A Regression Testing Method for Service Composition

Speaker: Huiqun Yu
Institute: East China University of Science & Technology
E-mail: yhq@ecust.edu.cn

September 28, 2011
Agenda

1. Motivation
2. Service composition modeling
3. Regression testing
4. A case study
5. Conclusion
Motivation

- **Web services**
  - Enable users to access business functionalities.
  - Enable streamlining and enhancement – heterogeneous enterprise application integration.

*E.g.* To sell a used car, a car dealer may need more than one Web service:

1. Car valuation service
2. Car auction service
3. Online financial service
Motivation (Con’d)

- **Service composition**
  - Individual components are implemented at different places.
  - Services execute in different contexts.
  - But, they need to communicate – to yield the desired behaviors.

- **Problem statement**
  - How to adapt service application to changes?
  - How to ensure adequacy of application testing?

- **The main idea:**
  - Aspect-orientation for better separation of concerns
  - Formalism of Petri nets for modeling and analysis of service composition
Aspect Orientation

● Aspect Oriented Programming (AOP)
  – Tyranny of the dominant decomposition
    • Impossible to modularize some concerns (e.g. logging) using current software engineering methodologies such as OO.
    • As a result code gets scattered all over the system

● AOP Concepts
  – Modularize these concerns in Aspects
  – An aspect defines a set of join points
  – Weavers to weave aspect logic in the core application
Formal definition: a Petri net (PN) is an algebraic structure \((P,T,F)\) composed of:

- Finite set of places, \(P = \{p_1, p_2, \ldots, p_n\}\)
- Finite set of transitions, \(T = \{t_1, t_2, \ldots, t_m\}\)
- A flow relation, \(F \subseteq (P \times T) \cup (P \times T)\)

- Allows to model events and states in a distributed system
- Cleanly captures sequentiality, concurrency and event based asynchronous control
The model structure of service composition is an 8-tuple: $\mathcal{E}=(C, WS, TC, RL, TW, RO, RN, TR)$, where:

- $C$, $WS$, $TC$: the sets of finite components, available services, and test suites;
- $RL$: the set of combinators, where $>$, $+$, $||$ represent sequence, choice and parallel;
- $TW$: the available service of component;
- $RO$: the interface and component binding before evolution;
- $RN$: is the evolution interface and binding;
- $TR$: is the action set of testing case.
The **BN model of component** $C_i$.
- Where the place $p_i^l$ and $p_o^o$ represent the input and output interface;
- $p_i^{in}$ and $p_i^{ou}$ represent the input and output interface in the top model.
Composition Model

- A hierarchical application model
  - Constructing $BN$ models of all components.
  - Combing $BN$ models according to their connecting relationship.
Aspect-oriented Elements

- **Introduction net**
  - Encapsulation of regression testing action.

- **Pointcut**
  - Interfaces of core and crosscutting concerns.

- **Advice**
  - Way to integration.

- **Aspect**
  - A pair \((\text{CutN}, \ \text{AD})\) is called where \text{CutN is a pointcut}, and \text{AD is an advice}. 
Aspect Model

Interface aspect

Binding aspect

Evolution notification aspect
Model Integration

According to the above model, the steps of aspect weaving are:

- Adding all the *advices* into composition net, thus forming a new composition net;
- Adding the *corresponding elements* according to the *weaving rules* and merging the common elements;
- Weaving *aspects* according to their priorities when more than one aspect is applied to the same pointcut.
Regression Testing

- Paths of regression testing $p$ reflecting model changes.
  - The path set $p_i$ affected by interface change
  - The path set $p_b$ affected by binding change
- An algorithm to generating regression testing paths.
  - Computing the set $p_i$: The system will compare the number of parameters in the interface, the definition of data type of each parameter in the interface, then do regression test in case any change occurs.
  - Computing the set $p_b$: According to the relationship between the components, if the binding of component $C_i$ has changed, then the associated components of $C_i$ should be retested to ensure that the binding is effective.
A Case Study

Components of a simplified Export Service:

- $C_1$: Destination checking
- $C_2$: Packaging service
- $C_3$: Commodity inspection
- $C_4$: Insurance processing
- $C_5$: Ship transport ordering
- $C_6$: Air transport ordering
- $C_7$: Customs checking
- $C_8$: Financial services

The composition process expression: $C_1 > C_2 > (C_3 || C_4 || (C_5 + C_6)) > C_7 > C_8$
The model is got by weaving binding changes and interface changes into composition net of Export Service.
Conclusion

- **Modeling regression testing method for service composition**
  - Separation of concerns: to reduce complexity and enhance flexibility
  - Rigorousness: to assure correctness

- **Further topics**
  - Performance, dependability, etc.
  - Tool support
Thank You!