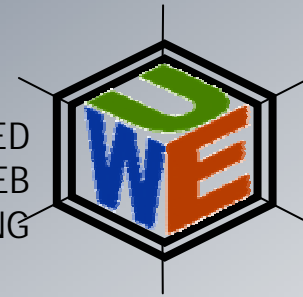


Object-Oriented
Hypermedia Method

UML-BASED
WEB
ENGINEERING



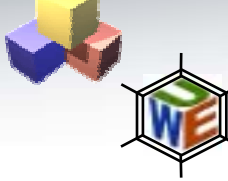
Modeling Web Business Processes with OO-H and UWE

Nora Koch
Andreas Kraus

Ludwig-Maximilians-Universität
München, Germany


Cristina Cachero
Santiago Meliá

Universidad de Alicante
Spain

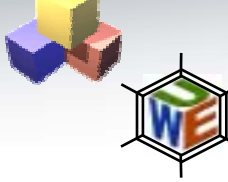


- Workshop goal:
 - Design & implementation of Web applications which support business processes or workflows

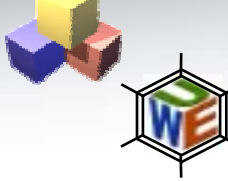
- Different Web Modeling Approaches
 - Include similar concepts although with different nuances
 - Use different modeling techniques
 - Represent similar concepts with different visual notation

- 
 - Prevent practitioners from their use
 - Prevent researchers from reaching agreements once the proposals have been settled

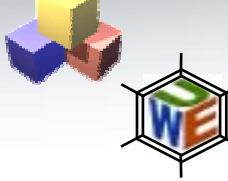
- Our additional goal:
 - A common solution for OO-H & UWE for Web Business Processes



- About OO-H & UWE
- Looking for a Common Solution for Modeling Web Business Processes
- Business Processes & Requirements Analysis
- Analysis of Web Business Processes
- Design of Web Business Processes
- Conclusions & Future Work



- OO-H is a Web design methodology which provides the designer with the semantics and notation necessary for the development of personalized Web-based interfaces
- The method is supported by the CASE tool **VisualWADE**
 - VisualWADE includes a set of model compilers to provide automatic code generation capabilities and a prototyping environment
- OO-H is made up of six complementary views
 - requirements
 - conceptual
 - navigational
 - presentational
 - **process**
 - architectural



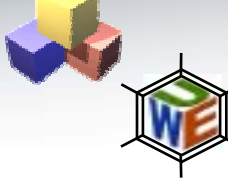
- UML-based Web Engineering (UWE) is an object-oriented approach based on **UML** and **Unified Process**
- Main focus of UWE is the systematic design followed by a semi-automatic generation of Web applications
 - Systematic design supported by CASE-tool **ArgoUWE** – an extension of ArgoUML
 - Semi-automatic generation supported by **UWEXML** – a model-driven Code Generator for deployment to an **XML** publishing framework
- The common language for data interchange is defined as a conservative extension of the UML metamodel



OpenUWE
development
environment

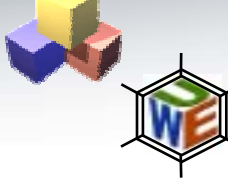


UWE
metamodel



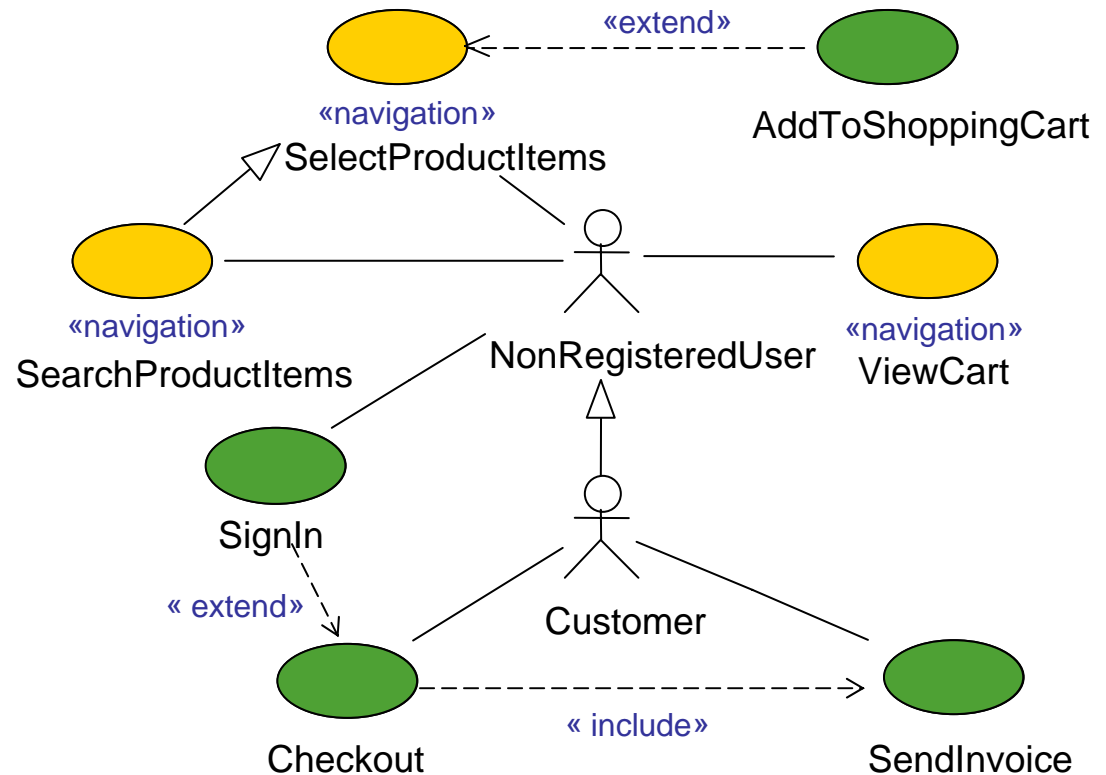
- Decisions made for process modeling
 - Analysis
 - a separate process model complementing other (domain) models
 - Design
 - navigation model supporting process vs.
 - process model complementing navigation

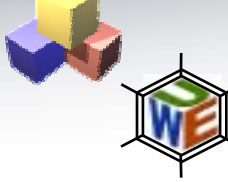
- Lessons learned
 - UML provides the necessary mechanisms to define process-aware Web applications
 - Navigation models are well suited both
 - to support and
 - to complement process concerns



