Requirements Driven Software Service Evolution

A feature based approach

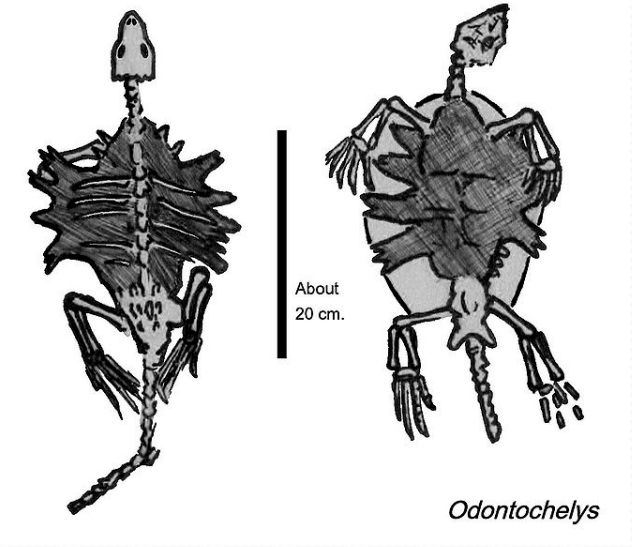
Feng-Lin Li
DISI, University of Trento, Italy
2012-06-21

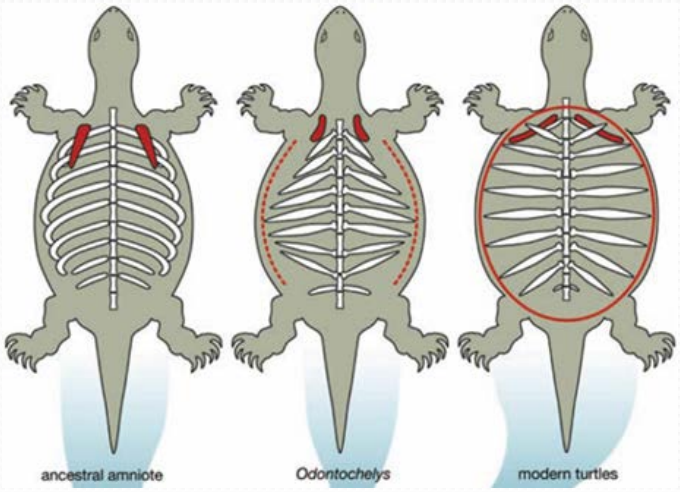
Outline

- *Motivation & Background*
- *Problem Statement*
- *The Feature based Approach*
- *A Case Study*
- *Discussion & Conclusion*

A Biological Evolution Example

- *The evolution of turtle shell*
  - *Odontochelys (oldest turtle)*
  - *The turtle shells formed from the underside - plastron (chest) first*
  - *And then grew bony extensions of ribs and bone formation above backbones*
  - *Existing features are modified and put into second use.*





Background

- $D, S \models R$
  - D: domain assumptions, S: specification, R : requirements
  - Specification S is made up of a set of services
  - A service is composed of a group of features
  - A feature can be interpreted as both a cohesive set of individual requirements or a unit of system functionalities
  - ***Feature is a set of cohesive specification items***.
- *Goal Oriented Requirements Model*
- *Feature & Feature Model*
- *Service*

