



Separating Per-client and Pan-client Views in Service Specification

The 2006 International Workshop on
Service Oriented Software Engineering
(IW-SOSE '06)

May 27-28, 2006, Shanghai - China

Outline



- Introduction
- Service Specification
- The Manager Pattern
- Per- and Pan-Client Interfaces
- Conclusion

Web Shop Example



- multiple customers
- each customer can have multiple orders

➤ Interface description

SessionedOrderManager
<pre>createSession return String deleteSession (String sessionID) sessionAuthenticate (String info) createOrder return String deleteOrder(String orderID) orderDefineCustomer (String info, String orderID, String sessionID) orderDefinePayment (String info, String orderID , String sessionID) orderAddItem (String name, String orderID , String sessionID) orderDeleteItem (String name, String orderID , String sessionID) orderCalculateTotal (String orderID , String sessionID) return Euro orderCheckPayment (String orderID , String sessionID) return Boolean orderPlace (String orderID , String sessionID)</pre>

- service description is central to SOA
 - how to find optimal service specification?

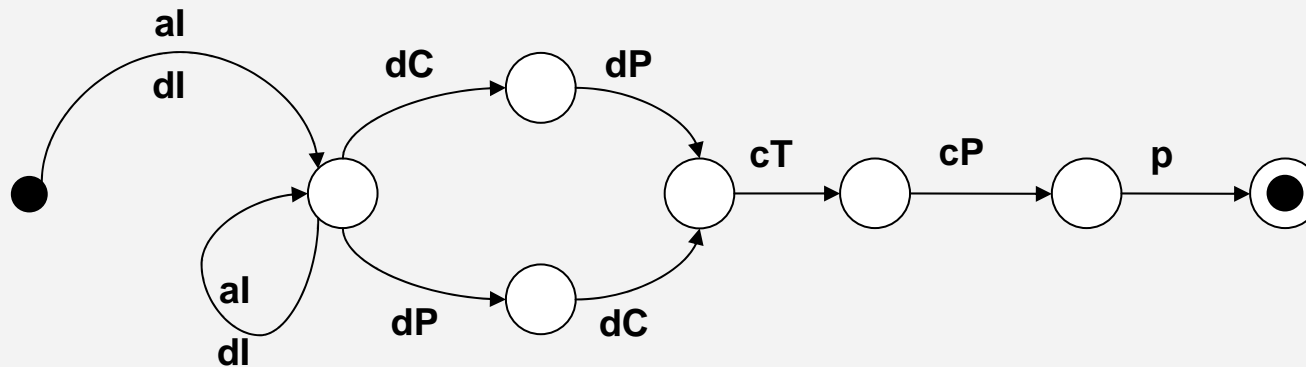
- current service description, e.g. WSDL
 - only procedures
 - low-level interface with all operations
 - not always optimal abstraction for user
 - context independent

- other standards for additional meta information
 - on top of WSDL

Protocol of Web Shop



$(\text{addItem} \mid \text{deleteItem})^+ .$
 $((\text{defineCustomer} . \text{definePayment}) \mid$
 $(\text{definePayment} . \text{defineCustomer})) .$
 $\text{calculateTotal} . \text{checkPayment} . \text{place}$



Protocol specification in WS-CDL



- Example: Specify the protocol for every customer in a Choreography Description Language (CDL), e.g. WS-CDL
- a channel represents a connection between one customer and one service provider
- uniquely identify order: attach token (orderID) to channel
- multiple orders and sessions at the same time: cannot be cleanly handled by WS-CDL

Taking id into account in WS-CDL



```
<package xmi:version="2.0" xmlns:om="http://www.example.com/order
  name="orderCDL" version="1.0" targetNamespace="..." ...>
```

...

```
<informationType name="addItemMessageType" typeName="om:addItemMessageType"/>
```

```
<informationType name="StringType" typeName="xsd:string"/>
```

```
<informationType name="EuroType" typeName="om:Euro"/>
```

```
<informationType name="addItemMessageType"
  typeName="om:addItemMessageType"/>
```

```
<informationType name="removeItemMessageType"
  typeName="om:removeItemMessageType"/>
```

```
<informationType name="paymentInfoType"
  typeName="om:paymentInfoType"/>
```

```
<informationType name="customerInfoType"
  typeName="om:customerInfoType"/>
```

```
<informationType name="paymentOkInfoType"
  typeName="om:paymentOkInfoType"/>
```

```
<informationType name="TotalType" typeName="om:TotalType"/>
```

...