Modeling requirements of the Amazon-example

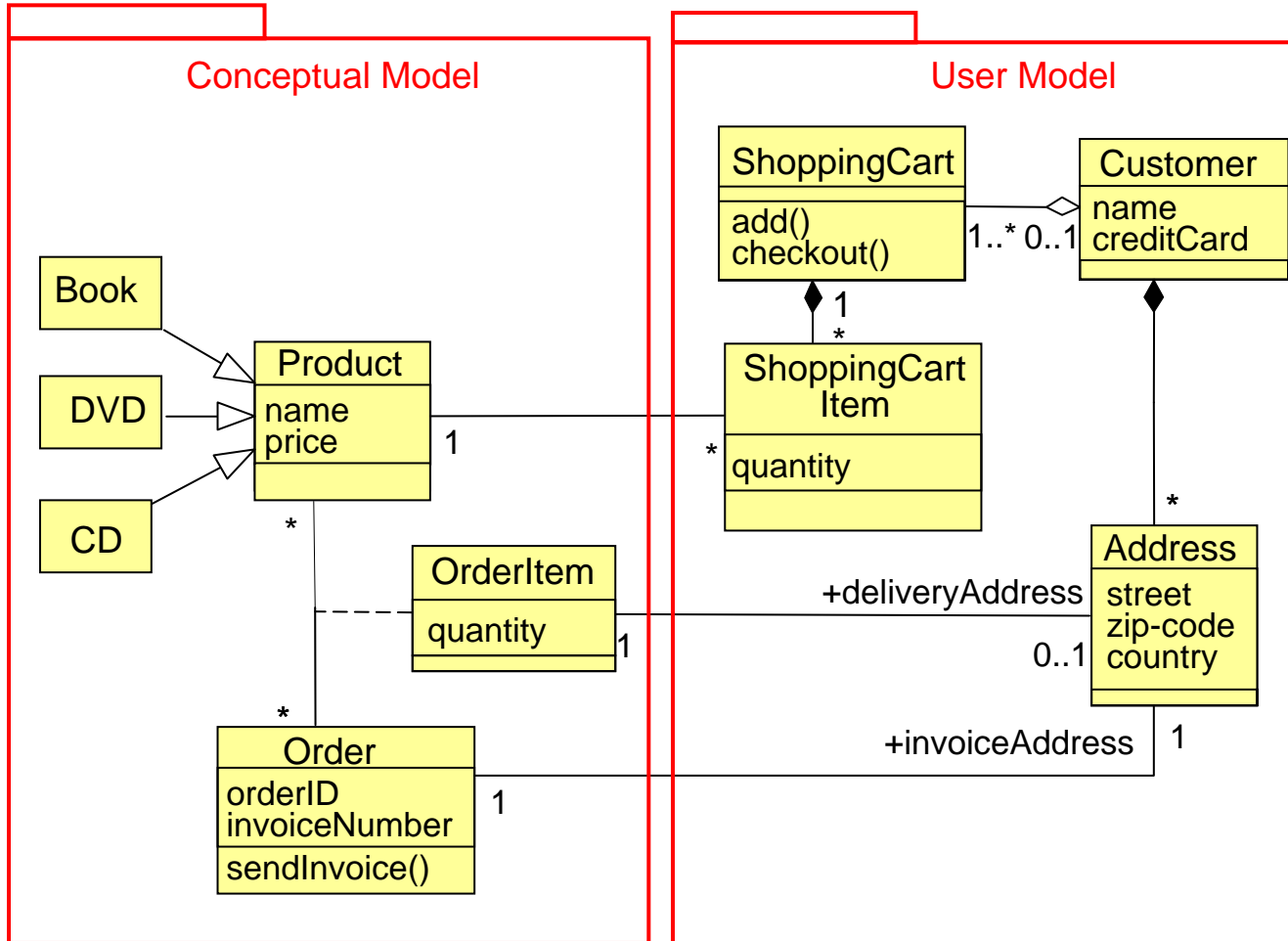
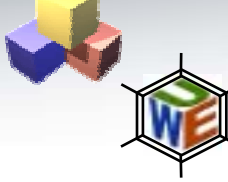
- Representation: UML Use Case Diagram
- Inclusion of a «navigation» stereotype associated with use cases (based on *Baresi et al. 2001*)
- Non-navigational use cases are candidates to further refinement by means of an associated process model





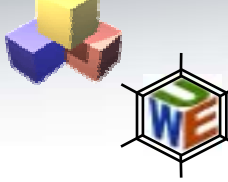
- About OO-H & UWE
- Looking for a Common Solution for Modeling Web Business Processes
- Business Processes & Requirements Analysis
- **Analysis of Web Business Processes**
 - Conceptual Model
 - Process Model
- Design of Web Business Processes
- Conclusions & Future Work

Conceptual Model

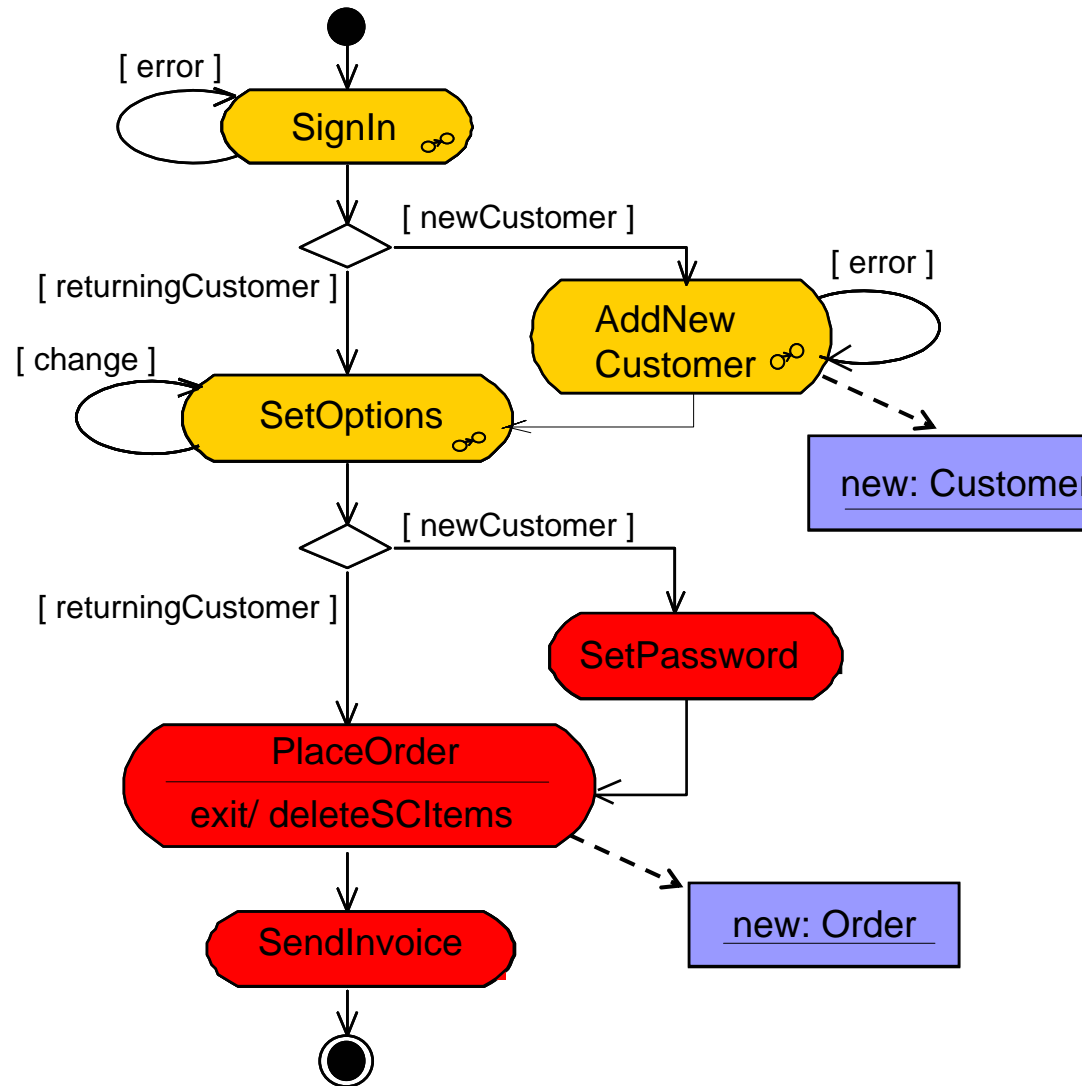


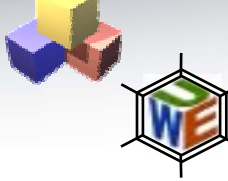
- Representation: UML Class Diagram
- Separation between domain and user concerns (personalization)

