

An Approach to Developing Web Services with Aspect-oriented Component Engineering



John Grundy^{1,3}

Thomas Panas²

Santokh Singh¹

Hermann Stöckle¹

¹Department of Computer Science and

**³Department of Electrical and
Computer Engineering,**

University of Auckland, New Zealand

²Department of Computer Science

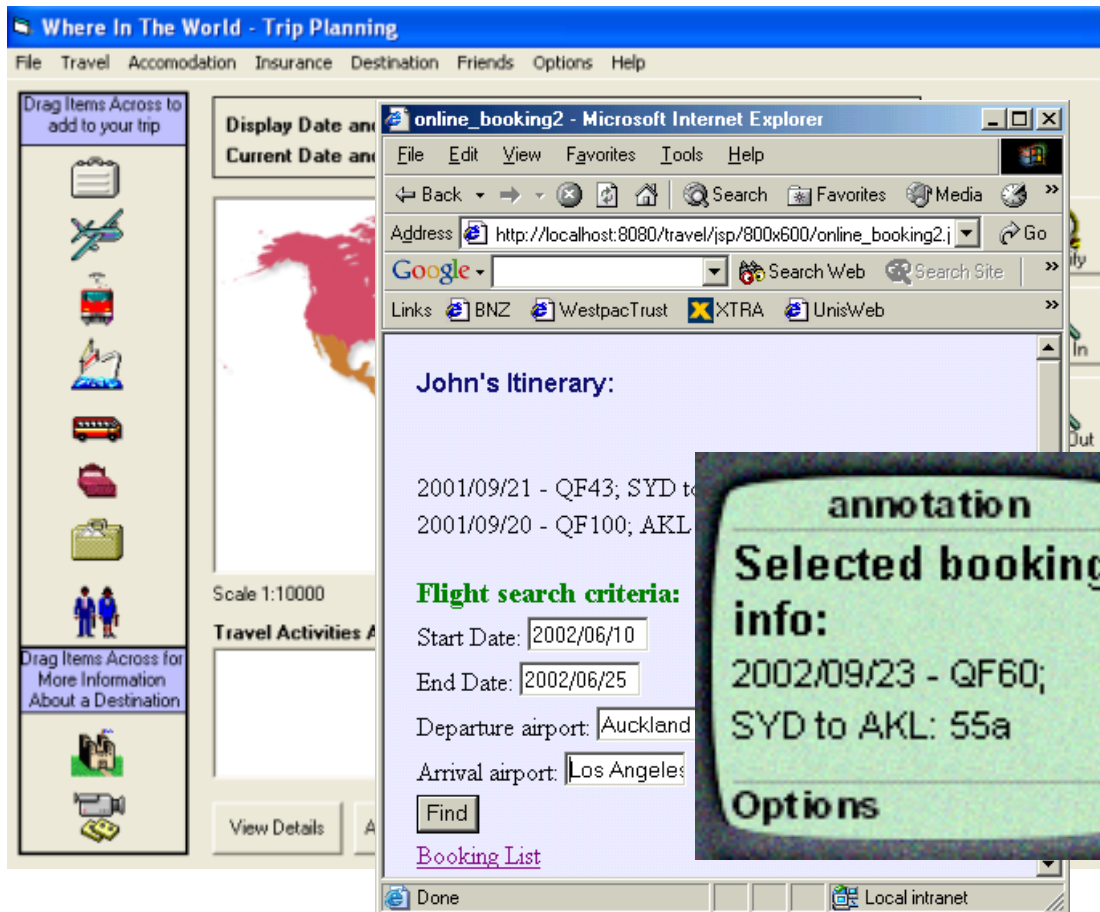
Växjö University, Sweden