Taking id into account in WS-CDL



...

```
<token name="orderID" informationType="StringType">
```

...

```
<tokenLocator token="orderID" informationType= "addItemMessageType"
  query="/AM/orderID"/>
<tokenLocator token="orderID" informationType= "removeItemMessageType"
  query="/RM/orderID"/>
<tokenLocator token="orderID" informationType= "paymentInfoType"
  query="/PI/orderID"/>
<tokenLocator token="orderID" informationType= "customerInfoType"
  query="/CI/orderID"/>
<tokenLocator token="orderID" informationType= "paymentOkInfoType"
  query="/PO/orderID"/>
<tokenLocator token="orderID" informationType= "TotalType"
  query="/TT/orderID"/>
```

```
<channelType name="customerOrderChannelType" referenceToken="customerRef" >
  <identities description="orderID" tokens="orderID"/>
</channelType>
```


Taking id into account in WS-CDL

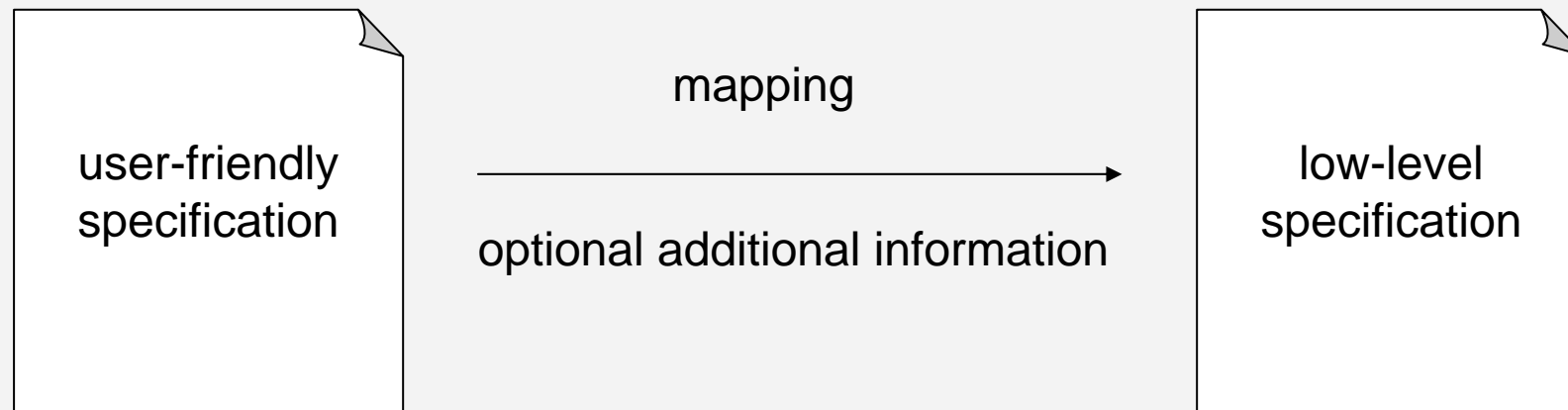


```
...
<channelType name="customerOrderChannelType" referenceToken="customerRef" >
  <identities description="orderID" tokens="orderID"/>
</channelType>
...
<choreography name="orderChoreo" root="true">
...
<activity type="Interaction" name="addItemActivity"
  operation="addItem" channelVariable="customerOrderChannelType" ...>
  <exchangeDetail name="addItem" type="addItemMessageType"
    action="Request"/>
</activity>
...
```