Problem Statement

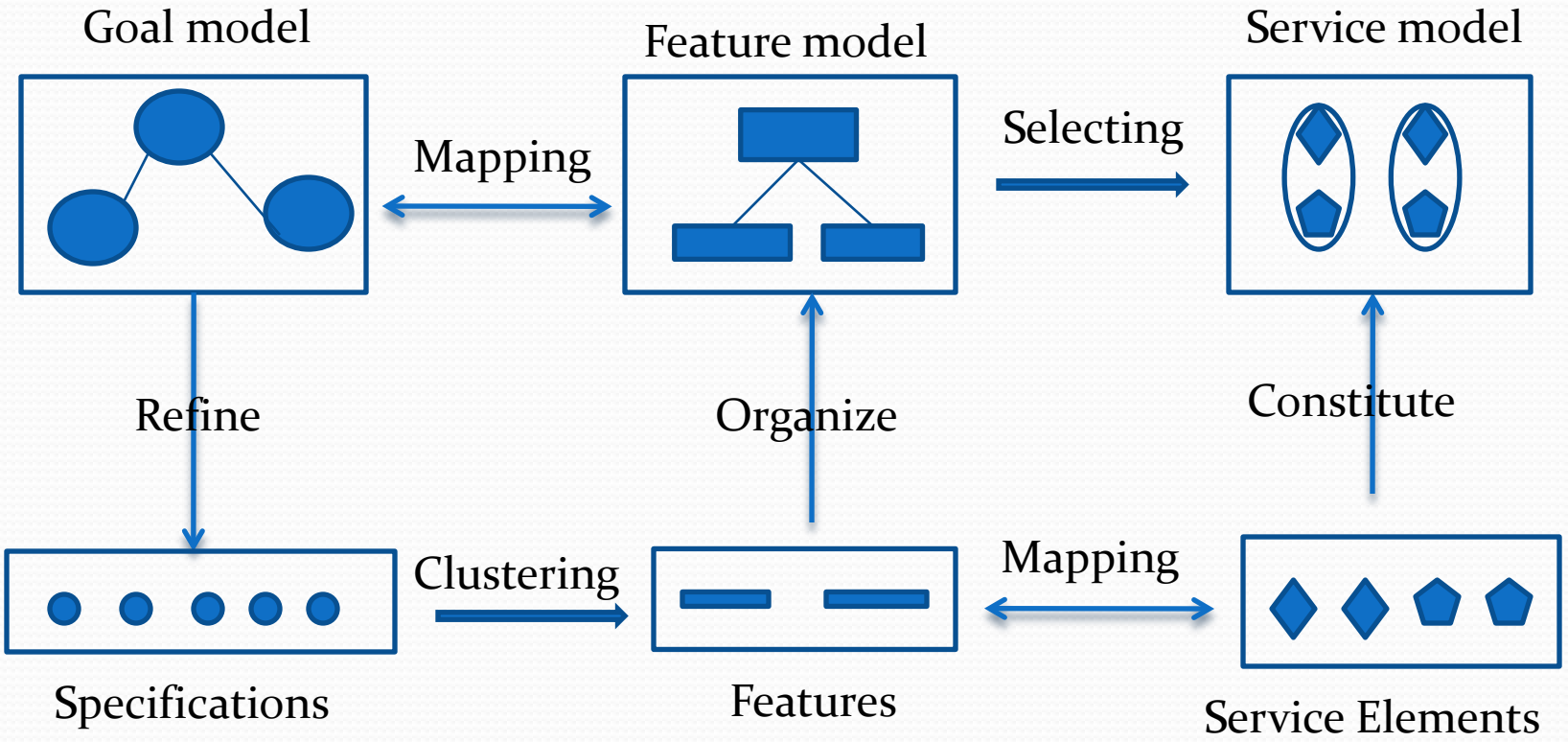
- ***Propagation***:
  - Given D, $S \models R$ holds, if requirement R changes to R', how can we find a new specification S' so that D, $S' \models R'$ holds?
- ***Traceability***:
  - Given D, $S \models R$ holds, if the domain assumption D changes to D', how to find a new specification S', so that D', $S' \models R$ still remain true?
  - Given D, $S \models R$ holds, if specification S changes to S', would the entailment D, $S' \models R$ still remain true?
- ***Non-Functional Requirements***:
  - Given D, $S \models R$ holds (i.e. functional requirements are fulfilled), for specified non-functional requirements R_q, how to find a differentiated specification S_d , so that D, $S_d \models R$, R_q holds?