Outline



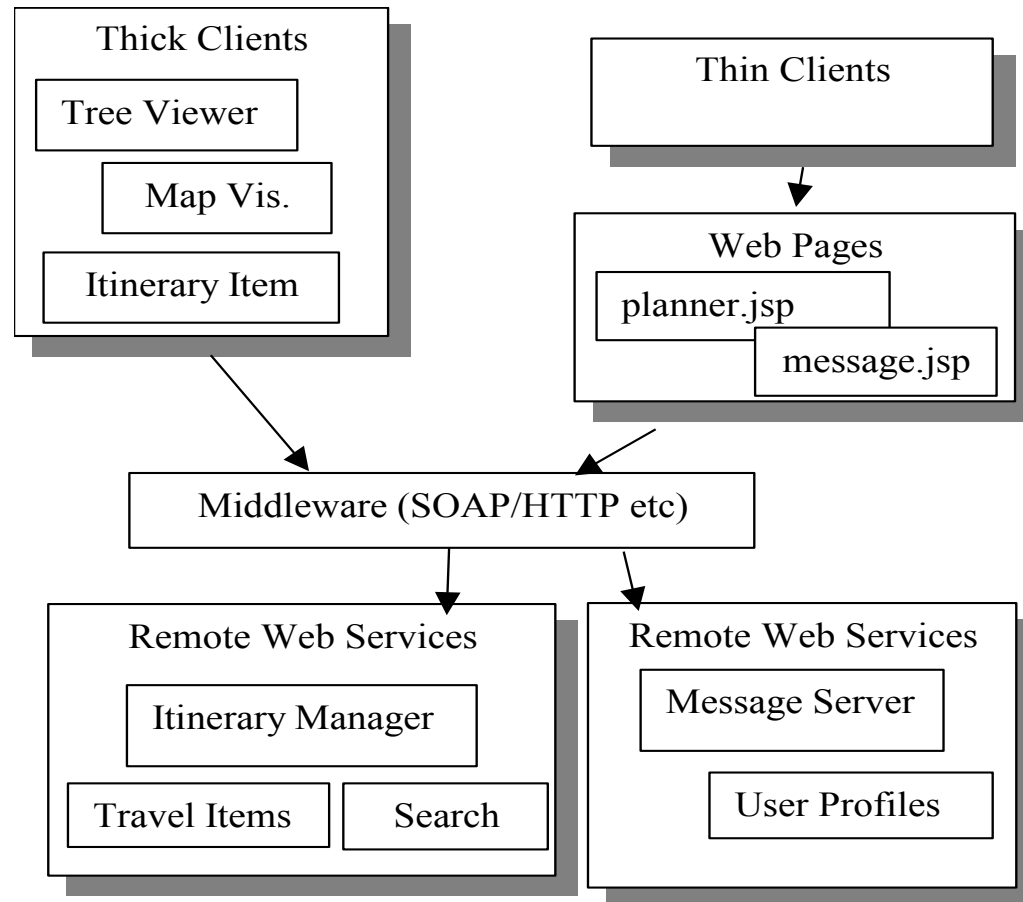
- Motivation
- Aspect-oriented Component Engineering
- AOCE – Designs
- AOCE – WSDL Extensions
- AOCE – UDDI Extensions
- Current and Future Work
- Conclusions

Motivation: Example



- ❑ Travel planner application
- ❑ Multiple views, different devices
- ❑ Multiple users
- ❑ Multiple web services to interact with...

