CITY UNIVERSITY LONDON

Academic excellence for business and the professions

A Requirements-led Approach for Specifying QoS-aware Service Choreographies: An Experience Report

Neil Maiden, **James Lockerbie**, Konstantinos Zachos

City University London

Antonia Bertolino, Guglielmo De Angelis, Francesca Lonetti

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" CNR

CITY UNIVERSITY LONDON

EST 1894

CHOReOS Large Scale Choreographies for the Future Internet

http://www.choreos.eu

Context of our work

- Part of the framework 7 EU funded CHOReOS project to implement a framework for **scalable choreography development**.

- *Goal: "...enable domain experts to develop decentralized ultra-large scale (ULS) solutions composed of heterogeneous services that are adaptable and QoS (Quality-of-Service) aware"*

- How to provide **user-centric** processes to support the whole life cycle of Service Oriented Computing systems from their:
  - **Design,** to their development, up to their maintenance and governance at run-time
- Based on business scenarios – we will use a *dynamic taxi management example*

CITY UNIVERSITY LONDON

Choreography

- **Service composition** – Orchestration
  - Services are arranged locally (centralized) to achieve a goal according to predetermined business logic and execution order
  - Often expressed in workflow notation such as BPEL
- Service composition - **Choreography**
  - Services interact in a global scenario to achieve a goal without a single point of control (decentralized)
  - Protocol for direct peer-to-peer interaction (no orchestrator)
- Increased flexibility can deliver more adaptive service-based systems that satisfy more ambition requirements of different quality types.

CITY UNIVERSITY LONDON

Requirements Challenges

1. Optimize the specification of choreography diagrams with respect to system requirements;
   - Design **flexible choreographies** that can be argued to satisfy system requirements
2. Associate specified system requirements with choreography activities in a choreography diagram;
   - To specify the required behaviour and qualities of choreography activities in a model.
3. Enhance choreography diagrams with quality properties that trace system requirements, to support analysis and monitoring facilities.
   - Means to trace service qualities back to the originating quality requirements on the systems.
   - Taking requirements that are not monitorable and transforming them into properties that are to show continued requirements satisfaction.

CITY UNIVERSITY LONDON

Requirements-led approach

- Future Internet vision – domain expert-centric process
- CHOReOS design components to specify QoS-aware service choreographies

Domain expert
Requirement specification

Specify natural language requirements

User task models

CHOReOS Requirements Tool

Generate first-cut choreography diagram

MagicDraw

Refine service choreography

Domain expert
Choreography design

Specify non-functional properties on choreography diagram

Q4BPMN

CITY UNIVERSITY LONDON

EST 1894

Expressing Natural Language Requirements

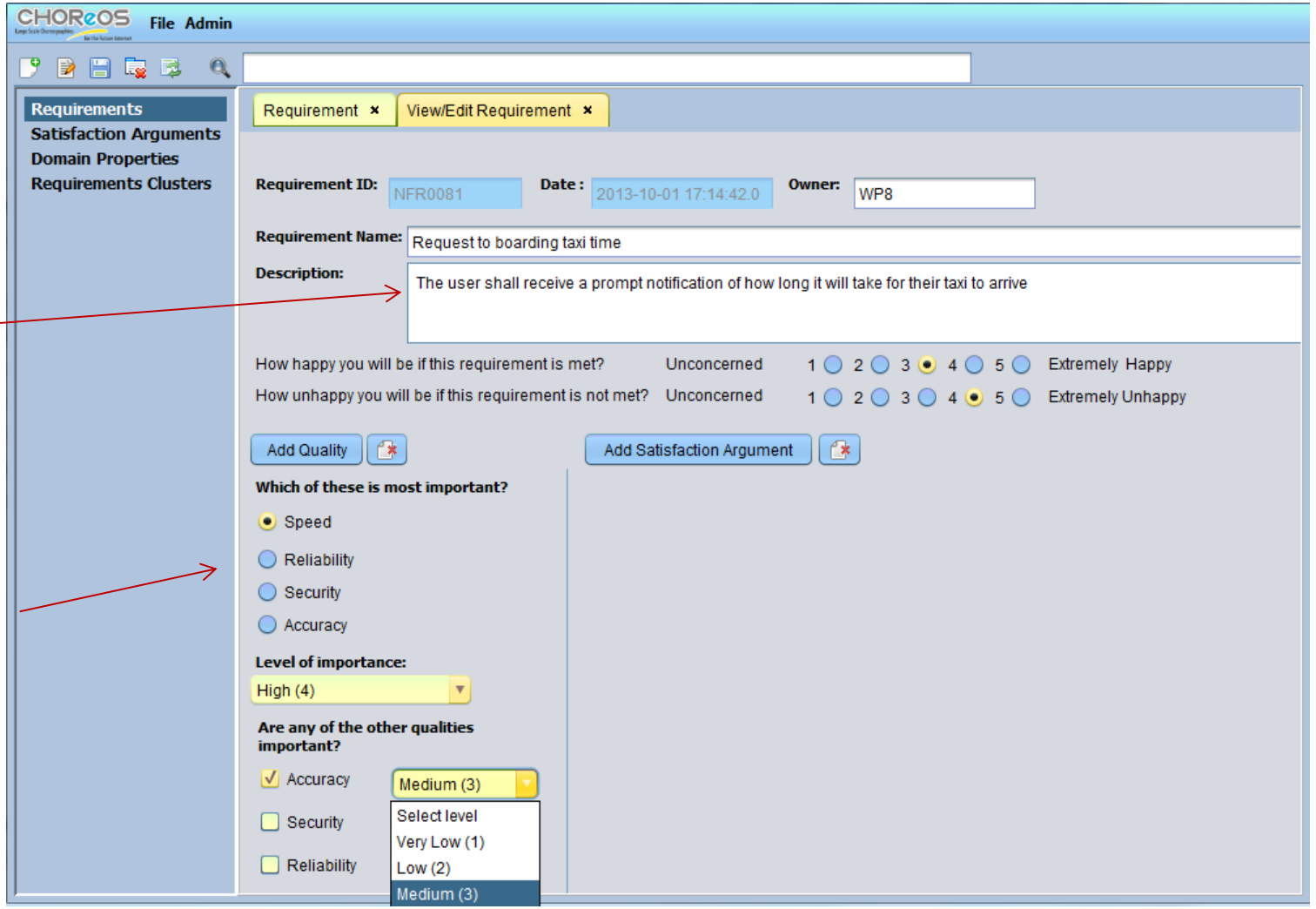
- Our experiences have shown that domain experts can write functional requirements in natural language
- However we do not believe that they can express measurable quality requirements such as performance and reliability as effectively.
- Therefore, need for tool support for expressing quality requirements with simple and clear UI