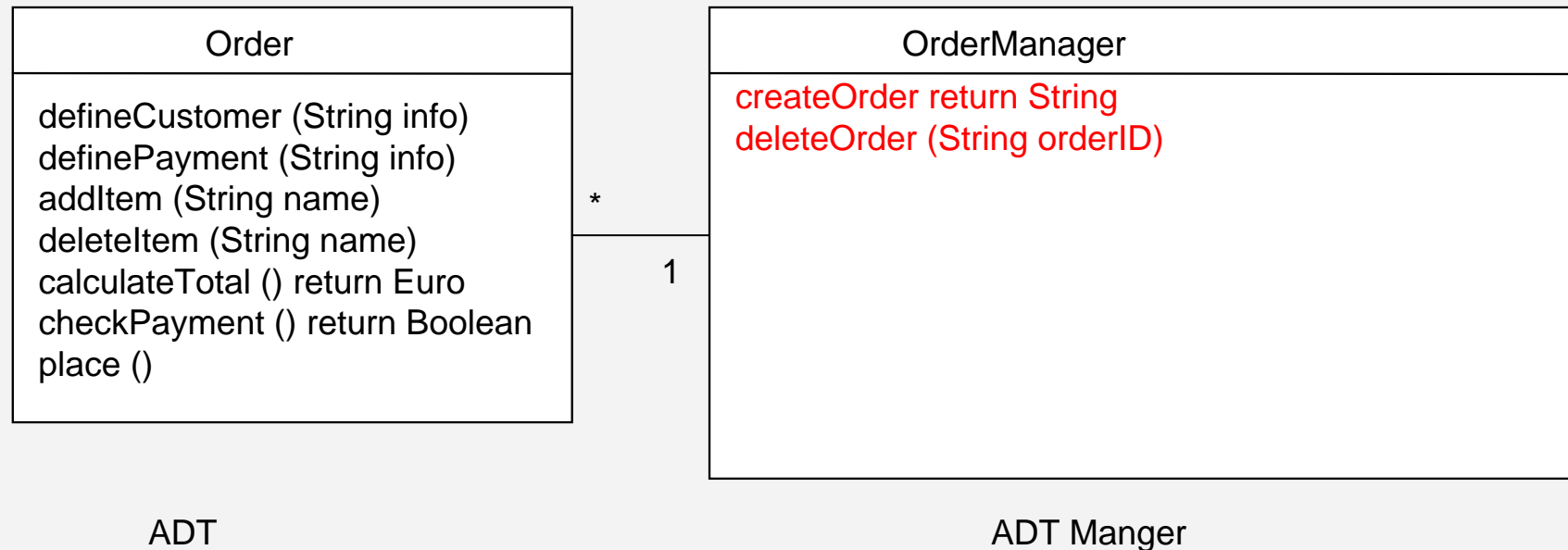
Different specification levels



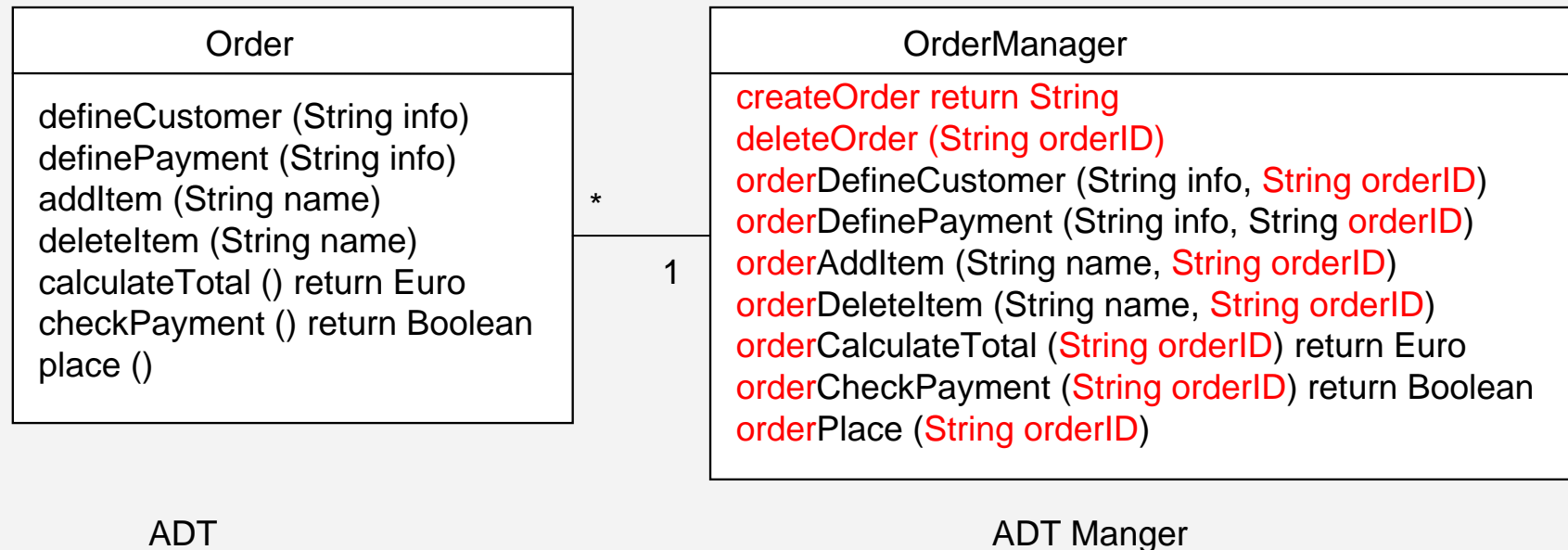
- goal: simplify specification
- not necessary to explicitly define low-level procedural interface – should be derived
- specify a service using different abstractions