Process Model

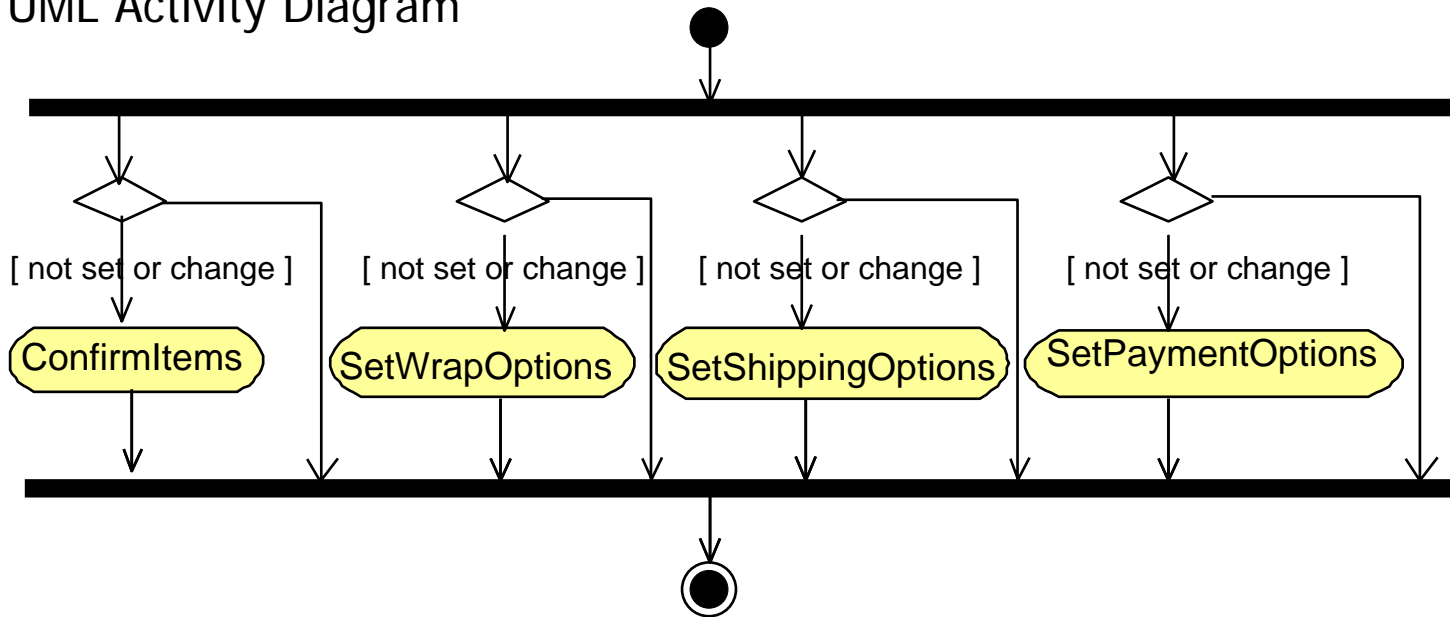


- Representation: UML Activity Diagram
- Influenced by «include» and «extend» dependencies (Use Case Model)
- Subactivity states express the hierarchical decomposition of a process
- Input/Output objects from crucial activities

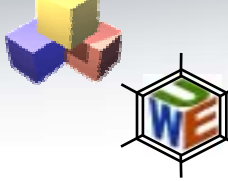




- Representation:
UML Activity Diagram

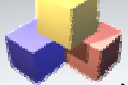


- Subactivity states are used
 - to reduce complexity
 - to define reusable components
- Specification of sequence of user activities is postponed to the design phase

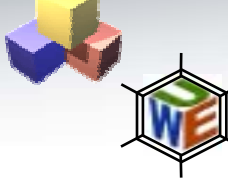


- About OO-H & UWE
- Looking for a Common Solution for Modeling Web Business Processes
- Business Processes & Requirements Analysis
- Analysis of Web Business Processes
 - Conceptual Model
 - Process Model
- **Design of Web Business Processes**
 - Two Approaches: Differences & Similarities
 - Modeling with OO-H
 - Modeling with UWE
- Conclusions & Future Work

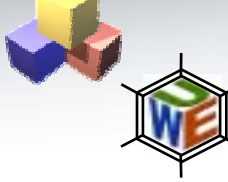
Two Approaches: Differences & Similarities



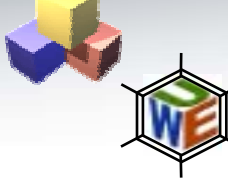
	OO-H	UWE
Model characteristics	Definition of a navigation model driven by a (refined) process flow model	Definition of integration points between navigation model and (refined) process model
Relationship between Navigation & Process Model	Navigation supports process (interplay between user interface and process steps)	Process model complements navigation model
Notation	NAD (proprietary)	UML (standard)
Advantages	<ul style="list-style-type: none">■ default navigation design diagram may be automatically obtained■ faster development process	<ul style="list-style-type: none">■ low coupling eases construction and maintenance■ better separation of concerns



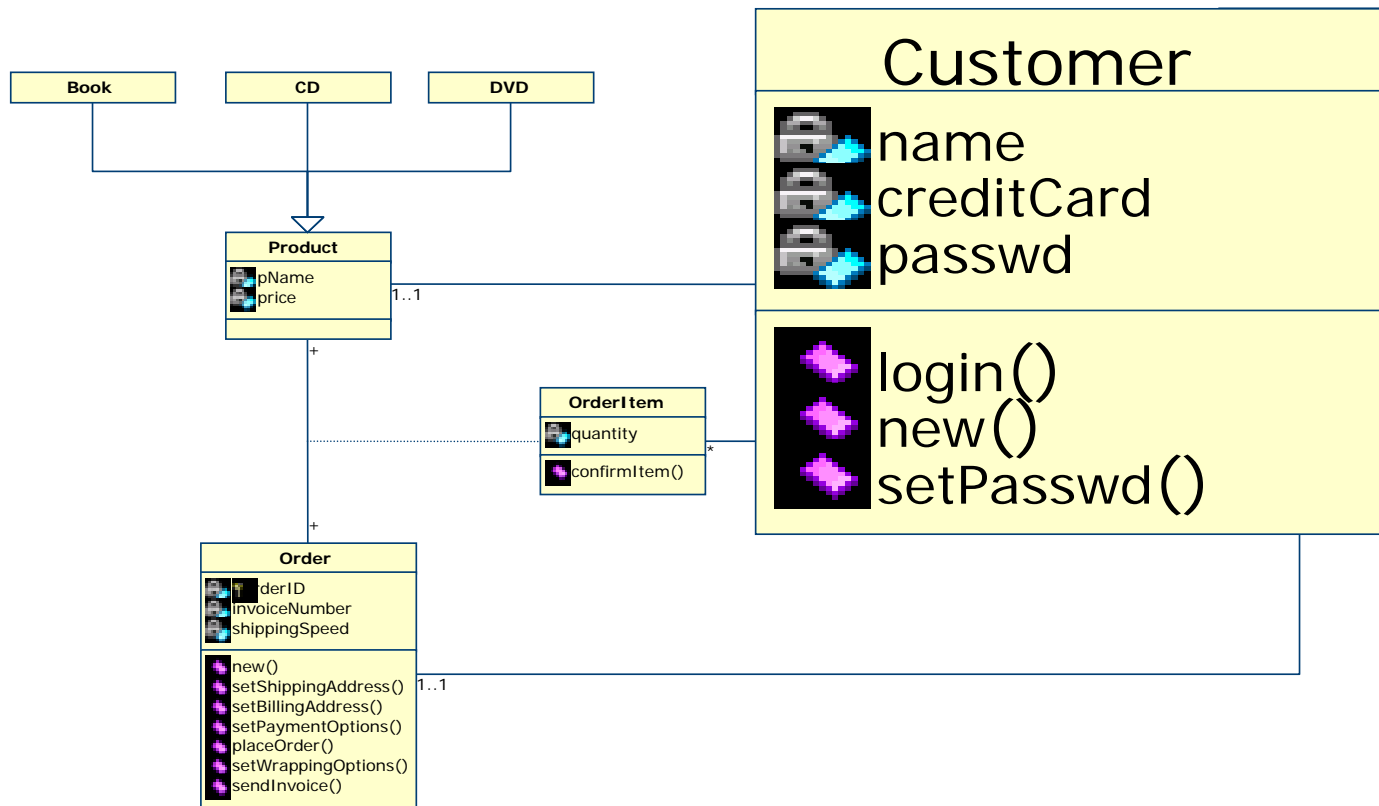
- About OO-H & UWE
- Looking for a common solution for modeling Web business processes
- Business Processes & Requirements Analysis
- Analysis of Web Business Processes
- Design of Web Business Processes
 - Two Approaches: Differences & Similarities
 - **Modeling with OO-H**
 - Process Model Refinement
 - Default Navigation Model
 - Modeling with UWE
- Conclusions & Future Work



