The 3rd Workshop on Automatic Service Composition

Co-Chairs: Ying (Jenny) Zou, Queens University; Bipin Upadhyaya, Queens University; Alex Lau, IBM Canada Ltd.; Joanna Ng, IBM Canada Ltd.

Theme: Dynamic Business Process Management and Smart SOA

Integrated Solution: Smarter Commerce

Abstract:

INTRODUCTION

Service-oriented architecture (SOA) is an architectural style for building software applications that uses services available in a network such as the web. An architectural style defines a vocabulary comprised of component and connector types as well as constraints on how they can be combined. SOA promotes loose coupling between software components so that they are able to be reused. Applications in SOA are built based on services, which means it is build on an implementation of well-defined business functionality. Such services can then be consumed by clients in different applications or business processes. SOA solutions have been created to satisfy business goals such as integration with legacy systems, reduced costs, and the ability to provide innovative services to customers. Web services are able to be used to realize SOA. Web services are defined as self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services range from comprehensive services such as storage management and customer relationship management to more specific services such as travel reservation, book purchasing, weather forecasts, financial data summaries, and news gathering. These fine grained services can be combined to solve a specific task such as locating cheaper airline tickes from a number of service providers. In short, services are everywhere in today's Internet.

When one or more services are connected, we define the compilation as a service composition. A service composition is a combination of collaborating services and/or resources that work together in order to obtain higher functionality. Service composition involves the development of customized services often by discovering, integrating, and executing existing services. Most of those services are fine grained for providing specific types of functionality. The fine granularity gives independence to end-users so that they can involve different kind of services. For example, a holiday-booking application from an end-user's perspective may consist of a flight-booking service, a hotel-booking service and local event service. The same holiday-booking application from another end-user's perspective may contain different variations on similar sets of services. End-users should be allowed to easily extend and customize different services because each end-user has their own specific needs and preferences. Selecting a trust-worthy service from a set of services and personalization services is still a research question.

Service composition programming models and tools are primarily designed to help professional SOA developers solve business problems in enterprises' complex IT environments. These technologies involve relatively strong developer skills and supporting infrastructures. Non-technical end-users with limited IT capabilities are unable to use these technologies to compose services. In the last few years, there has been rapid development in technologies with the goal of enabling end-users to aggregate Web content, such as data and functions from multiple Websites into an integrated unit to solve a specific problem (such as comparing prices between different Web sites). Mashups are popular tools of such enablement. A Mashup is generally referred to as a Web application that integrates data and sources from multiple sources to provide a unique service. However, mashups require some technical and programming skills which are not available for general end-users. All these illustrate a high barrier for general Internet users to freely and frequently construct data aggregation across Web sites for their everyday activity. There is no infrastructure that helps the user compose, share and distribute services among their peers. As the result, service composition from the perspective of the end-user remains a challenge.

In this workshop we investigated the research and practices in automatic service composition. We invited the speakers to discuss the following topics:

1) Service Discovery and Selection

The accuracy of service discovery increases the success in service composition and eventually enhances the satisfaction of the composition requirement. There are different ways to implement services (such as SOAP-based and RESTful services) and there is a large and rapidly growing number of heterogeneous services available on the Internet. It is time-consuming to search for appropriate services from a large number of available services. A Web service can be described by the interfaces (such as WSDL, WADL, and HAL), making it more difficult to discover and select an appropriate service.

2) Personalization in Web portals and Web services

Users' expectations and requirements constantly change so it is important to include their preferences in the composition and provisioning of Web services. Personalization is dependent on the features of the environment in which it happens. These features can be related to computing resources (such as fixed device, mobile device), time periods (such as in the afternoon or morning), and physical locations (such as a meeting room or cafeteria). The gathering and refinement of an environment's features will allow the ability to define its context. We looked into a different approach of personalization in Web portals and Web services and explored different context-aware model uses in Web services and Web portals.

3) Service Composition

Service composition is the process of constructing a complex service from atomic ones in order to achieve a specific task. The process of service composition inherently requires the specification of composition requests, a formal specification of static and behavioural properties of the service components, a matchmaking algorithm, and a modeling language expressing the logic of a composite service. Current approaches of service composition are very difficult for end-users to compose services. We explored different service composition techniques and frameworks.

WORKSHOP AGENDA

The workshop focused on research talks and interactive discussions. We invited 5 speakers to give 25 minutes presentations, followed by 5 minutes for clarification questions regarding the talk. We had two 15 minutes sessions to discuss more general topics that had arose during the talk. Finally, we held a wrap-up session to discuss goals for further research in the area of service composition.

CONCLUSION

The goal of this half-day workshop was to bring together researchers and practitioners to investigate state-of-the-art techniques, tools and methodologies to support the automatic service composition. The presentations and discussions identified challenges, developed ideas and presented approaches to provide a broader vision of service composition techniques. It also helped researchers to identify the open problems as well as to develop new approaches for addressing the problems.