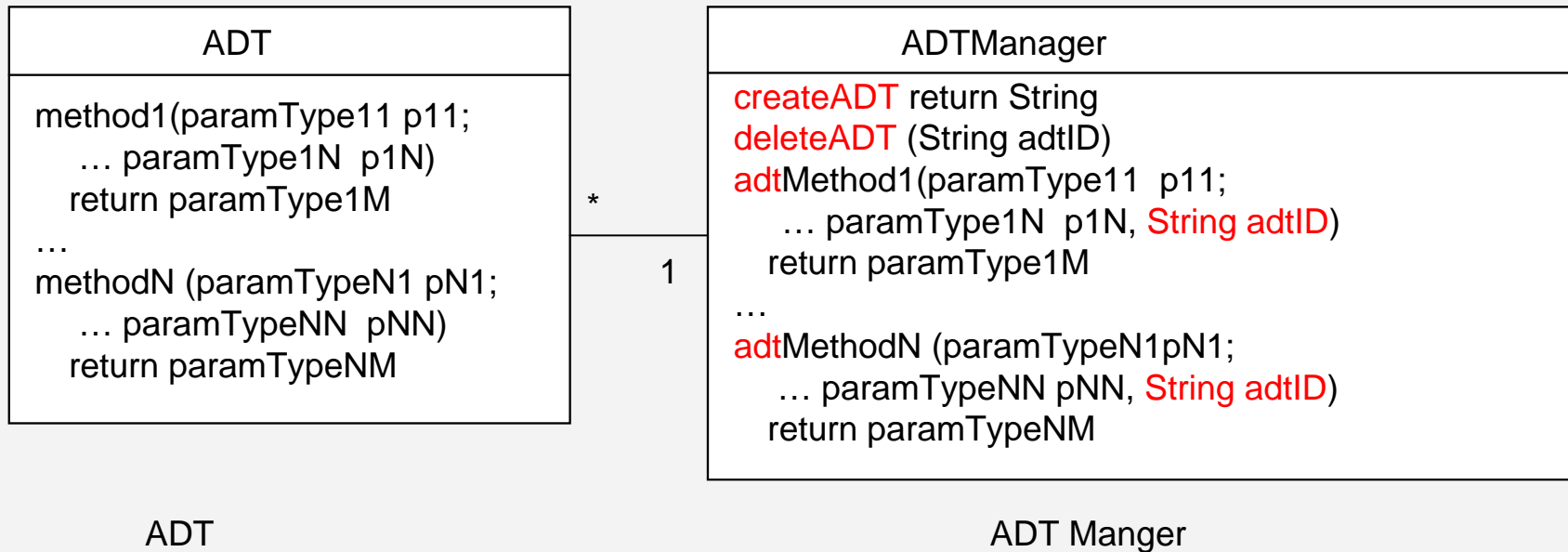
Manager Pattern example



Manager Pattern example



Manager Pattern structure



Manager pattern concepts

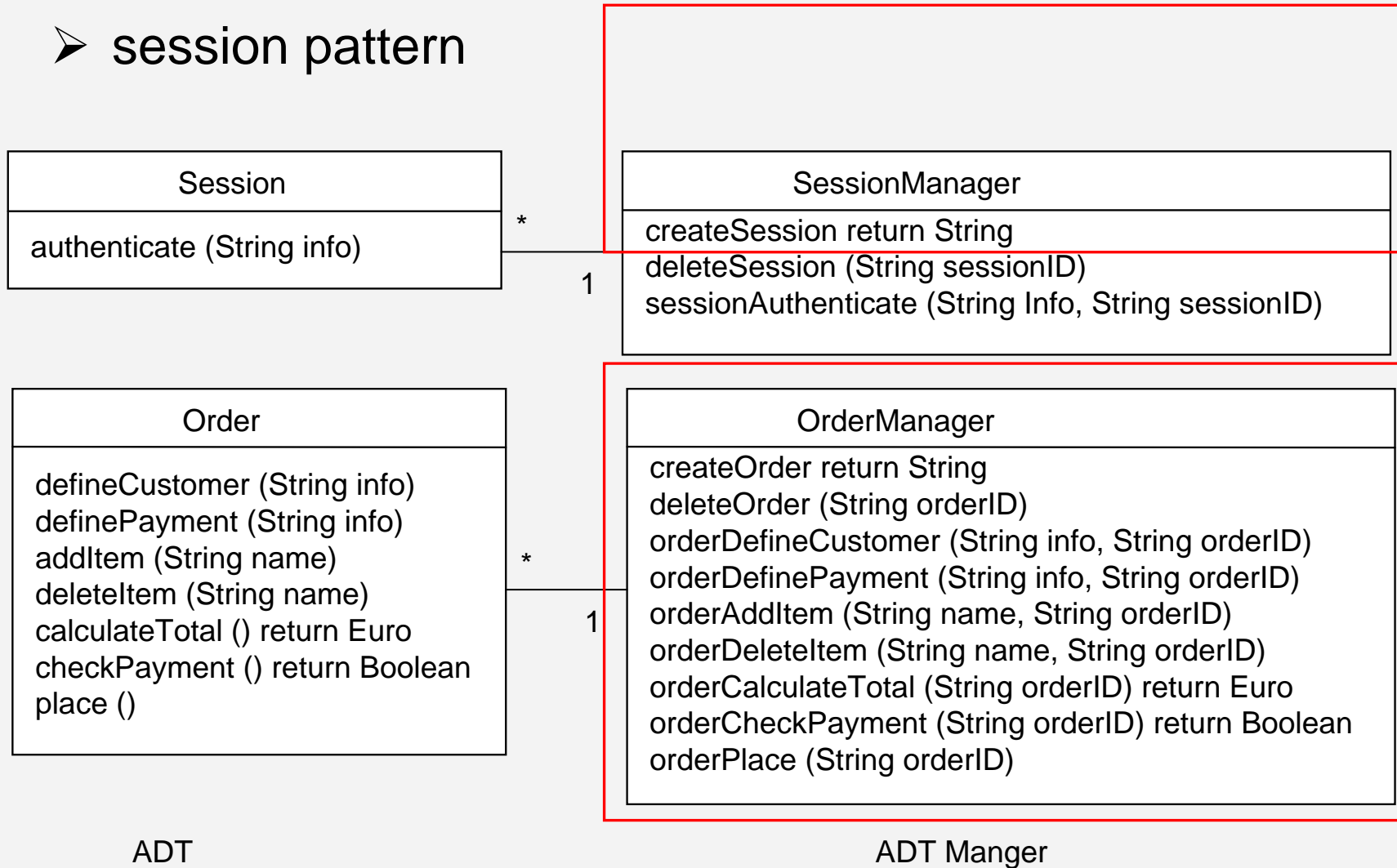


- add create and delete operation
- systematic signature change
- additional changes are possible, as long as mapping is well defined
- manager can be derived automatically from ADT

Composition of managed objects



➤ session pattern



Composition of managed objects



- global view of a service
 - naive way: union(SessionManager, OrderManager)

SessionedOrderManager
createSession return String deleteSession (String sessionID) sessionAuthenticate (String info, String sessionID)
createOrder return String deleteOrder (String OrderID) orderDefineCustomer (String info, String orderID) orderDefinePayment (String info, orderID) orderAddItem (String name, String orderID) orderDeleteItem (String name, String orderID) orderCalculateTotal (String orderID) return Euro orderCheckPayment (String orderID) return Boolean orderPlace (String orderID)

