Model formalization

Dr. ir. Igor Nikolic

26-11-12

Lecture goals

- Understand the 4th step of the method, model formalization
- Learn how to
  - make a model narrative
  - create action sequence diagrams
  - Write pseudocode

Model Formalization

- This is where the entire model is specified for the first time!
  - We know what and who is in the model
  - We need to establish who does what when and with whom
- Goal is to create a complete intermediate step between human and machine understandable description of the model

Two elements

- Model narrative
  - In textual form
  - In UML / activity sequence diagram form
- Pseudo code
  - “Code like” precise description in text
  - Full UML

Model narrative

- narrative is “an account of a series of events, facts, etc., given in order and with the establishing connection between them; a narration, a story, an account”
  - (Oxford English Dictionary).

- A narrative is an informal description of the generative theory of the system under study, leading to emergent patterns we are interested in exploring.

- a story which explains ***how agents and the environment do what they do, with whom and when.***

Model narrative – text form

- Be as precise and accurate as possible
- Exactly specify:
  - Which agent acts when
  - how the agents decision is made
  - Which states (agent or environment) are used
- Create a bullet list of each major action/decision
  - Sublist of how it is performed
- Guiding question :
  - Agent wakes up, gets a cup of coffee and does....

Model narrative – text form

- The narrative can and will be quite long and very detailed
- This is the actual conceptual model that you will be encoding
- Iterative, missing elements you will find:
  - actions (how do I get this number from the other agents/environment?)
  - states (I need this number in order to calculate something)
  - Decision algorithms (so how do I determine the price?)
- Modification is cheap at this stage!
  - Make sure that the description is complete before continuing !

Example model narrative text form

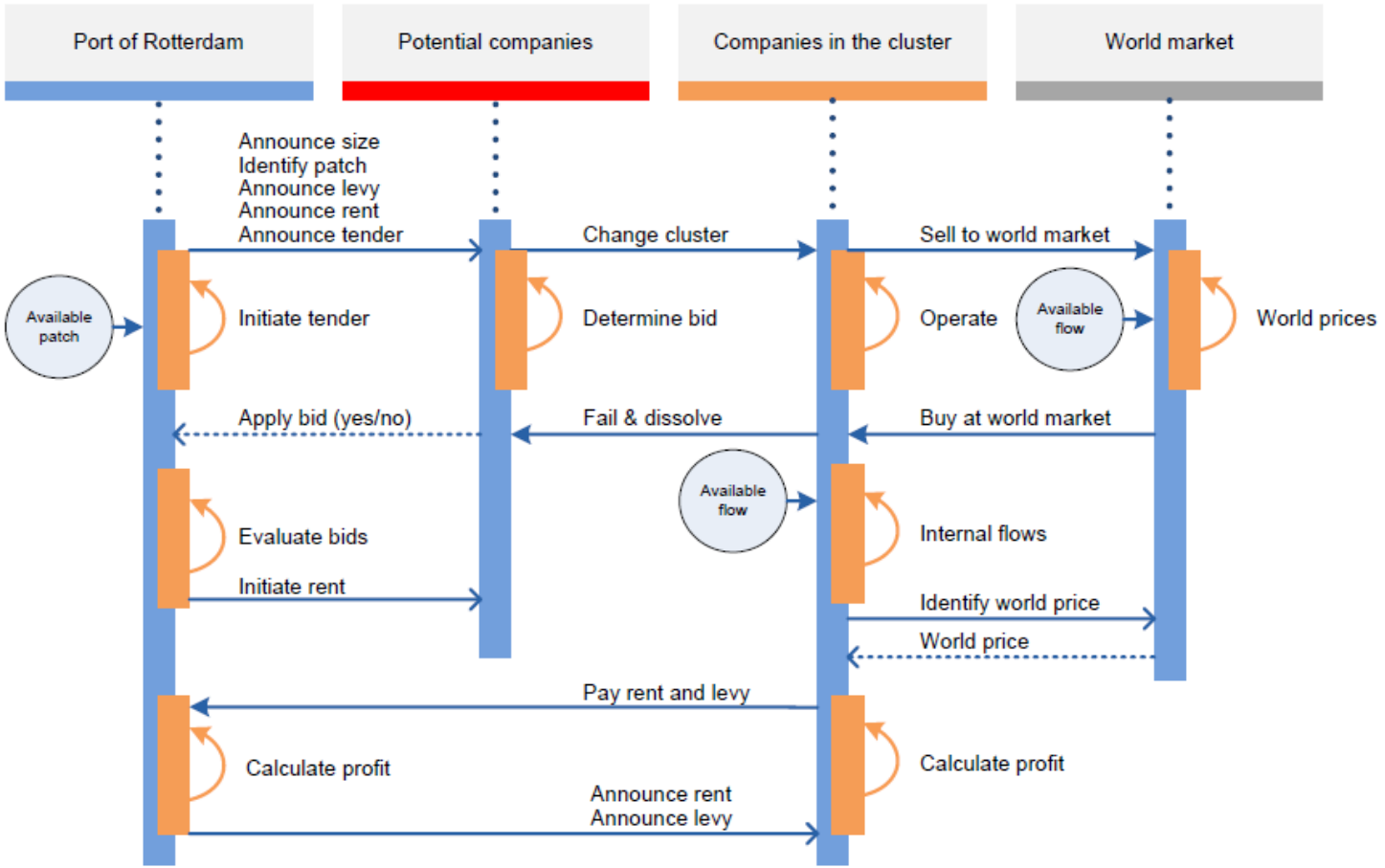
- ...
- calculate profit
  - profit = revenues - costs
  - revenues = price for their product * surface area * (base production + sum(performance))
  - price for products is dependent on the company type
  - base production is dependent on the company type
  - performance is dependent on the technologies owned and the company type
- Calculate costs
  - costs = basecost + surface area * (operating cost + sum(technology costs))
  - base cost and operating cost are company type dependent technology costs are dependent on the technologies owned as well as the company type
- Add profit to account balance
  Check to see if any of the neighbours has a new and unknown technology, and if so add it to Technology Library
- Update satisfaction proportional to profit made
- ...

