Business Process Management: Concepts, Languages, Architectures

Second Edition

Figures of Chapter 7

Mathias Weske

Legal Notice

- This PowerPoint file contains figures used in
  - Mathias Weske, Business Process Management: Concepts, Languages, Architectures 2012, XIV, 403 p. 300 illus., Hardcover ISBN: 978-3-642-28615-5, eBook ISBN: 978-3-642-28616-2 © Springer-Verlag Berlin Heidelberg 2012, 2007
- Each slide contains one picture in PNG-format with caption and copyright notice. It is illegal to remove the copyright notice from any of the figures.
- Commercial use in any form is not allowed without prior written approval by Springer-Verlag
- Teaching material can be found at BPM Academic Initiative, http://academic.signavio.com

Best regards,

Mathias Weske

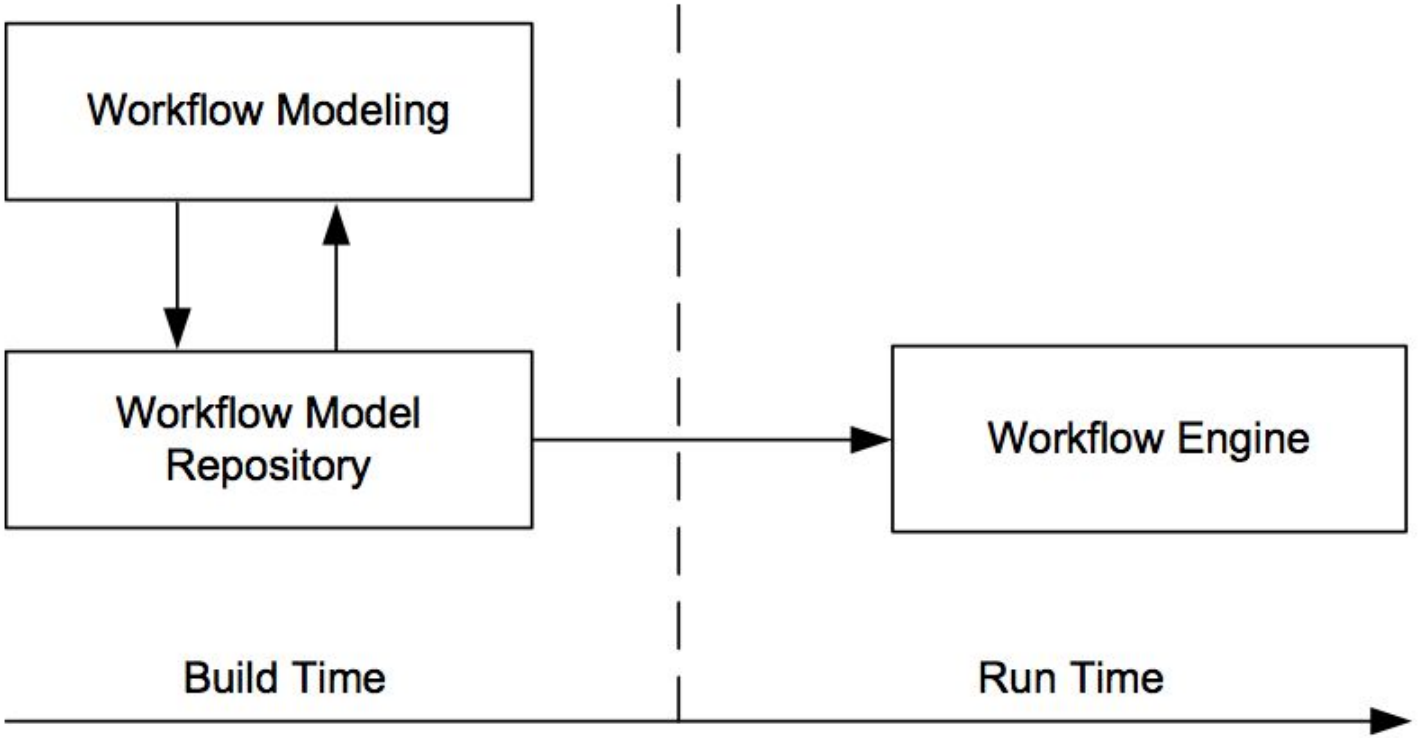
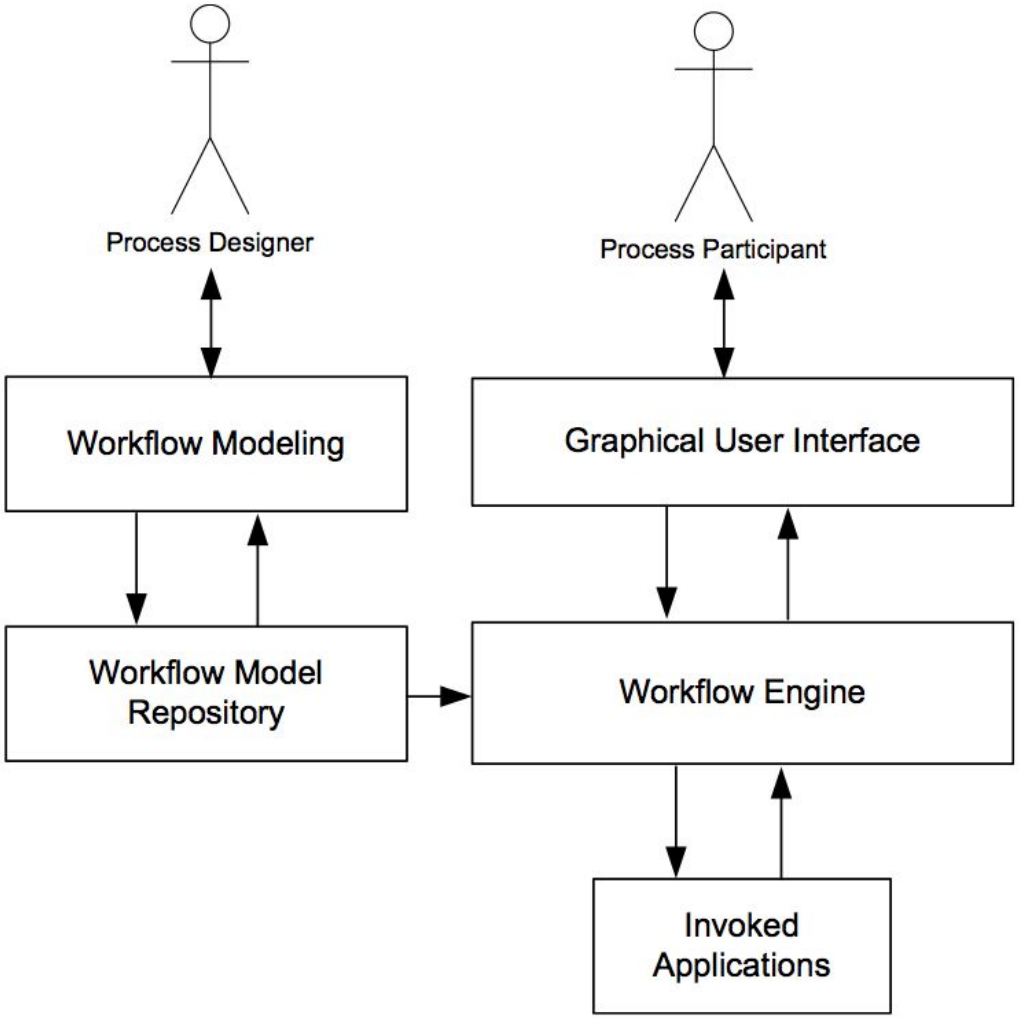
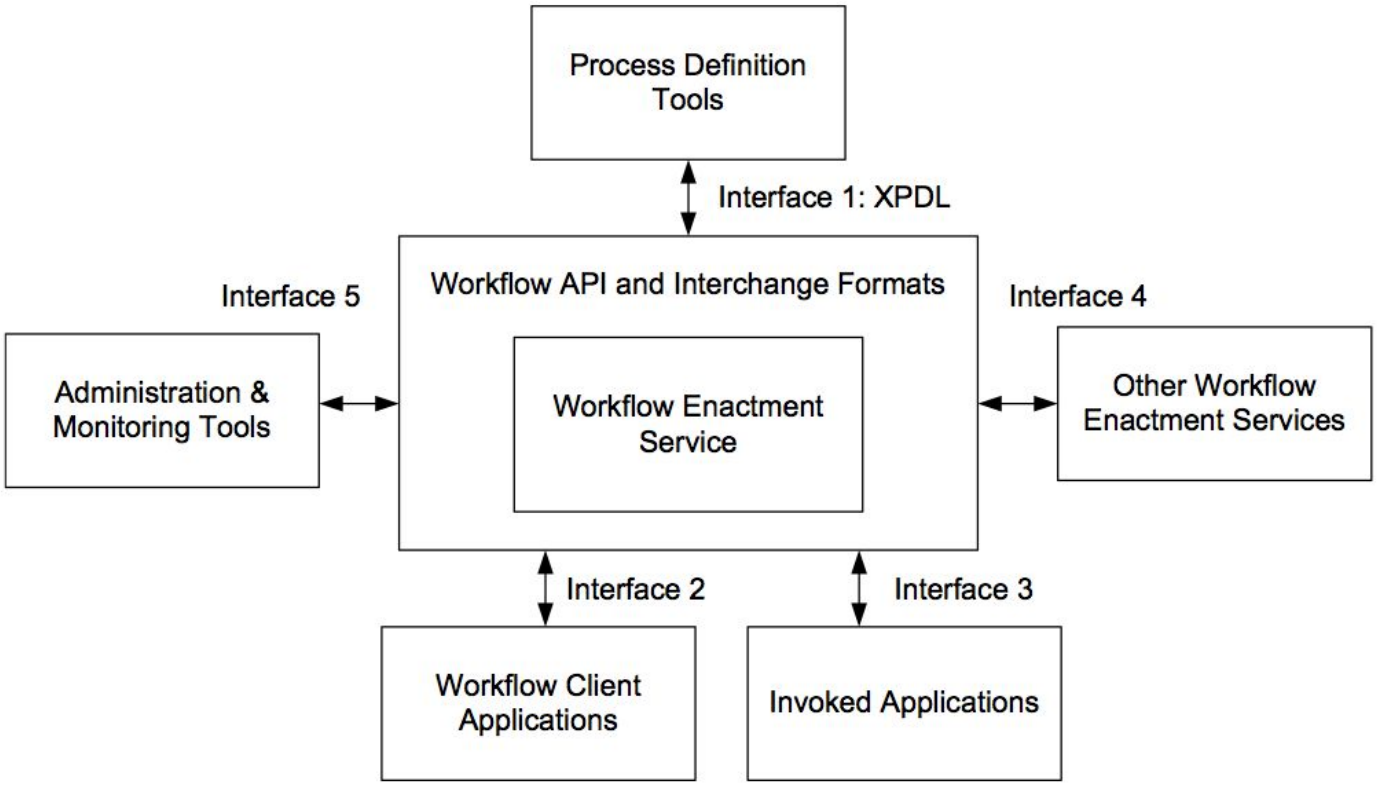


Fig. 7.1. Build time versus run time of a workflow



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 7.2. Workflow management systems architecture



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 7.3. Workflow reference architecture, proposed by the Workflow Management Coalition

