which concepts equivalents to query words which are considered acceptable by the system to generate an appropriate response.

So, the semantic matching algorithm [19] is based on the similarity function; it considers the semantic link between the terms by using ontology as external resource. We exploit in our case WordNet [7]. There are several methods to calculate semantic relations in WordNet. We chose to use the mathematical measure used in our previous model [19]. This work came to offer a solution for the process of semantic web services discovery according to user needs formulated under a request established in several terms. This allows calculating the length between two nodes representing two terms T_1, T_2 in WordNet ontology.

Modifying context is very important in our process; it decreases false answers and improves the overall quality of results. Our method ASWSCC is based on a context ontology [20] built and enriched with contextual data captured from different users. It allows us to consider changes in context in close moments during a single user session, dialing plans corresponding to the current use context are offered to user, respect to other plans related to

different instances are stored in a cache register during a change of state plans will be offered to him.

We use BPEL4WS (Business Process Execution Language for Web Services) specification or simply BPEL to express coordination between web services in the various plans proposed. It is the most industrially supported and well accepted by developers. In addition, there are several development tools that can help us in our work. For all these reasons, BPEL is the language that we have chosen for the implementation of this project. The result of the previous step is a set of execution plans in the form of finite automaton (state-transitions), states represent the different web services participating in the implementation plan, and transitions denote the semantic relationship between them.

For this reason a transformation "model to code" is necessary to generate the BPEL process. The work of Dautriche and Melliti [17] allows automatic generalization of BPEL code from an automaton and files corresponding WSDL service descriptions belonging to the automaton that describes the execution plan.

For this they have a tool for modeling behavior and then calculate the process model based on automaton. From this automaton a BPEL code generation is made. This work proposes a BPEL process executable generated by a tool developed in Java language.

To solve the problem of web services adaptation to change of user and service context, these services are invoked in a BPEL process, we have proposed an approach based on aspect weaving. We have presented a model for adaptable service composition characterized by adding two components, namely the context manager and aspect manager [5]. These are implemented as a web service, to allow the opportunity to easily integrate them in the same BPEL process where are services involved in adaptation operation.

We also proposed a model of context and we used a comparison algorithm of context variables based on a similarity computation algorithm proposed in the literature [18].

IV. CASE STUDY AND COMPARISON

To compare different service composition approaches, we realize an implementation of a process for web services composition; these services belong to different business domains (domain of travel organization which includes travel companies, hotels, restaurants, car rentals, and services organizing activities, library domain for purchase and rental services of books, DVDs and movies). Descriptions are published in a directory to provide best use and ease of access. We cite the example of travel organization agency which typically provides web services for consultation, reservation, payment and cancellation of travel tickets, hotel rooms, rental cars, restaurant reservations and organizing activities (Table I).

TABLE I. EXAMPLE OF WEB SERVICE DOMAIN AND ITS INPUT AND OUTPUT

WS domain	Input	Output
-----------	-------	--------

WS-Ticket booking	City departure, city arrival, date of departure and return date.	Code, Type, Price, city arrival, city arrival, departure and return date.
WS-Hotel	City, Input date, Output date.	Hotel Name, Address Hotel Class hotel, room type, duration and prices Hotel
WS-Restoration	Number of persons	Restaurant name, address, restaurant, restaurant class, many people, price
WS- Car rental	Duration rental, rental location	ID tenant, tenant Location, type and vehicle for hire
WS-Activities	City, Number of persons	List activities, duration, Region, activity price
WSCalculate order	List price	Total price
WS-Bank	Total price, payment, method, customer details	Customer details, Total price

Therefore, to provide these web services to its users, the travel agency must establish links with other companies: companies (airlines, agency of bus and train), car rental companies, hotel networks and catering, activities services that offers activities available in a given city. A bank is also required to facilitate financial transactions between users and agency organizing travels, or between the agency and other partners.

A. Experimentation

The prototype was developed under the operating system Windows Vista with open source graphics tools developed in Java and J2EE technology. We opted for Apache Tomcat application server that acts as JSP container, which allows its connection with a web server to deliver dynamic content to clients. We chose lexical database WordNet 2.0 as the global ontology, to compare terms and give similarity measure between concepts. We used the free dictionary Atla [8], which allows to translate words in French and whose target language is English, using its text files included with the dictionary.