Example Architecture



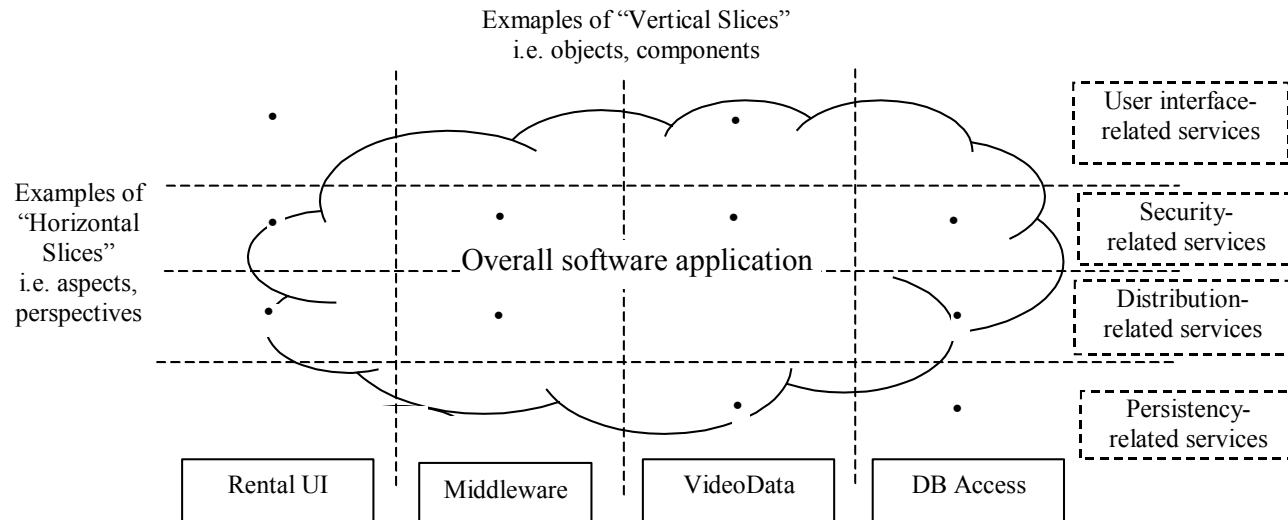
- Clients = thick-client (VB) planner; thin-client web pages and mobile devices
- Services = cars, hotels, flights, booking, payment etc

Aspect-Oriented Component Engineering



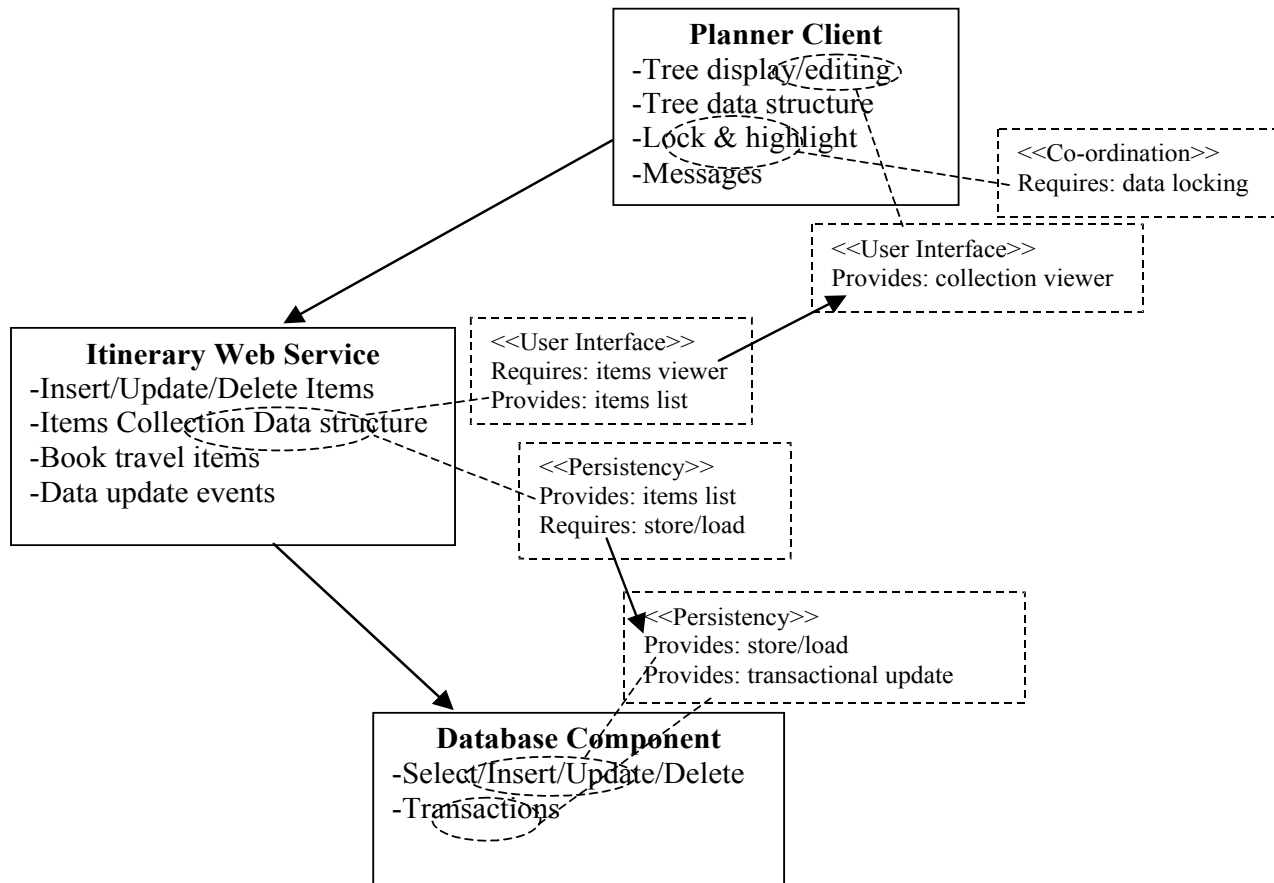
- Issues when engineering software components (including web service-based components):
 - How to identify components vs objects?
 - How to compose components?
 - How to make “reusable”, “tailorable”, “adaptable”?
 - How to reason about composed systems (statically and dynamically)
 - Reliability, trustability, performance etc issues
 - Plus all the usual: impl meets design meets spec etc