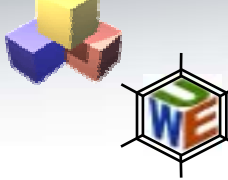
- The process view is based on **a set of activity diagrams** that supplement the information contained in the domain model
- During the design phase OO-H, in order to define the process model that complements the domain model, refines:
 - class diagrams
 - activity diagrams
- OO-H bases this process refinement on the concept of **service** as defined in [UML 2003]
- From the refined activity diagrams a set of **mapping rules** can be applied in order to get a **default navigation view**



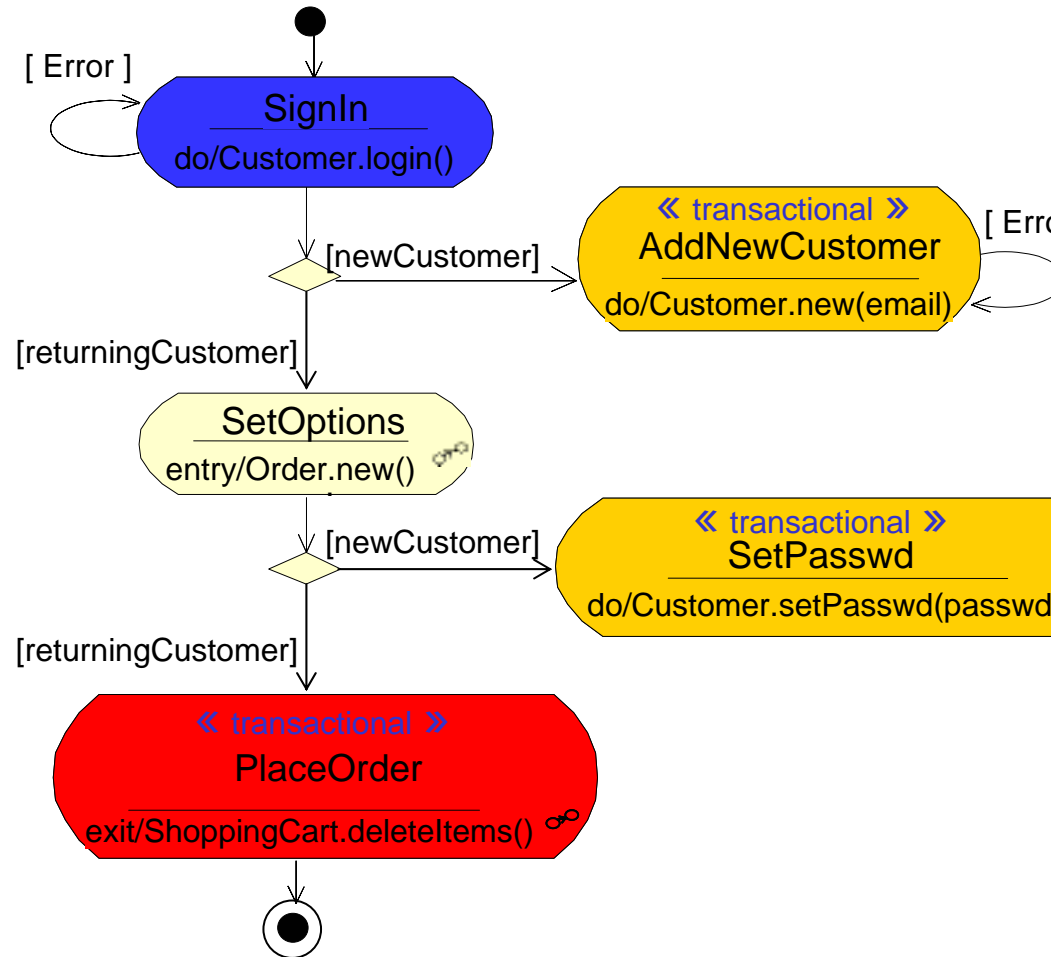
- Class diagram completed with new attributes and methods

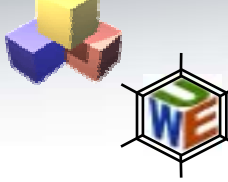


OO-H: Activity Refinement (second step)

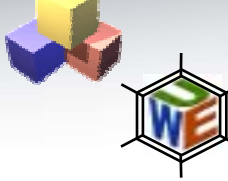


- OO-H redefines **call states** so that they can be mapped to underlying class operations (**do actions**)
- A new **«transactional»** stereotype may be applied both to call states and subactivity states
- Some call states may be merged under a common subactivity state
- Some subactivity states may be redefined as call states
- Subactivity states may be enriched with **entry/exit actions**



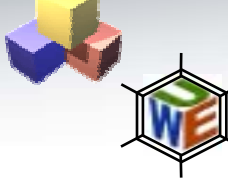


- The navigation view in OO-H is defined by means of a **Navigation Access Diagram (NAD)**
- This diagram is made up of collections, navigation targets, navigation classes and navigation links that specify a complete navigation view.
- We propose a predefined set of **mapping rules** to get a NAD from a refined activity diagram.
 - This default NAD not only speeds up the development cycle but also assures the interface guidance and support to this process.