The feature based approach

- General Framework
  - Assumption
    - A goal model
    - A domain feature model
  - Basic Idea
    - Mapping goal to features (*n*-to-*m* relationship)
    - Formalizing goal and feature
    - Reasoning the *support* relationship between feature and goal.
    - Mapping feature to service element
    - Clustering features into service

The feature based approach

- General Framework



The feature based approach

- Illustration
  - Mapping goal to feature

    $F_1 \cup F_2 \; supports \; G_1$

    $G_1 \; requires \; F_1$
  - Reasoning
    - Based on certain domain assumptions, together with the feature, it is able to satisfy a specified goal
  - Mapping feature to service
    - A feature *f* can be mapped to an operation in a WSDL service

The feature based approach

- Illustration

$$\textbf{Goal of Customer}: Happens(payOrder, t_0) \wedge \neg HoldsAt(ItemDelivered, t_0) \wedge t_0 < t \rightarrow HoldsAt(ItemDelivered, t)$$

$$\textbf{Goal of Seller}: Happens(startDeliver, t_1) \wedge \neg HoldsAt(RecepitAssigned, t_1) \wedge t_1 < t \rightarrow HoldsAt(ReceiptAssigned, t)$$

$$\textbf{Feature} - \textbf{1}: Happens(dispatchItems, t_2) \wedge \neg HoldsAt(ItemDelivered, t_2) \wedge t_2 < t \leq t_2 + d \rightarrow HoldsAt(ItemDelivered, t)$$

$$\textbf{Feature} - \textbf{2}: Happens(assignReceipt, t_3) \wedge \neg HoldsAt(ReceiptAssigned, t_3) \rightarrow HoldsAt(ReceiptAssigned, t_3 + 1)$$

The feature based approach

- Methodology
  - (1) Identify Goals (resolved)
    - Using goal model to represent requirements, the requirement problem could be resolved by reasoning on goal model (how to choose leaf goals so that the root goals will be satisfied) [1][2]
  - (2) ***Connecting goals with features*** (Key Challenge)
    - Transforming goal model to feature model [3][4]
    - Deriving specification from goal (requirements), clustering specification items into feature.