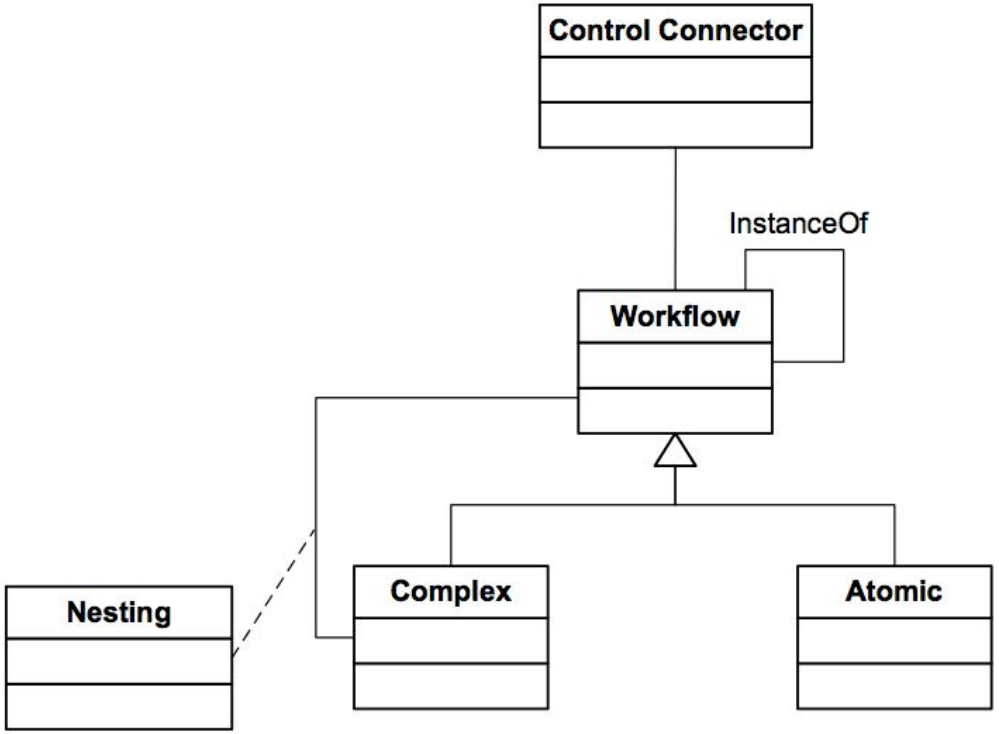


Fig. 7.4. Metamodel of flexible workflow management system, simplified version

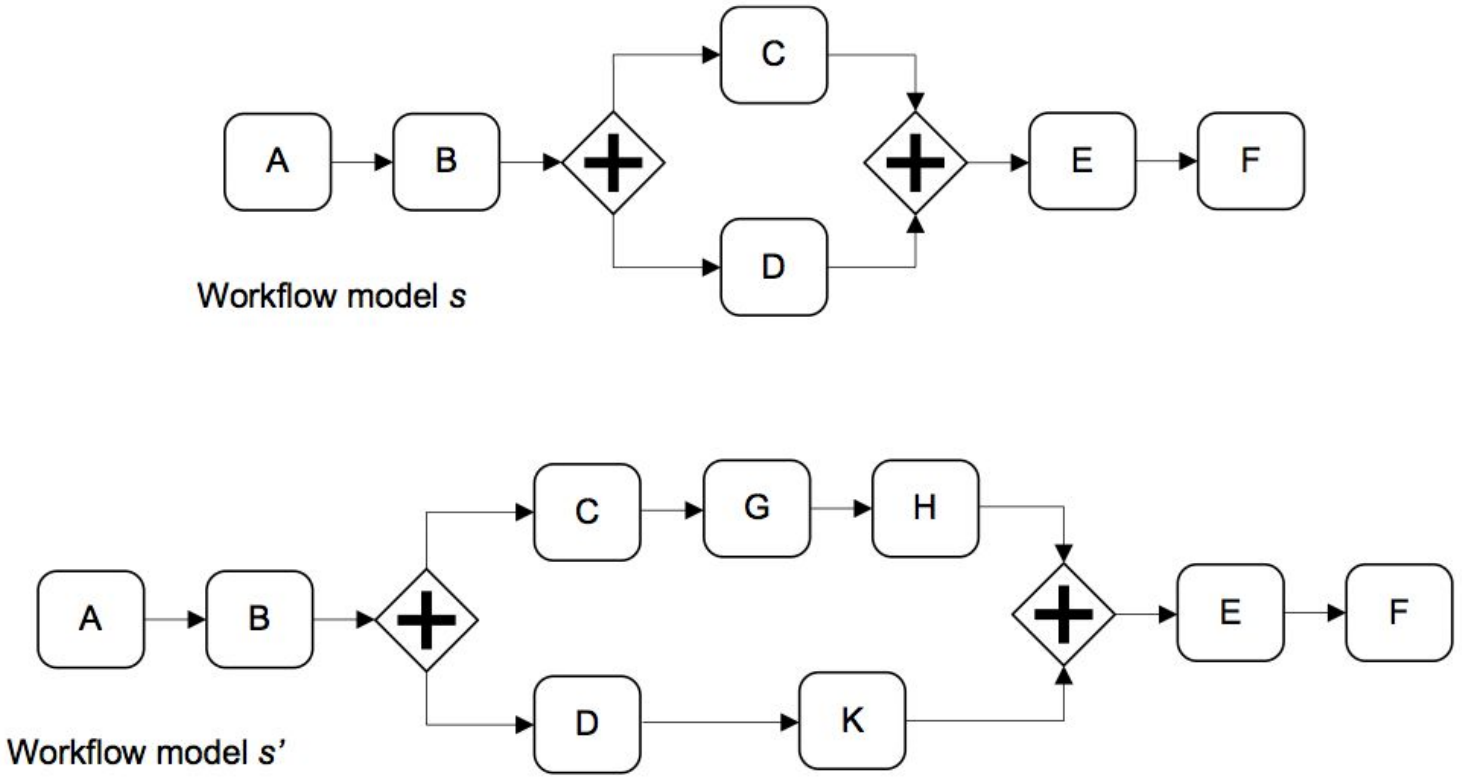


Fig. 7.5. Workflow model s and modified workflow model s'

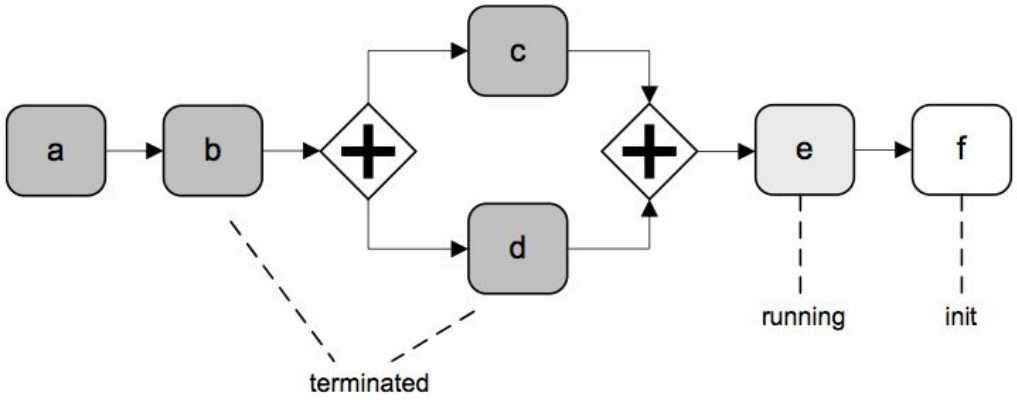


Fig. 7.6. Workflow instance i based on workflow model s with state information

M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

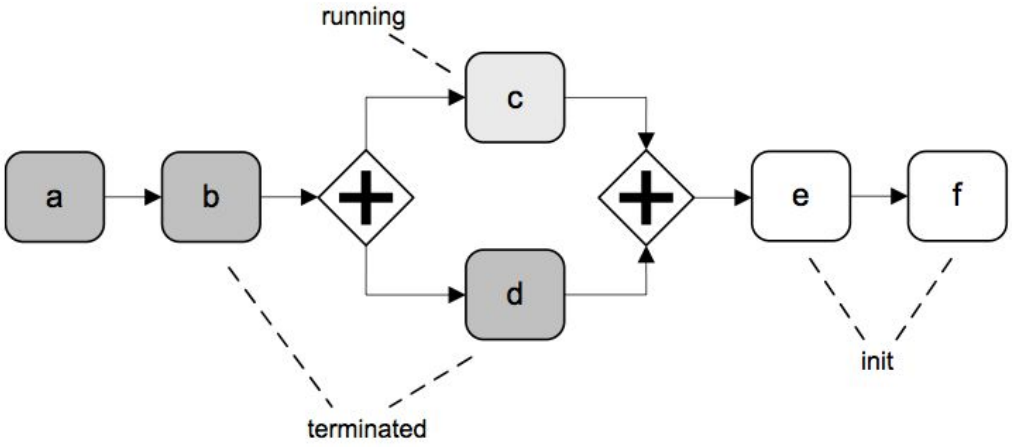


Fig. 7.7. Workflow instance j based on workflow model s with state information

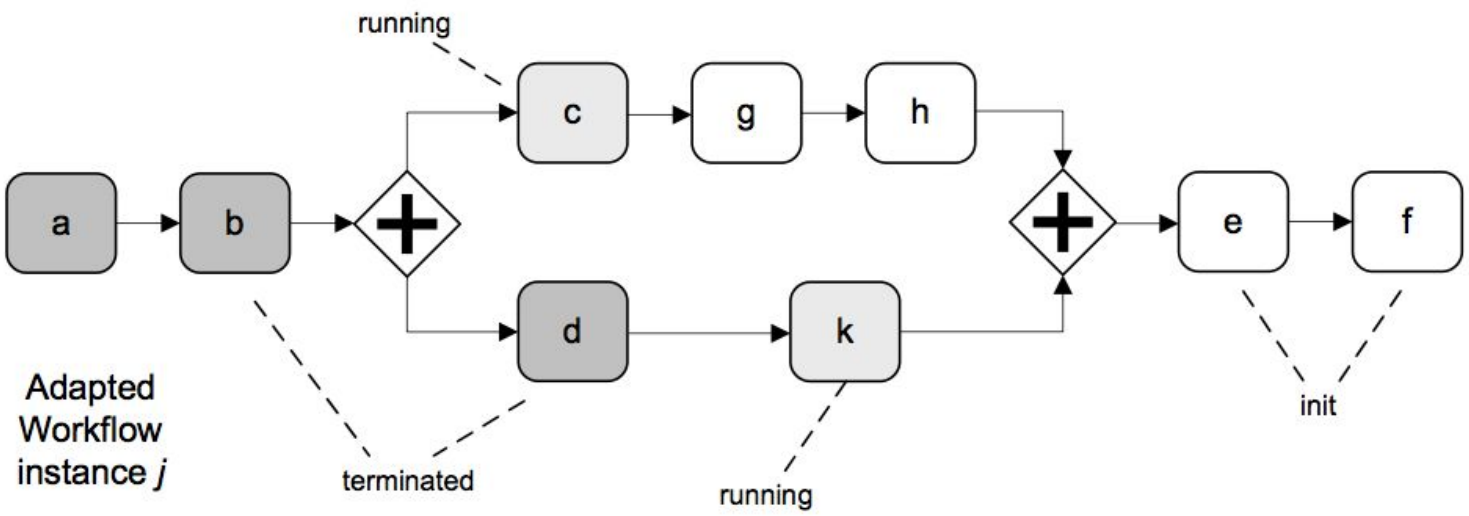
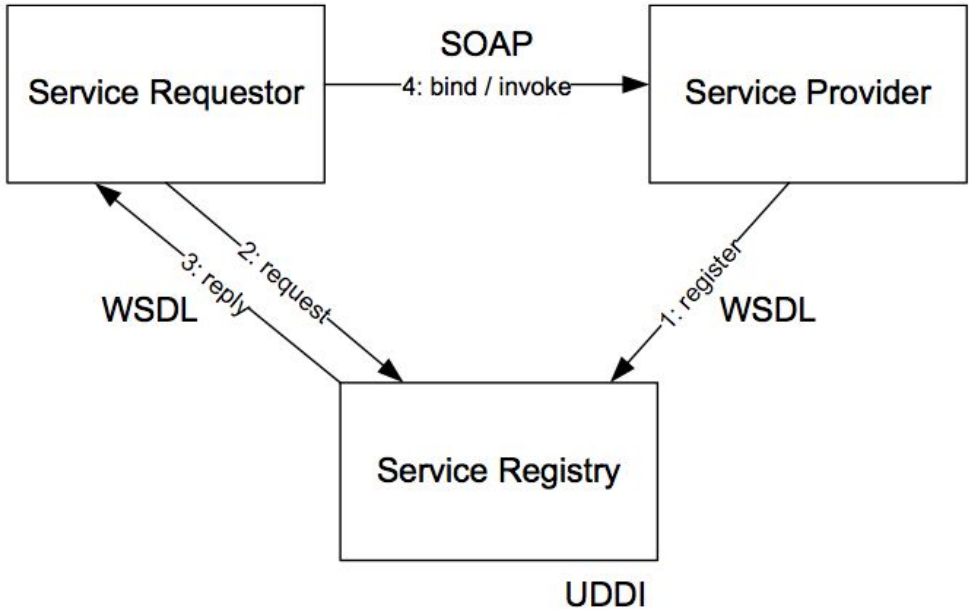


