Applying Ontological Similarity to Automatic Service Composition to generate alternative Business Processes

Phillip Luiz Viana¹,², Jorge Luis Risco Becerra¹
¹ University of São Paulo (USP)
Escola Politécnica – Computer Engineering Department
São Paulo, Brazil
² IBM Brazil Software Lab
São Paulo, Brazil
Email: phillipv@br.ibm.com, jorge.becerra@usp.br

Abstract—With the growing trend towards increased use of mobile technologies, companies have been providing their services on the web more than ever in order to fulfill user requests quickly in various situations. It is often necessary that two or more services from different providers are automatically composed in order to meet a single user request during runtime: this is called automatic composition of services. When a service composition is executed, it is automating a business process associated to the user request. If no compositions are found that can fulfill the request, the user might want to relax his requirements to accomplish his goal alternatively. The objective of this work is to present an automatic composition model that uses similarity between concepts of an ontology to generate alternative service compositions, thus enabling a single user request to be met in several different ways through different business processes. The proposal is validated through a software prototype that is exercised in the tourism domain.

Keywords – automatic service composition; similarity; ontology; semantic services

I. INTRODUCTION

With the growing trend towards increased use of mobile technologies, companies have increasingly been providing services on the web to quickly fulfill user requests in various situations. It is often necessary that two or more services from different providers are automatically composed by a software during runtime in order to meet a single user request: this is called automatic composition of services. The composition generated by the software automates a business process associated to the user request.

In order to have the composition built in an automatic way it is necessary that the user somehow expresses his request to the machine, which interprets his goal and searches in a repository for services that when composed together will satisfy the request. Currently there are automatic composition systems that receive the user request in natural language and interpret the purpose of the request using ontologies [1, 2, 3], which are explicit specifications of conceptualizations [4] and relations between them. Also a service can be defined with ontological information, in this case called semantic service. In a semantic service, the operations and the input and output parameters are mapped to ontology concepts, so that inferences about the services can be made. If both service and request are associated to ontology concepts, we can map the request to a service composition through an automatic composition algorithm and thus fulfill the user request in runtime with a service composition.

However, whenever a user request is not met by any composition due to the lack of suitable service compositions, the user might want to relax his requirements to accomplish his goal alternatively. In this sense, it is essential to have an approach that will generate alternative compositions, replacing the original candidate services to fulfill the user request. The model presented in this paper allows one to create an automatic composition system that provides alternative compositions based on similarity between services and ontology concepts, generating alternative business processes associated to that particular user request.

This paper is organized as follows: Section II refers to related works on automatic composition of services, similarity and proposals to integrate them; Section III details the problem of our research and presents an overview of our model; Section IV discusses the model in more detail and provides examples; Section V presents the experimental study. Finally, Section VI concludes this paper with future work and final considerations.

II. RELATED WORK

A. Automatic Composition of Services using Ontologies

In the field of automatic composition, the user request can be specified either syntactically or semantically. The syntactic approaches involve the specification of the service interface or explicit goal ontology manipulation by the user [5, 6]. In the latter, even though the approach uses an ontology, it is a syntactic approach since the ontology concepts are not related to the concepts of the words in the request, but to the target service interface, i.e. the user uses the ontology to specify the
entry parameters required by the service he wants to obtain and the return parameters that it should provide. Requiring this expertise is possibly prohibitive for end users, given that a considerable number of end users are lay over ontologies and service composition.

The semantic specification of the request concerns the semantic content of the user request. Many of these approaches obtain user requirements from natural language, either spoken or written. The use of ontologies to annotate service information in automatic composition is seen in several studies [6, 7, 8, 9, 10, 11], while there are also studies about the association of natural language and ontology to interpret end user requests [1, 5]. Other matching approaches associate words in lexical databases that do not support semantics (such as WordNet) or use pre-defined templates [12, 13, 14]. The use of natural language is shown as an appropriate approach due to its usability to final users.