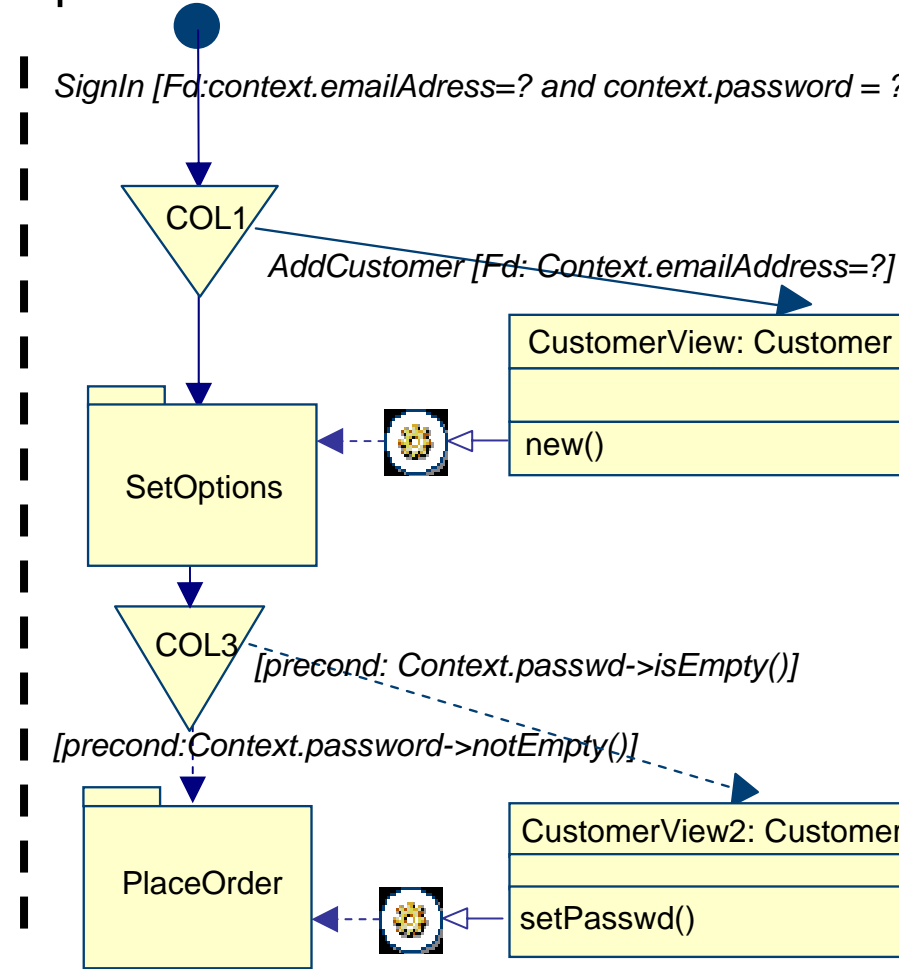
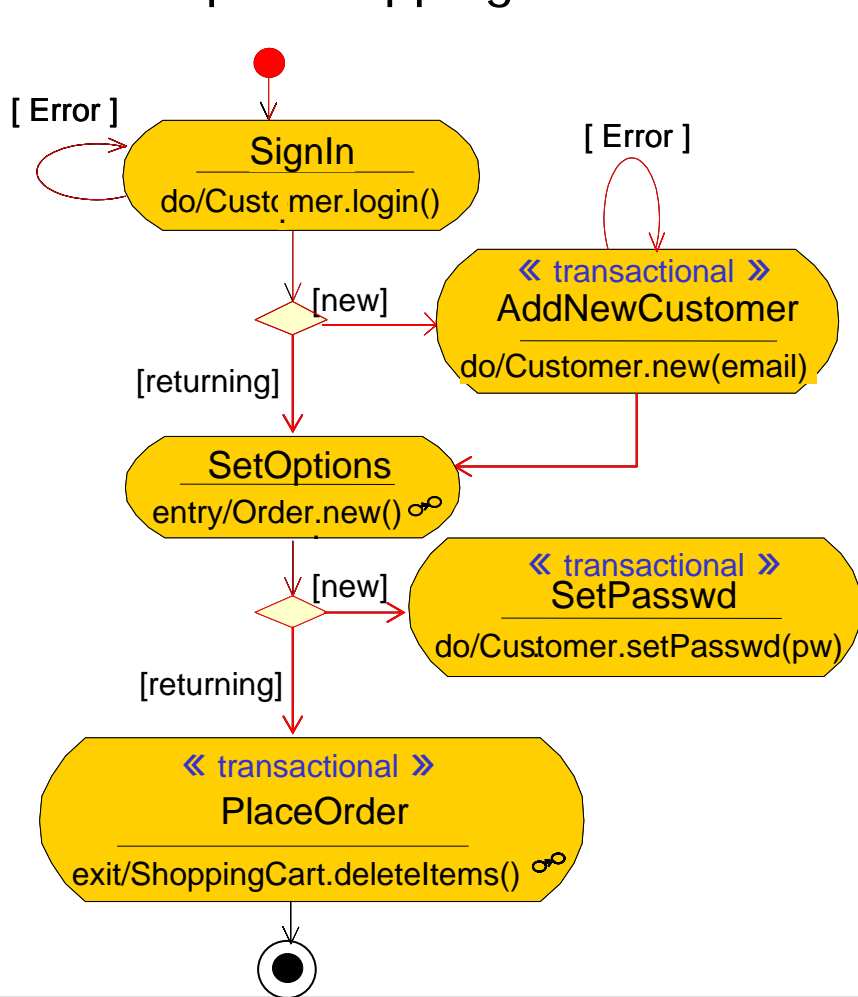


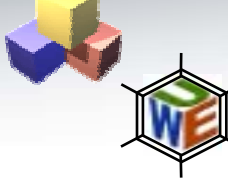
- Predefined set of mapping rules

Activity Diagram Element	NAD diagram element
Non-Transactional Activity	Navigational link refined with precondition filter
Transactional Activity	Service link associated with a Navigational class
Transition	Traversal link
Subactivity	Navigation target
Branch	Collection from which a set of Traversal links with exclusive filters departs
Merge	Collection at which a set of Traversal links with no filters arrives
Split-Join	Default path that traverses the concurrent activities sequentially

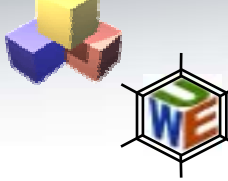


- Example: mapping of the Checkout process

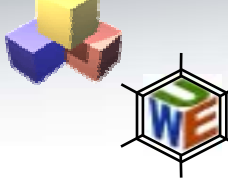




- The default navigation diagram may be refined iteratively, as long as **process steps are preserved**
- A presentation view that inherits the information provided in previous models completes the OO–H proposal
 - Based on XML templates that separate interface concerns
 - Graphically manipulable in the OO–H development environment
 - This view is used inside the OO–H development environment to prototype the application and provide feedback for the next iteration of the process
- The presentation view is fed to a set of **model compilers** that provide the mapping rules to actual interface constructs
 - Such rules depend on the target platform and language



- About OO-H & UWE
- Looking for a Common Solution for Modeling Web Business Processes
- Business Processes & Requirements Analysis
- Analysis of Web Business Processes
- Design of Web Business Processes
 - Two Approaches: Differences & Similarities
 - Modeling with OO-H
 - **Modeling with UWE**
 - Integration of Processes in the Navigation Model
 - Refinement of the Process Model
 - Support of Processes in the Presentation Model
- Conclusions & Future Work

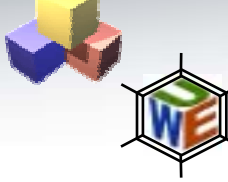


UWE proposes the following activities for the design phase:

- Refinement of the **conceptual model** by adding attributes and methods to the classes identified during analysis
- Construction of the **navigation model**
- Integration of the processes in the **navigation model** by defining entry and exit points to the processes
- Refinement of the **process model** building
 - a process **structural view**
 - a process **flow view** (behavioral view)
- Construction of the **presentation model** based on the navigation and process model