Fig. 7.8. Adapted workflow instance j



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 7.9. Main World Wide Web Consortium Web services recommendations

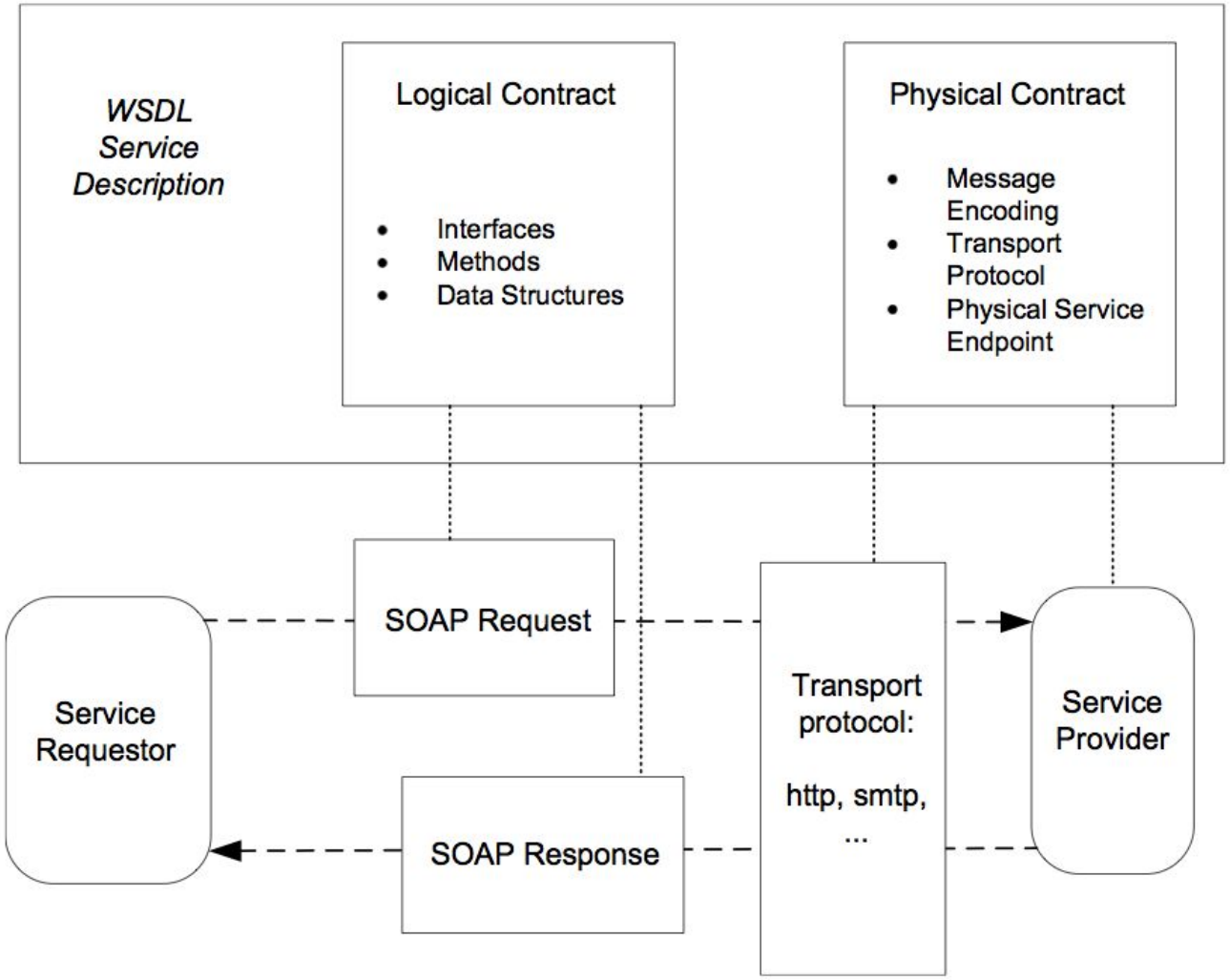


Fig. 7.10. Role of WSDL in Web service invocation

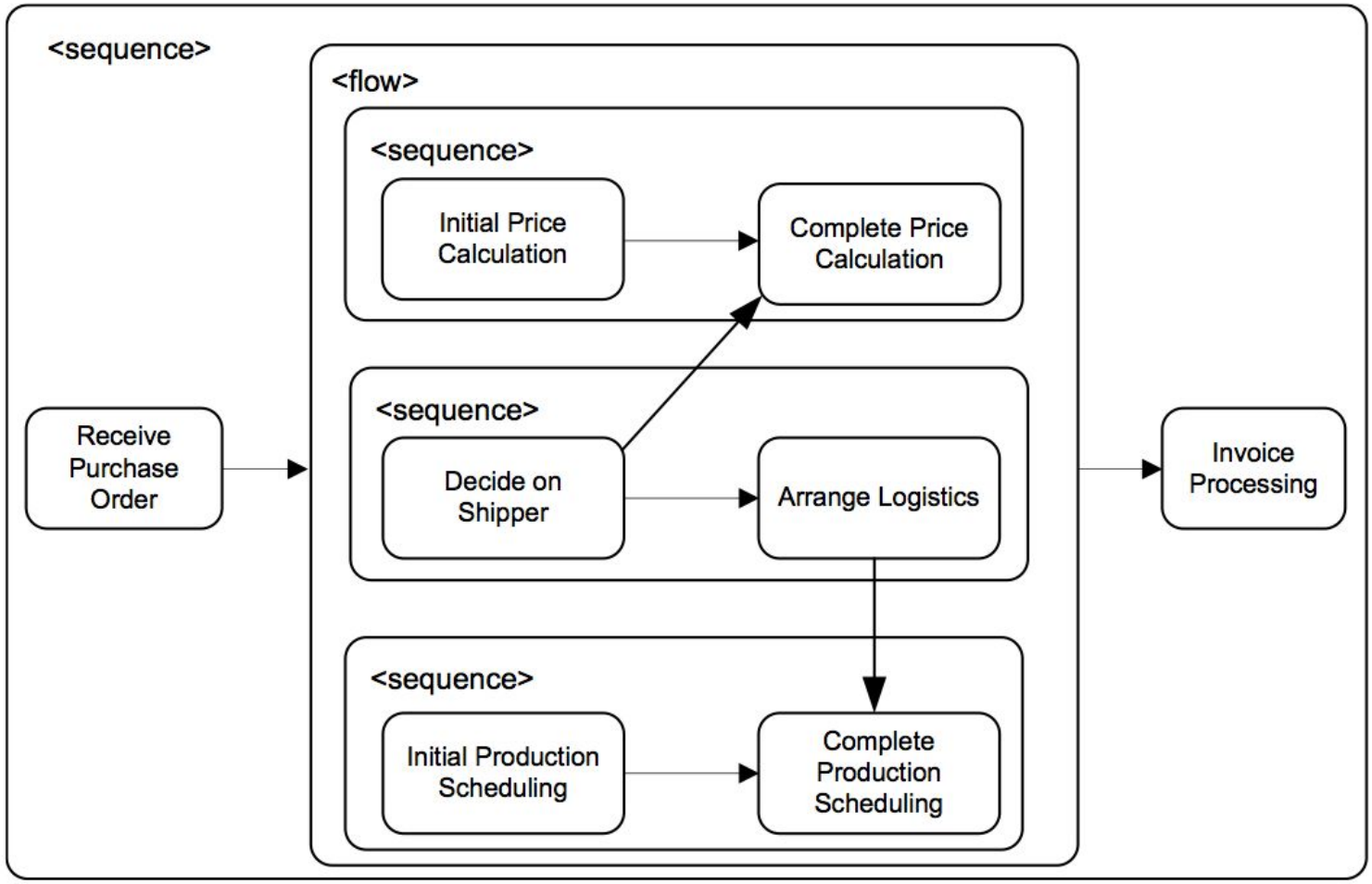
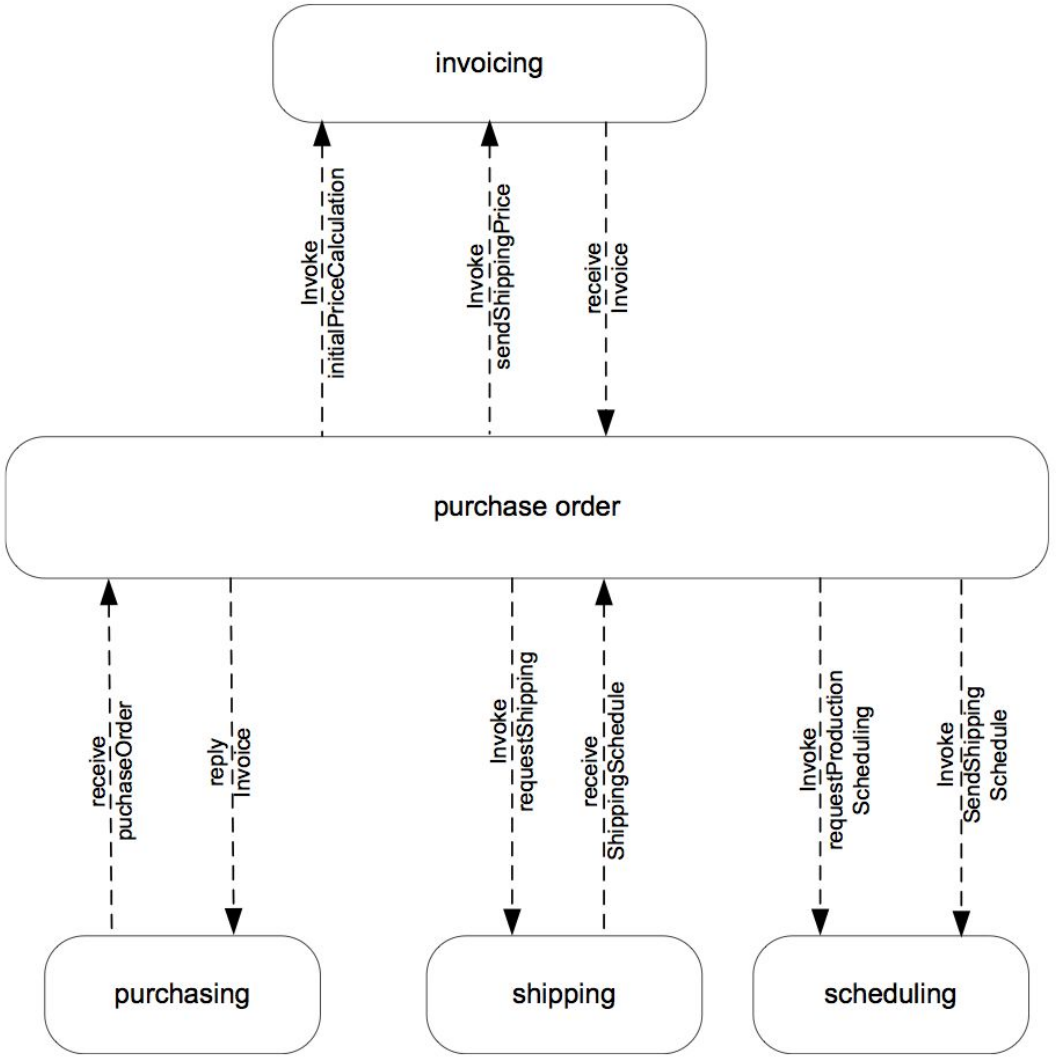


