

The UML is More than Boxes and Lines

Yvan Labiche

Software Quality Engineering Laboratory

Carleton University

Ottawa, Canada



Context

- The UML is more than Boxes and Lines
- It has semantics
 - Note: some would like it to have a stronger semantics but ...
- But do students (and all practitioners) know that?
 - Students often perceive the UML as a drawing tool
 - Some practitioners perceive the UML as a drawing tool



The Semantics of the UML

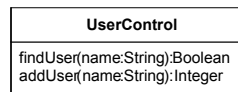
- Well-formedness rules in the standard documentation
 - Simple one: generally
 - Complex one: if the message in a sequence diagram is an operation, the signature of the operation, the class of the target instance of the message shall define that operation.
 - Additional semantics
 - Organization or project specific
 - Simple one: If an instance of class A sends a message to an instance of class B, there should be a path in the class diagram between A and B
 - Complex one: Assignment of responsibilities under the form of contracts (not necessarily in OCL) leads to specific design solution in sequence diagrams
- Consistency Rules

Missing one: if the signature of a message in a sequence diagram is an operation, the class of the target instance of the message shall define that operation.



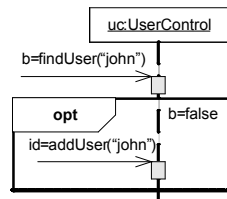
3

Coherence between Sequence and Contracts

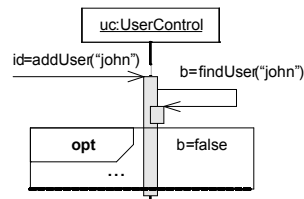


addUser(name:String):Integer
pre: findUser(name)=false
post: ...

(a)
Class diagram and contract excerpt



(b)
Correct interaction diagram



(c)
Incorrect interaction diagram



4

Possible Reasons for this Misconception

- UML semantics not necessarily taught to students during their (undergraduate or graduate) degree
 - few textbooks explicitly mention the problem of well-formedness of UML diagrams
 - Software engineering textbooks
 - UML textbooks
- Tools do not necessarily enforce well-formedness rules (even the ones of the standard)
- And as such they mask the need for a semantic understanding



Canada's Capital University



5

Attempt to Address the First Possible Reason



- Laboratory material
 - to heighten student awareness of the problem of consistency between UML diagrams
- Which lab material?
 - A UML analysis document for one software system
 - use case and use case descriptions (10 use cases)
 - class diagram (20 classes)
 - interaction diagrams (3 diagrams)
 - state machine diagrams (3 diagrams)
 - a data dictionary
 - Incomplete document (though with 40 pages)
 - e.g., not every use case is described and designed





