



UML for Global Computing

Lecture 2: Refinement

Martin Wirsing
LMU München
in cooperation with
Hubert Baumeister, Piotr Kosciuzenko, Nora Koch
Stephan Merz, Julia Zappe




GC Summer School, Edinburgh, July 2003



Contents

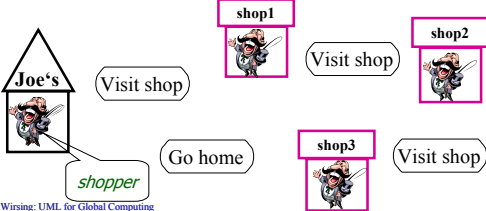
- Lecture 1: Introducing UML for Mobility
- **Lecture 2: Refining Mobility Designs**
 - Refining mobility activities
 - Refining mobility in sequence diagrams
 - A semantic approach to refinement: Mobile TLA
- Lecture 3: Property-driven Development of Mobile Systems

M. Wirsing: UML for Global Computing




The Running Example: Mobile Shopping Agent

- A mobile shopping agent is sent from his home "Joe" to visit several shops (at different network locations) and to collect offers for a particular good Joe wants to buy.
- If the shopping agent has collected three offers he returns to Joe.




M. Wirsing: UML for Global Computing



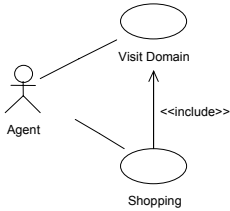
Goals of this Lecture

- Develop syntactic notions of refinement of mobile systems for UML which allow one to systematically refine locations and move actions.
- Base the syntactic refinement notions on a formal logic with explicit support for refinement.


M. Wirsing: UML for Global Computing



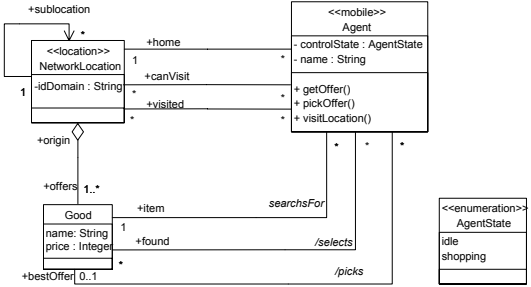
Mobile Shopping Agent: Use Case Diagram



M. Wirsing: UML for Global Computing

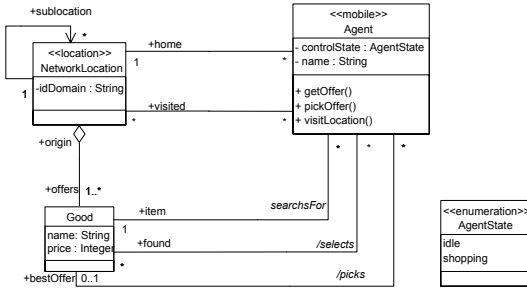


Mobile Shopping Agent: Class Diagram



M. Wirsing: UML for Global Computing

Mobile Shopping Agent: Class Diagram



M. Wirsing: UML for Global Computing

Contents



- Lecture 1: Introducing UML for Mobility
- **Lecture 2: Refining Mobility Designs**
 - Refining mobility activities
 - Refining mobility in sequence diagrams
 - A semantic approach to refinement: Mobile TLA
- Lecture 3: Property-driven Development of Mobile Systems

M. Wirsing: UML for Global Computing

Activity Diagrams



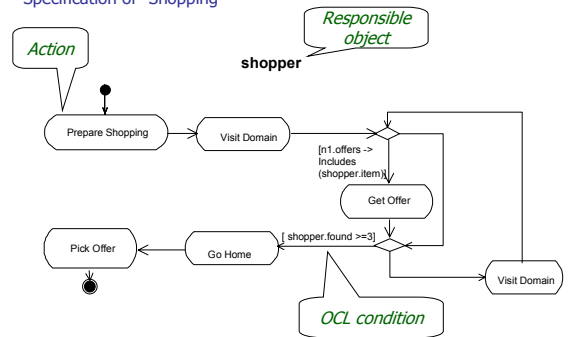
- Activity diagrams are intended for applications that need control flow or object/data flow models
- An activity graph describes a flow of actions which are connected by transitions
- Concurrency is expressed by fork and join states
- Swimlanes represent responsibilities of parts of the activity diagram

M. Wirsing: UML for Global Computing

Shopping Agent: Activity Diagram



Specification of "Shopping"

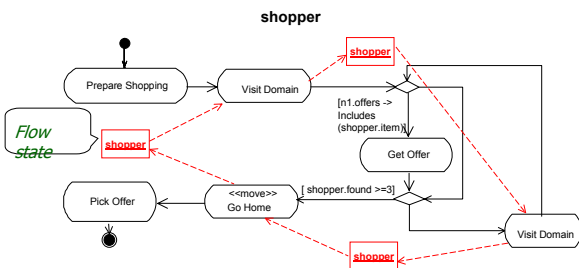


M. Wirsing: UML for Global Computing

Shopping Agent: Activity Diagram



Specification of "Shopping" (with Object Flow)