The directory publication of our web services has been implemented in a relational database for a more explicit treatment of data; the conceptual model formalism is under «entity association». The service is described by the following functional parameters (category, function, input, and output).

We enrich this description with the context parameter named ContexteService will be represented with a separate table with the following view: (ContexteServiceCode, serviceID, Attribute, Value).

The various attributes ContexteService belong to the space defined in Figure 1. We have a database of 1,200 web services. We illustrate the functionality of our system by doing some individual images that show us steps of our Semantic Web services dynamic composition based on semantic matching and use context. We have two access rights (administrator and client). The administrator is the web service provider that connects to publish Web services (Figure 2). The client is the user who connects to use our process, he seizes his search query that can be focused on areas offered by our process mentioned above, the search

query is a complex task generic, and it will be treated to compose the existing web services to meet the needs of the user.

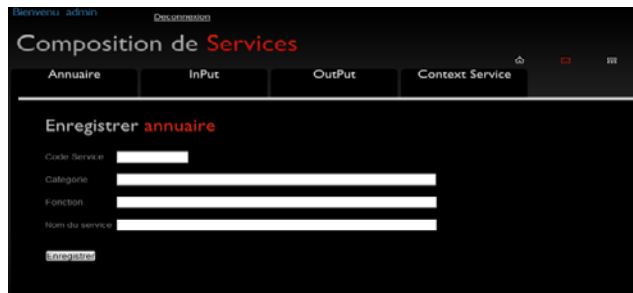


Figure 2. Administration Session

B. Dynamic User Interface

We opted for suggestion technique to help the user to edit his query; he starts typing his request, and a list box appears to propose him a list of tasks processed by our process loaded for a table « task».

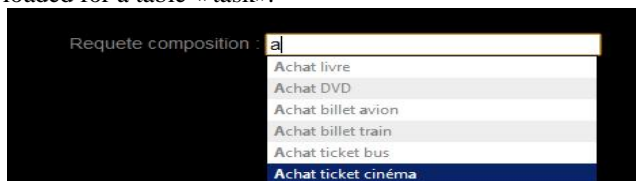


Figure 3. Menu suggestion

The user page also represents a source of contextual data collection, it gives the possibility to enter the type of device, its access mode, and through the IP address we can locate the network where the user is logged and can therefore face to location parameter.

This interface is 100% dynamic, information that appears in both fields of choice (device type, user type) are loaded from Context Utilization database, all instances of these two classes are extracted and offered to user.

This database is scalable and enriched by a domain expert to provide a broad definition of context. Use context is also about user preferences. According to his request, additional information must be completed.

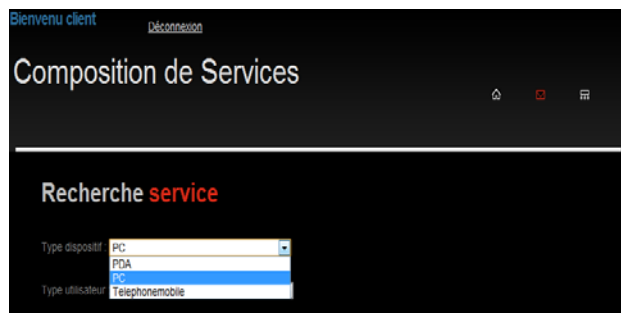


Figure 4. Collection of Context

Our user interface makes visible and invisible some input fields according to the request made. Example, for a query

Q1 = « travel organization» four boxes "hotel, restaurant, car rental, activities" appear after query formulation. For a query Q2 = « buy book» a checkbox "service pack" appears to propose to user if he wants or not packing the purchased item. This is done to make our dynamic interface based on a database tasks interconnected through a semantic link.