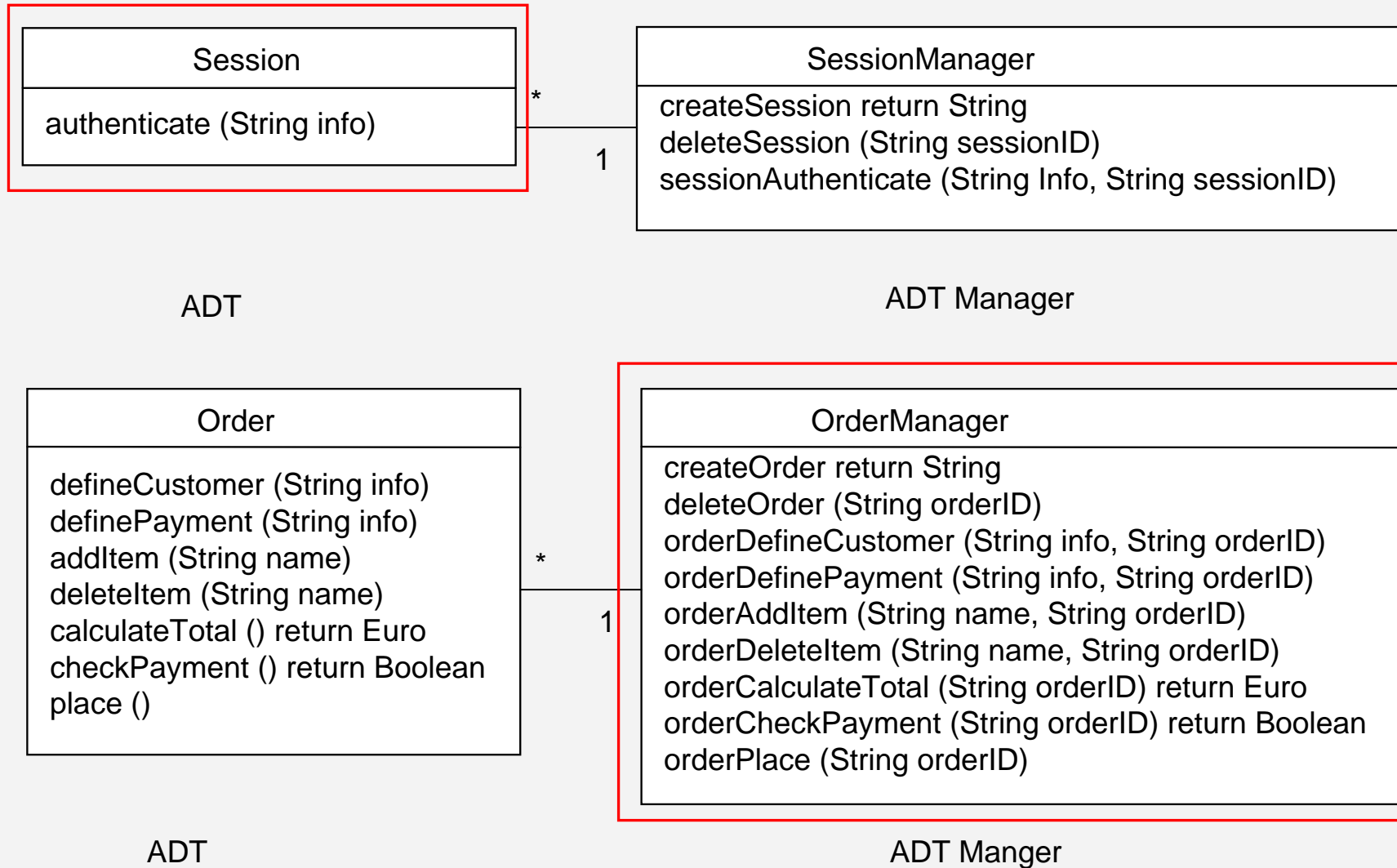
Composition of managed objects



- global view of a service: pan-client view
 - superimpose(SessionManager, OrderManager)

SessionedOrderManager
createSession return String deleteSession (String sessionID) sessionAuthenticate (String info, String sessionID)
createOrder(String sessionID) return String deleteOrder(String orderID, String sessionID) orderDefineCustomer (String info, String orderID, String sessionID) orderDefinePayment (String info, String orderID , String sessionID) orderAddItem (String name, String orderID , String sessionID) orderDeleteItem (String name, String orderID , String sessionID) orderCalculateTotal (String orderID , String sessionID) return Euro orderCheckPayment (String orderID , String sessionID) return Boolean orderPlace (String orderID , String sessionID)