B. Ontology Concept Similarity

Ontology concepts hold similarity degree with each other. While the technique to measure the similarity degree (usually with a numeric value) between two concepts may vary, one can say for example that two concepts car and bus, both subtypes of the same concept vehicle, hold a greater similarity degree than car and house, since house and vehicle are both subtypes of the root concept thing.

The calculation of similarity between concepts may be useful to find similarity between services especially when services are associated to ontology concepts. Proposals such as [15] and [16] define similarity metrics between semantic web sever requests to assist the service discovery mechanism through concept similarity. The former through ontologies, and the latter through mechanisms of data mining on the WordNet lexical database.

In our work the similarity between concepts is used to search for alternative services associated to concepts which are similar to the ones in the original user requirement.

C. The use of Similarity in Automatic Composition

In the realm of automatic composition, similarity between services is used to perform service discovery and suggest alternative services in the case of unavailability of a candidate for a composition. In figure 1, suppose a user has a request R that can be resolved by business processes BP1, BP2 and BP3, and BP1 contains two different compositions C1 and C2 that can automate it. Faced with the unavailability of a resource in C1 (e.g., a service is down), C2 can be used as a substitute due to the similarity it holds to C1.

Suppose a user has a request R. He wants to travel from São Paulo to Rio to watch a samba school rehearsal. The business process BP1 associated with this request comprises a multitude of services and providers such as air companies, hotels, taxis and credit card. If taxi company A is unavailable then taxi company B can be found since their services are similar. If there are no more air tickets available between the origin and the destination, a bus ticket might be proposed since an airplane and a bus are both transportation vehicles and thus there is a similarity degree between them. In this example, C1 can be a composition that uses taxi company A, while C2 uses taxi company B; or C1 can be the business process using an air ticket while C2 uses a bus ticket.

Nevertheless, the business process associated to those compositions is basically the same, with equivalent activities such as buying a ticket (either air ticket or bus ticket), booking a hotel, booking a taxi (either from company A or B), paying with credit card and so on. Therefore, current proposals use similarity to enable the generation of alternative compositions that automate the same business process. A composition C1 is said to be similar to a composition C2 if it partially satisfies a request R, compared to R2.

The key difference of our work is that our model aims at generating alternative compositions that span across different business processes rather than alternative compositions associated to a single business process as detailed in section III. Existing automatic composition models replace services that fulfill the exact same requirement. Our goal is to provide an automatic composition approach which is capable of mutating the requirement (thus changing the associated user business process), satisfying the user request partially rather than not fulfilling it entirely.

III. PROBLEM CHARACTERIZATION

In the previous example, suppose there were no more air tickets available and the user would not arrive on time for the samba school rehearsal if he went by bus. In this case, no compositions are available and the request is not met.

Nevertheless, there exists the possibility that this user wants to relax some of his requirements to be able to watch the...
rehearsal. Suppose the rehearsal is being shown in a 3D cinema room in São Paulo. This alternative solution is associated to different business processes since many activities will no longer be needed (such as book taxi, buy ticket and book hotel room) and new activities will be added.

Figure 2 represents the change from live show to 3D cinema. The business process BP3 associated to watching the rehearsal in a 3D cinema room did not originally resolve request R, since the request involved live show attendance. However, BP3 holds some similarities with BP1, which genuinely fulfills the request but can not be automated by a service composition due to the lack of available air tickets. BP3 alternatively resolves the request if the user relaxes his requirement of watching the rehearsal in person.

![Figure 2. Similarity between business processes](image)

We propose an automatic composition model that supports the generation of alternative business processes if there are no available compositions that automate a business process that fully satisfies the user request. The model takes user request in natural language, maps the request to ontology concepts, searches for semantic services associated to those concepts (and similar concepts) and finally provides compositions that might serve the user request.