M. Wirsing: UML for Global Computing

Activity diagrams for Mobility



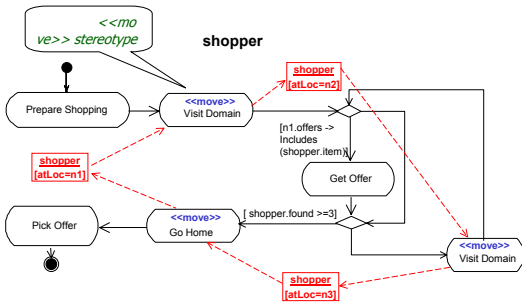
- Goal of the extension
 - To model how actions change the location of mobile objects
- Extensions
 - Stereotypes <<move>> / <<clone>> for actions
 - Two Notations
 - Responsibility Centred
 - Location Centred

M. Wirsing: UML for Global Computing

Mobile Shopping Agent: Activity Diagram



Responsibility centred view

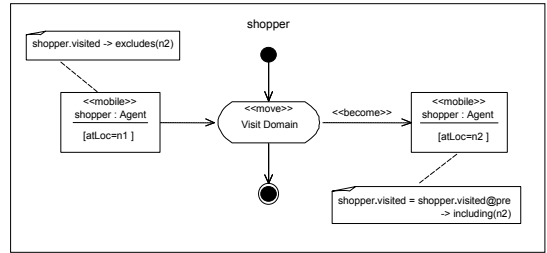


M. Wising: UML for Global Computing

Mobile Shopping Agent: Move Action



Specification of "Visit Domain" using Responsibility Centred Notation

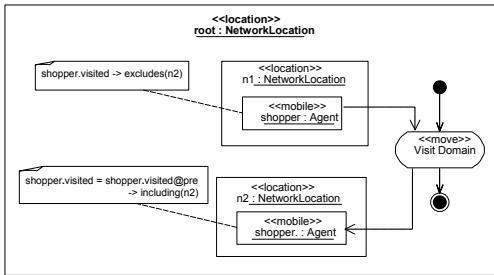


M. Wising: UML for Global Computing

Mobile Shopping Agent: Move Action



Location Centred Notation

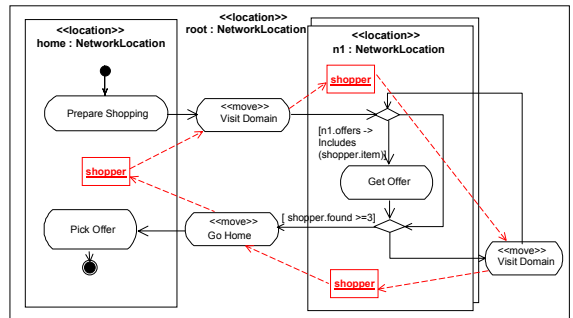


M. Wising: UML for Global Computing

Mobile Shopping Agent: Activity Diagram



Specification of "Shopping" (with Object Flow)



M. Wising: UML for Global Computing

Refining Mobility in Activity Diagrams



- An activity graph describes a flow of actions which are connected by transitions
- Refining mobility in an activity diagram amounts to
 - Representing a move action by several actions
 - Refining the location hierarchy by adding enclosing locations
- For the shopper example we present
 - the specification of the activity of the shopper
 - an activity refinement
 - a location refinement

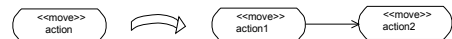
M. Wising: UML for Global Computing

Transformation Rules for Refinement



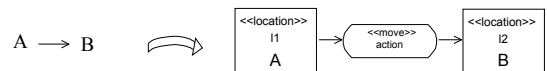
Activity Refinement

Replace one action by several actions



Location Refinement

Add additional locations and move activities

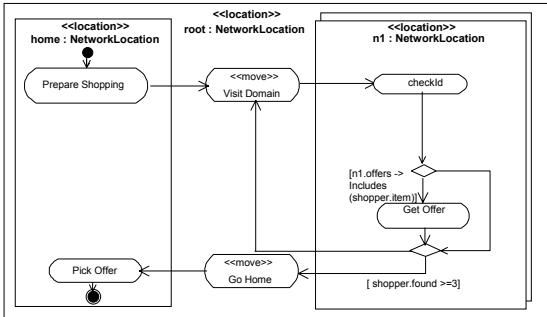


M. Wising: UML for Global Computing

Mobile Shopping Agent: Activity Diagram



Activity Refinement of "Shopping"

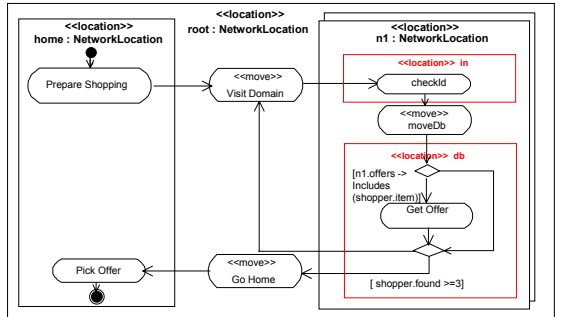


M. Wirsing: UML for Global Computing

Mobile Shopping Agent: Activity Diagram



Location Refinement of "Shopping"