Composition of managed objects



Per- and Pan-Client Interfaces

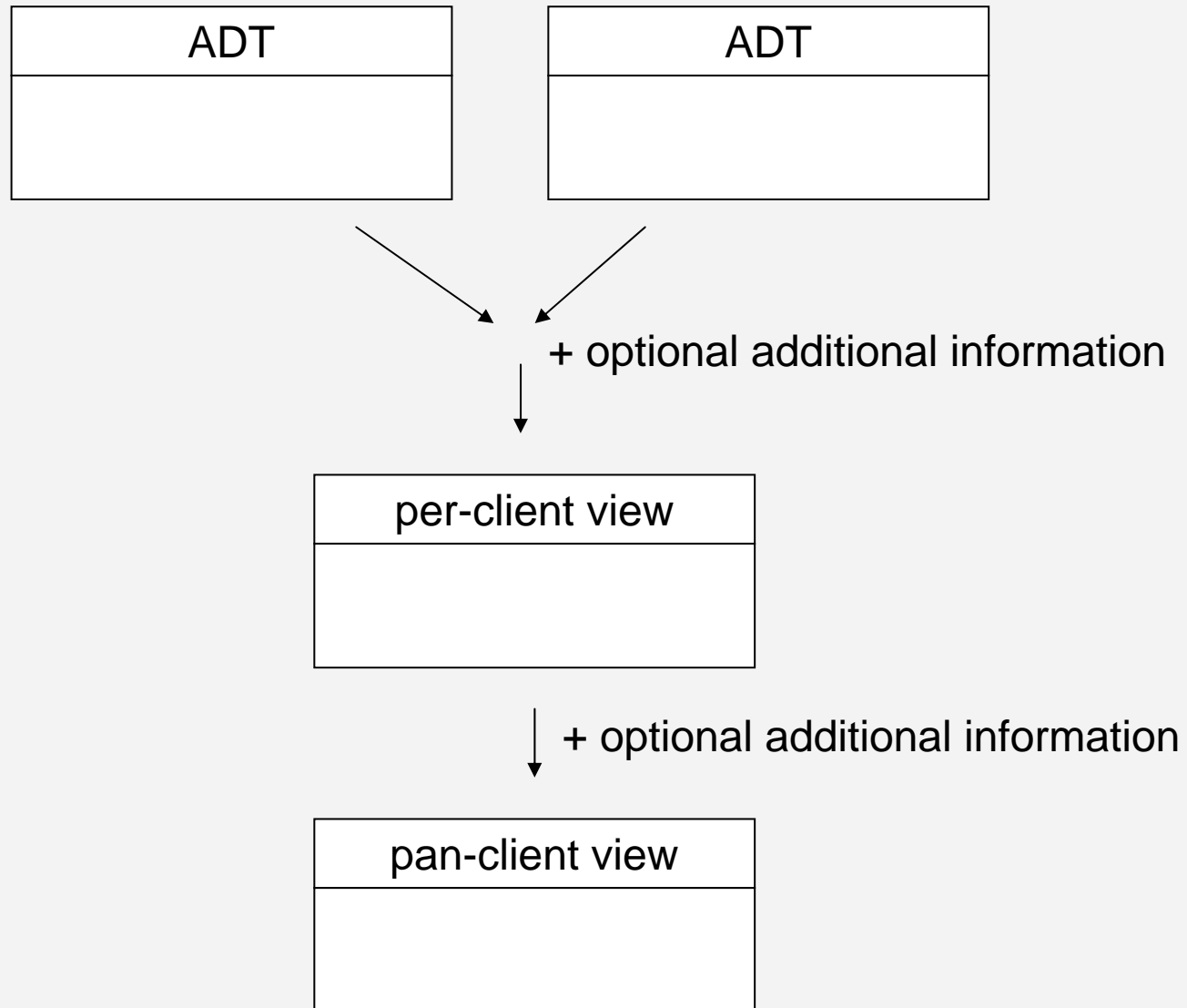


- point of view of a client: per-client view
 - union(Session, OrderManager)

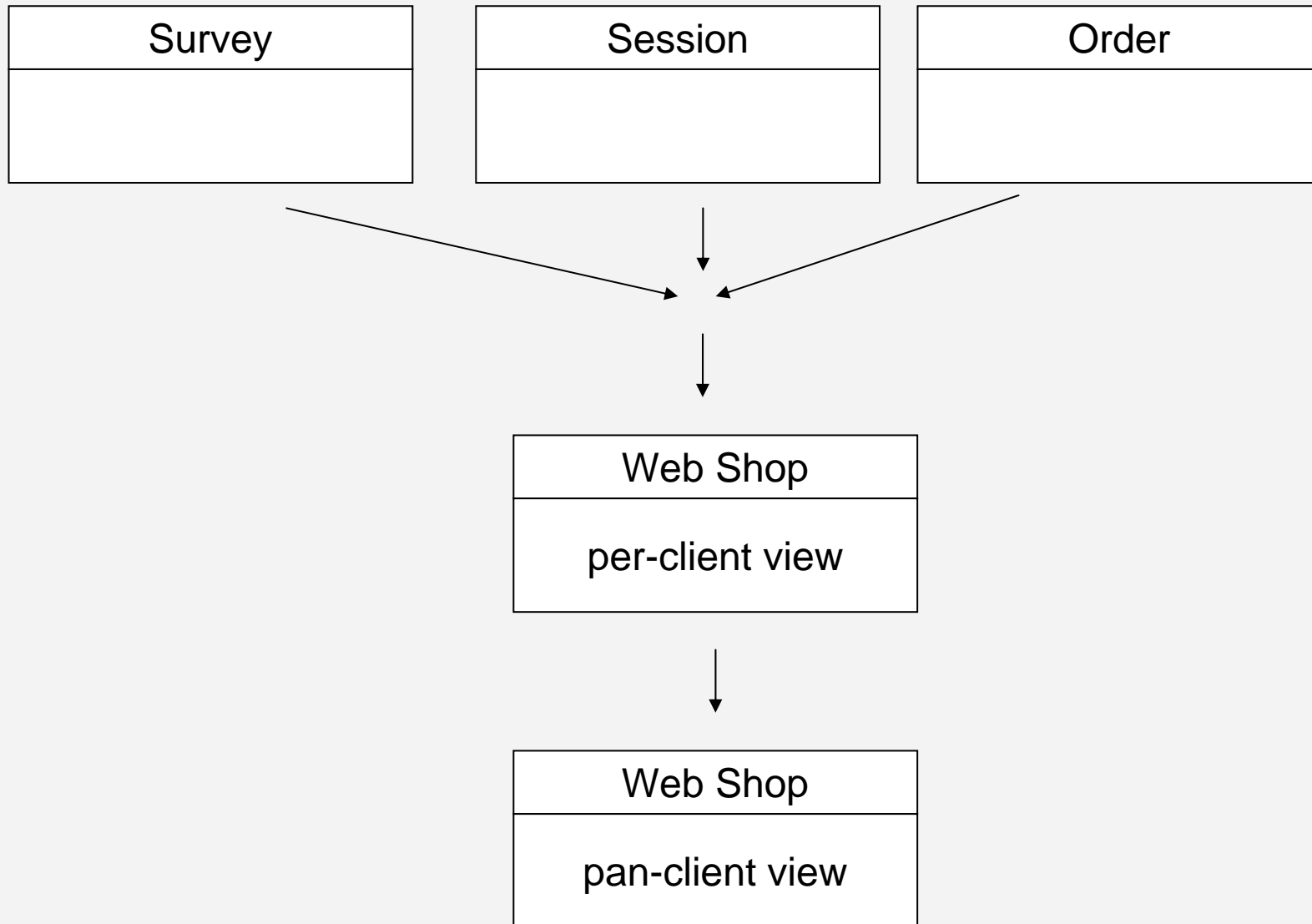
SessionedOrderManager
authenticate (String info) createOrder () return String deleteOrder(String orderID) orderDefineCustomer (String info, String orderID) orderDefinePayment (String info, orderID) orderAddItem (String name, String orderID) orderDeleteItem (String name, String orderID) orderCalculateTotal (String orderID) return Euro orderCheckPayment (String orderID) return Boolean orderPlace (String orderID)