CITY UNIVERSITY LONDON

Expressing Natural Language Requirements

Free-text

Quality Prompts – Decided by project consortium



CITY UNIVERSITY LONDON

EST 1894

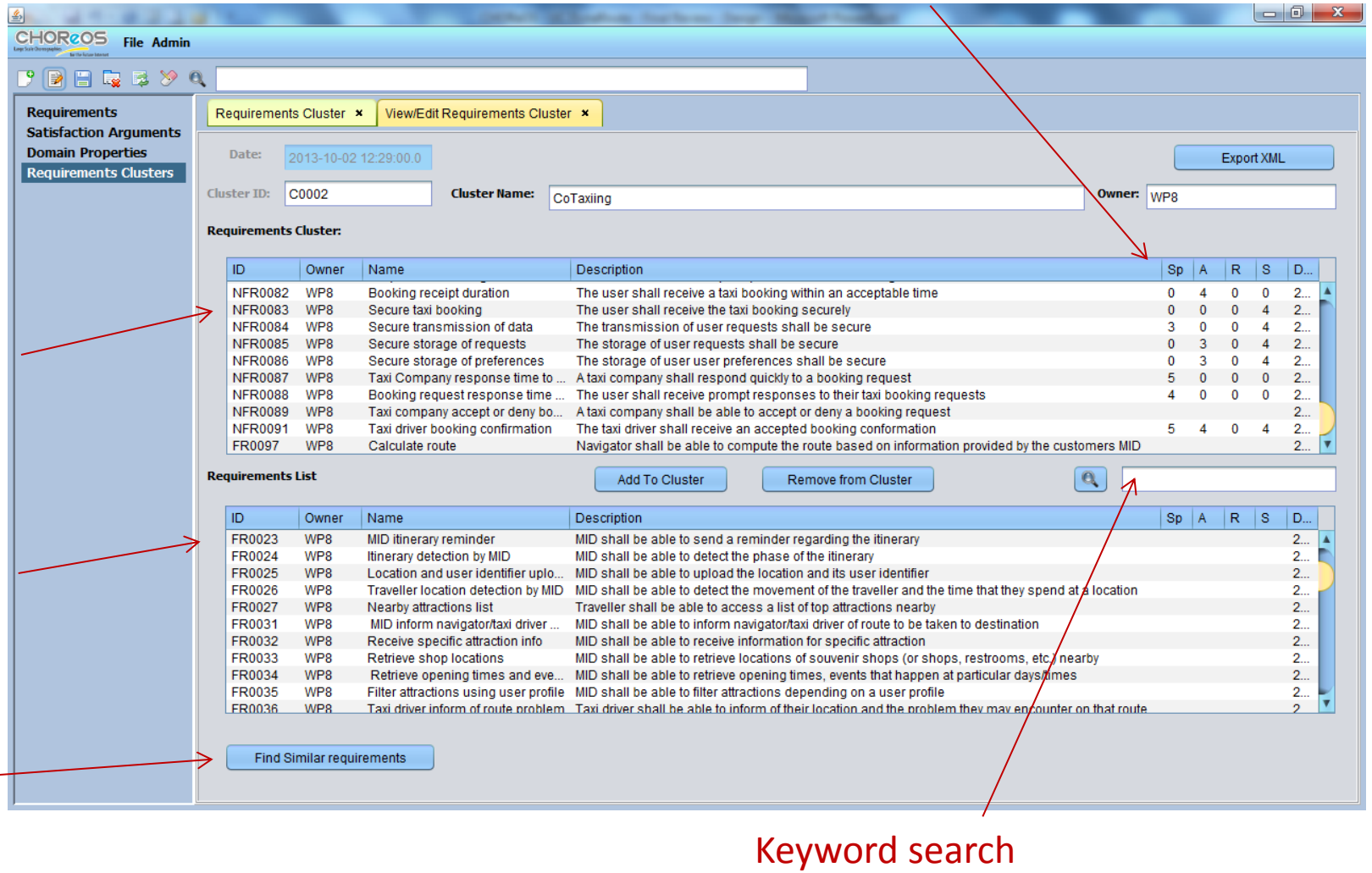
Clustering Requirements

- Users express requirements on systems rather than service choreographies.
- Potential for requirements expressing similar functions or qualities – overlaps and duplications need to be discovered and handled. (NB surrogate for service consumers)
- We wanted to provide them with support to cluster similar requirements that will map onto a single choreography diagram and its elements such as activities and roles.