The feature based approach

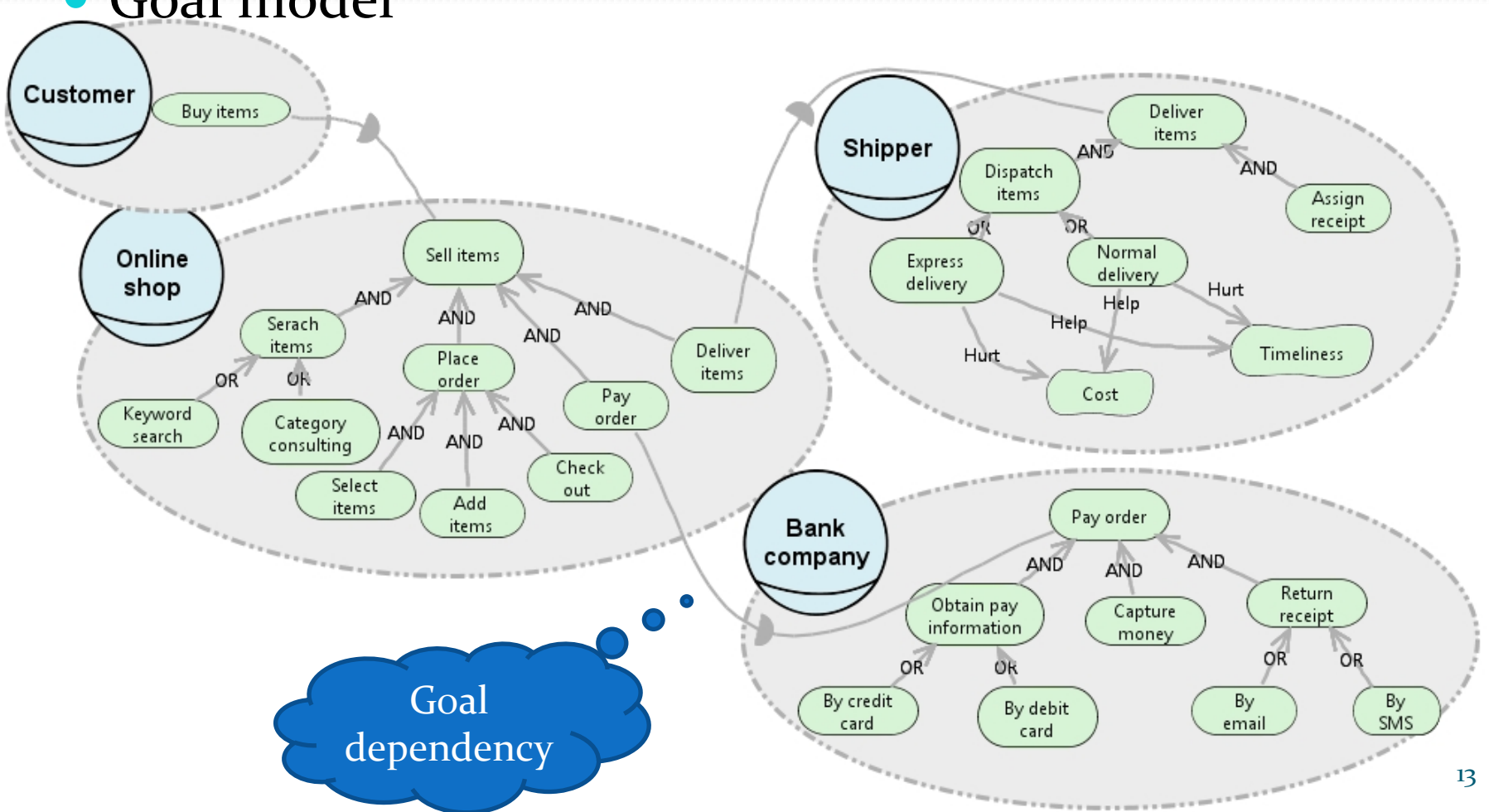
- General Ideas (cont.)
  - (3) Modeling feature [6]
    - Types: capability, quality, environment, implementation
    - Attributes: bind-time,
    - Logical Expression - LTL, pre- and post- condition
    - State machine/chart
  - (4) Modeling Service
    - Input-Output-Precondition- Effect(Post-condition) IOPE (i.e. WSDL)
    - Finite State machine
  - (5) Mapping feature to service [5]
    - Structural: mapping feature to service elements

A case study

- Online shop
  - It is owned by a store selling different kinds of items, such as book, audio tape and CD.
  - Roles: *customer*, *merchant*, *bank*, and *shipper*. For each role, there would be corresponding software service(s) play it.
  - Customers are able to query items and specify their orders; *merchant* could handle orders, use the *bank* service to deal with payment transactions and depend on *shipper* to deliver physical items to customers.

