

# Dynamic Web Service Composition: eBusiness Just for You!

Steffen Higel  
Steffen.Higel@cs.tcd.ie

## Overview

- Quick introduction to Service Composition.
- The influence of workflow.
- Pseudo-Runtime Automatic Service Composition.
  - A.I. Planning
  - Semantic Standards
  - Adaptivity
- The Pattern Based Approach
  - Why is it good?
  - What can it do for us?
  - How do we maintain adaptivity?
- The Big Picture (M-Zones)

## Quick Introduction to Service Composition

- World Wide Web Consortium:
  - *“The World Wide Web is more and more used for application to application communication. The programmatic interfaces made available are referred to as Web services.”*
- *Web services are small pieces of software that use rigidly defined methods for self description and inter-communication.*
- *What if we develop larger pieces of software by feeding the outputs of one web service into the inputs of another?*

## The Influence of Workflow

- Joosten, 1994:
  - “The set of relationships between all the activities in a project, from start to finish. Activities are related by different types of trigger relation. Activities may be triggered by external events or by other activities.”*
- *Obvious parallels with Web-Service composition.*
- *Web-Service Composition languages like BPEL4WS are heavily influenced by workflow.*

## Automatic Service Composition

- Imagine that software automatically generates a suitable string of services to achieve a user task.
- The concepts in this are much the same as those in A.I. planning.
  - We need to make a suitable set of changes to the state of the world to leave it in a way that the user deems acceptable.
  - In reality, the task is extremely difficult to do reliably without strong and detailed descriptions of each individual service.

## Automatic Service Composition (continued)

- Ontology driven service description using OWL/DAML provides the way forward.
  - not too pleasant to work with, lacking sufficiently robust and simple editing tools.
  - However, very expressive and readily understood by computer software.
- How does this help?
  - Software can readily reason on whether the outputs of one service match up to the inputs of another.
- Adaptivity in Web Service Composition
  - Structural vs. Material adaptivity.

## The Pattern Driven Approach

- Automatically generating compositions can be trouble prone!
- While a composition is being generated, it would be great to bridge the gaps using previously used, verified compositions.
- We could save a lot of time and money by providing a framework which helps to achieve this.
- Reuse should improve the quality of the compositions and the speed at which they are generated.

## The Pattern Driven Approach (continued)

- Typically, reuse of knowledge can result in a loss of adaptivity.
- How do we store and use the patterns such that we minimise this loss?
  - refer to general service functionality rather than specific services (helps material adaptivity).
  - never assume that an appropriate pattern match is going to be perfect. We shouldn't be afraid of a bit of hard work (helps structural adaptivity).
- This should allow an appropriate balance (as our experimentation will hopefully verify)

## Worked Example

- I'd like to buy an electronic version of a book explaining how exactly to configure my expensive new Cisco router.
- The composition engine understands that I'm me and that I'd like to alter the world in such a way that I have a copy of the book on my laptop. When formulating the request, I made it clear that I don't mind paying.
- What happens next?

## Worked Example (continued)

- Composition engine starts a breadth first search of all book recommendation services.
- From there, it searches for services which sell electronic books.
- I want the book sent to me in a zipped, DRM-free PDF file.
- The composition engine notices that it has a pattern for converting from eBook to a plain old PDF. It adds this to the composition.
- It finds a service for zipping and sending files by email which it then invokes.

## The Big Picture (M-Zones)

- Although eBusiness a valid use of automatic service composition, UbiComp also has much to gain.
- Management of UbiComp environments using web service composition is a tidy way of modelling many problems in that area.
- Integration with concurrent research into Context Aware computing, Quality of Service and Policy.
- Formulation of initial requests non-trivial, so integration with HCI research very important.