CITY UNIVERSITY LONDON

EST 1894

Clustering Requirements



CITY UNIVERSITY LONDON

User task models for choreography specification

Domain expert
Requirement specification

Specify natural language requirements

User task models

CHOReOS Requirements Tool

Generate first-cut choreography diagram

MagicDraw

Refine service choreography

Domain expert
Choreography design

Specify non-functional properties on choreography diagram

Q4BPMN

CITY UNIVERSITY LONDON

EST 1894

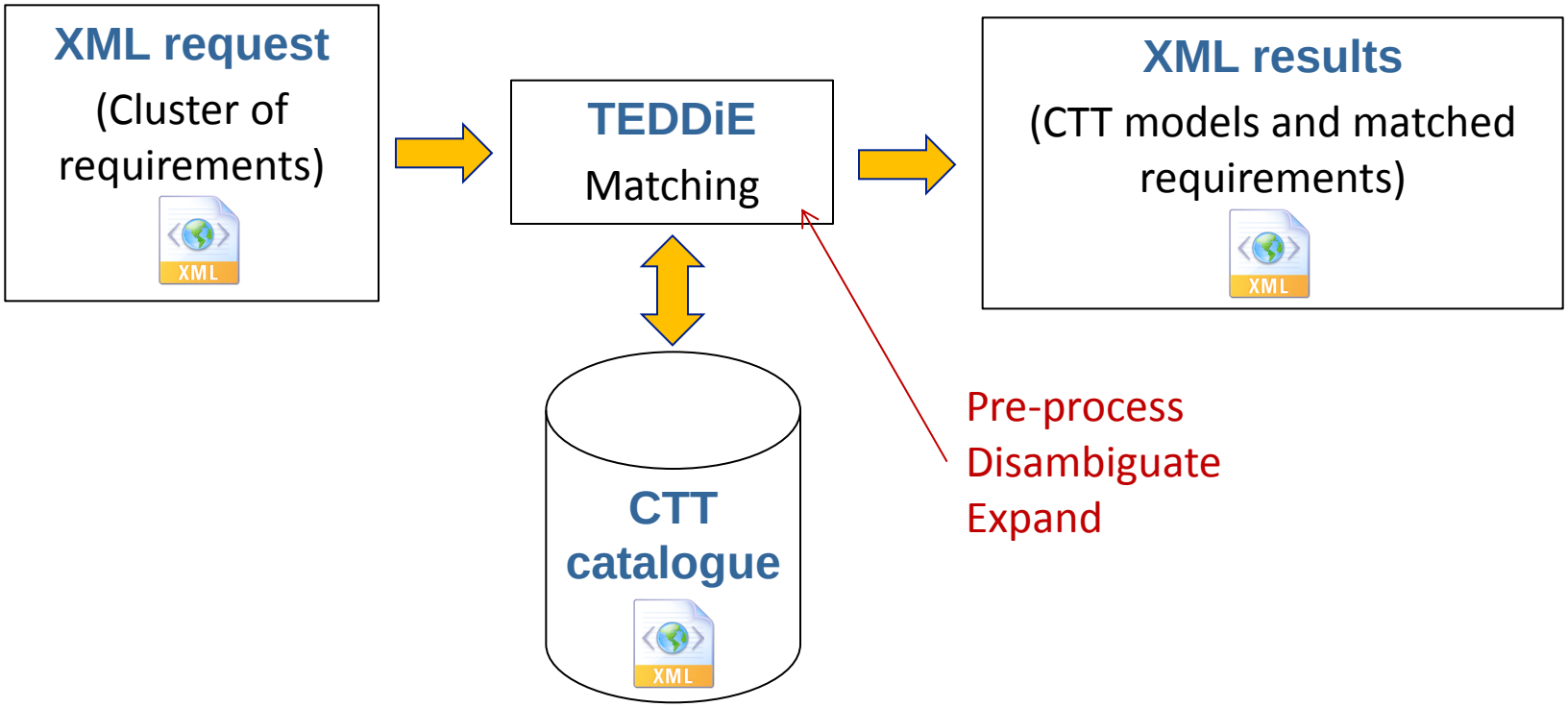
User task models for choreography specification

- Aim to bridge gap between the requirements and the choreography specification.
- Expand the user problem described by the requirements with classes of user tasks
- Help to provide an idea of the choreography tasks based on the set of requirements