Concept of Aspects



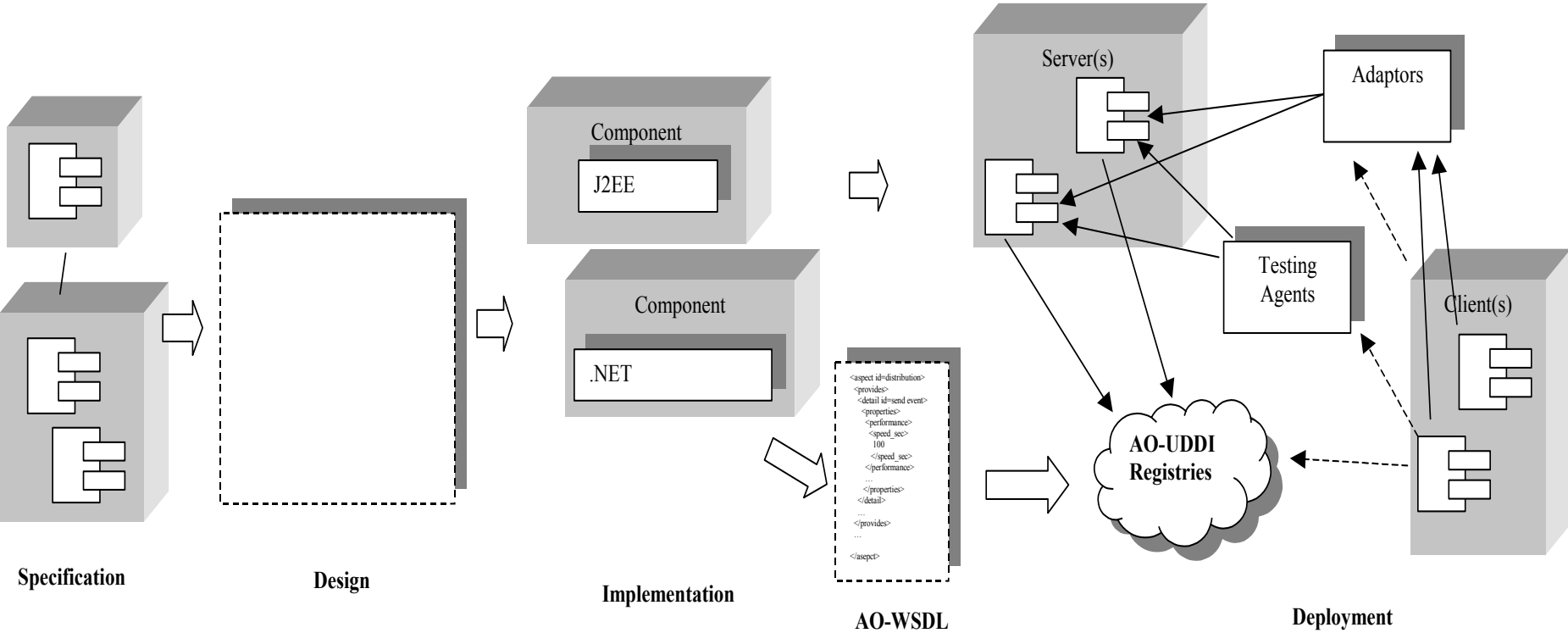
- ❑ Functional decomposition - normal approach
- ❑ Alternatives: parts of system contributing to "systemic" properties e.g. user interfaces, distribution, transactions, security, resource usage, persistency etc
- ❑ Systemic properties of system get spread...

