



Specification and Verification of Composite Web Services

Simon Woodman

University of Newcastle upon Tyne, UK

Work with Santosh Shrivastava, Stuart Wheater
and Doug Palmer

EDOC 2004, Monterey, CA



Outline

1. Aims and Approach
2. Notations
3. Pi-calculus
4. Example: online store
5. Example: asynchronous multi-party interaction
6. Conclusions

EDOC 2004, Monterey, CA

Aims

- Create Composite, value added Web Services from existing Web Services
- Used within the context of Virtual Organisations
 - Collaborating to provide a service
 - Integrating business processes
- Composite Services require:
 - Fault tolerance
 - Interoperability
 - Modularity
 - Adaptability
 - **Verification**
- Flexible execution model
 - Centralised
 - Distributed peer-to-peer

EDOC 2004, Monterey, CA

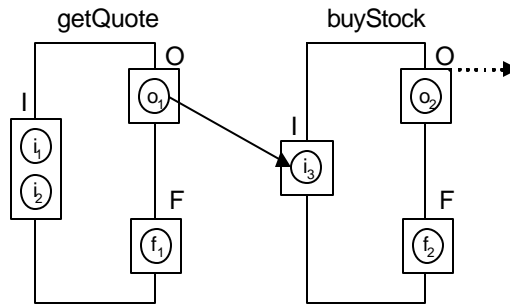
Approach

- Initially have a set of web services
 - Public: UDDI or other registry
 - Private: business logic not publicly exposed
- Design and create Composite Service from the available services
- Check static and dynamic correctness of the Composite Service
 - Uses the correct interface for each service
 - Lack of deadlocks and livelocks
 - The operations of each service are invoked in the correct order
 - e.g. Login before order
 - e.g. ensure transaction protocols followed correctly
- Latter needs a new language – “Sequencing Constraints”
 - Combine the composition with sequencing constraints for verification purposes

EDOC 2004, Monterey, CA

Composition Notation

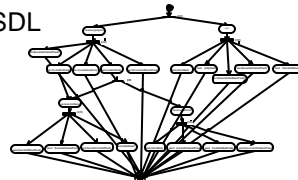
- Graphical Notation (creation, monitoring) inherited from OPENflow
- XML representation (machine processing/deployment)
- p -calculus basis (verification)



EDOC 2004, Monterey, CA

Sequencing Constraints Notation

- No easy visual notations
 - Possible: UML Activity or Sequence Diagrams
 - Graphical State explosion
- XML Representation for human readability and machine processing
 - Sequential document based rather than pre-conditions
 - Consists of sequence, parallel, choice, multiple constructs
 - Primitives for expressing RPC as well as sending and receiving messages
 - Presented as an extension to WSDL
- p -calculus basis (for verification)



EDOC 2004, Monterey, CA

Sequencing Constraints Notation

- Designer can decide on the level of detail/encapsulation that they wish to use
 - Similar to ODP
- Breaking encapsulation can give causality in a multiparty asynchronous environment
- Allows message inspection but this is not verified
- Sequencing constraints are service-centric – only refer to the service exposing them
 - Gives distributed view, contrasting to WS-Choreography centralised view

EDOC 2004, Monterey, CA

Verification of the Composition

- Want to verify that an arbitrary composition is
 - Free of deadlocks
 - Free of livelocks
 - Respects the sequencing constraints of each service under all conditions
- Construct a *global picture*
- Analyse the global picture using ρ -calculus reaction rules

EDOC 2004, Monterey, CA

[p -calculus]

- p -calculus is a formalism developed by Robin Milner
- Basics:
 - Processes
 - Channels (for communication between processes)
 - Names (similar to tokens)
 - Send/Receive operators to allow message (name) passing
 - Four operators to specify parallelism, sequence, choice and replication
- A **reaction** is a matching send/receive pair
- Extension of CCS with the addition of mobility (sending a channel name along a channel)
- Use of mobility is useful in some protocols such as transactional ones and clarifying ambiguities in callbacks
- Automatically translate from XML to p -calculus

EDOC 2004, Monterey, CA

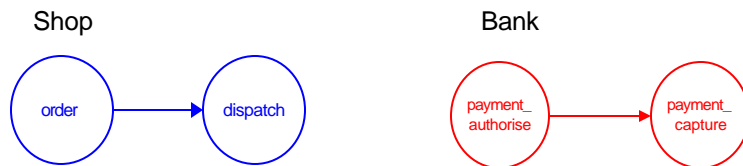
[Algorithm]

- It is necessary to show the following:
 1. Following any reaction, either another reaction can occur or the system is in a completion state
 2. From every state it is possible to reach a completion state
- **Completion State Definition:**
 - The composition has been reduced to an empty expression (no terms are left)
 - The sequencing constraints have been reduced to either an empty expression or they are in the starting state

