A Comparison of Three Service Composition Approaches for End Users

Usman Wajid, Abdallah Namoune, Nikolay Mehandjiev
Manchester Business School
The University of Manchester
United Kingdom
{firstname.lastname}@manchester.ac.uk

1. ABSTRACT
This paper compares user preferences for three alternative approaches to service composition, namely: control flow, data flow, and assisted composition approach. The end user perspective is gathered by organizing three focus groups that include discussions and subjective questionnaires, involving 35 non-technical participants. The comparison of alternative composition approaches yielded results confirming that users favour system-driven or assisted composition which deals with technological complexities such as service compatibility problems while allowing user control and involvement. The results also define the requirements for user-friendly design of a service composition tool, which is being developed in the EC-funded project SOA4All.

Categories and Subject Descriptors
D.1.7 [Visual Programming]

General Terms
Design, Human Factors, Verification.

Keywords
Service Composition Approaches, Service Oriented Architectures, End User Development, Focus Group.

2. INTRODUCTION
Service-oriented technologies are gaining widespread popularity amongst Internet users by offering independent services that can serve as building blocks for constructing composite services or service-based applications. The activity of connecting different services to achieve combined functionality is known as service composition. Key benefits of service composition include producing combined functionality which is not yet available as a service, improved reusability because the same service can be part of many composite services and improved flexibility since modifications can be made by replacing an individual service in a composition. However, despite the advantages of service composition only a small proportion of users, often with considerable modelling and programming skills, can construct service-based applications. Majority of Internet users are unable to exploit the benefits of service-oriented technologies and create service-based applications tailored to their specific needs. This limitation can be attributed to the complexity of the existing composition approaches and the limited technical knowledge of ordinary end users. Thus, the research challenge lays in simplifying service composition and abstracting this process from any unnecessary technical complexity, with the general aim of promoting service reuse and consumption especially among end users who have no or very limited technical background. The EC-funded project SOA4All1 addresses this research challenge by developing a framework and set of tools to support the complete service lifecycle from service discovery to composition and consumption. In particular, to simplify and engage end users in service composition, this paper reports on user opinions about three different service composition approaches. The user opinions are important in making informed selection on a user preferred approach that can be used in the composition editor of the SOA4All studio. This will enable the enhancement of this service composition environment allowing end users to develop composite services tailored to their needs and requirements.

3. EXISTING WORK ON SERVICE COMPOSITION
At present, Internet users can add independent web services as widgets to their personalized pages on iGoogle2 and Facebook. However, this cannot be regarded as service composition because users cannot wire services together to exchange and share data or functionality. The produced service-based personalised pages are very trivial and offer autonomous services. It is more useful and interesting if end users are enabled to produce rich and complex service-based systems that fulfil their specific needs, but also allowed to easily extend and customise applications. An integrated mini holiday-booking application which consists of a flight-booking service, a hotel-booking service and a car-booking service is a good and realistic example of interactive service-based application that users can build and customise by adding or removing specific services.

An interesting example of service composition is Yahoo Pipes3, which enables users to combine various information sources and perform different filtering operations to achieve a desired outcome. The major drawback of this approach is its reliance on modelling skills which most end users do not have. In addition, the nature of information available to users and the type of

1 http://www.soa4all.eu
2 http://www.google.com/ig
3 http://pipes.yahoo.com/pipes/
operations that users can perform on that information is quite limited.

Existing research efforts for service composition mostly focus on the development and implementation of latest techniques and languages for realizing service composition. Such initiatives are typically targeted towards business developers, with virtually no attentions to the ordinary end-users’ needs and perspective. In this respect, existing approaches for service composition employ top-down techniques where the users are expected to get familiarized and use the approach that has been presented to them. For example, the surveys in [1] and [2] discuss existing approaches for service composition with a particular focus on the technical aspects such as the use of various languages (e.g. OWL-S, BPML) and technologies (e.g. Petri-nets) employed by the existing approaches. However, if ordinary users are to capitalise on the benefits offered by Service Oriented Architectures (SOAs) are not the technical aspects, but how these are presented and how they can use these advanced technologies to perform their desired tasks.

Despite rapid developments in service-oriented technologies, service composition by end-users is an area in its infancy. In this respect, identifying the needs and specific requirements of ordinary users is a crucial prerequisite to the design of an “easy to use” and “easy to understand” service composition approach. Such a bottom-up approach can be helpful in promoting the uptake of service-based technologies by ordinary users.

4. ALTERNATIVE SERVICE COMPOSITION APPROACHES

The user opinions about three alternative service composition approaches have been gathered in the SOA4All project in order to simplify the service composition process and to hide the underlying technical complexity from ordinary users. These approaches are inspired by (1) engineering methods for analysing computer programs and (2) human interface design methods. An overview of each of the three approaches is given below:

4.1 Control Flow-based Composition Approach

Control flow (an example shown in Figure 1) represents a sequential composition process where a task or service requires completion before the next task / service is executed. In other words, control flow defines the order in which atomic services are executed. Moreover, control flow approach enables users to define various useful relations between services, namely: (1) unconditional branching, (2) conditional branching, (3) iterative execution (i.e. loops), and (4) unconditional stopping.