CITY UNIVERSITY LONDON

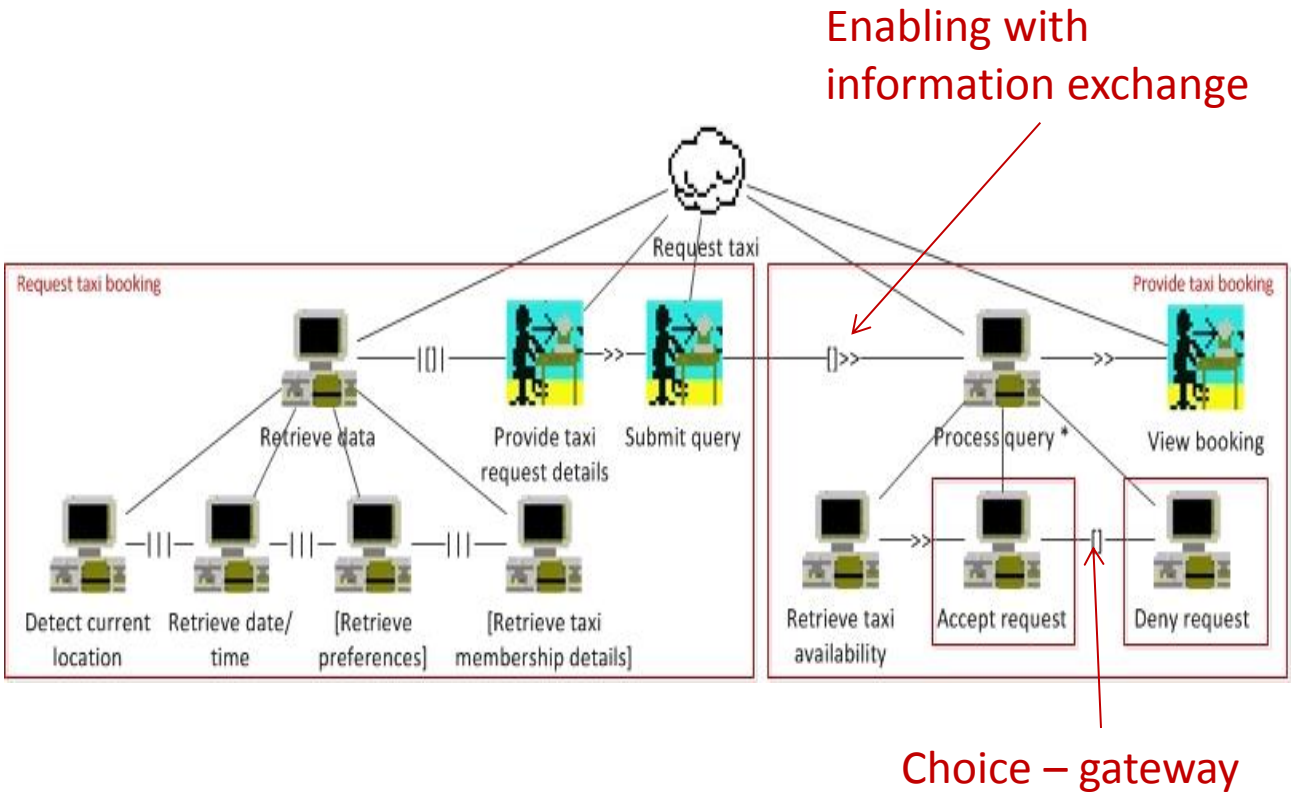
Match requirements to user task models

- The cluster of requirements is matched to a catalogue of user task models using a service called TEDDiE
- The retrieved user task models are returned with the matched requirements



CITY UNIVERSITY LONDON

Request Taxi CTT model



User task models expressed in the ConcurTaskTrees (CTT) formalism

Paterno F., Santoro C., Preventing User Errors by Systematic Analysis of Deviations from the System Task Model. International Journal of Human-Computer Studies, Vol. 56(2), 225-245 (2002)

XML output

```
<?xml version="1.0" encoding="UTF-8"?>
<Choreography name="CoTaxiing">
  - <ChoreographyTasks>
      <ChoreographyTask name="Request route" id="1">CTT tasks: Retrieve input data, Input request details, Submit query, Detect current
        location, Retrieve Date and time, [Retrieve preferences], [Retrieve existing routes], Verify address, Provide preferences. Service
        Classes: LocationService, TimeDateService, RoutingService, AddressService, TrafficDetection, MapData</ChoreographyTask>
      <ChoreographyTask name="Provide route" id="2">CTT tasks: Compute route, View route, Access map services, Retreive traffic data.
        Service Classes: LocationService, TimeDateService, RoutingService, AddressService, TrafficDetection, MapData</ChoreographyTask>
      <ChoreographyTask name="Request taxi booking" id="3">CTT tasks: Retrieve data, Provide taxi request details, Submit query, Detect
        current location, Retrieve Date and time, [Retrieve preferences], [Retrieve taxi membership details]. Service Classes:
        LocationService, TimeDateService, TaxiNotification, UserPreferences</ChoreographyTask>
      <ChoreographyTask name="Deny request" id="4"/>
      <ChoreographyTask name="Accept request" id="5"/>
      <ChoreographyTask name="Provide taxi booking" id="6">CTT tasks: Process query, View booking, Retrieve taxi availability, Accept
        request, Deny request. Service Classes: LocationService, TimeDateService, TaxiNotification, UserPreferences</ChoreographyTask>
      <Gateway name="Accept/deny" id="7" type="Exclusive"/>
    </ChoreographyTasks>
  - <Relationships>
      <Relationship type="sequence" startID="1" endID="2"/>
      <Relationship type="sequence" startID="2" endID="1"/>
      <Relationship type="sequence" startID="3" endID="7"/>
      <Relationship type="sequence" startID="7" endID="4"/>
      <Relationship type="sequence" startID="4" endID="3"/>
      <Relationship type="sequence" startID="7" endID="5"/>
      <Relationship type="sequence" startID="5" endID="6"/>
      <Relationship type="trace" startID="REQ2" endID="1"/>
      <Relationship type="trace" startID="REQ3" endID="1"/>
      <Relationship type="trace" startID="REQ6" endID="1"/>
        Etc ...
      <Relationship type="trace" startID="REQ39" endID="3"/>
    </Relationships>
  - <Requirements>
      <Requirement name="Pre arrange taxi arrival time(FR0003)" id="REQ1">Traveller shall be able to arrange beforehand when the taxi
        shall arrive</Requirement>
      <Requirement name="Reservation time for Taxi Company(FR0004)" id="REQ2">Taxi company shall be able to receive a confirmation a
        couple hours before the reservation time from MID</Requirement>
      <Requirement name="VIP taxi subscriptions(FR0006)" id="REQ3">Traveller shall be able to subscribe to VIP taxi subscriptions to
        enhance prioritization</Requirement>
      <Requirement name="MID location coordinates(FR0008)" id="REQ4">MID shall be able to transmit location coordinates to taxi
        company</Requirement>
      <Requirement name="Taxi type(FR0009)" id="REQ5">Traveller shall be able to choose type of car, big boot, air-conditioning,
        etc.</Requirement>
      <Requirement name="Taxi Driver English language(FR0010)" id="REQ6">Traveller shall be able to request a taxi driver who speaks
        English</Requirement>
      <Requirement name="Taxi green card(FR0011)" id="REQ7">Traveller shall be able to request a taxi with green card</Requirement>
      <Requirement name="MID send customer details(FR0012)" id="REQ8">MID shall be able to transmit traveller (customer) details to the
        taxi company</Requirement>
      <Requirement name="Taxi number(FR0013)" id="REQ9">MID shall be able to display the taxi number that will pick up the traveller
        (customer)</Requirement>
      <Requirement name="Taxi arrival time(FR0014)" id="REQ10">Traveller shall be able to know how long it will take for the taxi to
        arrive</Requirement>
        Etc ...
```

