Framework for Supporting an Interactive Web Service Composition

Mohamed Boukhebouze, Waldemar P. Ferreira Neto, Lim Erbin, and Philippe Thiran
PReCISE Research Center, University of Namur
Agenda

• Introduction

• UsiWSC Framework
  • User interaction expression
  • UI & WSC generation
  • UI & WSC deployment
  • UI & WSC coordination management

• Related work

• Conclusion & future work
Introduction
Introduction

• Web service composition
  • Orchestration or choreography of a set of Web services in order to implement a business process
  • Example: purchase order process scenario
Introduction

• Web service composition
  • Web service composition description languages
    • Defining the Web service composition aspects
      • Behaviour aspect (control flow)
      • Information aspect (data flow)
  • Several languages have been proposed
    • E.g. XLANG (Microsoft), WSFL (IBM) and BPEL (OASIS standard)
• BPEL is the de facto standard for the description of the service composition execution
Introduction

• BPEL standard

  • WS-BPEL Web Services Business Process Execution Language (BPEL for short)
  • BPEL is currently maintained by the OASIS
  • BPEL is an XML based language

```
<process>
  <partnerLinks> .. Process Partenaires.. </partnerLinks>
  <variables> ..Process Variable.. </variables>
  <faultHandlers> .. Activities to be executed when a fault occurs.. </faultHandlers>

[Activities]* example:
  <invoke> .. Web service invocation..</invoke>
  <assign> ..variable value assignment.. </assign>
  <sequence> .. Activities order..</sequence>
  <if> ..condition expression..</if>
  <pick>.. .. Activities to be executed when a specific event occurs .. </pick>
</process>
```
Introduction

- User interaction expression challenge
  - BPEL is not able to express the user interactions at runtime
  - Illustration: purchase order process scenario

```
Data input

Data selection

Data output

Data selection

Data output

Cancel the process
```
Introduction

- User interaction expression challenge
  - BPEL meta-model should be extended to support the user interaction types in runtime
    - Data input interaction
    - Data output interaction
    - Data selection interaction
    - User event interaction
  - User interface should be generated for the Web service composition
    - The user interface should be adapted to the user context
UsiWSC Framework
UsiWSC Framework

- UsiWSC Framework keys features
  - User interaction expression
  - UI & WSC generation
    - Adapted UI & WSC
    - Compliant with BPEL & UsiXML
  - UI & WSC deployment
  - UI & WSC coordination management
UsiWSC Framework

- UsiWSC Framework Architecture
UsiWSC Framework

• UsiWSC Framework keys features
  • User interaction expression
  • UI & WSC generation
    • Adapted UI & WSC
    • Compliant with BPEL & UsiXML
  • UI & WSC deployment
  • UI & WSC coordination management
UsiWSC Framework

- User interaction expression
  - Using a BPEL extension: UI-BP
    - New BPEL activities
      - DataInputUI
      - DataOutputUI
      - DataSelectionUI
    - New type of BPEL event
      - OnUserEvent
    - New BPEL attributes
      - UserRole
UsiWSC Framework: Design-time

- User interaction expression
  - Illustration:

  Data input
  Data selection
  Data output
  Data selection
  Data output
  Data selection
  Data output
  Cancel the process
User interaction expression
- UI-BPEL Designer
  - The designer is an Eclipse plug-in
  - The designer is built based on the Eclipse BPEL Designer
UsiWSC Framework

- UsiWSC Framework keys features
  - User interaction expression
  - UI & WSC generation
    - Adapted UI & WSC
    - Compliant with BPEL & UsiXML
  - UI & WSC deployment
  - UI & WSC coordination management
UsiWSC Framework

• User Interface generation
  • At deployment-time:
    • Abstract user interface (AUI) generation
    • Description of a UI independently to:
      • Any modality (e.g., graphical modality or vocal modality)
      • Any user context (user preference, user device or user environment)
    • The derivation rules transform each UI-BPEL user interaction into a set of abstract abstract compounds
    • The derivation rules are able to define a role-specific AUI
User Interface generation

- The AUI is described using the **UsiXML** language
  - Good expressiveness to describe UIs
  - Wide use in different research projects
  - Standardization action plan in the context of a European project
  - Availability of different tools including AUI and UI code generators
UsiWSC Framework

• User Interface generation
  • *At runtime:*
    • Final User Interface (FUI) generation (HTML)
      • Description of a UI specific to the user modalities and the user contexts
    • FUI are obtained from the AUI
      • Each abstract UI compound is transformed into a HTML widget based on the user context
UsiWSC Framework

• WSC generation
  • **At deployment-time:**
    • Abstract BPEL generation
      • Description of the composition control flow independently to any concrete Web service
      • Each user interaction is transformed into an invocation to a particular Web service: UI Manager
  • **At runtime:**
    • Executable BPEL generation
      • Selecting Web services that are involved in the composition based on the user context and a set of predefined selection rules (e.g. user experience, and user QoS preference)
      • Supported by any BPEL engine
UsiWSC Framework

- UsiWSC Framework keys features
  - User interaction expression
  - UI & WSC generation
    - Adapted UI & WSC
    - Compliant with BPEL & UsiXML
  - UI & WSC deployment
  - UI & WSC coordination management
UI & WSC deployment

- The UI & WSC deployment feature is provided as cloud computing services (Platform as a Services):
  - **UI Service**
    - The generated UI is automatically deployed as a HTML 5 code
  - **WSC Service**
    - The generated interactive WSC is automatically deployed as a specific BPEL process
    - Using Apache BPEL ODE to execute the composition
UsiWSC Framework

- UsiWSC Framework keys features
  - User interaction expression
  - UI & WSC generation
    - Adapted UI & WSC
    - Compliant with BPEL & UsiXML
  - UI & WSC deployment
  - UI & WSC coordination management
UI & WSC coordination management

- A UI manager routes data between the executable BPEL and its CUI
  - Using the mappings between the related UI-BPEL interactions and the CUI compounds
- Illustration: Data Input request Use Case
Related Work
Related Work

- **ActiveBPEL for People Approach**
  - Using the BPEL4People extension (OASIS)
    - Introducing a new type of BPEL activities for the specification of human tasks
  - Pro:
    - Expressing the data interactions
  - Con:
    - Do not deal with the generation of an adapted user interface for the Web service composition
    - Do not express the event interaction
Related Work

• BPEL4UI Approach
  • Using the BPEL4UI extension (Daniel et al., BPM 2010)
    • Introducing new the *Partner Link part Type*
  • Pro:
    • Defining a binding between BPEL activities and an existing user interface
  • Con:
    • This user interface is developed separately from the composition
    • Do not allow the generation of a user interface adapted to the user context
Related Work

- Comparison between the UsiWSC approach with others approaches
Conclusion & future work
Conclusion & future work

• Conclusion
  • UsiWSC Version 1.0
    • Expressing a user interaction: UI-BPEL designer
    • Generating an adapted UI to the user device
    • PaaS for deploying UI & WSC
    • Managing the coordination between the UI & WSC

• Future work
  • Generating an adapted UI to the user model and user environment
  • Generating an adapted WSC
    • Selecting Web services based on the user context
  • WSC store
    • Registry of the deployed interactive WSC
Web Site

http://webapps.fundp.ac.be/usiwsc/

Demo

http://www.youtube.com/watch?v=_py7E9zqqg4
Thank you for your attention
Appendix

- BPEL meta-model
Appendix

- UsiXML AUI meta-model