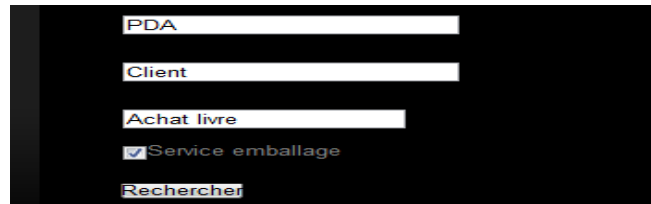


Figure 5. Dynamic Interface

Our composition system begins to execute by decomposing user query in simple tasks using a knowledge base representing a composition pattern and discover web services by basing on their functional parameters, and use context, which reduces false answers. This process constructs and proposes then plans in order of composition sub queries in the pattern by matching input-output of services. This matching issue is based on domain ontology. At the end, we show to user the final result in a set of plans, each plan is a set of services. For context C1 = (PC, Client, IP @) the result of the query Q1 composition has 6 plans; see Figure 6.

Context change on one of C1's parameters implies a change in the proposed plans. For example, if the contextual situation changed C1 to C2 = (PDA, visitor, @ IP), the result is shown in Figure 7.

Some services are only available to customers and do not display the result on a type device «PDA». Through this example, we have seen that the steps proposed in our prototype composition gave us satisfactory results.

V. ASPECT WEAVING TOOL USING THE TEMPLATE

Changes can affect the context of web services during their executions and the need to adapt their dynamically becomes increasingly crucial.

Liste des services composés		
NomService	NomService	NomService
Le plan N° : 1		
agile azure	Chercher billet avion	Voyage
Service calculatrice	Calculer commande	Calculer
DesJardin	Payer commande	Paierement
Le plan N° : 2		
agile azure	Chercher billet avion	Voyage
Service calculatrice	Calculer commande	Calculer
LCL	Payer commande	Paierement
Le plan N° : 3		
agile azure	Chercher billet avion	Voyage
Service calculatrice	Calculer commande	Calculer
Banque nationale de crédit	Payer commande	Paierement
Le plan N° : 4		
agile azure	Chercher billet avion	Voyage
Maple Calcule	Calculer commande	Calculer
DesJardin	Payer commande	Paierement
Le plan N° : 5		
agile azure	Chercher billet avion	Voyage
Maple Calcule	Calculer commande	Calculer
LCL	Payer commande	Paierement
Le plan N° : 6		
agile azure	Chercher billet avion	Voyage
Maple Calcule	Calculer commande	Calculer
Banque nationale de crédit	Payer commande	Paierement

Figure 6. Result of composition before context change

Liste des services composés		
NomService	NomService	NomService
Le plan N° : 1		
SNCF voyage	Chercher billet train	Voyage
Service calculatrice	Calculer commande	Calculer
DesJardin	Payer commande	Paierement
Le plan N° : 2		
SNCF voyage	Chercher billet train	Voyage
Service calculatrice	Calculer commande	Calculer
LCL	Payer commande	Paierement
Le plan N° : 3		
SNCF voyage	Chercher billet train	Voyage
Maple Calule	Calculer commande	Calculer
DesJardin	Payer commande	Paierement
Le plan N° : 4		
SNCF voyage	Chercher billet train	Voyage
Maple Calule	Calculer commande	Calculer
LCL	Payer commande	Paierement
Le plan N° : 5		
Keep Exploring	Chercher billet train	Voyage
Service calculatrice	Calculer commande	Calculer
DesJardin	Payer commande	Paierement

Figure 7. Result of composition after context change

To solve the problem of the adaptation of web services invoked in a BPEL process to change of the client context and web service context, we have proposed an approach based on the aspect weaving. Its originality is the addition of two components namely the context manager and appearance manager [5]. To illustrate our model adaptation composition of semantic web services based on the weaving of aspects we

are working on a concrete example of use. We have chosen the example of a process of buying books.

This process begins with the search and purchase books that result in a decision-command followed by the calculation of the amount of the order and finally the delivery of this order.

The BPEL processes represent complex service consists in this case of all three web services and the two components that will ensure adaptation where the context of the client or the web service change during process execution.

VI. COMPARISON WITH OTHER APPROACHES

We present now a comparative table between our approach and existing approaches explained above. This table is created according to some evaluation criteria to justify, evaluate, and positioning our proposal in comparison with other methods and techniques (see Table II).