Fig. 7.11. Graphical representation of Web services composition in the WS-BPEL format

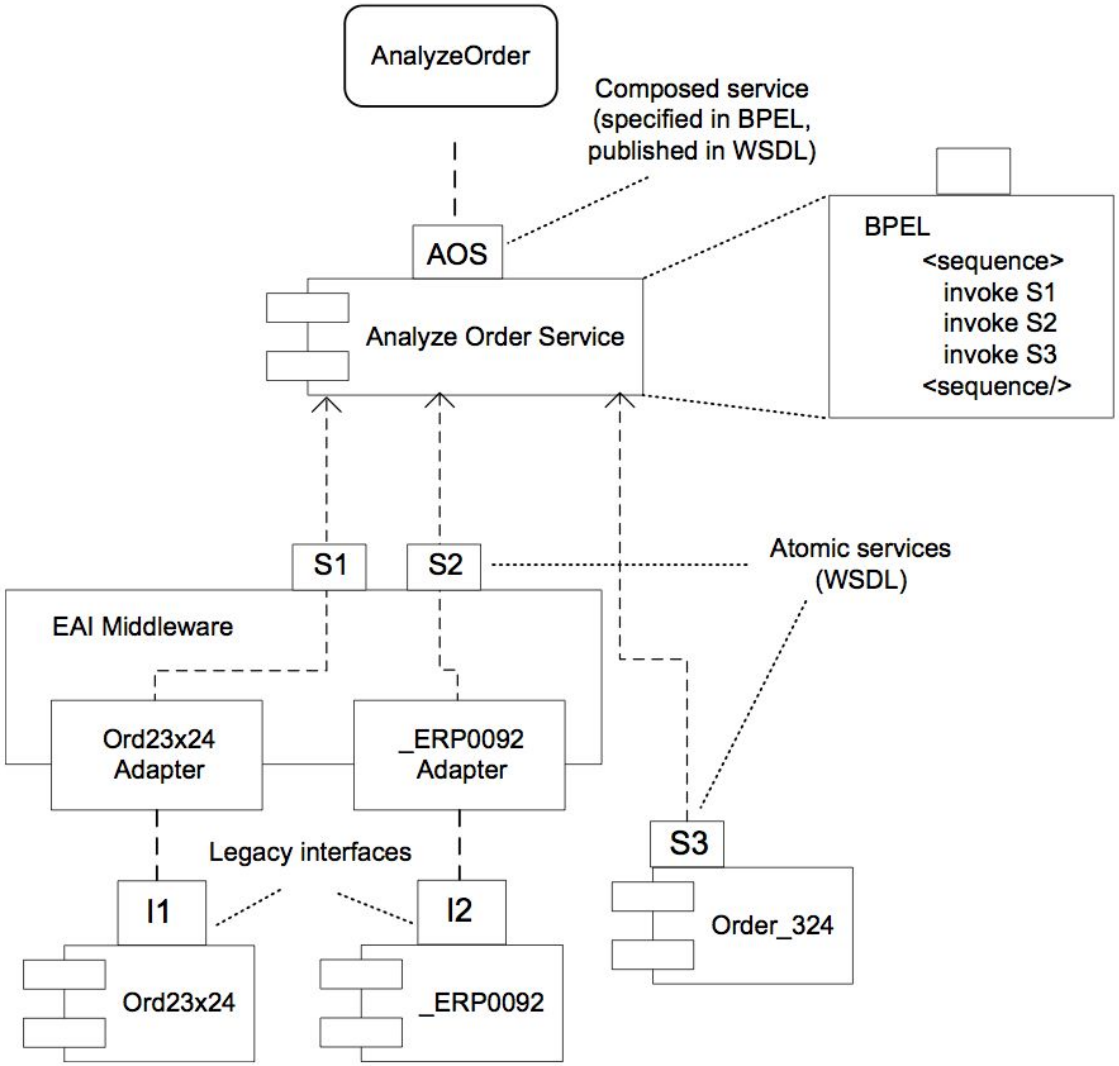


M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 7.12. Communication behaviour of purchase order WS-BPEL process

```
sequence
// Receive Purchase Order, partner link purchasing
receive PO
flow
   // defining links between activities
   links
      link name = ship-to-invoice
      link name = ship-to-scheduling
   sequence
      // Decide on Shipper, partner link shipping
      invoke requestShipping(in: shippingRequest, out:shippingInfo)
         source ship-to-invoice
      // Arrange Logistics activity, partner link shipping
      receive shippingSchedule
         source ship-to-scheduling
   sequence
      // Initial Price Calculation, partner link invoicing
      invoke initialPriceCalculation (in: PO)
      // Complete Price Calculation, partner link invoicing
      invoke sendShippingPrice (in: shippingInfo)
         target ship-to-invoice
      // Receive invoice, partner link invoicing
      receive Invoice
   sequence
      // Initiate Production Scheduling, partner link scheduling
      invoke requestProductionScheduling (in: PO)
      // Complete Production Scheduling, partner link scheduling
      invoke sendShippingSchedule (in: shippingSchedule)
         target ship-to-scheduling
// Invoice Processing, partner link purchasing
reply Invoice
```


Fig. 7.13. Structure of Web services composition expressed in WS-BPEL (simplified)



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 7.14. Web services standards in service-enabling

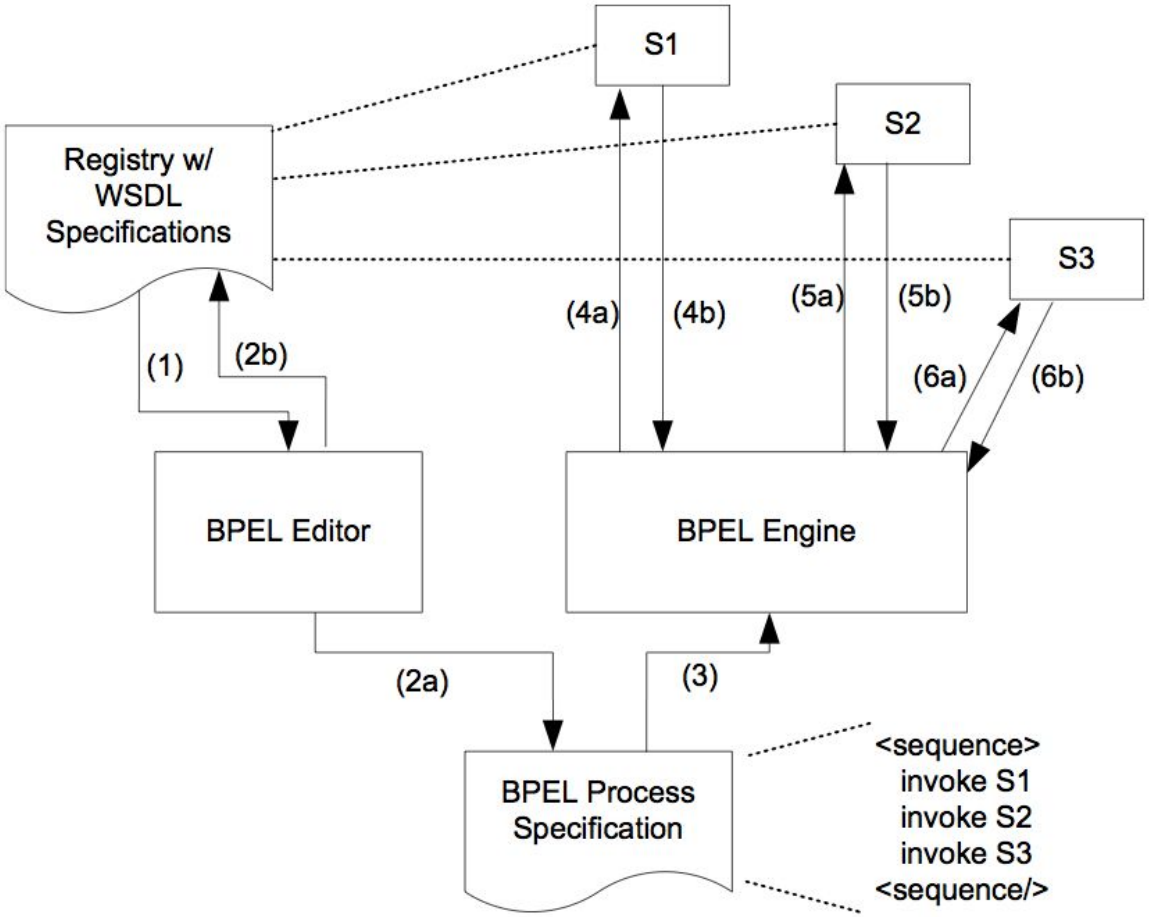


Fig. 7.15. Composed service design and enactment using Business Process Execution Language