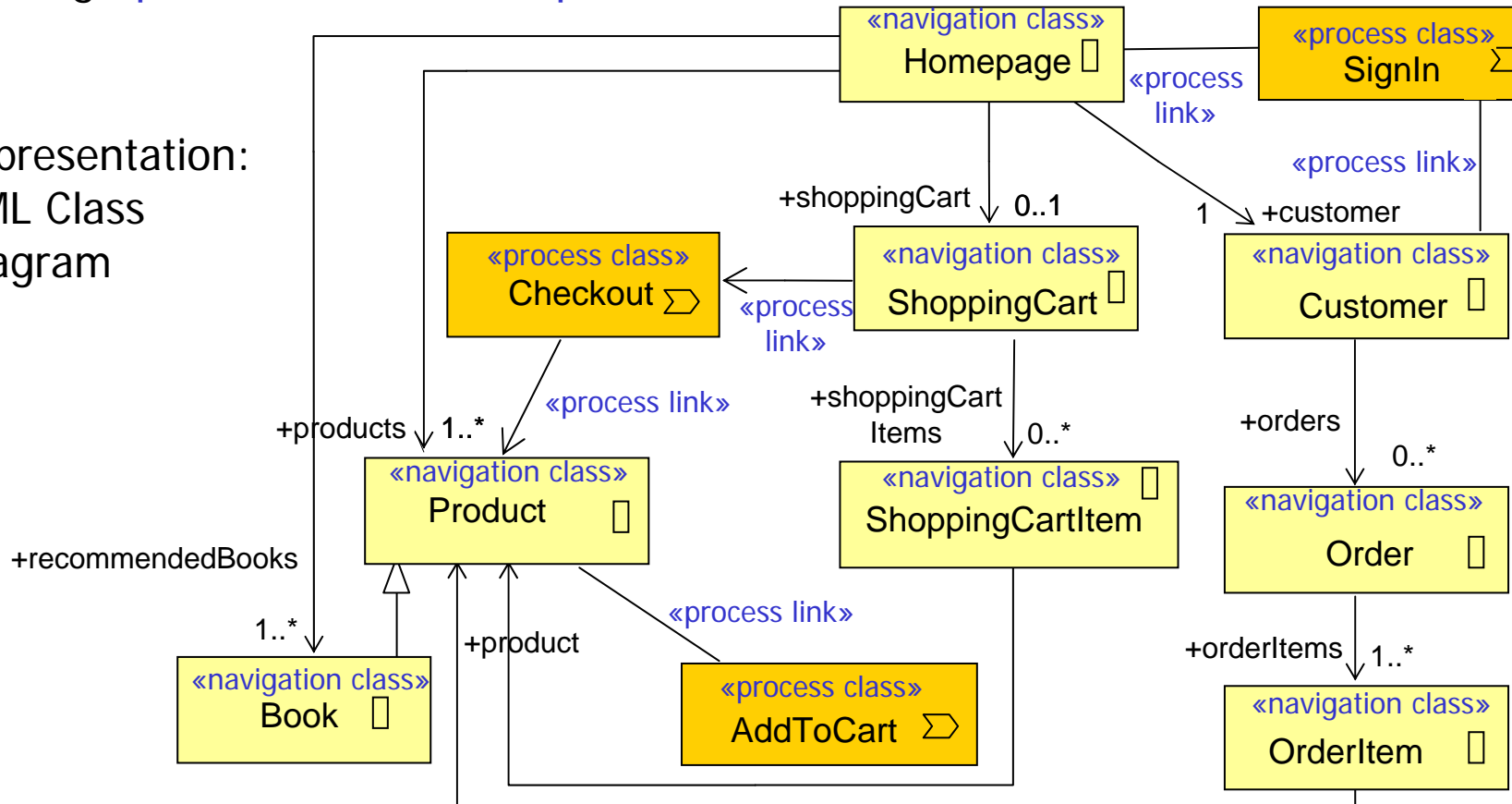
UWE uses UML for all models

JWE: Navigation Model (first step)

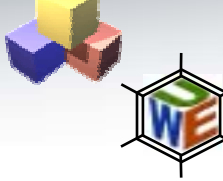


- Nodes and links are modeled with stereotyped classes «navigation class» and «navigation link»
- Integration of the process in the navigation structure by adding «process class» and «process link»

Representation:
UML Class
Diagram



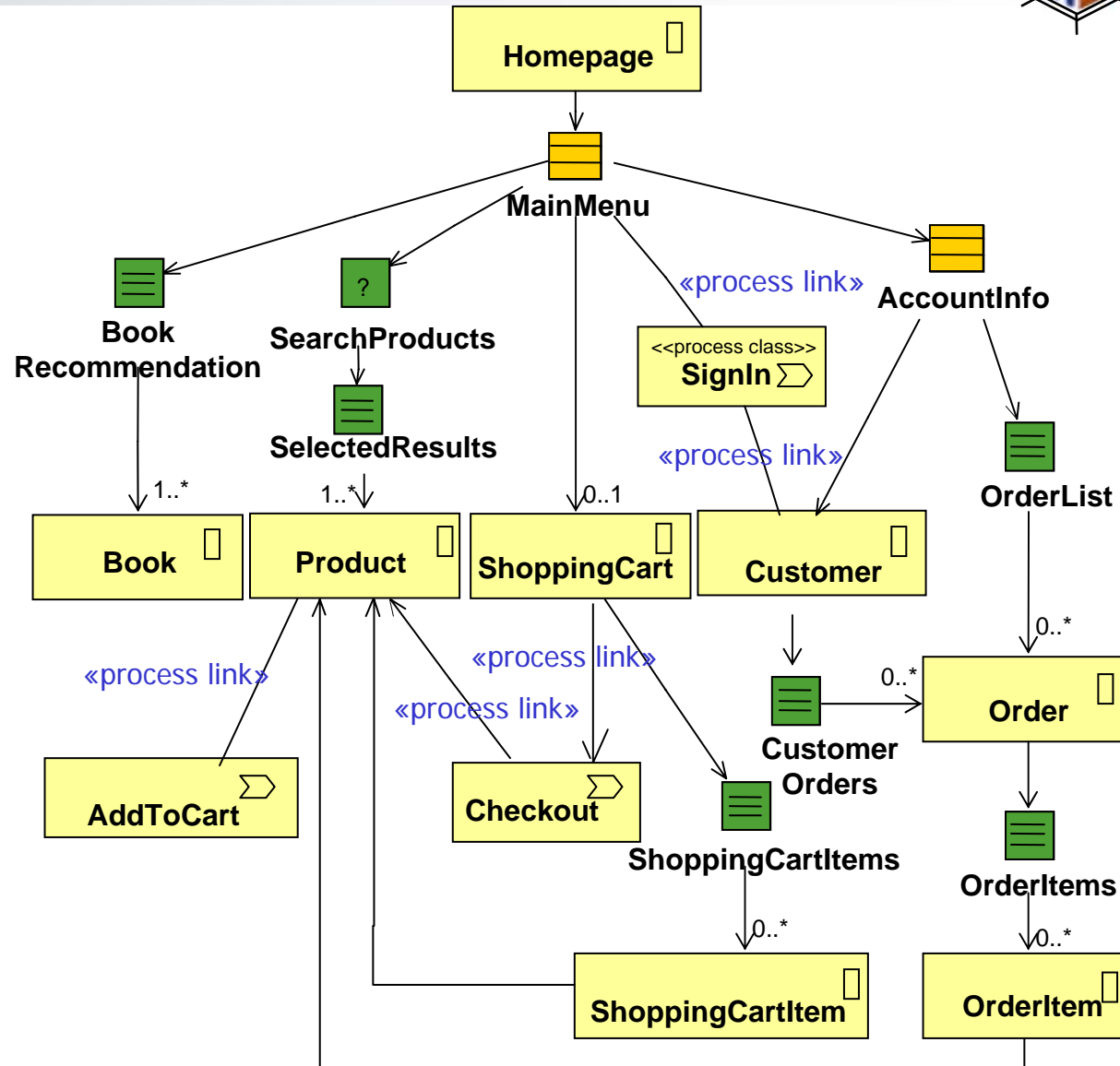
UWE: Navigation Model (second step)

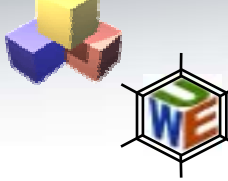


- Enhancement of the navigation model by access structures:

- indexes, guided tours and queries
- menus

- No changes in the integration between processes & navigation model



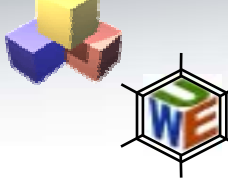


Processes are represented by ...