AOCE Design

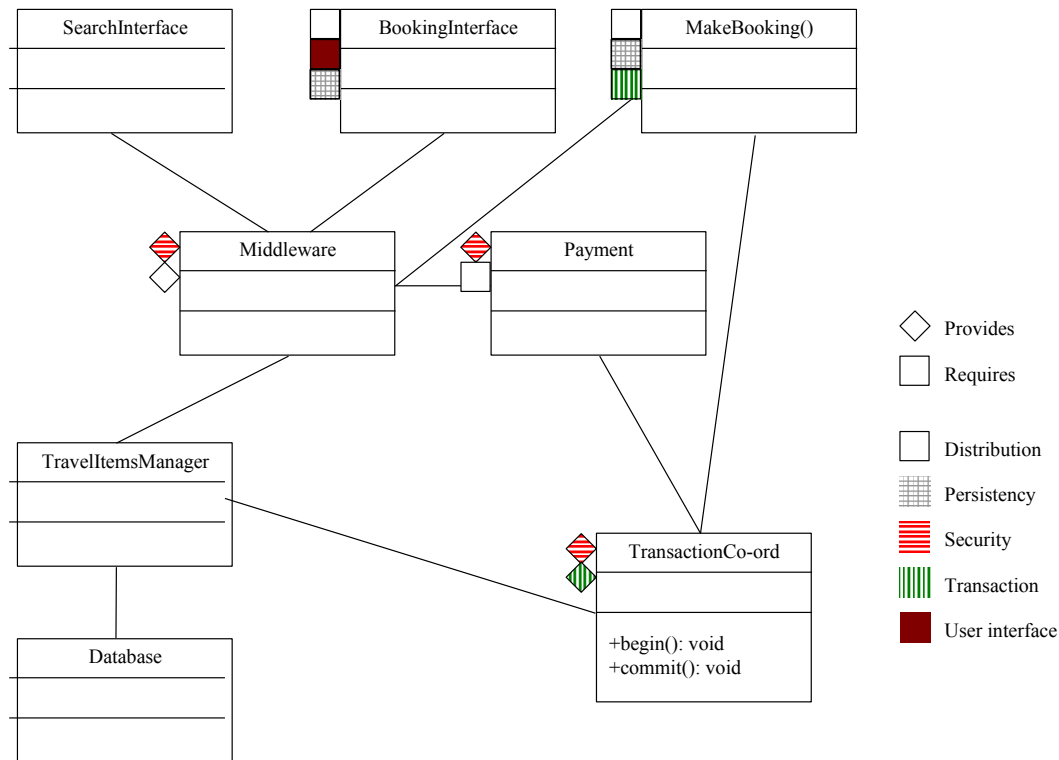


- Each service has various “aspects” that can be used to characterise it
- Examples are UI, co-ord, persistency, security etc