CITY UNIVERSITY LONDON

Generate first-cut choreography diagram

Domain expert
Requirement specification

Specify natural language requirements

User task models

CHOReOS Requirements Tool

Generate first-cut choreography diagram

MagicDraw

Domain expert
Choreography design

Refine service choreography

Specify non-functional properties on choreography diagram

Q4BPMN

CITY UNIVERSITY LONDON

EST 1894

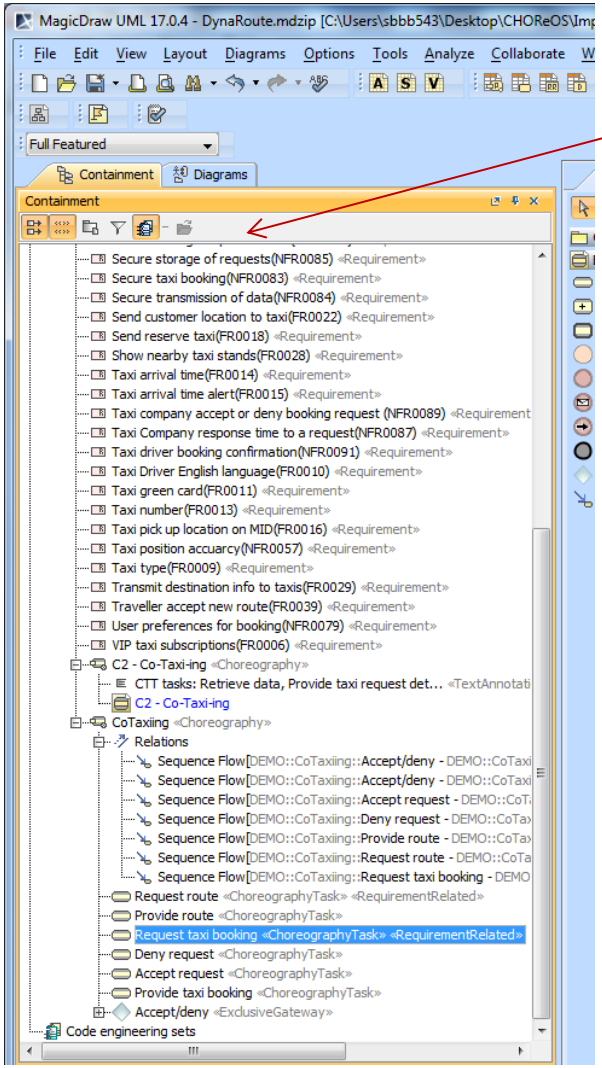
First-cut choreography diagram

- Our approach uses the **Business Process Model and Notation** (www.bpmn.org), which is an emerging standard for business process modeling and the specification of service choreographies.
- The choreography designer designs BPMN choreographies using the **MagicDraw visual modeling tool** (www.nomagic.com), which we configured to accept the XML files output from the Requirements Tool.
- As a result, the choreography designer receives **explicit requirements-based guidance for designing a service choreography** based on a first-cut template model annotated with requirements information.