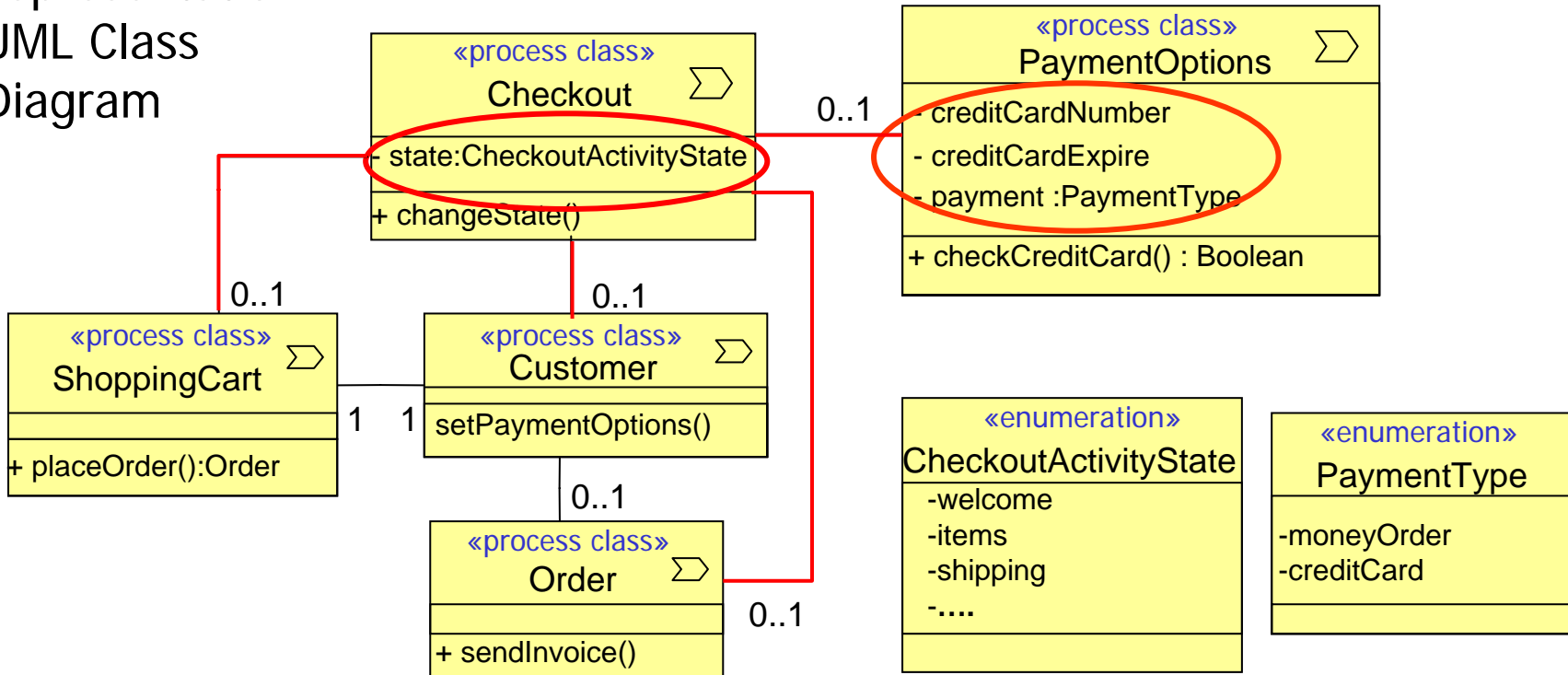
- «process class»
 - models a class whose instances are needed by the user during execution of a process
 - mapping function between not navigational use cases and process classes

- «process link»
 - models the association between a «navigation class» and a «process class» indicating **entry and exit points** of processes within the navigation structure
 - associates information about the process state which allows for re-initiation of the process after an interruption

UWE: Process Model – Structural View

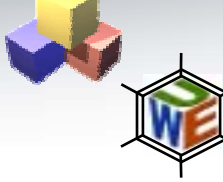


Representation:
UML Class
Diagram



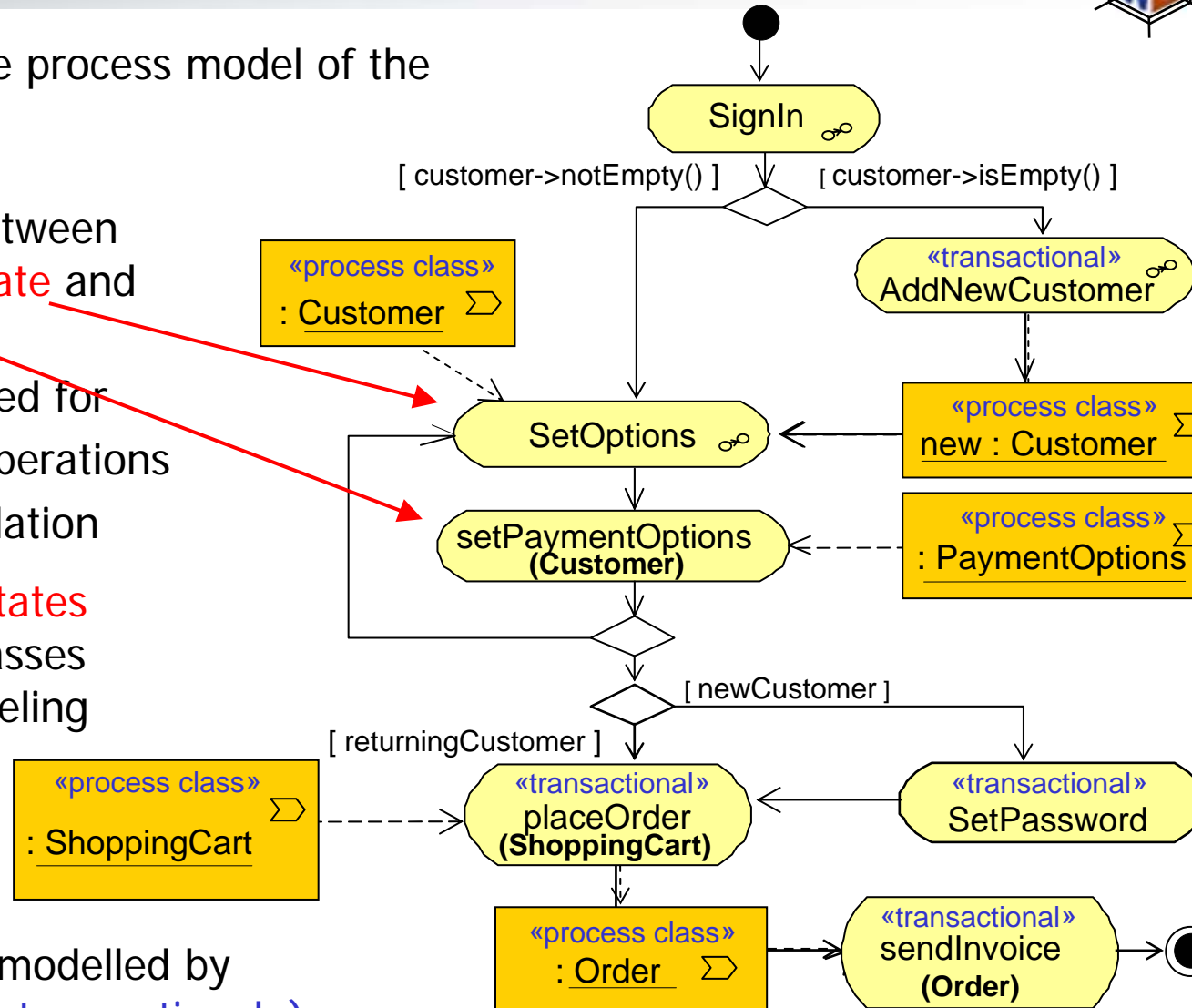
- Structural view is partially derived from the conceptual model
- Process state is defined explicitly and implicitly by associations
- Includes user input data

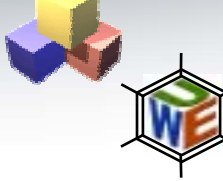
UWE: Process Model – Process Flow View



Refinement of the process model of the analysis phase

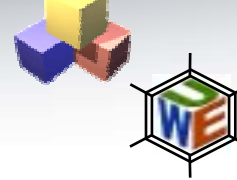
- Distinction between **subactivity state** and **call state**
- Call states used for
 - change operations
 - data validation
- **Object flow states** of process classes used for modeling user input/output
- Transactions modelled by stereotypes («**transactional**»)