M. Wirsing: UML for Global Computing

Contents



- Lecture 1: Introducing UML for Mobility
- **Lecture 2: Refining Mobility Designs**
 - Refining mobility activities
 - Refining mobility in sequence diagrams
 - A semantic approach to refinement: Mobile TLA
- Lecture 3: Property-driven Development of Mobile Systems

M. Wirsing: UML for Global Computing

Refining Mobility in Sequence Diagrams



- A Sequence Diagram describes ...
- Refining mobility in a Sequence Diagram amounts to
 - Refining a lifeline (Object Refinement)
 - Refining an action box
 - Refining a move action
 - **But not** to refine sending a message, terminating or creating an object.
- Examples will be
 - Sending a virus
 - Shopping

M. Wirsing: UML for Global Computing

A Grammar for Refinement



- **Goal**
 - Develop a refinement calculus for mobility in Sequence Diagrams
 - Refinement steps are treated as graphical transformations
 - The set of derivations from a diagram (containing nonterminals) defines all possible refinements
- **The Refinement Grammar**
 - has a graphical form and
 - is context free with
 - constraints which rule out inconsistent expressions (diagrams)

M. Wirsing: UML for Global Computing

A Grammar for Refinement: Symbols



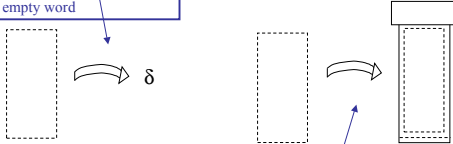
- **Terminal symbols**
 - action box, δ (empty box), $\bullet \rightarrow$ move arrow
 - object
- **Nonterminal symbols**
 - nonterminal action box,
 - nonterminal move arrow

M. Wirsing: UML for Global Computing

Location (Action Box) Refinement Rules



Terminate location refinement:
a nonterminal can be replaced by an empty word

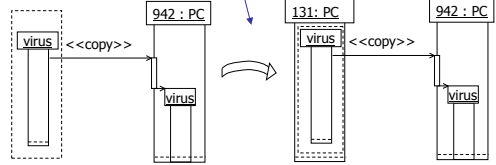


Add location: a new location can be added

Example Location (Action Box) Refinement



Add location



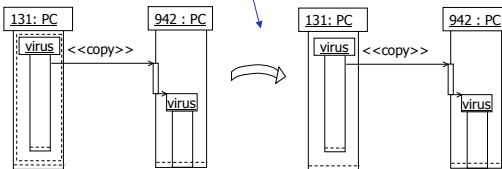
Virus from an unknown location

is identified to come from somewhere in PC 131

Example Location (Action Box) Refinement

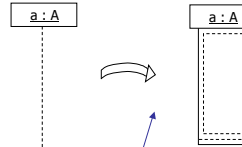


Terminate location refinement



The virus is identified to come directly from PC 131

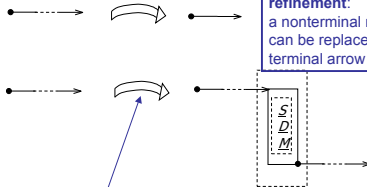
Object Refinement Rule



Zoom-into object:

The content of an object can be refined.
This rule can be understood as a special case of the previous rule where the dotted box is bound to one object.

Refinement of Move Actions



Terminate move refinement:
a nonterminal move arrow can be replaced by a terminal arrow

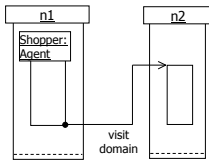
Move refinement:
A move can be refined by showing details like breakpoints or actions performed within the moving object

Move Constraints



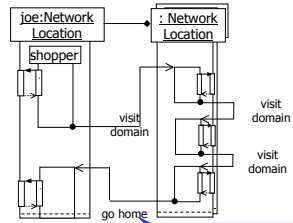
- A move can not create or terminate objects
 - objects residing in a moving object before the move can not be terminated during the move
 - a move can not introduce new objects
- The SDM within the move arrow must be a complete SDM except of the objects residing within the moving object which must satisfy the constraint above
- Other objects within the move arrow must be created during the move and terminated before its completion

Shopper Example



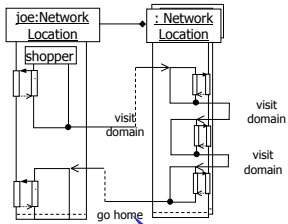
The shopper visits 1 domain

Shopper Example



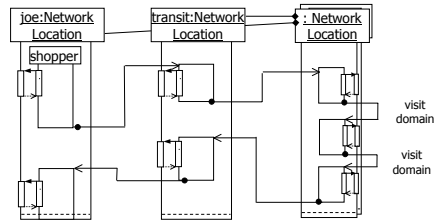
Terminal Moves:
no refinement possible

Refining the Shopper



Nonterminal Moves

Shopper Example: Move and Location Refinement



Refine move by
transit location and
internal actions