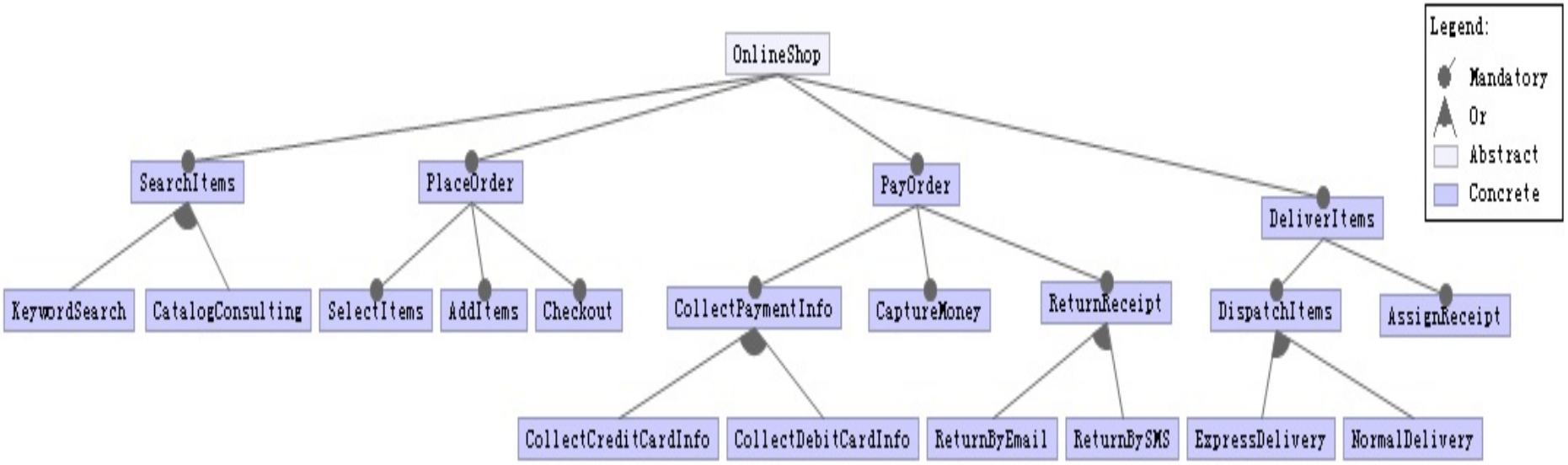
A case study

- Goal model



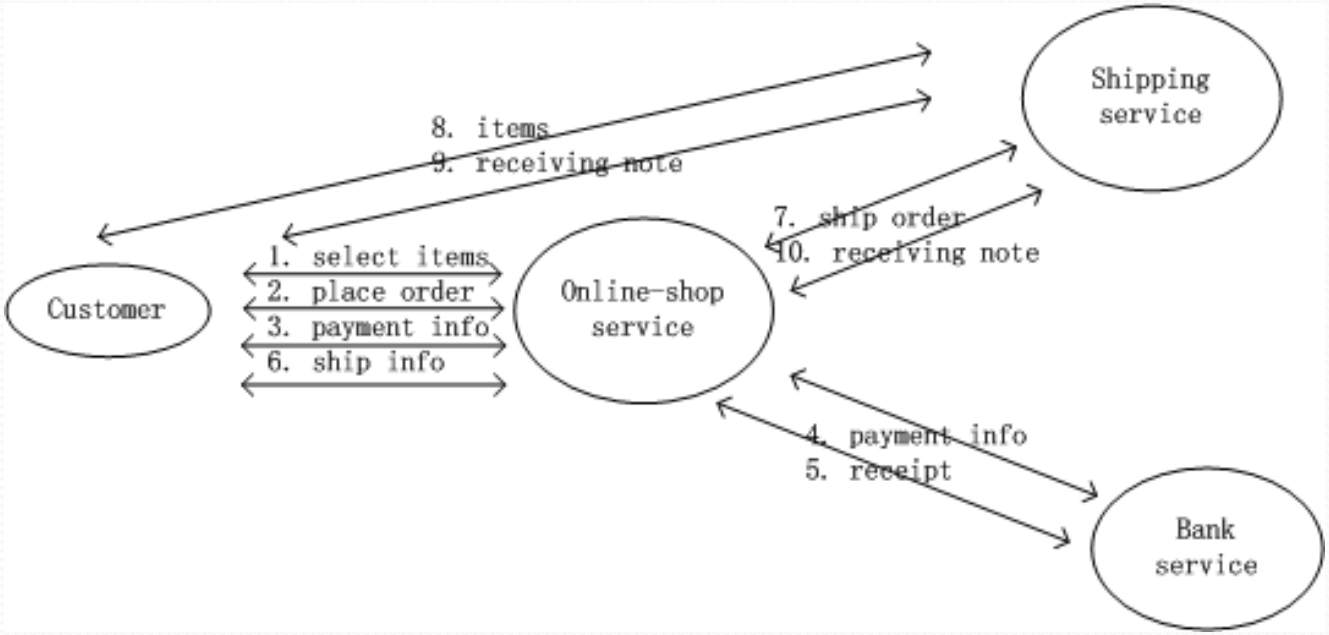
A case study

- Feature Model



