From goals to statecharts

João Pimentel
{jhcp@cin.ufpe.br}

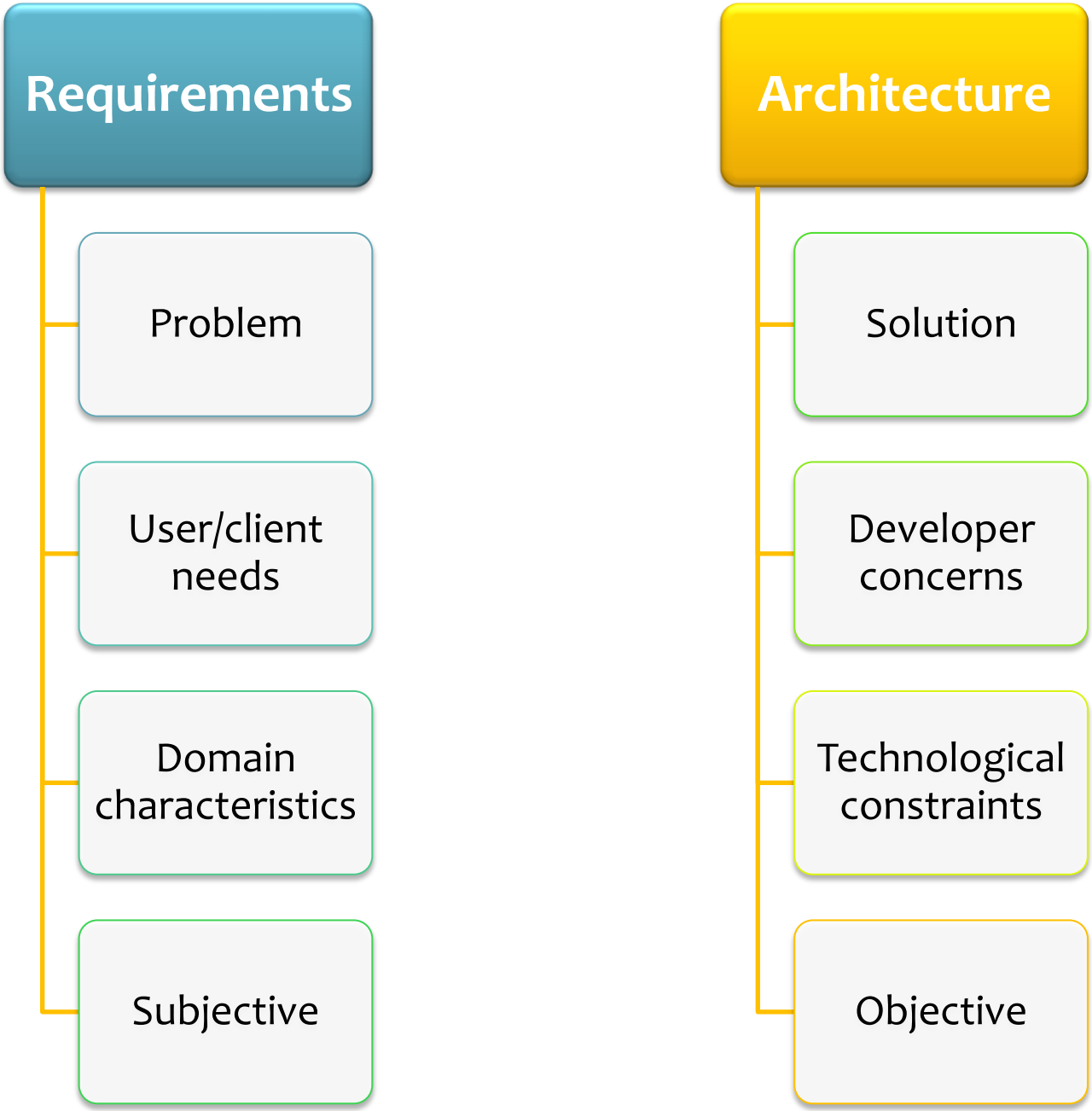
UNIVERSITÀ DEGLI STUDI
DI TRENTO