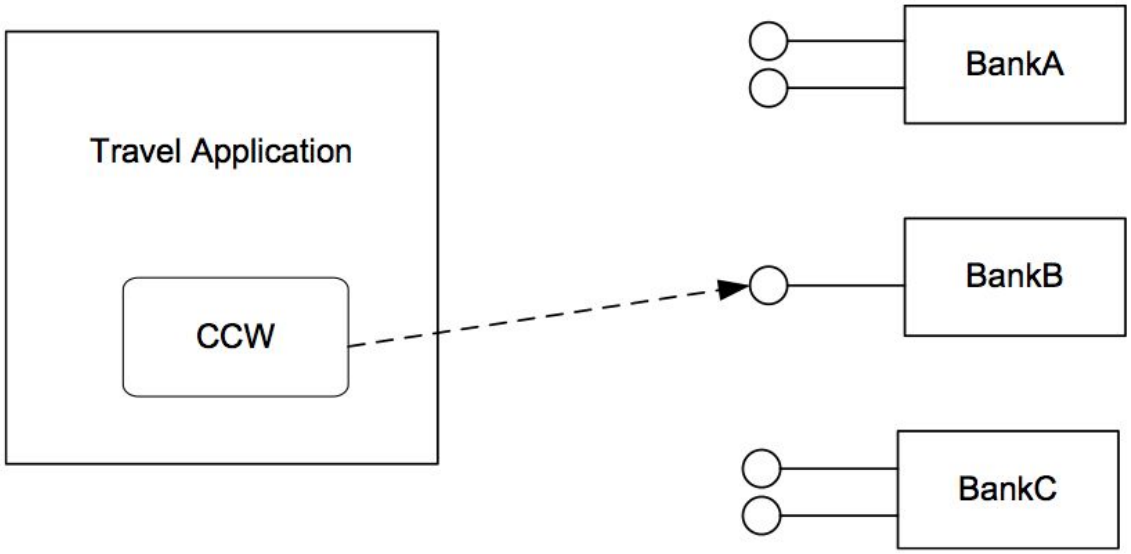


Fig. 7.16. Static binding: service provided by *BankB* coded in the travel application

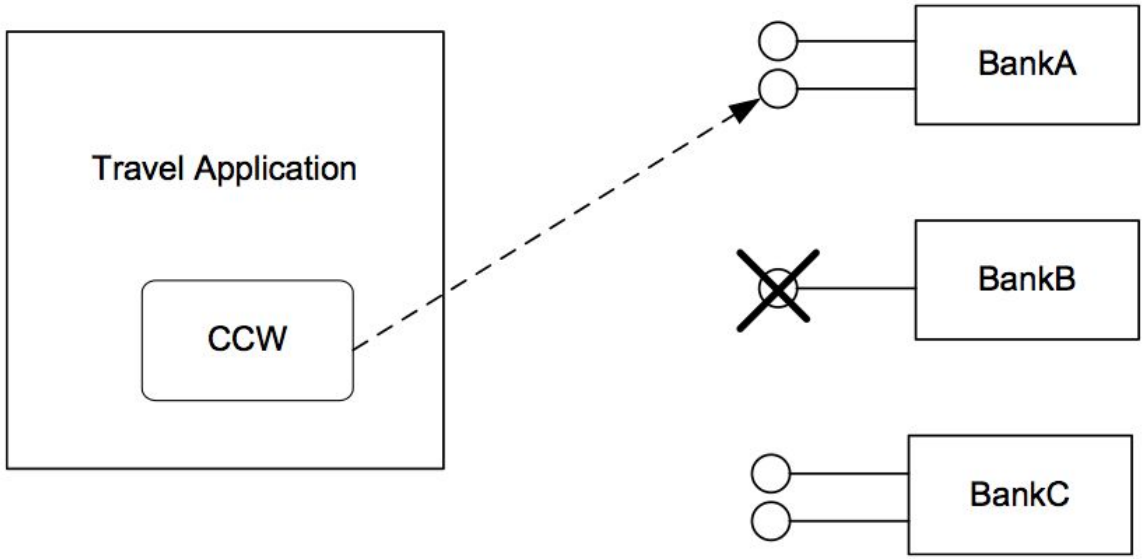


Fig. 7.17. Dynamic binding: service implementation by *BankA* is bound dynamically to travel application, due to failure of service implementation by *BankB*

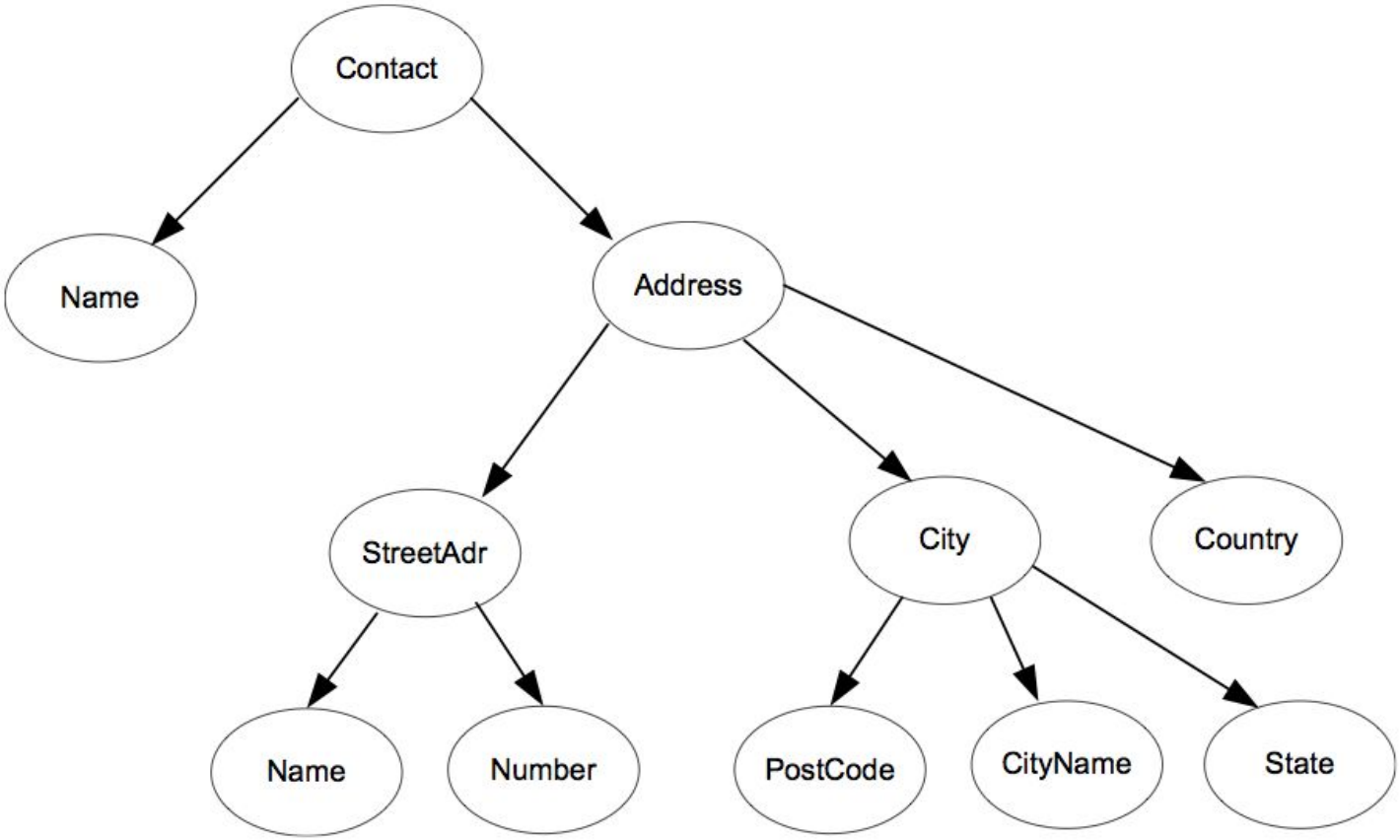
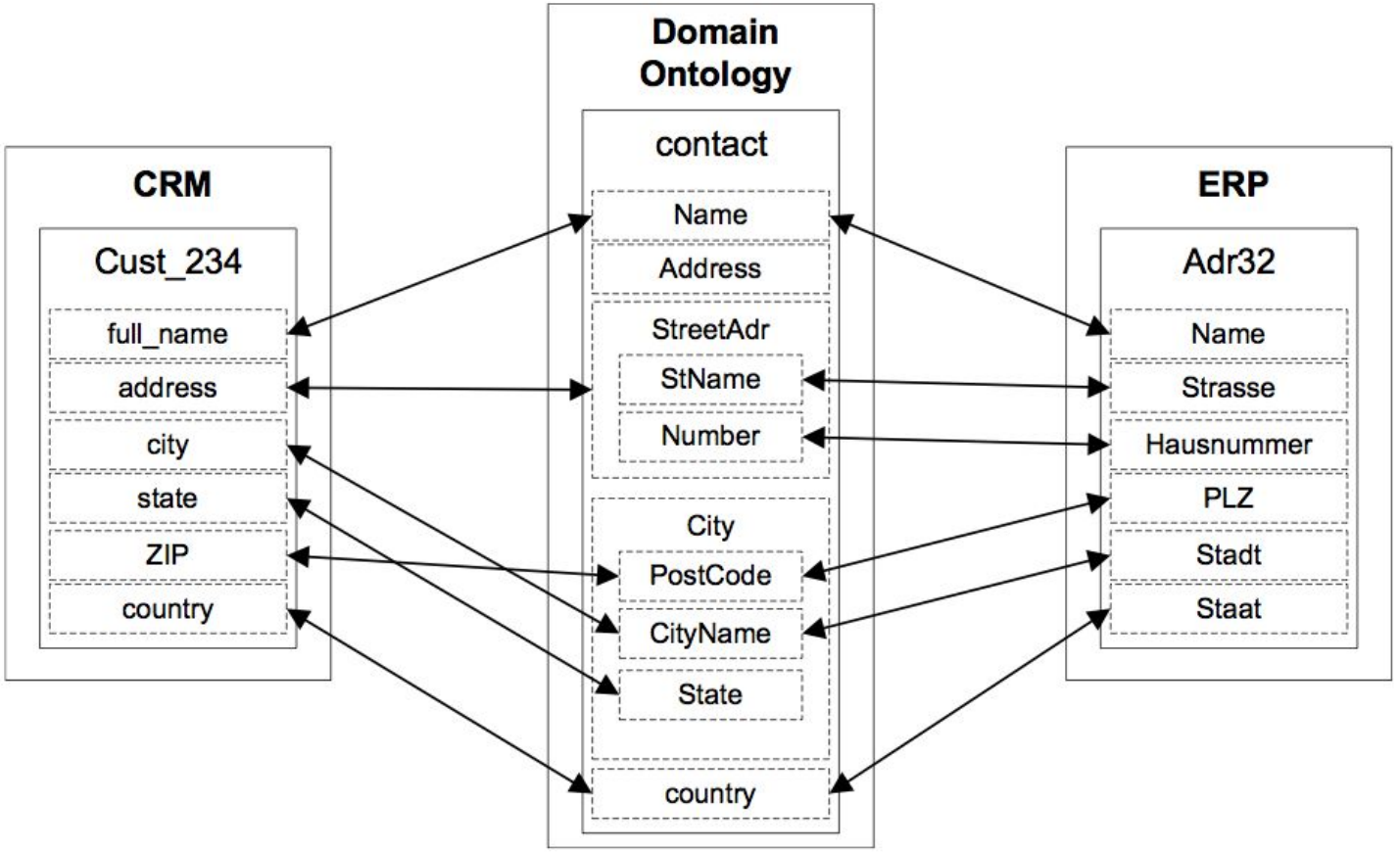


Fig. 7.18. Domain ontology for contacts



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 7.19. Domain ontology facilitates data mapping, Kuropka et al. (2006)