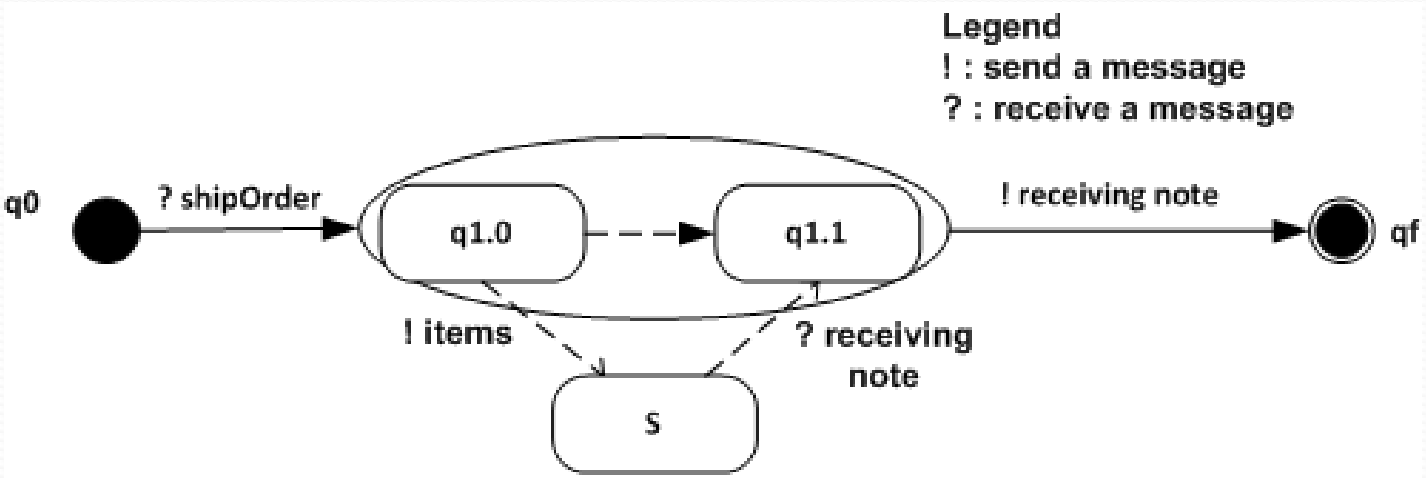
A case study

- A possible process



A case study

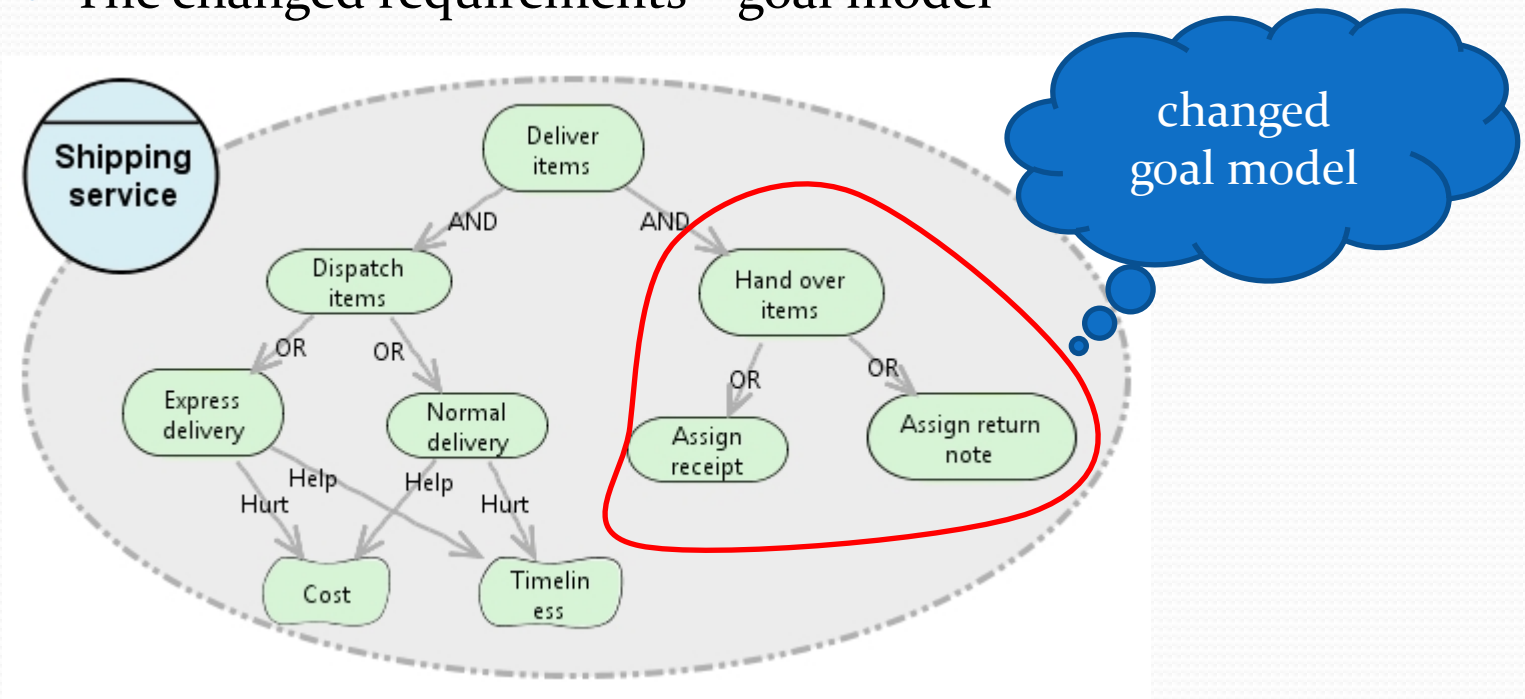
- A FSM model for the *shipping* service