TABLE II. COMPARISON OF SERVICE COMPOSITION APPROACHES

	Manual Method	Workflow based method	Planning based method	Our approach
Automation level	Null : No automated process composition	Tool: the search and link to services are only automatic	High : Practically, all the process is done automatically	High : Practically, all the process is done automatically
Dynamic layer	All is done statically	Process model is done statically. Link to concrete services is dynamic	The generation of composition plan is done dynamically	High : Interface user 100% dynamic. The generation of a plans of composition is dynamic
Formulation of the query	Very low level : the user that defines the composition plan	Low level : the user should implement the process model using flow language	Specifies the initial state, the final state, and with certain constraints mixture of very low level (logic language)	The scheme of composition is defined as sub queries using a knowledge base, each sub query models of semantic web services
Layer of Granularity of generated composition plans	One generated plan (executed plan)	Abstract Workflow and executable workflow	One generated (executed plan)	Many composition plans that are generated
Used supports and abstraction level	No semantic and contextual support l (UDDI registries)	Using of syntactic descriptions	Using of logic formalisms	Using ontologies. Extraction and management context. Enriching service descriptions with context
Discovery web service	Primitive (in the case of UDDI registries)	Using a research motor UDDI	In the profiles DAML-S or OWL-S	Dynamic discovery based on semantic matching, taking into account the context of use.

We identify six evaluation criteria to be considered (first column). Our approach uses AOP to compose services dynamically. This approach depends also on mathematical formula of semantic matching algorithm [19].

Compared to other models listed in the table below, and with respect with these criteria, our approach has demonstrated automation in all the process steps, its trigger element is the user query.; It is inputted in a dynamic interface changing according to the user's needs. This request can be divided into several simple tasks based on a decomposition model proposed and explained in our anterior

work [5]. This process has been also proof of wealth due to a semantic layer that contains all the necessary data to complete the semantic web services discovery. These web services will belong to the composition plan. This discovery process is based too on a parameter that we consider very important: the context. In the end several composition plan are available to the user.

VII. CONCLUSION AND FUTURE WORKS

Web services adoption is a major advance in the development of interoperable information systems. In particular, the composition of services can meet the needs

increasingly complex of user, by a combination of web services within a single business process. However, despite this widespread adoption of Web services, many obstacles prevent their reconciliation in the composition, or may occur within a BPEL process in a state change, the context for example. To solve this problem of the Web service adaptation invoked in a BPEL process and to satisfy the client context and web service context, we have proposed approach based on the aspect weaving, its originality is the addition of two components namely the context manager and appearance manager. These two managers are discussed in detail in Mcheick et al., 2012 [5].

In this paper, we presented the detailed steps (method) we followed to model semantic Web service composition. Then, this composition method is compared briefly with other service composition methods such as manual, workflow and planning methods. This comparison shows that our approach has many advantages in terms of automation and dynamic level, layer of granularity of generated composition plans and others.

This platform needs more investigation in terms of semantic layer and comparison with all the platforms of service composition, such as METEOR-2 [11], SELF-SERV [9], and SHOP2 [13]. As of our perspectives, we need to consider more applications to measure the satisfaction of user requests based on the context and weaving aspects.

ACKNOWLEDGMENT

This work was sponsored by the University of Quebec at Chicoutimi (Quebec), Canada.