Model narrative – UML form

- Action sequence diagrams
- UML trees for the flow of the decision making
- We won't be using them in the course
  - If you already know UML, feel free to use it !

Sequence diagram

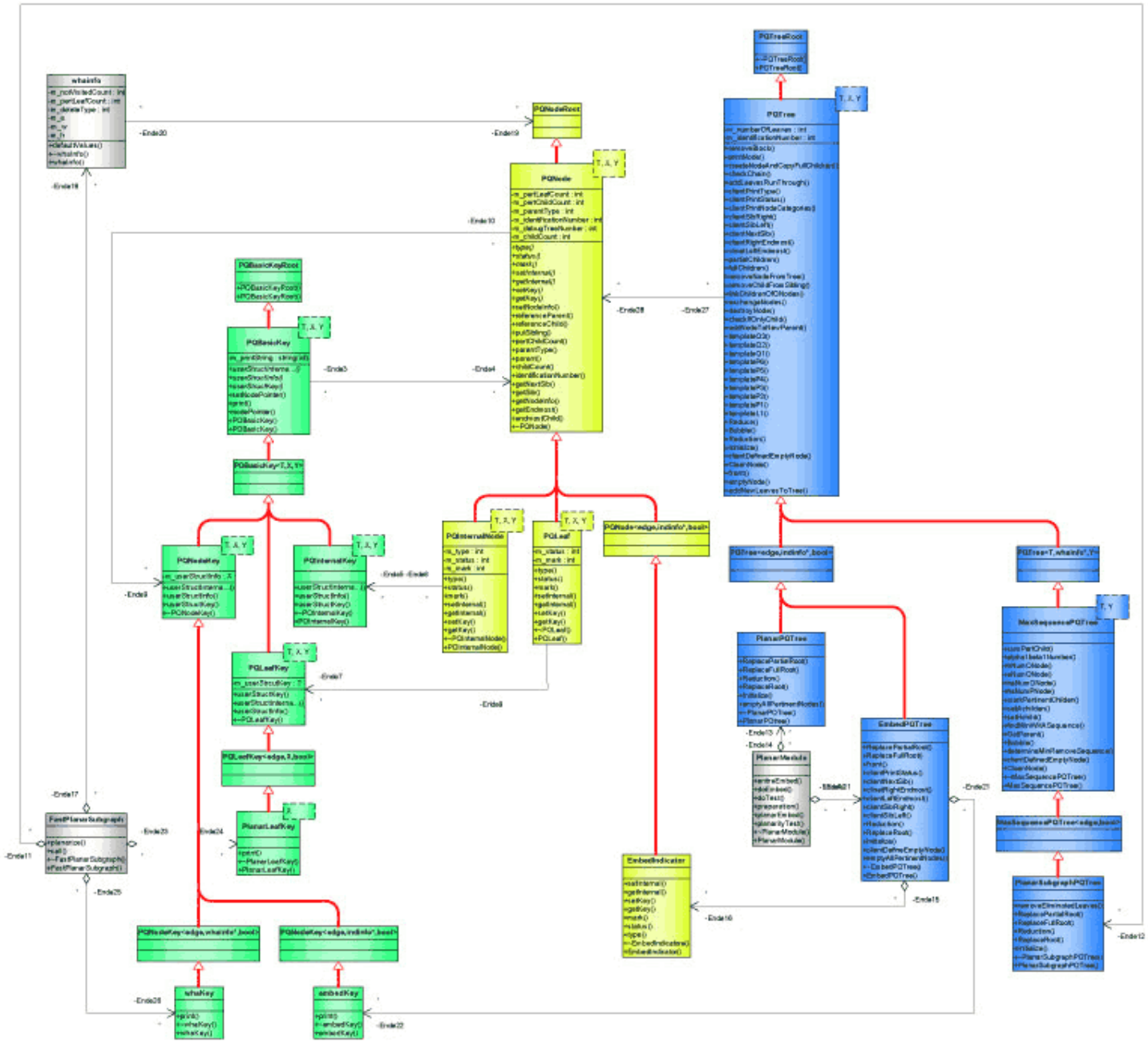
Fig. 4.2.1



Legend

- Actor activity
- Actor sub-activity
- Message
- Message response
- Action
- Condition

Example from
http://www.oreas.com/shared_img/example_big_uml_orth.gif



Pseudo code

- Last “informal” step before programming
- Textual description of the exact flow of the algorithm
- This step is frowned upon in “normal” software engineering, as it predetermines how the goal is achieved
- We care a lot about HOW the decision process is described → part of the generative theory.

Elements of pseudo code

- Computation and assignment
- Iterations and loops
- Conditions
- Input/output operations

Computation and assignment

- use words such as:
  - Generate, Compute, Process, Calculate, Set
  - Reset, Increment, Get, Add, Sum, Multiply, Print, etc
  - Mathematical expressions

- set the value of “variable" to : "arithmetic expression"
  "variable" equals "expression "
- "variable" = "expression"

Iterations and loops

- While. . . End while; Do until. . . End until; Case. . . End case; Call . . . with (parameters); Call; Return . . . ; Return; When.

- do while "condition"
  - Statement 1
  - statement 2
- end w h i l e

- for <variable> from <firstvalue> to <lastvalue> by <step>
  - statement 1
  - statement 2
- end for

Conditions

- if. . . then. . . else structure.
  - Can be deeply nested

- if "condition"
  - statement 1
  - statement 2