A Finite State Machine model for the *Shipping service*

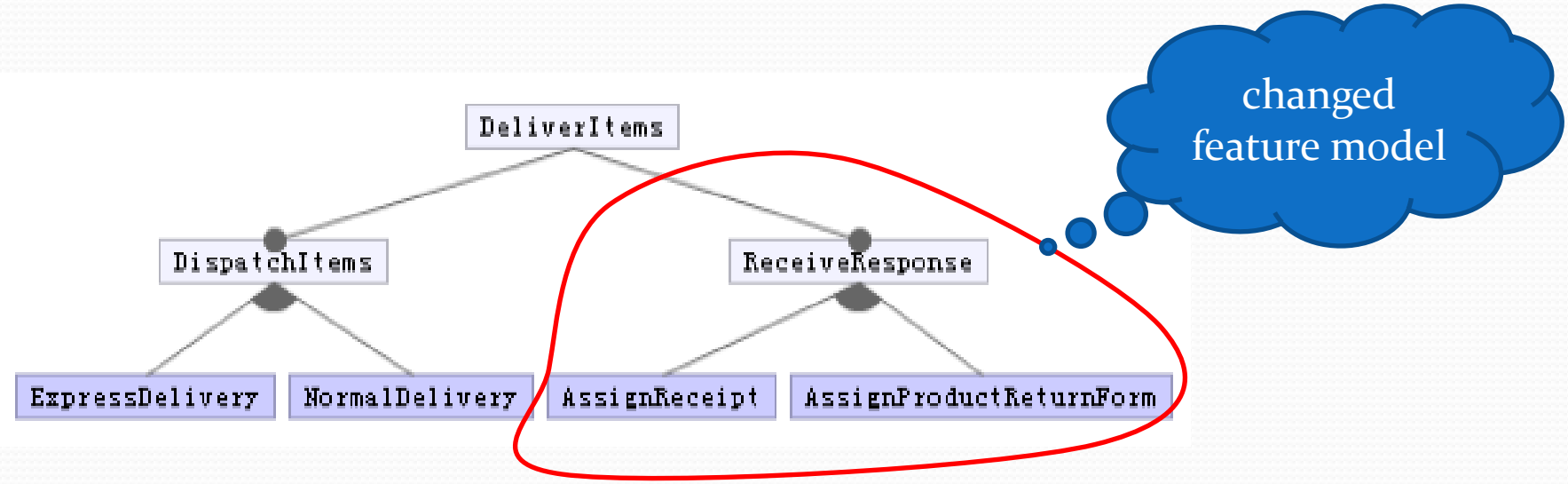
A case study

- A functional-requirements driven evolution scenario
  - When a customer finds out that the items are broken, he/she may won't accept the items and assign the receiving note.
  - The changed requirements – goal model