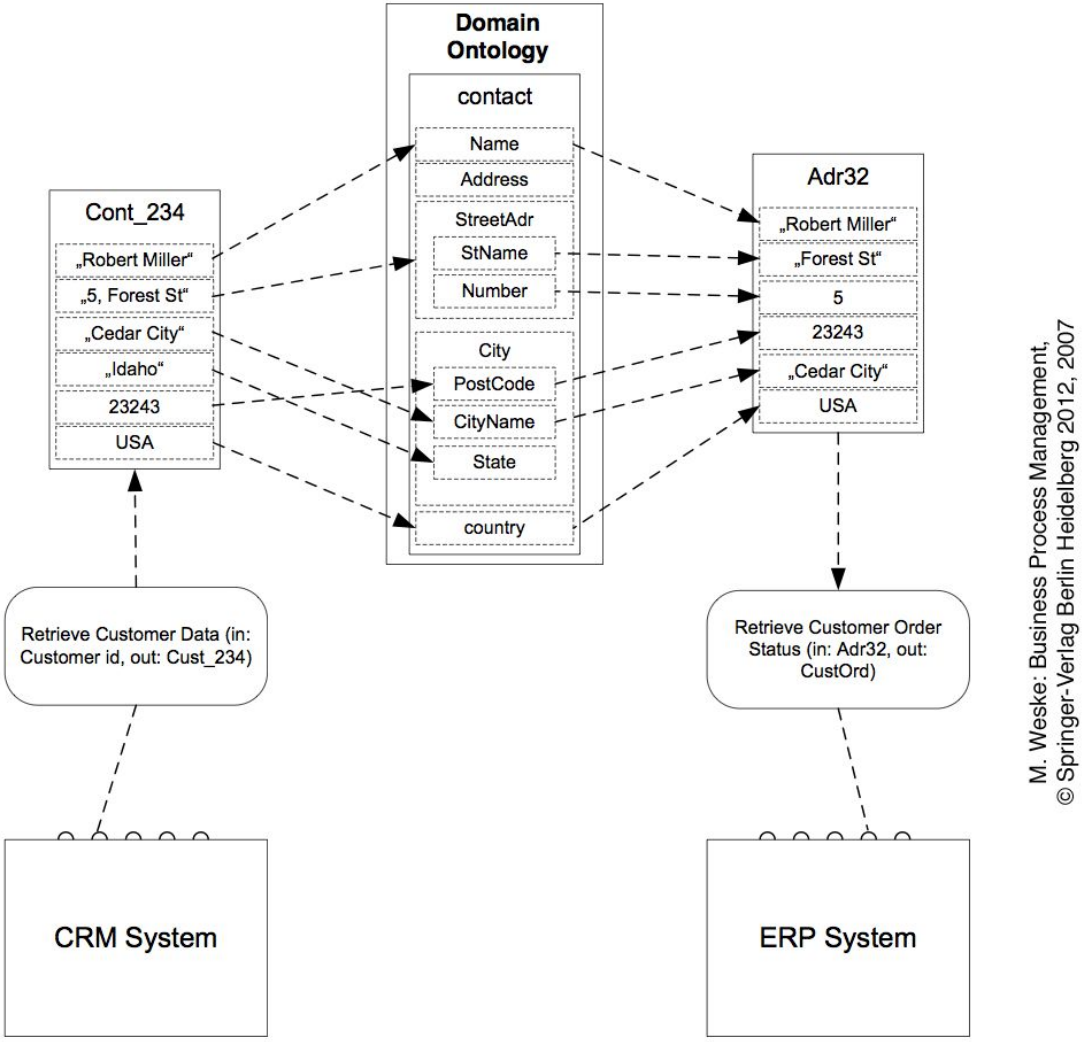


Fig. 7.20. Domain ontology used for automatic data mapping of CRM customer data to ERP customer data

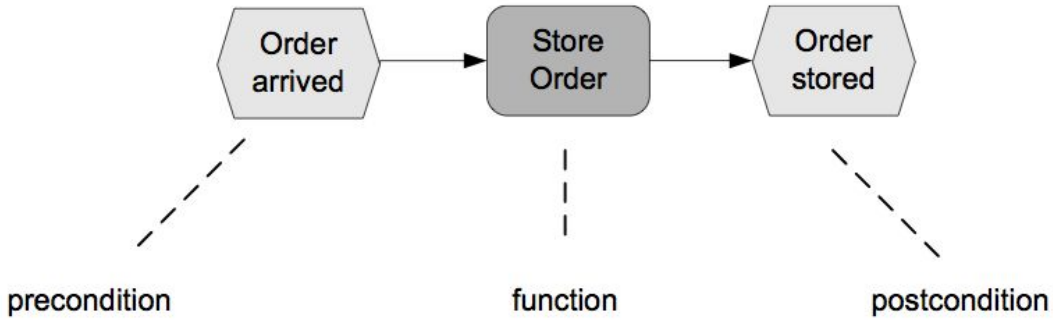
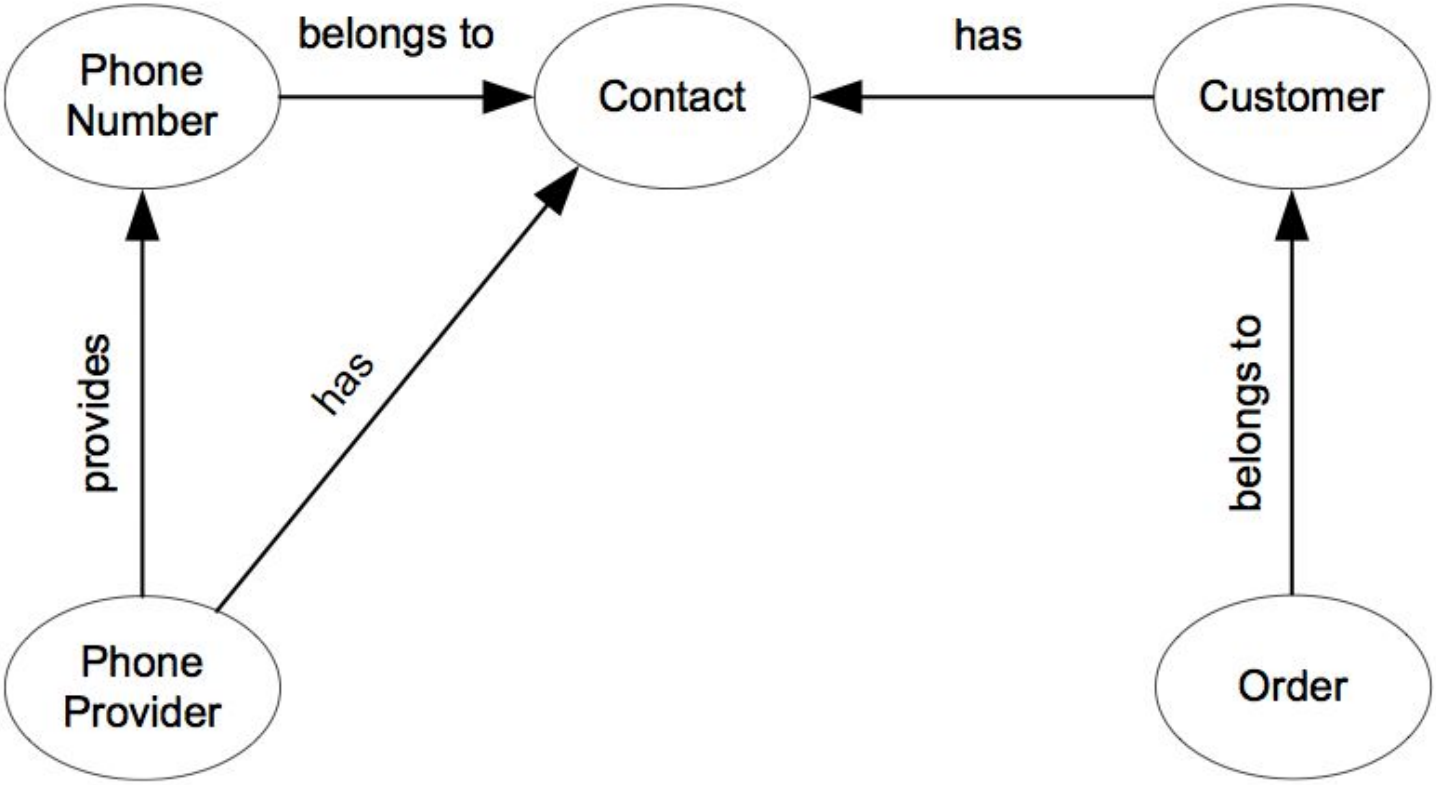
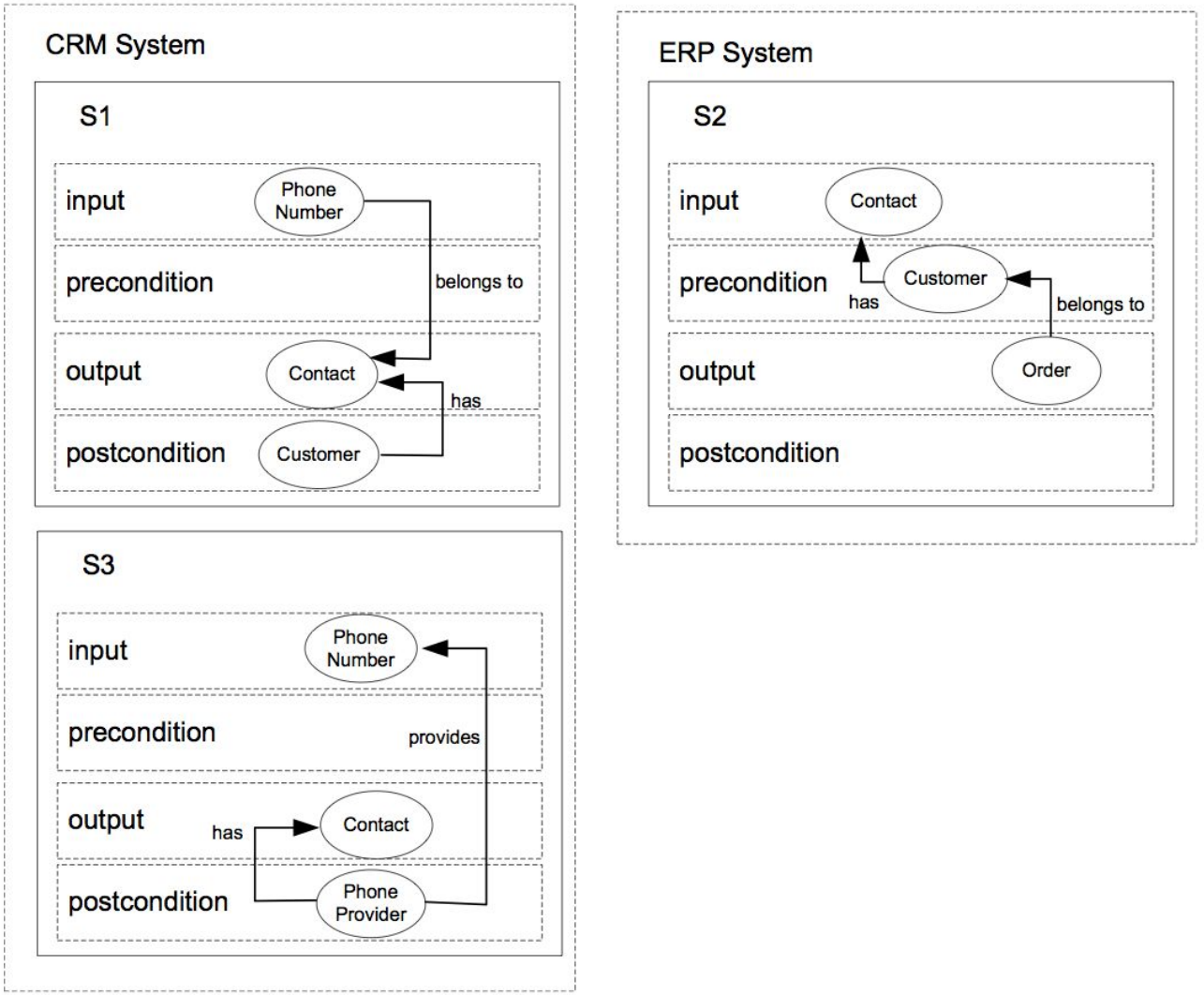