IV. AUTOMATIC SERVICE COMPOSITION AND SIMILARITY TO GENERATE ALTERNATIVE BUSINESS PROCESSES

This section shows the approach of our research, first explaining the basic elements of the model (Section A), how the user request is interpreted from natural language to ontology (Section B), how the similarity between concepts is calculated (Section C), the hypotheses that rule how the alternative business processes will be obtained using similarity (Section D), the new approach for similarity search and the algorithm for automatic composition (Section E), and finally the extended rules to generate alternative business processes, derived from the hypotheses and enabled by the new similarity search approach (Section F).

A. Basic Elements of the Model

The model is based on similarity, semantic services and automatic composition techniques. It defines the elements needed to implement an automatic composition system that generates alternative business processes.

In this section we will describe the basic elements that an automatic composition system should comprehend according to the model:

Ontology: a set of concepts and relations between them, expressed through one or more documents in markup language. The ontology is specific to a domain

Similarity: a numeric value that represents the degree of similarity between two nodes of the ontological tree. An ontological tree is a representation of an ontology. In an ontological tree, the root concept is Thing and other concepts are derived from it through inheritance

Atomic semantic service: a computer service with interfaces (inputs and outputs) and one or more operations, which is not composed of subservices. The interfaces are associated to ontology concepts

Final service: an atomic semantic service which encapsulates the semantics of a concept. A final service might be compared to a class with its attributes, but it is abstracted to a service to enhance and facilitate the automatic composition algorithm. For example, a final service Airplane encapsulates the semantics of the concept of an airplane, and the attributes of the airplane are implemented as return parameters of its operations. Non-final services have semantics embedded into their operations and parameters, but they don’t represent a concept as a whole

Composite service: a service consisting of atomic semantic services and/or other composite services. The goal of any automatic composition system is to generate composite services

Request: a user request that must be met by the system. It is passed in written natural language. The request must obey to a construction rule that will be detailed later

Automatic Composition Algorithm: the algorithm responsible for finding the appropriate services that match the semantics of the request passed by the user and suggests service compositions according to the requirement

Similarity rules: the rules used by the automatic composition algorithm that determine when two services are considered similar. These rules involve the similarity between the concepts represented by each service and/or operation parameters, and some rules to generate alternative business processes according to three hypotheses

B. Request in Natural Language

As mentioned before, the model assumes that a user will express his request in written natural language. The request needs to comply with a fixed generation rule based on [1]:

\[
\text{Request ::= Predicate Noun (Concept Noun)*}
\]

This rule is expressed in BNF notation [17] and means that every user request must contain at least two words: a predicate, indicating an action; and a noun, indicating the object towards which the action will take place. The request may optionally contain one or more sequences of two words; one general concept followed by a noun. A request such as “buy ticket from São Paulo to Rio” contains the word buy as predicate;
ticket as first noun; from and to as general concepts (links), and São Paulo and Rio as nouns.

The request keywords are searched in the ontology and the concepts associated to these words, as well as similar concepts, are kept in a concept buffer that will later be used to perform service discovery. We search on the ontology for the concepts with the same name as the words in the requirement, and then we look for similar concepts. This process is known as similarity search.

C. Concept Similarity
The calculation of similarity between the concepts of the ontology is based on [15]. The similarity between concepts is used to indirectly find similar services according to the concepts associated to a service and its operations.

The similarity between concepts is important in the research because it will allow the discovery of similar services that will enable the generation of similar (alternative) business processes.

D. Hypotheses for the Generation of Alternative Business Processes
Several typical requests of the tourism domain were analyzed and were assigned correspondent business processes that fully met the request. Through the heuristic method, similar business processes that alternatively or partially met the original request were obtained. The generation of alternative business processes was only possible because the semantics of the request were changed to different semantics.

After the fully satisfactory business processes were transformed into similar business processes, we detected the basic principles that were implicitly used for that transformation, leading us to three hypotheses for the generation of alternative business processes:

Hypothesis H.1: in order to generate alternative business processes, the request semantics must be altered to similar semantics

Hypothesis H.2: the first noun of the request holds most of the request semantics

Hypothesis H.3: starting the similarity search by the first noun, rather than the predicate, enables the generation of similar semantics for the request

To illustrate the hypotheses with an example, in the request “buy air-ticket from São Paulo to Rio”, we could say that these hypotheses lead to the three conclusions below:

Conclusion 1: in order to generate an alternative business process to meet the request (other than buying an air ticket, considering that there are no air tickets available), the request semantics must be altered to similar semantics, for example, changing the transportation method, renting rather than buying, etc. (H.1).