Canada's Capital University



6

 Carleton UNIVERSITY Canada's Capital University  Engineering Better Software Squall <small>SOFTWARE QUALITY ENGINEERING LABORATORY</small> <small>Carleton University, Ottawa, Canada</small> <small>www.squall.carleton.ca/squall</small>		<h1 style="color: red;">A Cab Dispatching System</h1>		<ul style="list-style-type: none"> 1. Summary2 2. Use Cases4 <ul style="list-style-type: none"> 2.1. Use Case Diagram4 2.2. Actors5 2.3. Use Case Descriptions6 <ul style="list-style-type: none"> 2.3.1. Activate Driver Use Case Description6 2.3.2. Deactivate Driver Use Case Description6 2.3.3. Update Location Use Case Description7 2.3.4. Handle Immediate Job Use Case Description7 2.3.5. No Cab Available Use Case Description8 2.3.6. Log Pre-booked Job Use Case Description8 2.3.7. Dispatch Pre-booked Job Use Case Description9 2.3.8. Add Account Customer Use Case Description9 2.3.9. Charge Job To Account Use Case Description10 2.3.10. Print Invoice Account Customer Use Case Description10 3. Class Diagram11 4. Data Dictionary13 <ul style="list-style-type: none"> 4.1. Entity classes13 <ul style="list-style-type: none"> 4.1.1. Cab13 4.1.2. Customer18 4.1.3. Driver19 4.1.4. Job20 4.1.5. ImmediateJob20 4.1.6. PrebookedJob22 4.2. Control Classes25 <ul style="list-style-type: none"> 4.2.1. CustomerControl25 4.2.2. DriverControl26 4.2.3. JobManagerControl27 4.2.4. UpdateLocationControl28 4.3. Boundary Classes29 <ul style="list-style-type: none"> 4.3.1. AddAccountForm29 4.3.2. ChargeAccountForm29 4.3.3. DispatchJobForm30 4.3.4. InvoiceAccountCustomer30 4.3.5. ReportCabLocation31 4.3.6. ChangeDriverStatusForm31 4.3.7. RequestCabForm32 4.3.8. NotifyNewJob33 4.3.9. ReportCabETA33 4.3.10. ClockBoundary33 4.4. Associations between entity classes35 5. Sequence diagrams36 <ul style="list-style-type: none"> 5.1. Abbreviations36 5.2. HandleImmediateJob Sequence Diagram36 																							
		<table border="1"> <tr> <td>Total Number of classes</td> <td>20</td> </tr> <tr> <td>Number of entity classes</td> <td>6</td> </tr> <tr> <td>Number of control classes</td> <td>4</td> </tr> <tr> <td>Number of boundary classes</td> <td>10</td> </tr> <tr> <td>Number of attributes</td> <td>25</td> </tr> <tr> <td>Number of operations</td> <td>59</td> </tr> <tr> <td>Number of class relationships</td> <td>21</td> </tr> <tr> <td>Number of use cases</td> <td>10</td> </tr> <tr> <td>Number of actors</td> <td>7</td> </tr> <tr> <td>Number of interaction diagrams</td> <td>3</td> </tr> <tr> <td>Number of state machine diagrams</td> <td>3</td> </tr> <tr> <td>Descriptions of use cases and actors</td> <td>7 pages</td> </tr> <tr> <td>Data Dictionary</td> <td>23 pages</td> </tr> </table>	Total Number of classes	20	Number of entity classes	6	Number of control classes	4	Number of boundary classes	10	Number of attributes	25	Number of operations	59	Number of class relationships	21	Number of use cases	10	Number of actors	7	Number of interaction diagrams	3	Number of state machine diagrams	3	Descriptions of use cases and actors	7 pages	Data Dictionary
Total Number of classes	20																										
Number of entity classes	6																										
Number of control classes	4																										
Number of boundary classes	10																										
Number of attributes	25																										
Number of operations	59																										
Number of class relationships	21																										
Number of use cases	10																										
Number of actors	7																										
Number of interaction diagrams	3																										
Number of state machine diagrams	3																										
Descriptions of use cases and actors	7 pages																										
Data Dictionary	23 pages																										