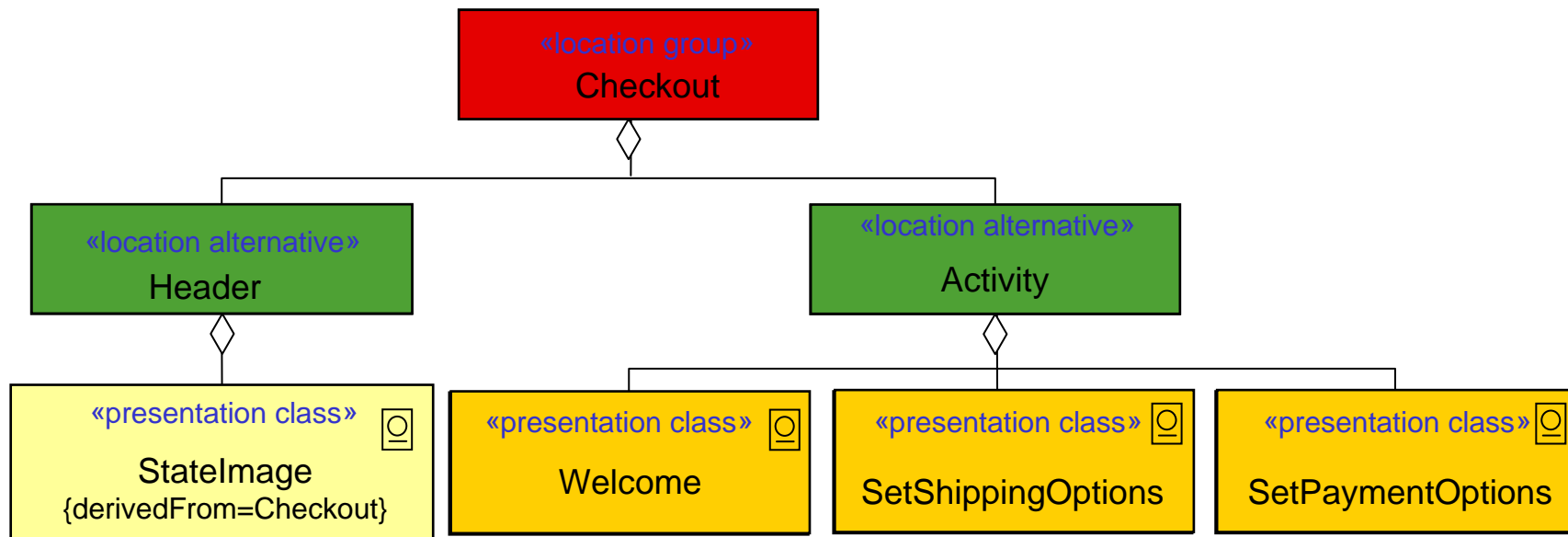


Two different views:

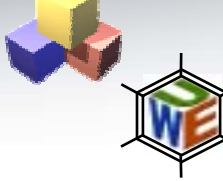
- **Structural view** shows the structure of the presentation space
- **User interface view** presents the user UI elements of a page
- «presentation class» represents logical page fragments
- an instance of a «presentation class» is related to exactly
 - one «navigation class» element of the navigation model or
 - one «process class» element of the process model
- «location group» used to model the presentation sub-structure aggregating a list of sub-locations
- «location alternative» used to model presentation alternatives among «location» classes



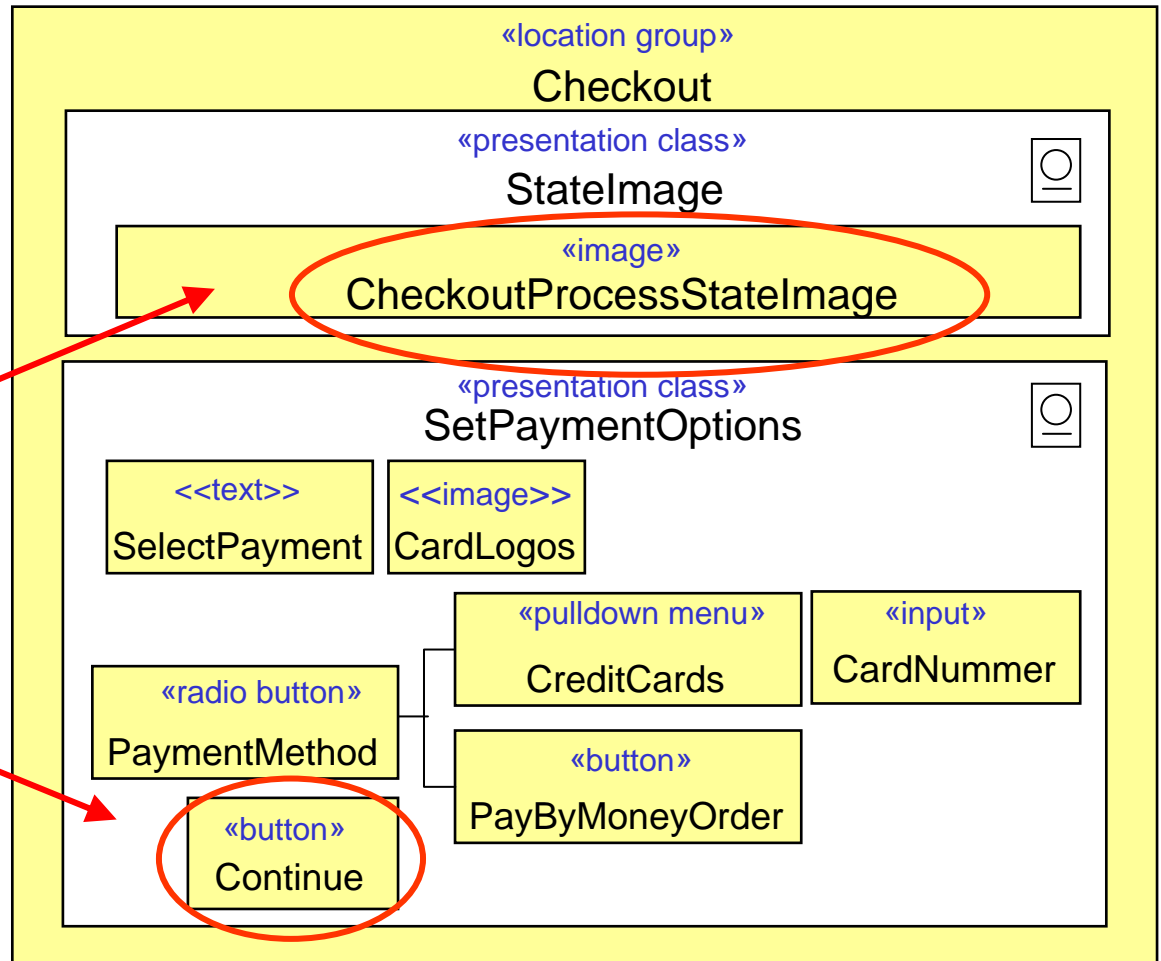
- Representation: UML Class Diagram
- Presentation structural view models shows
 - *how* presentation elements are grouped
 - *how* the presentation space is partitioned
 - *which* presentation elements are displayed in the same space (not = time)

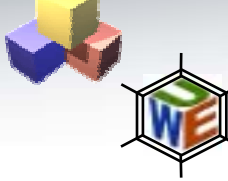


IWE: Processes & User Interface View



- Representation: UML Class Diagram
- Composition of UI elements (UML container notation)
- Current state of the Checkout process
- Continue «button» triggers the SetPaymentOptions call state in the process model

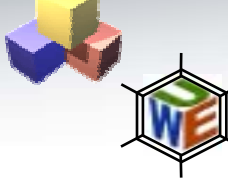




- Lessons learned

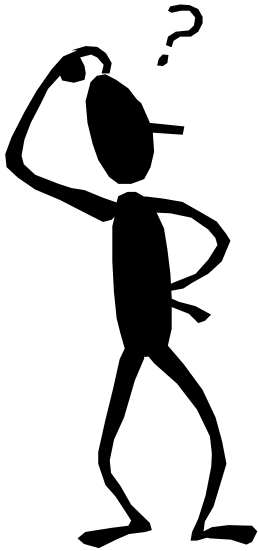
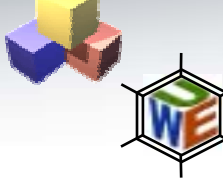
- Use of standards facilitates reuse of analysis models in different Web modeling proposals
- A separate analysis process model eases the construction and maintenance of process-aware Web applications
- Working together in the extension of methods
 - provides a common vocabulary for the inclusion of concepts
 - facilitates the comparison

even though OO-H and UWE chose different solutions
for the design models



- Further work
 - Evaluation of changes due to the new standard UML 2.0
 - Process personalization
 - **Metamodel extension** to support Web business processes
 - Extension of CASE tools **VisualWADE** and **ArgoUWE** to support process modeling

Thanks for your attention!!



Questions?

For further comments...

ccachero@dlsi.ua.es

kochn@informatik.uni-muenchen.de

krausa@informatik.uni-muenchen.de

santi@dlsi.ua.es



Time for lunch?