Fig. 7.21. Precondition and postcondition, expressed in event-driven process chain



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 7.22. Domain ontology of call centre example



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 7.23. Semantic specification of services

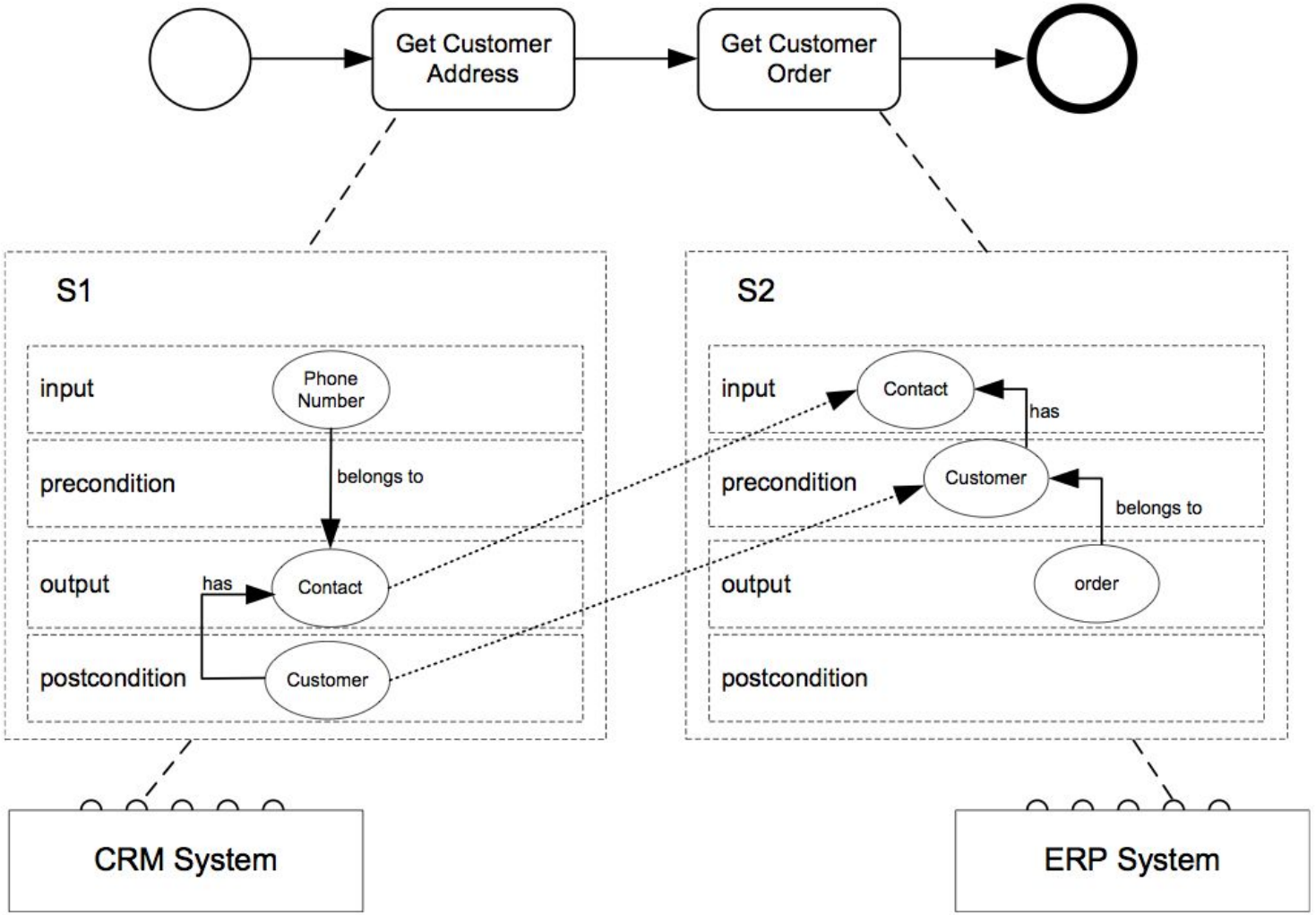


Fig. 7.24. Semantic service composition

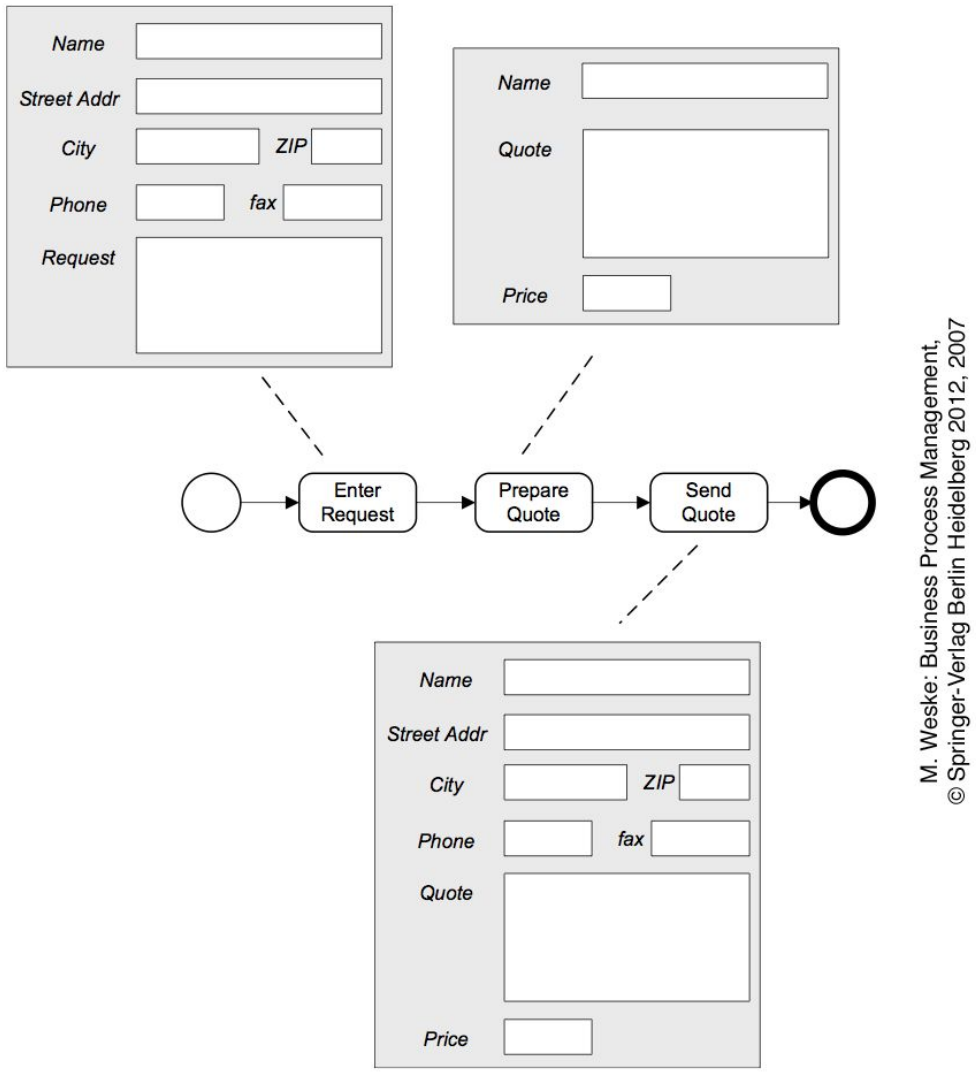
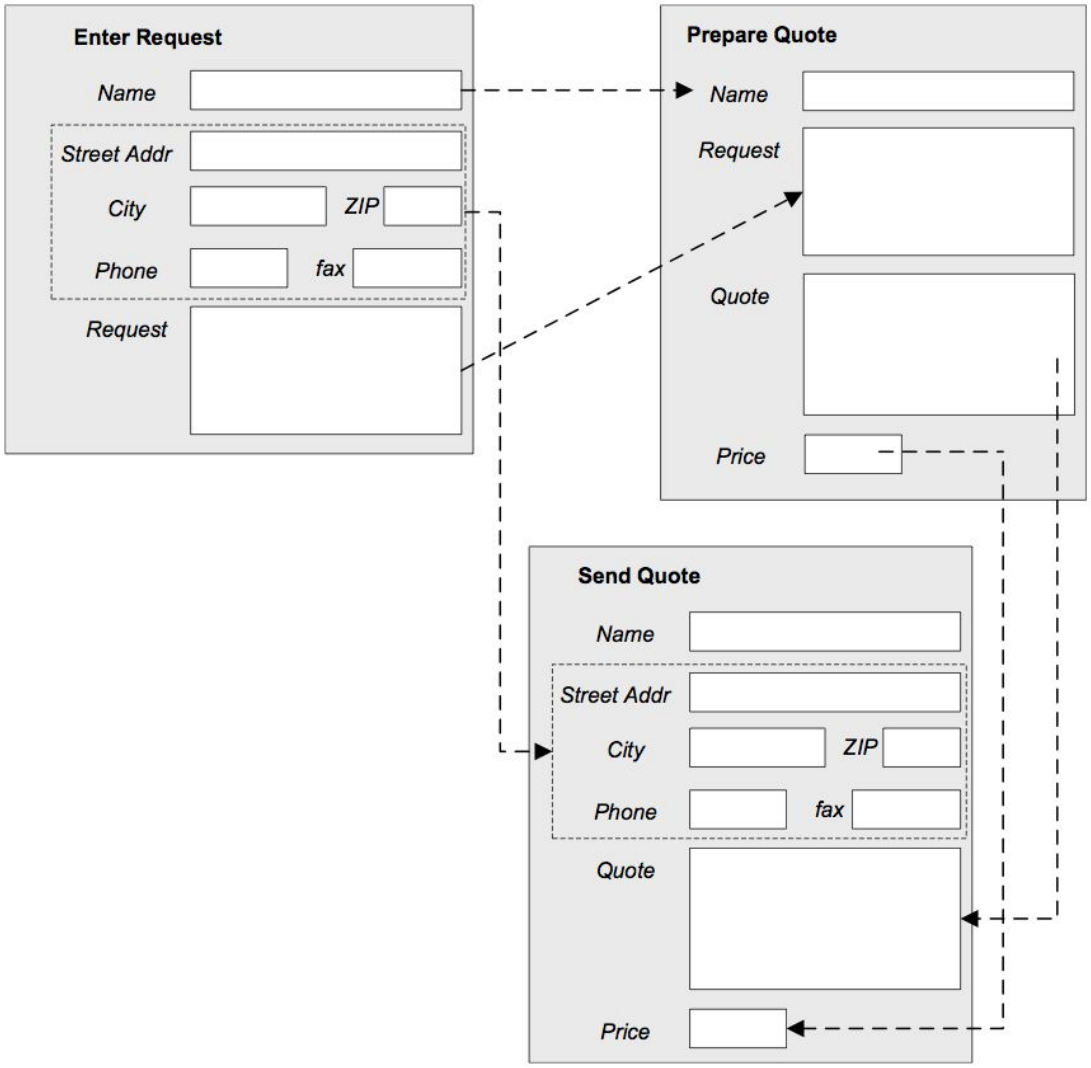
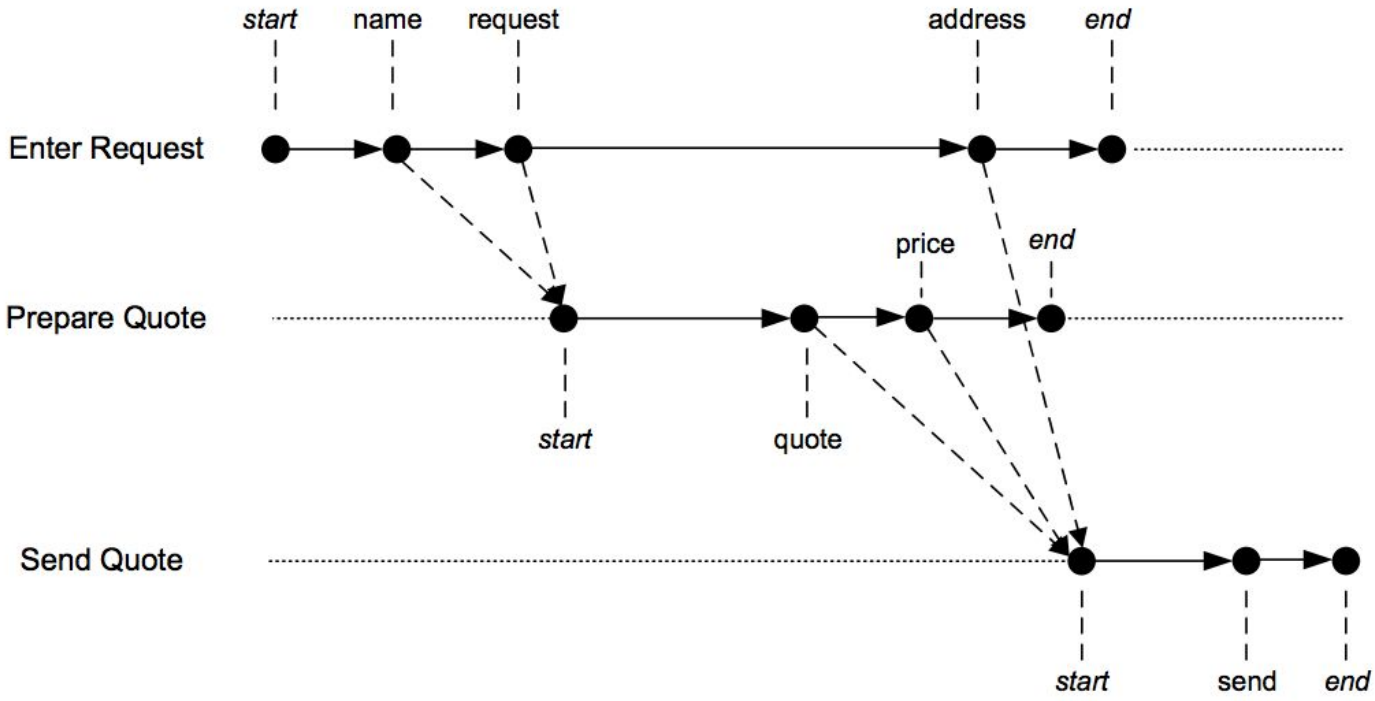