 Carleton UNIVERSITY Canada's Capital University  Engineering Better Software Squall <small>SOFTWARE QUALITY ENGINEERING LABORATORY</small> <small>Carleton University, Ottawa, Canada</small> <small>www.squall.carleton.ca/squall</small>		<h1 style="color: red;">Inconsistencies</h1>													
		<ul style="list-style-type: none"> ▪ Document seeded with (57) inconsistencies! <ul style="list-style-type: none"> – From a taxonomy of inconsistencies – In and between different parts of the document 													
		<table border="1"> <thead> <tr> <th></th> <th>Total number</th> </tr> </thead> <tbody> <tr> <td>In the data dictionary (e.g., erroneous operation description, missing or incorrect contract)</td> <td>14</td> </tr> <tr> <td>Between interaction and class diagrams (e.g., message suggests navigation between objects but there is no association)</td> <td>2</td> </tr> <tr> <td>Between class diagram and data dictionary (e.g., missing operation in data dictionary, inconsistent signatures in class and data dictionary, inconsistent scopes in class and data dictionary, missing attribute description in data dictionary, inconsistent association role names or multiplicities, attribute redundant with association, missing association)</td> <td>17</td> </tr> <tr> <td>Between interaction diagram and data dictionary (e.g., wrong operation used in message, inconsistency of message sequence with operation contracts, wrong sequence of messages, precondition not checked in message sequence, wrong target object type for message)</td> <td>12</td> </tr> <tr> <td>Between state machine diagram and data dictionary (e.g., missing state, missing transition, missing guard condition)</td> <td>12</td> </tr> </tbody> </table>		Total number	In the data dictionary (e.g., erroneous operation description, missing or incorrect contract)	14	Between interaction and class diagrams (e.g., message suggests navigation between objects but there is no association)	2	Between class diagram and data dictionary (e.g., missing operation in data dictionary, inconsistent signatures in class and data dictionary, inconsistent scopes in class and data dictionary, missing attribute description in data dictionary, inconsistent association role names or multiplicities, attribute redundant with association, missing association)	17	Between interaction diagram and data dictionary (e.g., wrong operation used in message, inconsistency of message sequence with operation contracts, wrong sequence of messages, precondition not checked in message sequence, wrong target object type for message)	12	Between state machine diagram and data dictionary (e.g., missing state, missing transition, missing guard condition)	12	8
	Total number														
In the data dictionary (e.g., erroneous operation description, missing or incorrect contract)	14														
Between interaction and class diagrams (e.g., message suggests navigation between objects but there is no association)	2														
Between class diagram and data dictionary (e.g., missing operation in data dictionary, inconsistent signatures in class and data dictionary, inconsistent scopes in class and data dictionary, missing attribute description in data dictionary, inconsistent association role names or multiplicities, attribute redundant with association, missing association)	17														
Between interaction diagram and data dictionary (e.g., wrong operation used in message, inconsistency of message sequence with operation contracts, wrong sequence of messages, precondition not checked in message sequence, wrong target object type for message)	12														
Between state machine diagram and data dictionary (e.g., missing state, missing transition, missing guard condition)	12														

Introduction Lecture to UML Consistency

- To help students find consistency problems
- This is not a (formal) inspection technique

	Use case diagram	Use case descr.	Sequence diagram	Class diagram	Statechart	Data dictionary
Use case diagram	NA	For each use case there should be	Must be accompanied by			Actors and use cases must appear in
Use case descr.	Corresponds to each use case in	NA	Services described find realization in	Entity class, methods and attrib appear in		Actors and use case name must appear in
Sequence diagram	Corresponds to each use case in	Corresponds to each use case in	NA	Objects must be instances of classes in	Scenarios in statecharts and sequence diagrams should match	Scenarios and contracts should match
Class diagram			Each class and operation must appear in	NA		classes, attributes, operations described
Statechart			Scenarios in statecharts and sequence diagrams should match	Describes instance behav. op. and attr. defined in	NA	Each class, attribute, operation identified as in
Data dictionary	Has entries for actors and use cases in	Has entries for use case names in	Scenarios and contracts should match	Has entries for all elements in		NA

Observations and Lessons Learned

- Observations of laboratory results of, and comments made by, 437 computer systems and software engineering students (since 2003)
- 11.5 inconsistencies found on average
 - Mostly the easiest ones are found
 - Difficult task:
 - document perhaps too long (40 pages)
 - lab perhaps too short (3 hours)
 - First time they are presented with such a “complete” document
 - No systematic inspection technique taught
- But ... increased awareness on UML consistency
 - “Before that laboratory session, I did not have a clear understanding of what the UML was all about.”

Future Work

- Investigating series of consistency rules
- Implementing consistency rules in CASE tool
- Using inspection technique in classroom
- Improving laboratory material
 - More focused (perhaps even less complete)
 - UML 2.0
 - Using a UML CASE tool instead of hardcopies (or in conjunction)



11

? Questions ?