per-client view

Combining multiple ADTs



Combining multiple ADTs



Combining multiple ADTs



Survey
getTopicList () return String selectTopic (String topic) getQuestions () return String answerQuestions (String answers)
<i>getTopicList+ . selectTopic+ getQuestions . answerQuestions</i>

Order
defineCustomer (String info) definePayment (String info) addItem (String name) deleteItem (String name) calculateTotal () return Euro checkPayment () return Boolean place ()
<i>(addItem deleteItem)+ . ((defineCustomer . definePayment) (definePayment . defineCustomer)) . calculateTotal . checkPayment . place</i>

user view - managed objects

Combining multiple ADTs



OrderCustomerManager
sessionAuthenticate (String info)
createOrder return String
deleteOrder(String orderID)
orderDefineCustomer (String Info, String orderID)
orderDefinePayment (String info, String orderID)
orderAddItem (String name, String orderID)
orderDeleteItem (String name, String orderID)
orderCalculateTotal (String orderID) return Euro
orderCheckPayment (String orderID) return Boolean
orderPlace(String orderID)
sGetTopicList (String surveyID) return String
sSelectTopic (String topic, String surveyID)
sGetQuestions (String surveyID) return String
sAnswerQuestions (String answers, String surveyID)

per-client view

Combining multiple ADTs



OrderCustomerManager
createSession return String deleteSession (String sessionID) sessionAuthenticate (String info) createOrder return String orderDefineCustomer (String info, String orderID, String sessionID) orderDefinePayment (String info, String orderID, String sessionID) orderAddItem (String name, String orderID, String sessionID) orderDeleteItem (String name, String orderID, String sessionID) orderCalculateTotal (String orderID, String sessionID) return Euro orderCheckPayment (String orderID, String sessionID) return Boolean orderPlace (String orderID, String sessionID) sGetTopicList (String surveyID , String sessionID) return String sSelectTopic (String topic, String surveyID , String sessionID) sGetQuestions (String surveyID , String sessionID) return String sAnswerQuestions (String answers, String surveyID, , String sessionID)

pan-client view

Conclusion



- highlighted shortcomings of current approaches in specifying context sensitive information
- manager pattern and pan/per-client interfaces
- benefits:
 - service discovery
 - matching from client's viewpoint, may include protocol and pre/post-conditions
 - service specification
 - simpler and more concise
 - semantic information:
 - meaning of the id parameter explicit

Thank you



Thank you for your attention!

Questions?