- else
  - statement 3
  - statement 4
- end if

Input/output operations

- get, store, save, print, display

- get "variable"
- display "variable"
- save variable

Algorithm 3.5 Model pseudo-code example – Company opinion update

```
1  All companies
2   calculate-profits
3     set profit (ProductPrice * SurfaceArea *
4          (BaseProduction + sum(TechologyProductionEffect
               ))) -
5          (BaseCost + SurfaceArea * (OperatingCost + sum(
               TechCosts)))
6     set AccountBalance AccountBalance + profit
7     set ProfitPerSurfArea profit / SurfaceArea
8
9   update-statisfaction
10    for each neighbour in AllMyneighbours
11         ask for ProfitPerSurfArea
12    calculate ProfitPerSurfArea - (sum(ProfitPerSurfArea
          of AllMyNeighbours) / number of companies in
         AllMyNeighbours)
13    normalise the difference by converting to a value
         between -1 and 1
14    save normalised difference in 'statisfaction'
15
16  update-TechLibrary
17    for each neighbour in AllMyNeighbours
18         Ask what are their current technologies
19    end for
20    for each Technology, compare to TechnologyLibrary
21         if not in TechnologyLibrary, add to
               TechnologyLibrary
22    end for
```

Few pitfalls

- Use clear and descriptive variable names !
- Think of language features that you could be using already now.
- Really make sure that you have written out all of the model!
  - “This is simple, we will do this later in code” is a very bad idea
- If you can not figure it out on paper, you will NOT be able to figure it out in code

Break

Group progress