Our Approach to WS Development

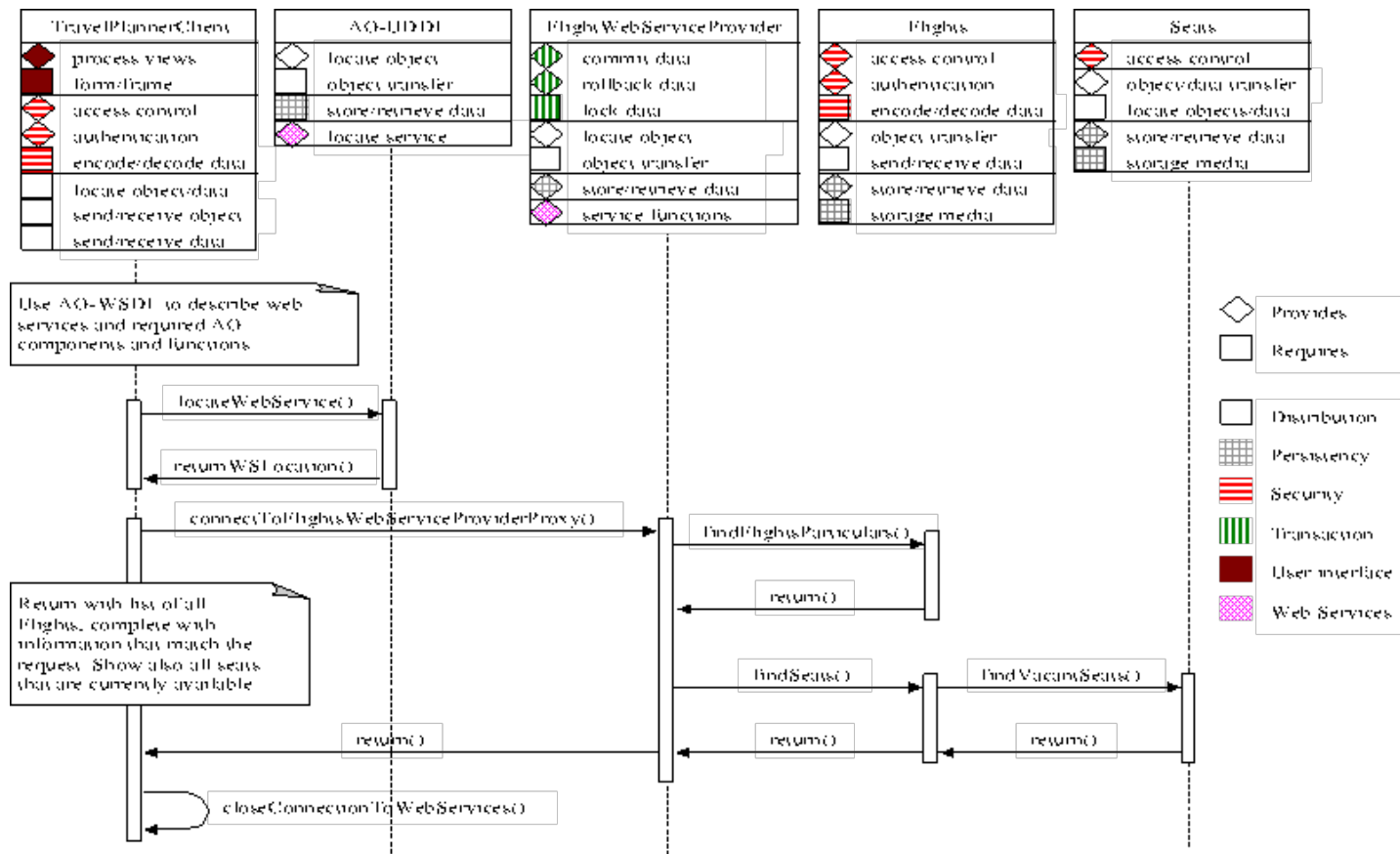


AOCE WS Component Design



- Represent WS components & relationships
- Annotate with aspect indications
- Provides vs requires
- Can also add to individual WS operations