A case study

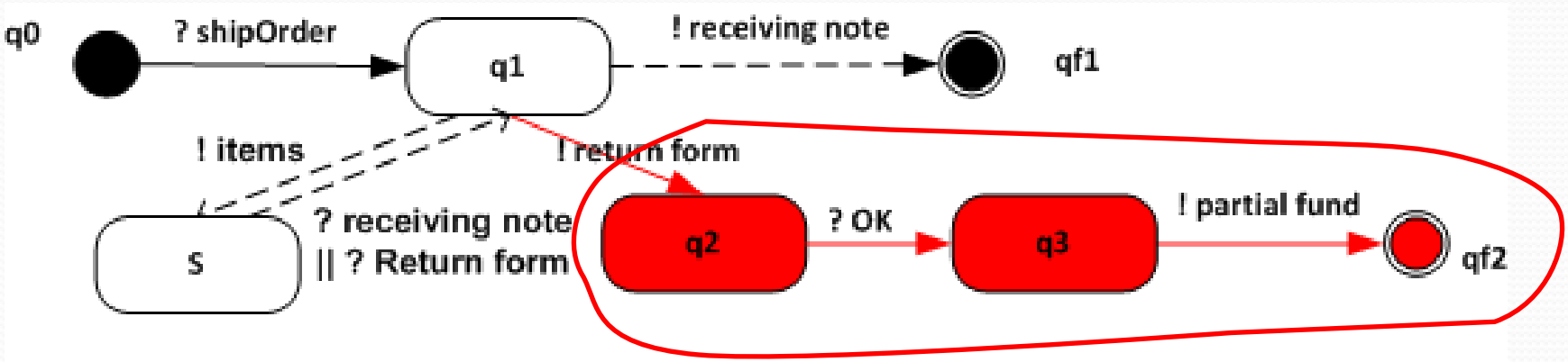
- A functional-requirements driven evolution scenario
  - The changed feature configuration



The changed feature model

A case study

- A functional-requirements driven evolution scenario
  - The changed service model
  - In addition, the service can described use a WSDL model, i.e., a set of operations with input and output, a feature can be mapped to operations.



The evolved Finite State Machine model for the *Shipping service*

Discussion and Conclusion

- Key Challenges:
  - Transforming goal into feature
    - How to establish the connection between goal and feature ?
  - Specifying feature and modeling service
    - How to relate feature with service?
  - How to predict and measure the non-functional requirements?
    - Predicting : Bayesian Net-Work, Personal Construct Theory
    - Monitoring : streaming event processing

Reference

- [1] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani, "Reasoning with goal models," *Conceptual Modeling—ER 2002*, pp. 167–181, 2003.
- [2] R. Sebastiani, P. Giorgini, and J. Mylopoulos, "Simple and minimum-cost satisfiability for goal models," in *Advanced Information Systems Engineering*, 2004, pp. 675–693.
- [3] Y. Yu, J. C. . do Prado Leite, A. Lapouchnian, and J. Mylopoulos, "Configuring features with stakeholder goals," in *Proceedings of the 2008 ACM symposium on Applied computing*, 2008, pp. 645–649.
- [4] Y. Yu, A. Lapouchnian, S. Liaskos, J. Mylopoulos, and J. C. S. . Leite, "From goals to high-variability software design," in *Proceedings of the 17th international conference on Foundations of intelligent systems*, 2008, pp. 1–16.
- [5] T. Nguyen and A. Colman, "A Feature-Oriented Approach for Web Service Customization," in *Web Services (ICWS), 2010 IEEE International Conference on*, 2010, pp. 393–400.
- [6] A. P. Felty and K. S. Namjoshi, "Feature specification and automated conflict detection," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 12, no. 1, pp. 3–27, 2003.