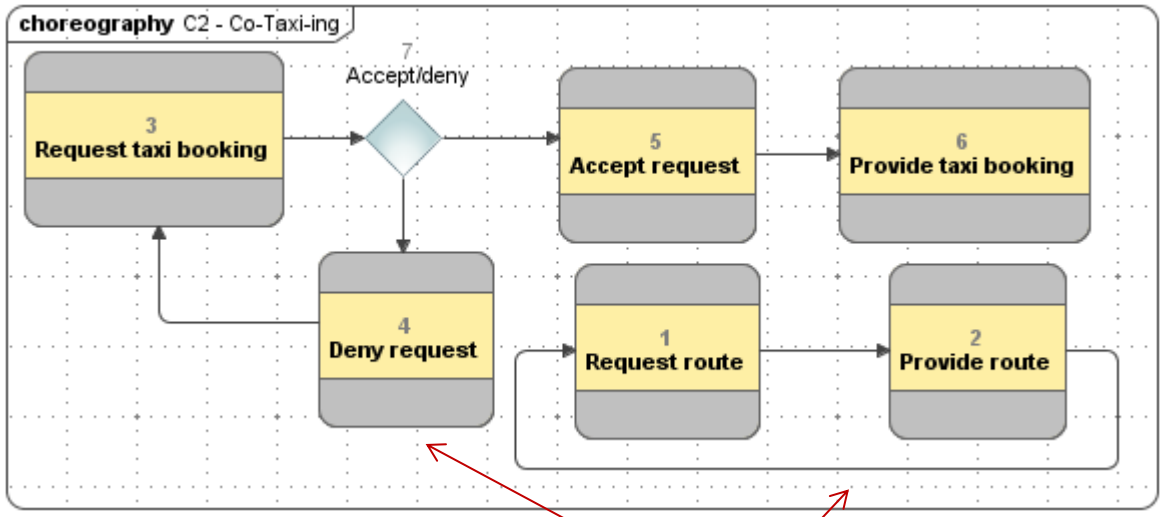
CITY UNIVERSITY LONDON

First-cut choreography diagram

Design advice



Containment tree



Output from two separate CTT models

CITY UNIVERSITY LONDON

EST 1894

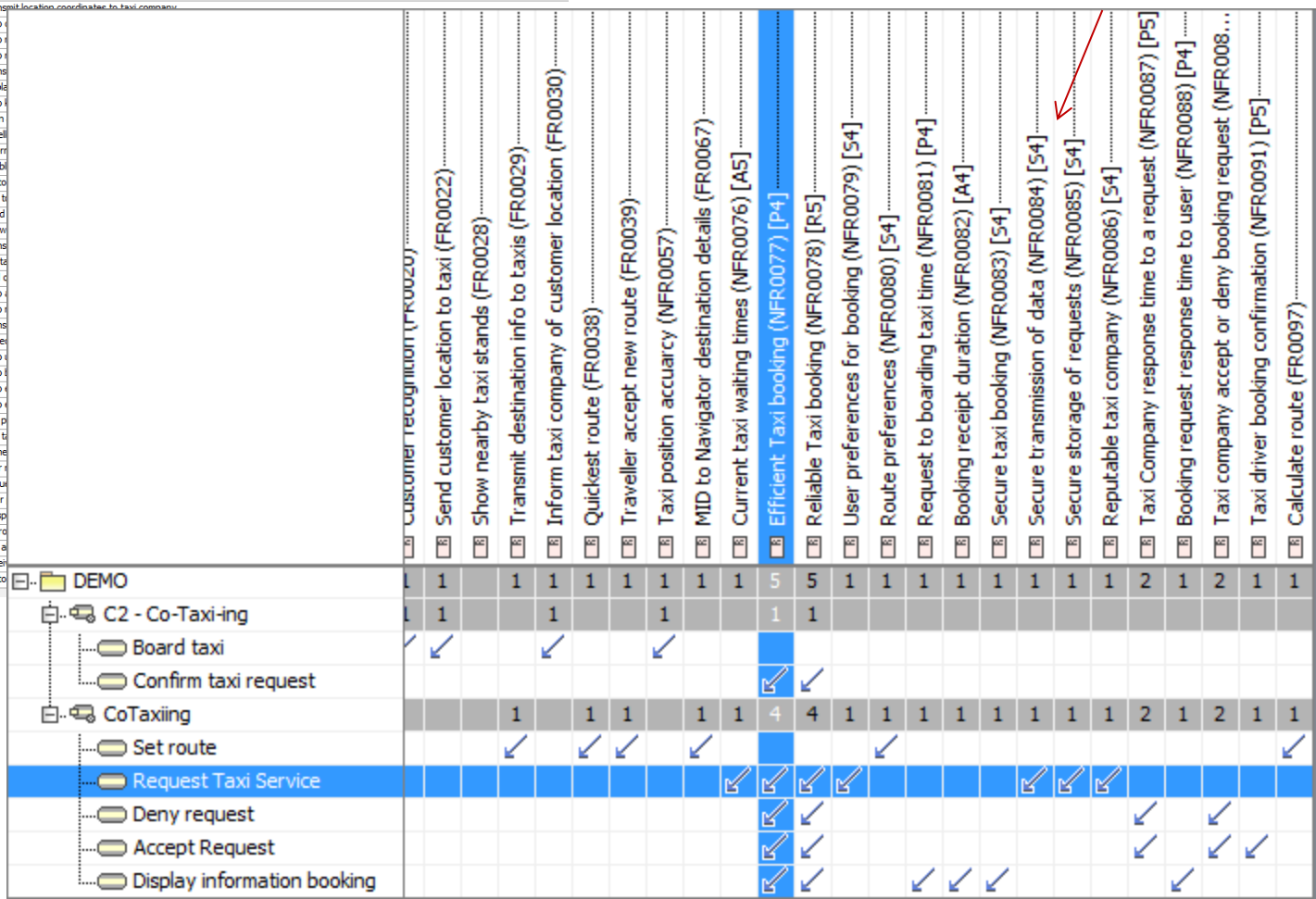
Mapping requirements to the choreography tasks

Requirements table

Quality scores

Requirements - task matrix

#	Id	Name	Text
1	REQ1	Pre arrange taxi arrival time(FR0003)	Traveller shall be able to arrange beforehand when the taxi shall arrive
2	REQ2	Reservation time for Taxi Company(FR0004)	Taxi company shall be able to receive a confirmation a couple hours before the reservation time from MID
3	REQ3	VIP taxi subscriptions(FR0006)	Traveller shall be able to subscribe to VIP taxi subscriptions to enhance prioritization
4	REQ4	MID location coordinates(FR0008)	MID shall be able to transmit location coordinates to taxi company
5	REQ5	Taxi type(FR0009)	Traveller shall be able to
6	REQ6	Taxi Driver English language(FR0010)	Traveller shall be able to
7	REQ7	Taxi green card(FR0011)	Traveller shall be able to
8	REQ8	MID send customer details(FR0012)	MID shall be able to trans
9	REQ9	Taxi number(FR0013)	MID shall be able to displa
10	REQ10	Taxi arrival time(FR0014)	Traveller shall be able to
11	REQ11	Taxi arrival time alert(FR0015)	Traveller shall receive an
12	REQ12	Taxi pick up location on MID(FR0016)	MID shall allow the travell
13	REQ13	MID inform of near taxi arrival(FR0017)	MID shall be able to inforr
14	REQ14	Send reserve taxi(FR0018)	Taxi company shall be abl
15	REQ15	Graphical coordinates on Navigator(FR0019)	Navigator shall be able to
16	REQ16	Customer recognition(FR0020)	Taxi driver shall be able t
17	REQ17	Send customer location to taxi(FR0022)	MID shall be able to send
18	REQ18	Show nearby taxi stands(FR0028)	MID shall be able to show
19	REQ19	Transmit destination info to taxis(FR0029)	MID shall be able to trans
20	REQ20	Inform taxi company of customer location(FR0030)	MID shall be able to conta
21	REQ21	Quickest route(FR0038)	MID shall be able to find c
22	REQ22	Traveller accept new route(FR0039)	Traveller shall be able to
23	REQ23	Taxi position accuarcy(NFR0057)	Traveller shall be able to
24	REQ24	MID to Navigator destination details(FR0067)	MID shall be able to trans
25	REQ25	Current taxi waiting times(NFR0076)	The user shall be provide
26	REQ26	Efficient Taxi booking(NFR0077)	The user shall be able to
27	REQ27	Reliable Taxi booking(NFR0078)	The user shall be able to
28	REQ28	User preferences for booking(NFR0079)	The user shall be able to
29	REQ29	Route preferences(NFR0080)	The user shall be able to
30	REQ30	Request to boarding taxi time(NFR0081)	The user shall receive a p
31	REQ31	Booking receipt duration(NFR0082)	The user shall receive a t
32	REQ32	Secure taxi booking(NFR0083)	The user shall receive the
33	REQ33	Secure transmission of data(NFR0084)	The transmission of user
34	REQ34	Secure storage of requests(NFR0085)	The storage of user requ
35	REQ35	Secure storage of preferences(NFR0086)	The storage of user user
36	REQ36	Taxi Company response time to a request(NFR0087)	A taxi company shall resp
37	REQ37	Booking request response time to user(NFR0088)	The user shall receive pro
38	REQ38	Taxi company accept or deny booking request (NFR0089)	A taxi company shall be a
39	REQ39	Taxi driver booking confirmation(NFR0091)	The taxi driver shall recei
40	REQ40	Calculate route(FR0097)	Navigator shall be able to