EDOC 2004, Monterey, CA

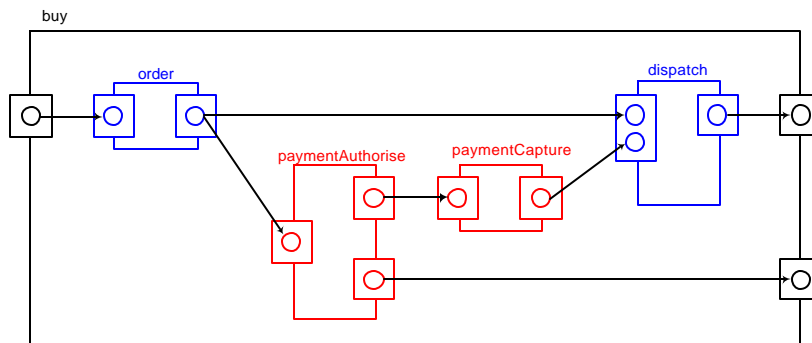
Example Sequencing Constraints

- Example: two simplified web services bank and shop
- Aim is to combine them to offer a composite shop web service



EDOC 2004, Monterey, CA

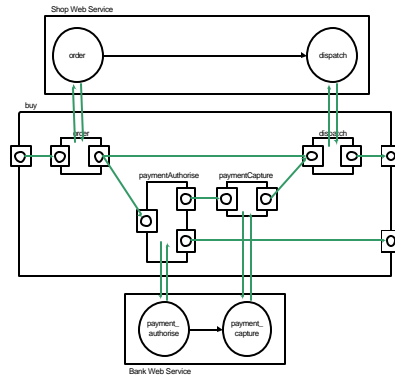
Example Composition



EDOC 2004, Monterey, CA

The Global Picture

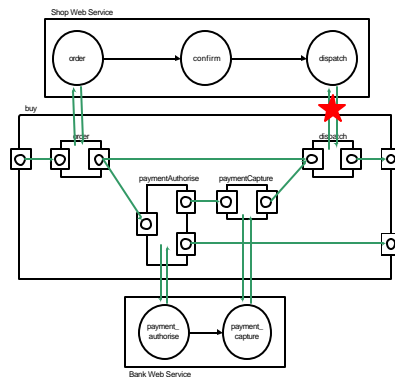
- Composition merged with sequencing constraints
- Composition: one process per task, one channel per dependency, one name per part
- Sequencing Constraints: one process per operation, one channel per message type, one name per message



EDOC 2004, Monterey, CA

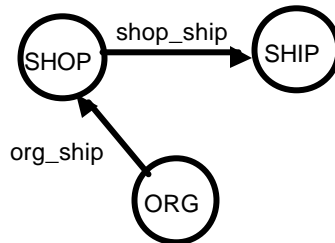
The Global Picture 2

- Addition of confirm operation
- Now the composition does not respect the sequencing constraints
- The system shows a deadlock when analysed using p - calculus as there is nothing ready to receive the dispatch message



EDOC 2004, Monterey, CA

Asynchronous multi-party Example



$ORG = org_shop < order, ship_org > . ship_org(del_info)$

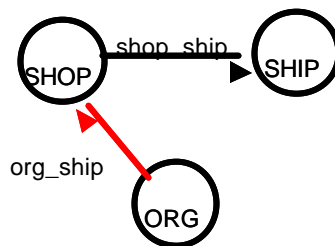
$SHOP = org_shop(order, ship_org). shop_shop < goods_info, ship_org >$

$SHIP = shop_shop(goods_info, ship_org). ship_org < del_info >$

$SYSTEM = ORG \mid SHOP \mid SHIP$

EDOC 2004, Monterey, CA

Asynchronous multi-party Example



$ORG = org_shop < order, ship_org > . ship_org(del_info)$

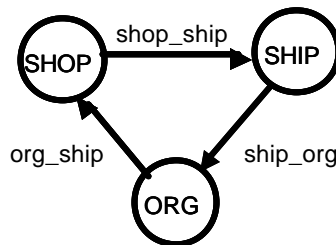
$SHOP = org_shop(order, ship_org). shop_shop < goods_info, ship_org >$

$SHIP = shop_shop(goods_info, ship_org). ship_org < del_info >$

$SYSTEM = ORG \mid SHOP \mid SHIP$

EDOC 2004, Monterey, CA

Asynchronous multi-party Example



ORG = c

SHOP =

SHIP = s

SYSTEM = *ORG* | *SHOP* | *SHIP*
EDOC 2004, Monterey, CA

Current Status and Future Work

- Investigate the use of pi-calculus restriction to show that data private to an organisation is not released as part of the composition
- Partially developed Enactment Engine (DECS)
 - Allows completely distributed coordination
 - Flexible deployment options configurable for organisational autonomy or performance
- Current lack of tooling for π -calculus mean that automatic verification is currently not possible
- Automatic translation to Promella and verification using SPIN model checker

EDOC 2004, Monterey, CA

[Questions?]

EDOC 2004, Monterey, CA