Unconditional branching is simply a continuation to the next service. Conditional branching is specified using predicates, for instance: GetFriendLocation service (in Figure 1), which retrieves the exact location of friends, is executed if and only if, GetFriendAvailability service satisfies a precondition “Friends are available”. Iterative execution defines the execution of a particular service zero or more times until a condition is satisfied. Unconditional stopping signifies the end of the composite service. This composition approach does not deal with how and what type of information will be passed between single services.

4.2 Data Flow-based Composition Approach

Data flow represents an information-oriented view of service composition, where data is passed between multiple services without the requirement of a specific sequence. Data flow diagrams contain a set of concurrently executing services that exchange messages. This approach enables users to define how data flows from source service(s) to destination service(s). Each connection in the diagram ought to carry the data that is passed from one service to another. Using this approach, no information about service execution order and conditions can be defined or elicited.

4.3 Assisted Composition Approach

The Assisted Composition approach is drawn from human-computer interface design methods and principles. This approach enables users to choose individual services from a wide range of available categories in the composition template (as shown in Figure 3). However, the selection of services is supported by the underlying computational algorithms that manage the interoperability and compatibility of already selected services. For example if a user selects a service ‘Set Place Google’ from ‘Set Location Task’ category then only compatible services are highlighted in the subsequent categories (incompatible services are greyed-out). Users can select from a wide range of
customisable system templates and do not have to define control and data flow among services as these aspects are managed automatically.

<table>
<thead>
<tr>
<th>Templates</th>
<th>Set Location Task</th>
<th>Set Friends</th>
<th>Friends’ Location Services</th>
<th>Travel Route Services</th>
<th>Weather Services</th>
<th>Messages Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Birthday</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conference</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Movie</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wedding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Family</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Food</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Travel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: Assisted Composition Approach

5. COMPARISON OF THREE SERVICE COMPOSITION APPROACHES

Three focus groups were organized with non-programmers (35 participants in total) to capture their perceptions about the aforementioned service composition approaches. The participants discussed the advantages and disadvantages of each composition approach from an end user perspective, followed by filling in rating questionnaires. We used inductive content analysis [3] to analyse the qualitative results gathered in the focus groups.

The results show that users liked control flow-based approach for its ease of use. However, users expressed concerns about the strict nature of the approach as it does not allow them to drag and drop services in a composition without revising the overall composition i.e. how control will be passed through one service to another, thus it requires attention to details to make it work. Additionally, participants argued that due to the sequential nature of a control flow a composition involving various services can get too complex for them to keep track of.

On the other hand, data flow-based approach ranked high for the flexibility it offered to users, but was regarded as a difficult approach for non-programmers due to the underlying complexity of managing different data sources. Moreover, when data flow diagrams (Figure 2) were shown, some participants pointed out that it was difficult for them to understand how the overall structure works. Other opinions reflected similar worries about the disorganized representation of data flow and the resulting complexity in understanding relationships between various services. Hence, data flow approach was regarded as more suitable approach for users with technical background and programming experience.

In the end, the assisted composition approach came out as the easiest approach to use and scored the highest ranking for acceptability by the participants. Overall participants were highly interested in the assisted composition approach, since it reduces the complexity of service composition by providing a systematic way of composing compatible services. The interest in assisted composition can be linked to the non-programming background of users, as most of the users regarded control flow and data flow as “programmer’s way of building software artefacts”. In this respect, participants preferred this approach over the other two approaches because it saves time and is perceived as the least complex by hiding technical details from the user.

However, few issues were raised during the discussions about the scalability and flexibility of the assisted composition approach. For instance, participants showed interest in customising and creating their own templates and extending the list of available services. Respectively, questions were raised about the scalability of the approach in accommodating many services on a template and how the approach will be able to accommodate user preferred services in a template that are not available in the list of services. Moreover, probing the limited flexibility offered by the approach participants were interested if the approach could allow them to select more than one service from a category, e.g. rather than getting contact details from one service provider i.e. Gmail, the user may like to add some contacts from other service providers as well such as Facebook. This could make it difficult for users to compose a service-based application using only the available set of services within the template.

Subjective ratings showed that assisted composition ranked higher than control flow and data flow-based composition on all usability measures, except “easy to understand”, as shown in Table 1.

Table 1: Users’ subjective rating of the three composition approaches on a 5-point rating scale

<table>
<thead>
<tr>
<th>Usability Measures</th>
<th>Control flow</th>
<th>Data flow</th>
<th>Assisted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to use</td>
<td>4</td>
<td>3.45</td>
<td>4.27</td>
</tr>
<tr>
<td>Easy to understand</td>
<td>4.81</td>
<td>3.45</td>
<td>4.09</td>
</tr>
<tr>
<td>Effective</td>
<td>3.63</td>
<td>3.36</td>
<td>4.27</td>
</tr>
<tr>
<td>Overall rating</td>
<td>4</td>
<td>3.54</td>
<td>4.45</td>
</tr>
</tbody>
</table>

6. CONCLUSION

The paper reports on the user opinions about three alternative service composition approaches. The aim of the study was to identify approaches which can enable ordinary users who have no significant modelling and programming skills to compose service-based applications. Focus groups revealed that end users favour system-driven composition owing to its ease of use. Whilst both control flow and data flow approaches require profound knowledge of programming and modelling concepts, assisted composition hides away all the technical details and complexities. Nevertheless, for the later approach to succeed it has to offer the right level of control to users by allowing them to customise templates and add services according to their needs. This requirement is important for designing tools that realise the benefits of end-user based service composition.

7. REFERENCES