REFERENCES

- [1] B. Benatallah, R. Dijkman, M. Dumas and Z. Maamar, "Service Composition?: Concepts, Techniques, Tools and Trends," In: Z. Stojanovi and A. Dahanayake, Eds., *Service-Oriented Software System Engineering: Challenges and Practices*, Idea Group, pp. 48-66, 2005.
- [2] M. Keidl and A. Kemper, *Towards context-aware adaptable web services*, Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, May 19-21, New York, NY, USA, doi:10.1145/science.1013367.1013378, 2004.
- [3] T. Chaari, F.Laforest, and A. Celentano, "Design of Context-Aware Applications Based on Web Services," LIRIS. Dipartimento di Informatica. INSA Lyon, France, 2005.
- [4] M. Vukovic and P. Robinson, "Adaptive, planning based, web service composition for context awareness," Proc. 2nd International Conference on Pervasive Computing, Vienna, Austria, June 2004.
- [5] H. Mcheick and A. Hannech, "Modèle de composition des services web sémantiques par orchestration à la demande," Proc. of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 12), April 29-May 2, Montreal, Canada, 2012.
- [6] A. Dey, G. Abowd, and D. Salber, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," *Human-Computer Interaction Journal*, vol. 16, issues 2-4, pp. 97-166, 2001.
- [7] <http://fr.wikipedia.org/wiki/WordNet>, [retrieved : 04, 2013].
- [8] <http://atla.revdanica.com/>, [retrieved: 05, 2006].
- [9] Q. Sheng, B. Benatallah, M. Dumas, and E. Mak, "SELF-SERV: A Platform for Rapid Composition of Web Services in a Peer-to-Peer Environment," Proc. of the 28th International Conference on Very Large Data Bases (VLDB 2002), Aug., Hong Kong, China. Morgan Kaufmann 2002, pp.1051-1054, 2002.
- [10] B. Benatallah, Q. Sheng, and M. Dumas, "The SELF-SERV environment for Web services composition," *IEEE Internet Computing*, vol.7, n°1, pp.40-48, 2003.
- [11] N. Oldham, C. Thomas, A. Sheth, and K. Verna, "METEOR-S Web Service Annotation Framework with Machine Learning Classification," Proc. of the 1st International Workshop on Semantic Web Services and Web Process Composition, SWSWPC 2004 (SWSWPC 2004), San Diego, CA, USA. LNCS Springer, 2004, pp.137-146, 2004.
- [12] R. Aggarwal, K.Verma, J. Miller, and W. Milnor, "Constraint Driven Web Service Composition in METEOR-S," Proc. of the IEEE International Conference on Services Computing (SCC'04), Sept. 2004, Shanghai, China. IEEE Computer Society, pp.23-30, 2004.
- [13] E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau, "HTN Planning for Web Service Composition Using SHOP2," *Journal of Web Semantics*, vol.1, n°4, pp.377-396, 2004.
- [14] L. Cabral, J. Domingue, S. Galizia, A. Gugliotta, V. Tanasescu, C. Pedrinaci, and B. Norton, "IRS-III: A Broker for Semantic Web Services Based Applications," Proc. of the 5th International Semantic Web Conference (ISWC 2006), Nov. 2006, Athens, GA, USA. LNCS Springer, pp. 201-214, 2006.
- [15] Y. Charif and N. Sabouret, "Coordination in Introspective MultiAgent Systems," Proc. of the International Conference on Intelligent Agent Technology (IAT'07), pp. 412-415, Silicon Valley, California, USA, 2007.
- [16] F. Pourraz, "Diapason une approche formelle et centrée architecture pour la composition évolutive de services Web," Thèse de Doctorat, LISTIC : Laboratoire d'Informatique, Systèmes, Traitement de l'Information et de la Connaissance, Dec. 2007.
- [17] A. Dautriche and T. Melliti. "Génération automatique de code pour services web composites," Rapport de TER, 2010.
- [18] S. Lajmi, C. Ghedira, K. Ghedira, and D. Benslimane, "WeSCo CBR: How to compose Web Services via Case Based Reasoning," *e-Business Engineering ICEBE*, pp. 618-622, 2006.
- [19] H. Mcheick, M. Adda and A. Hannech. "Web Service Discovery Model Based on Context," *American Journal of Software Engineering and Applications* issued by Science Publishing Group, Vol.1, No.1. December 2012.
- [20] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parisia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara, "Owl-s : Semantic markup for web services," Technical report, W3C, 2004.
- [21] F. Curbera, Y. Golland, J. Klein, F. Leymann, and S. Weerawarana, "Business Process Execution Language for Web Services (BPEL4WS)," Version 1.1, May 2003.
- [22] <http://www.ibm.com/developerworks/webservices/library/ws-bpelcol4/>, [retrieved: 11, 2002].
- [23] T. Chaari, F. Laforest and A. Flory "Adaptation des applications au contexte en utilisant les services WEB " *UbiMob 2005*: 111-118