Conclusion 2: the first noun of the request, air-ticket holds most of the request semantics (H.2). Intuitively one can say that the is the central piece of this request is actually the air ticket.

Conclusion 3: starting the similarity search by the air ticket enables the generation of similar semantics for the request (H.3), because we can look for different concepts that might replace the first noun, such as car and bus-ticket; as a consequence, the search for predicates that use these similar concepts might lead to different business processes that alternatively solve the user need, such as rent car.

E. Automatic Composition Algorithm and Similarity Search
The three hypotheses culminated in a fundamental change in the automatic composition algorithm. Our algorithm is based mostly on the algorithm proposed by [1], which processes the request and searches for concepts in the same order as the words are shown: Predicate, Noun, (Concept, Noun, Concept, Noun...).

According to the three hypotheses found H.1, H.2 and H.3, the lookup for concepts associated to the contents of the first noun of the request will enable the generation of new semantics for the request and, consequently, new business processes.

In this sense, the search for concepts associated to the request semantics must also change. Originally, in models that do not aim at the use of similarity for finding alternative business processes, the search for concepts began with the predicate, as seen in figure 3.

Request ::= Predicate Noun (Concept Noun)*

Figure 3. Concept search approach not focused on similarity

In this approach, we first look for services containing operations that meet the predicate. For example, in the request “buy ticket from São Paulo to Rio”, the first step in the search would be finding services associated to the concept of buying (or paying for) something. After the search is finished for the predicate, we need to find the services in the pool of candidate services that meet the first noun, which generally means they receive an input parameter to fulfill the operation found for the predicate.

As per the hypotheses H.1, H.2 and H.3, the similarity search is driven by the first noun. Our approach begins with the search for concepts associated to the first noun and concepts with a minimum similarity degree \( \sigma_{min} \) to the first noun, according to [15]. Then candidate services which are associated to these concepts are kept in a pool. In the example, suppose that the concepts bus-ticket and car were found as having acceptable similarity degrees to air-ticket, so the pool must contain candidate services associated to the concepts air-ticket, bus-ticket and car.

After the pool of candidates for meeting the first noun is done, we then look for candidate services that contain operations that receive them as input parameters, and keep them in a predicate pool (this search is depicted as the row number 1 in figure 4). For example, services with operations called buy that receive air-ticket or bus-ticket, or services with an operation rent that receive car as an argument.

Also, concepts that meet the predicates associated to the first noun (not its similar concepts) are searched (arrow 2).
This enables us to find different predicates associated to the first noun passed in the request.

\[
\text{Request ::= Predicate Noun (Concept Noun)}^* \\
\]

Figure 4. Similarity search approach

The similarity search continues complementing the entry parameters of the services associated to the predicate and the first noun (and similar concepts). These entry parameters have to be associated to the set of optional Concept-Noun concepts of the request (arrows 3, 4 and 5). In the example, we need to find services that provide origin and destination information for São Paulo and Rio, since services associated to buy air-ticket and buy bus-ticket must receive origin and destination parameters associated to these locations.