AOWS-Sequence Diagram



AO-WSDL

```
<component name="Itinerary Management">
  <services name="" /> <!-- no web services implementing this component />
  <components name="" />
  <property name="caching">
    <value type="boolean" />
    <getter operation="getCaching" />
    <setter operation="setCaching" />
  </property>
  ...
  <operation name="findItinerary" style="rpc">
    <arg name="ID" style="in" type="LongInt" />
    <arg name="itinerary" style="out" type="itinerary:ItineraryData" />
  </operation>
  ...
  <aspects namespaces="www.travelplanner.com/aspects/namespaces/itinerary" >
    <aspect name="ItineraryData"
      detail="itinerary:ItineraryDataManagement" type="provided" >
      <impacts operations="all" />
    </aspect>
    <aspect name="Persistency"
      detail="common:DataManager" type="required" >
      <impacts operations="findItinerary!addItinerary!..." />
      <property name="Performance"
        type="common:OperationSpeed">
          <common:lessThan units="ms">100</lessThan>
        </property>
    </aspect>
    <aspect name="TransactionSupport"
      detail="common:TransactionsRequired" type="required" >
      <impacts operations="findItinerary!addItinerary!..." />
      <property name="TransactionScope"
        type="common: TransactionDemarcation">
          <common:transactionState>IN_TRANS</transactionState>
        </property>
    </aspect>
    <aspect name="BookingManager"
      detail="booking:TravelBookingManager" type="required" >
      <impacts operations="addItinerary!updateItinerary!..." />
      <property name="BookingCommittalApproach"
        type="booking:BookingCommittal">
          <booking: BookingCommittal value="BTP" />
        </property>
      <property name="Timeout" type="booking:Timeout" >
          <booking:Timeout units=days>
            <max>5</max></booking:Timeout>
        </property>
    </aspect>
  ...
</component>
```

□ WS component characterisation


□ Low-level aspects

□ Medium-level aspects

□ High-level aspects

□ Use in WS-UDDI to help locate, adapt...

Current/Future work; Conclusions



- ❑ AO-WSDL, AO-UDDI development
- ❑ Re-designing and re-implementing C#.NET travel planner using AO-WS approach
- ❑ Investigating automated AO-WS component discovery, testing, integration/adaptation

- ❑ Key idea is to provide/use more information about WS components via an aspect-based ontology
- ❑ AO-WS extension of AO component engineering
- ❑ Using AO-WS to develop .NET WS components

References



- ❑ Grundy, J.C., Panas, T., Singh, S., Stoeckle, H. An Approach to Developing Web Services with Aspect-oriented Component Engineering, In Proceedings of the 2nd Nordic Conference on Web Services, 2003.
- ❑ Wang, Y., Singh, S., Hosking, J.G. and Grundy, J.C. An Aspect-Oriented UML Tool for Software Development with Early Aspects, Proceedings of ICSE 2006 Workshop on Early Aspects at ICSE: Aspect-Oriented Requirements Engineering and Architecture Design, Shanghai, May 2006.
- ❑ Singh, S. Chen, H.C. Hunter, O., Grundy, J.C. and Hosking, J.G. Improving Agile Software Development using eXtreme AOCE and Aspect-Oriented CVS, in Proceedings of the 12th Asia-Pacific Software Engineering Conference, Taiwan, December 2005, IEEE CS Press.
- ❑ Singh, S., Grundy, J.C., Hosking, J.G. and Sun, J. An Architecture for Developing Aspect-Oriented Web Services, In Proceedings of the 2005 European Conference on Web Services, Vaxjo, Sweden, Nov 14-16 2005, IEEE Press.
- ❑ Singh, S., Hosking, J.G. and Grundy, J.C. Deploying Multi-Agents for Intelligent Aspect-Oriented Web Services, In Proceedings of the 2005 Pacific Rim Workshop on Intelligent Multi-agents, Kuala Lumpur, 14-16 September 2005, Lecture Notes in Artificial Intelligence, Springer.
- ❑ Grundy, J.C. and Hosking, J.G. Developing Software Components with Aspects: Some Issues and Experiences, Chapter 25 in Aspect-Oriented Software Development, Prentice-Hall, October 2004, pp. 585-604.
- ❑ Singh, S., Grundy, J.C., Hosking, J.G. Developing .NET Web Service-based Applications with Aspect-Oriented Component Engineering , In Proceedings of the Fifth Australasian Workshop on Software and Systems Architectures, Melbourne, Australia, 13-14 April 2004.
- ❑ Grundy, J.C. Aspect-oriented Requirements Engineering for Component-based Software Systems, 1999 IEEE Symposium on Requirements Engineering, Limerick, Ireland, 7-11 June, 1999, IEEE CS Press.