CITY UNIVERSITY LONDON

Refine choreography diagram

Domain expert
Requirement specification

Specify natural language requirements

User task models

CHOReOS Requirements Tool

Generate first-cut choreography diagram

MagicDraw

Refine service choreography

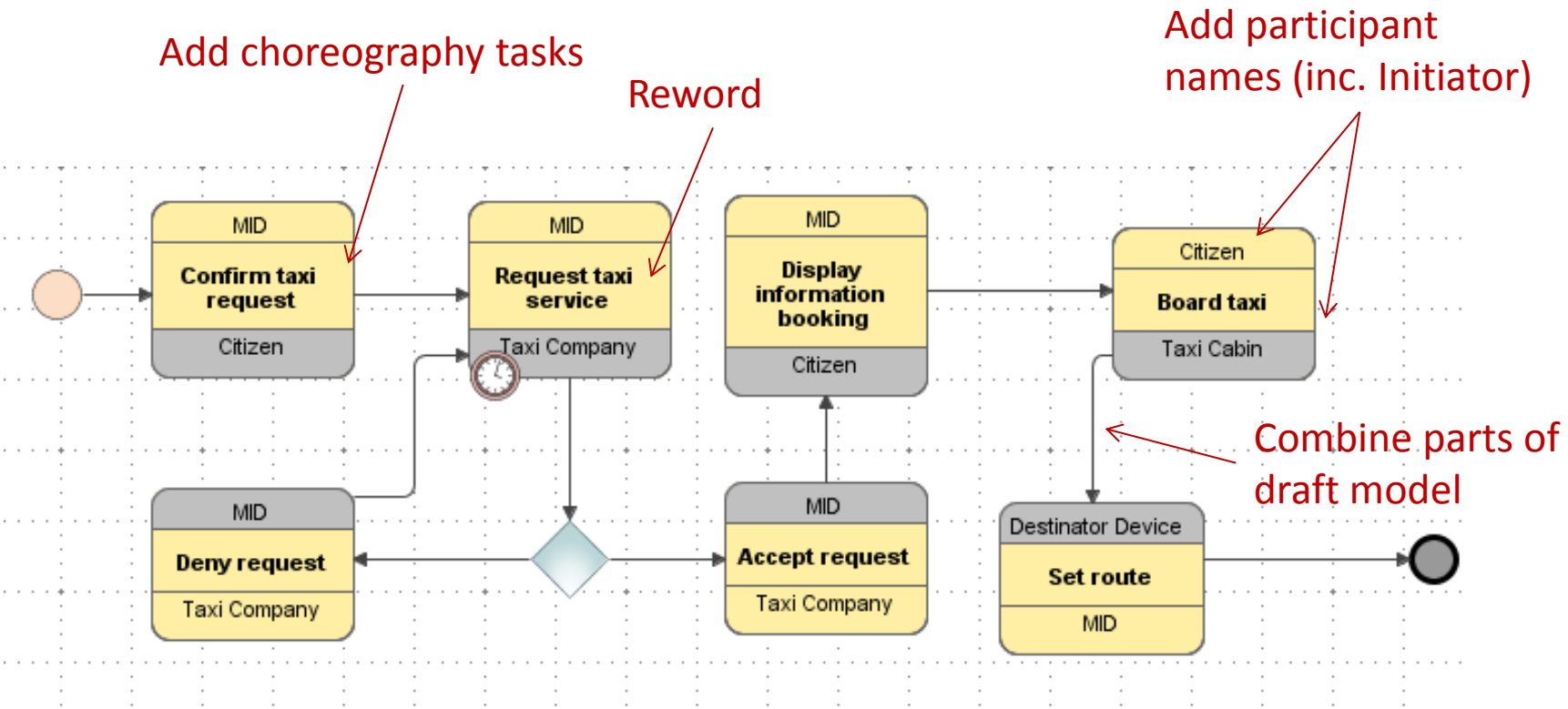
Domain expert
Choreography design

Specify non-functional properties on choreography diagram

Q4BPMN

CITY UNIVERSITY LONDON

Refine choreography diagram



CITY UNIVERSITY LONDON

Specify non-functional properties

Domain expert
Requirement specification

Specify natural language requirements

User task models

Generate first-cut choreography diagram

CHOReOS Requirements Tool

MagicDraw

Domain expert
Choreography design

Refine service choreography

Specify non-functional properties on choreography diagram

Q4BPMN

CITY UNIVERSITY LONDON

Q4BPMN (Quality for BPMN)

- Implemented in MagicDraw as a design tool for specifying non-functional properties within a BPMN choreography diagram
- The quality annotations can be applied to:
  - a single task («Q4Task»);
  - specific participant of a task («Q4Participants»); or
  - whole choreography («Q4Choreography»)
- It currently defines four classes of properties:
  - dependability properties
  - performance properties
  - security properties
  - accuracy properties
- Properties services entering the choreography will have to abide by.
- Potential participants can understand quality level required on their part.