This adaptation enables us to make important transformations in the request semantics:

- Other predicates associated to the first noun
- Other services with at least one entry argument associated to a concept that is similar to the first noun
- Other services that return a parameter associated to a concept that is similar to the first noun

These transformations led to three important rules that derived to sixteen similarity rules on the service level:

- Rule 1: Two final services are similar if they return similar parameters
- Rule 2: Two services are similar if one of them has an operation that takes an argument associated to a concept C, while the other is a final service that represents C or a similar concept
- Rule 3: Two services are similar if they return the same parameters, and the set of entry parameters of one service is a subset of the set of entry parameters of the other service

These rules were implemented in the automatic composition algorithm as shown below:

### Rule 1
Discover services that return a parameter similar to Ci;  
Add service to SEARCH_PREDICATE_LIST;

### Rule 2
Discover services that take Ci as input parameter;  
Add service to SEARCH_PREDICATE_LIST;

---

<table>
<thead>
<tr>
<th>C is (P(S_1))</th>
<th>Final service (S_1) contains a property of type (C) ((C) is a property of (S_1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_1 &gt; C)</td>
<td>Service (S_1) contains an operation that returns a type (C) ((S_1) returns (C))</td>
</tr>
<tr>
<td>(C_1 \sim C_2)</td>
<td>The concept represented by (S_1) is similar to the concept represented by (S_2) ((S_1) is similar to (S_2))</td>
</tr>
<tr>
<td>(O(S_1) &lt; C)</td>
<td>The operation (O) of the service (S_1) receives an argument that represents (C) ((S_1) receives (C) as input)</td>
</tr>
<tr>
<td>(E(S_1) \subseteq E(S_2))</td>
<td>The set of entry arguments of an operation of service (S_1) is a subset of the set of entry arguments of an operation of (S_2) ((S_2) contains all the arguments of (S_1))</td>
</tr>
</tbody>
</table>

---

In this sense, the generation of alternative business processes is enabled by semantic changes in the request in natural language. These hypotheses change the generation of compositions, as we will see in detail in section IV, and the new compositions have direct impact on the business process they automate.
V. EXPERIMENTAL STUDY

We adopted a mix approach of investigative surveys and the experimental model on software engineering presented in [18] for the experimental study.

Even though we adopted the tourism domain as the experimental boundary of our research, we believe that other domains can benefit from the use of similarity with automatic composition. Thus, even though the experiment is limited to the tourism domain, we would like to emphasize that proper evaluation needs to be performed to establish which other domains could benefit from our research, and this is included as a topic of future work.

The experimental application was divided into five sections:

1 – Identification of the tourism sector: here we present the field of tourism as our experimental limit, justifying its use to prove the validity of the proposed model.

2 – Definition of the purpose of the experimental application: we define the purpose of the application and how the research can be useful to the society

3 – Application model: details the experimental application of the model: what are they and how test scenarios will be run, the constraints on the scenarios, as well as result analysis

4 – Operation of the experimental application: practical details of the experiment

5 – Data analysis: the data collected during the experiment was analyzed to assess the contribution of our proposal

A. Tourism Domain

The tourism domain has been selected as the experimental scope of research. Some specific reasons to the field of tourism make automatic composition interesting for this area. Firstly, users are mostly using mobile technologies (because they are exploring the areas through which they pass) and therefore, their needs vary with frequency. The need for certain types of value-added services is higher in comparison to other areas, for not only tourists know where you are, but generally has the objective to know as much as possible. In addition, many consumers are not even aware of what product or service they are looking for concrete [19], and want to be surprised with products and services that they are not yet aware. Several systems for automatic composition of the tourism sector were analyzed and reported before, like [5], [19], [20], [21], [22] and [23], among others. These systems are usually tour guides that compose services at runtime to meet a request, ideally instantaneously. None of these systems, however, suggests alternatives at integration of business processes, while defining and using similar services in the event of unavailability of any necessary service to the composition.

The experiment of similarity in a feasible way in the tourism domain is justified by the heterogeneity of service providers and non-urgent requests of the users, typically the field of tourism has several different providers for the same services (airlines, taxi companies, hotels among others), while the requests can be loosened by the users, since in general are not critical or emergency.
B. Purpose of the Experimental Application

The objective of the experimental application is to demonstrate that our model of automatic composition with similarity represents a significant contribution in respect of models of automatic composition no similarity, using metrics that demonstrate quantitatively that contribution within an experimental limit, which is the field of tourism.