Fig. 7.25. Motivating example case handling



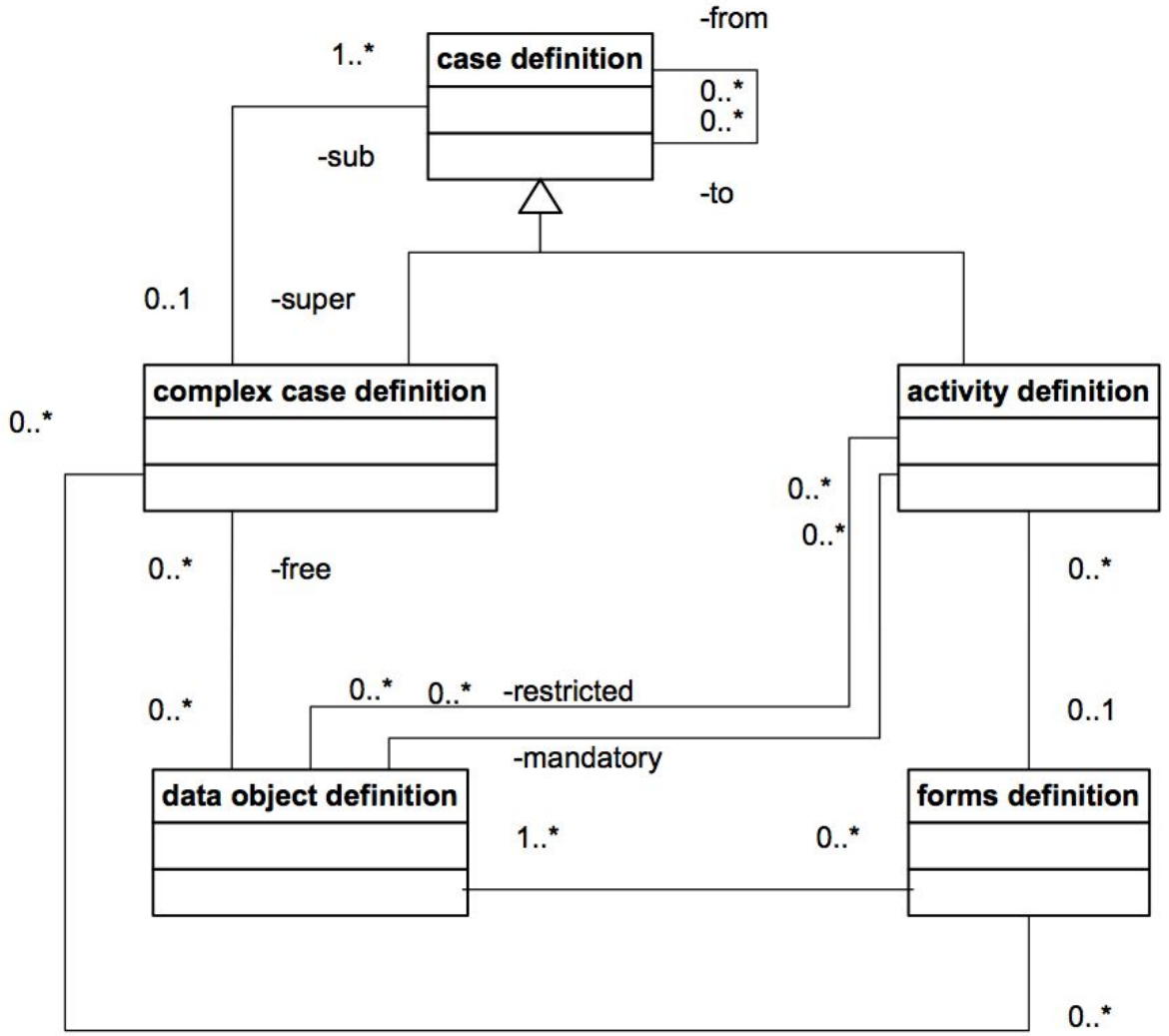
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 7.26. Data dependencies in case handling example



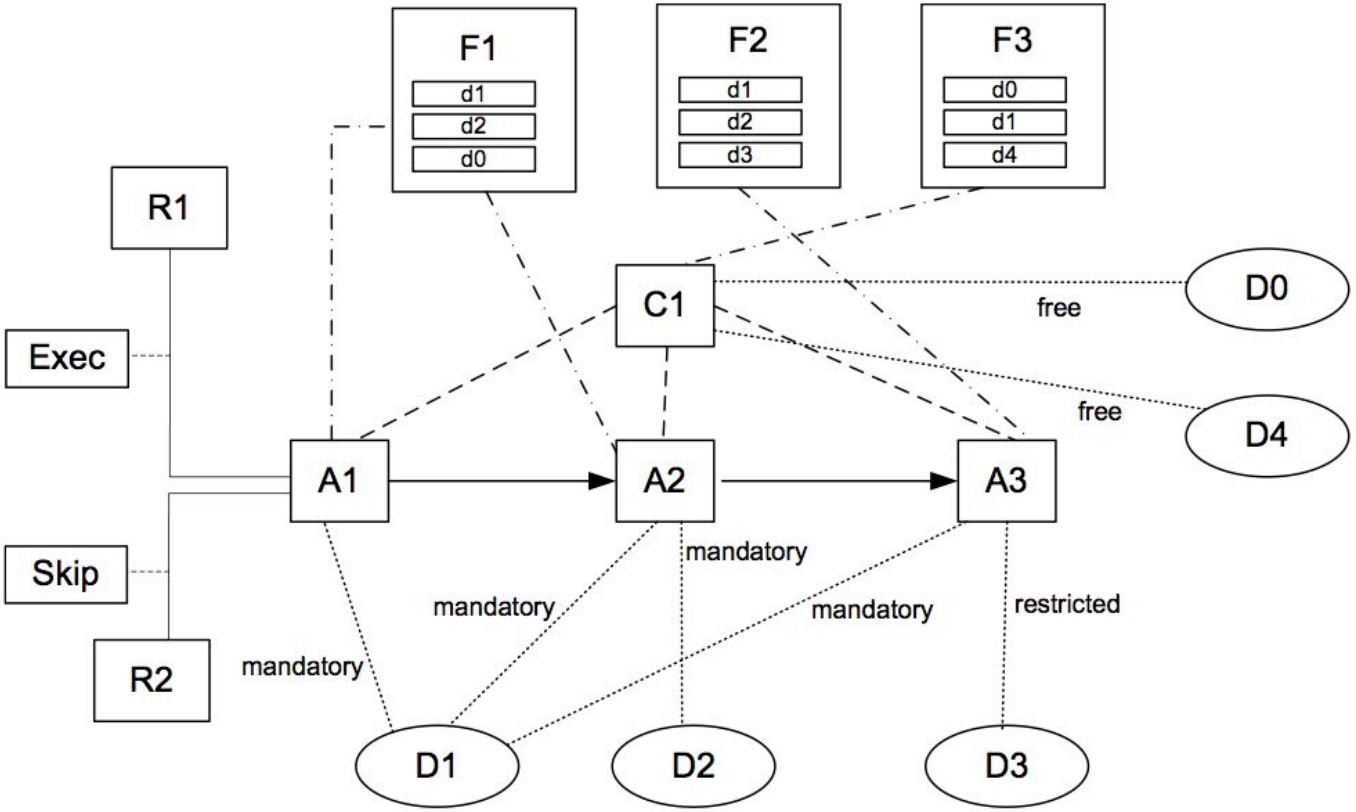
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 7.27. Temporal behaviour in case handling example: overall execution time is reduced, since prepare quote can start before enter request is completed



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 7.28. Case metamodel, simplified version without roles



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 7.29. Abstract example to illustrate case handling metamodel, van der Aalst et al. (2005b)