CITY UNIVERSITY LONDON

Q4BPMN (Quality for BPMN)

- We identified **three main concerns** representing the different contexts of the quality requirements:
- ***Software System:*** quality attributes quantifying either the behavior of software components, or their interactions;
- ***Human–Computer Interaction***: quality attributes quantifying any interactions between a human and any part of the considered software system;
- ***Business Activities:*** quality attributes quantifying the admissible constraint used in order to characterize the activities of both the whole system (i.e. software + human related activities), and its actors from a business perspective.

CITY UNIVERSITY LONDON

EST 1894

Mapping values

- Some examples from the taxi management scenario

	Property Type	Dimension	Property Name	Levels				
				1	2	3	4	5
Software System	Accuracy	Time	timeAC_Software	3 sec	2 sec	1 sec	30 msec	5 msec
	Security	Encryption	encryptionSEC_Software			optional SSL	SSL	SSL
	Dependability	Downtime	downtimeDEP_Software	30 sec/1 d	30 sec/2 d	30 sec/1 w	30 sec/1 m	30 sec/6 m
	Performance	Time	timePER_Software	10 sec	5 sec	2 sec	500 msec	50 msec
	Trust							
Human–Computer Interactions	Accuracy	Time	timeAC_HCI	10 sec	8 sec	7 sec	5 sec	2 sec
	Security	Encryption	encryptionSEC_HCI			optional SSL	SSL	SSL
	Dependability	Downtime	downtimeDEP_HCI	30 sec/1 d	30 sec/2 d	30 sec/1 w	30 sec/1 m	30 sec/6 m
	Performance	Time	timePER_HCI	15 sec	5 sec	2.5 sec	1 sec	500 msec
	Trust							
Business Activities	Accuracy	Time	timeAC_Business	2 h	1 h	30 min	15 min	5 min
		Space	spaceAC_Business	optional 7.8 m	7.8 m	5 m	3 m	3 m
	Security							
	Dependability	Downtime	downtimeDEP_Business	5 min/1 d	5 min/2 d	5 min/1 w	5 min/1 m	5 min/6 m
	Performance	Time	timePER_Business	optional 1 h	optional 30 min	15 min	5 min	90 sec
	Trust	Number of Re-sources	resourcesTRU_Business	10 Units	30 Units	50 Units	70 Units	100 Units

CITY UNIVERSITY LONDON

EST 1894

Q4BPMN (Quality for BPMN) Example

Choreography element, Name	Original requirement Name, ID, description, quality	Quality Property
Task Request Taxi Service	MID send customer details (NFR0012): *MID shall be able to transmit traveller (customer) details to the taxi company* [Performance 5]	`timePer_Software_L5` ▪ isHard = true ▪ metrics – DefaultMaxDurationMetric ▪ nature = PRESCRIPTIVE ▪ operator = LESS EQUAL ▪ propertyClass – PERFORMANCE ▪ unit = "msec" ▪ value = "50"
Participant property Taxi Company	Reputable taxi company (NFR0086): *The taxi company shall have an established reputation* [Security 4]	`taxiCompanyBusinessTrust` ▪ NF Properties = resourcesTRU_Business_L4 ▪ ParticipantRef = Taxi Company

CITY UNIVERSITY LONDON

Software monitors from Q4BPMN

- Q4BPMN enables us to map quality requirements into monitorable properties on a choreography model, a task, participants in a task
- Generated monitoring module determines whether a property associated to a quality requirement is satisfied using observed data and messages.
- The approach uses event-based monitor includes a complex event-processing engine based on Drools Fusion.
- E.g. Latency property
  - e_1 event starts incomingRequest, and finishes with outgoingResponse event
  - Rules monitor the completion of events and calculate average completion times.

CITY UNIVERSITY LONDON

Lessons Learned (1)

- Expressing requirements
  - 97 requirements successfully specified and used to generate meaningful first-cut choreography diagrams.
  - Secondary qualities not used in practice.
- Clustering requirements
  - Natural language processing on similarity algorithm took too long (over 5 minutes per invocation on the requirements set)
  - Matrix a solution?
- Retrieving task models for draft BPMN model
  - Varying success on CTT models to draft BPMN. Need to expand catalogue and evaluate the effectiveness in further studies.

CITY UNIVERSITY LONDON

Lessons Learned (2)

- Mapping requirements to choreography tasks
  - In our example, 38 out of the 40 clustered requirements were automatically mapped to choreography tasks and provided a useful starting point for the user.
  - Mapping issues however. Mapped to CTT model, not individual choreography tasks
  - Requirements granularity issues – high level requirements over several tasks vs low level requirements.
  - ***NFR0077: The user shall be able to use the taxi booking system efficiently*** mapped across 5 choreography tasks
  - Need decomposition and requirements trace. Via Satisfaction Arguments?

CITY UNIVERSITY LONDON

Lessons Learned (3)

- Reconciling the user expressed quality scores with actual values for the quality properties – a big challenge.
  - Tried to use values from literature but wasn't too successful.
  - A lot of input from domain experts needed. Ultimately those with knowledge of the taxi domain set the values.
- Integration of tools
  - Successful in a forwards process, with trace to the original requirements but...
  - no backwards compatibility between MD and the requirements tool – reduces the usefulness of the tool support.
- No need for dedicated IT professionals to provide the skills needed for architectural design and software engineering? – yes and no

CITY UNIVERSITY LONDON

Conclusion

- End-to-end approach for generating service-based systems that can be traced to their originating requirements
- Integrated toolkit based on BPMN modelling in MagicDraw
- Task models provide domain knowledge and functional constraints for choreography design
- QoS – quality requirements that are not measurable are mapped to measurable, and thus monitorable, non-functional properties (Q4BPMN profile)
- Future work?
  - Address lesson learned
  - Further application to new case studies