It also aims to allow implementation of the survey is compared with other surveys or generalized from future work based on it. It will set an experimental context, i.e., a set of parameters (characteristics or constraints) within the experimental limits, while ensuring the comparability of different experiments in the area and allow them to have the notion of the applicability of the results for other needs. Thus, the parameters will be explained the context to be stable or variable during the experiment, so that it is determined whether the observed differences in results may or may not be assigned to these parameters.

C. Application of the Experimental Model

According to [24], investigative research (surveys), case studies and experiments are the three most research strategies used in experimental applications. In surveys, investigative information is collected evenly amongst a group of people or projects; an experiment has a high level of control, and all the variables that will be experienced and that will be fixed in order to obtain results that can be compared with other experiments or expected results; a case study is an investigation of one or a set of related cases and their effects.

Our research used the strategies of investigative research and experiment together for the experimental application. We found the experiment as an interesting approach, since in our case we can define the variables to be determined (services and their parameters, number of requests, degree of similarity, among others), and one can define the expected results.

The investigative research was used to assess the aspect of practical use of the tool by practitioners of the application domain, that is, tourists (experimental subjects) can use the tool in familiar contexts of the field of tourism and provide information on use, thus enabling a quantitative evaluation of the research.

The case study does not appear as an interesting approach at first, since there is no practical cases of using other tools with similar functionality, and the first cases to be run are the test cases define in our research.

D. Operation of the Experimental Application

For the experiment, we implemented a prototype architecture according to our model and populated this architecture with a few dozens of illustrative services for the tourism domain in a region of the city of São Paulo.

A set of 23 practitioners of the tourism domain experimented this prototype with real requests during their visits and explorations of the city. They received training and were asked to provide actual requests that met their desires during their visit to the city. The prototype was exercised in two operation modes: enabled similarity mode and disabled similarity mode. The system provided fictitious service compositions that might serve the user needs during run time, including alternative business processes in the enabled similarity mode.

For the survey, after their exercise of the prototype, practitioners of the domain were asked to fill a form with a few questions about their experiences with and without the similarity feature. The questions on the form included qualitative options in Likert scale to assess from what scale the compositions suggested met their requirements.

E. Data analysis

The experiment was run with 23 practitioners from the tourism domain, coming from 11 different countries, between men and women, ages from 17 to 42.

The comparison between the answers in the survey show an increase of 4% in answers of “All” or “Most” for the question “How many compositions suggested by the software met exactly your requests?”, comparing the prototype with disabled and enabled similarity feature; we noticed significant increase of these answers to the question “How many compositions suggested by the software did not meet exactly your requests, but were alternatives that sufficed your needs?”, and significant increase to the question “How many compositions suggested by the software met partially your requests?”. However, in the comments section, many users reported that, even though the similarity feature provided many more compositions numerically, comparing to the prototype without the similarity feature enabled, many of the compositions offered did not meet their request at all.

VI. CONCLUSIONS AND FUTURE WORK

We consider that the contribution of our research on applying similarity to automatic service composition was shown in our work.

The model for the process of experimental software engineering (ESE) was pointed out in [18] as having been used in most areas of science, the experimental procedure is used to explain phenomena and accept or reject one or more hypotheses, this practice in science computing and software engineering is not widely used.

We used the experimental strategy according to the experimental model in [18] and investigative research strategy through the collection of the quantitative results of this research. The quantitative paradigm sought to quantify the phenomenon through metrics with the aim of identifying cause and effect. For this, several fictitious services of the field of tourism were defined, and actual practitioners of the domain (subjects of the experiment, i.e., tourists) ran the tool and provided objective information about their perceptions regarding the operation of the tool in the context in question.

Results show that the generation of alternative semantics to the requests contributes to the generation of alternative business processes, which validates our three original hypotheses.
However, more work needs to be done in order to guarantee higher quality of the compositions provided. Two approaches that might solve this issue are more rigorous semantic reasoning and choice learning by the history of composition executions. These are identified as topics for future research.

REFERENCES