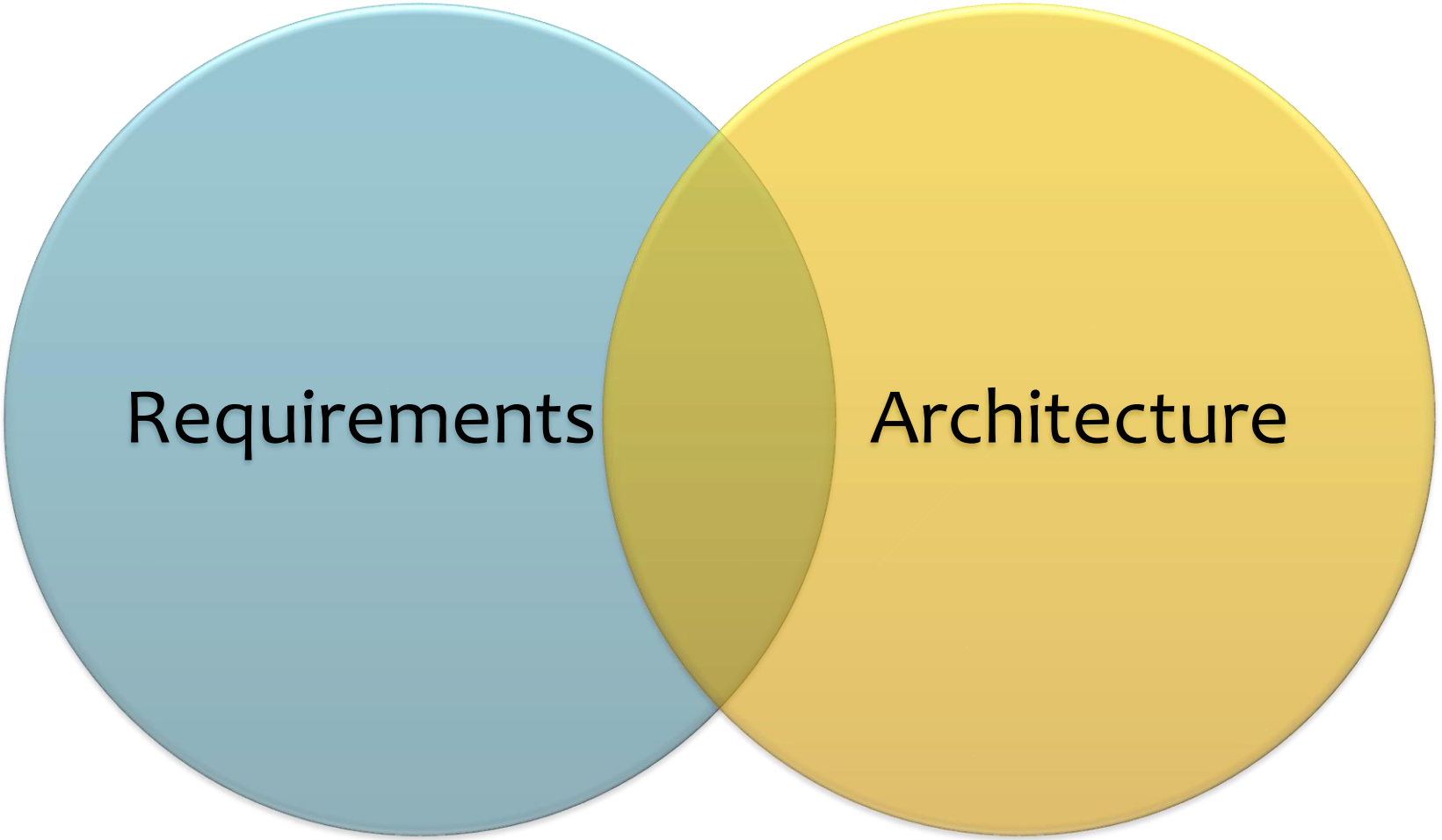
João Pimentel

- 3rd year Ph.D – UFPE/Brazil
- In trento for 1 year
  - From September 2012

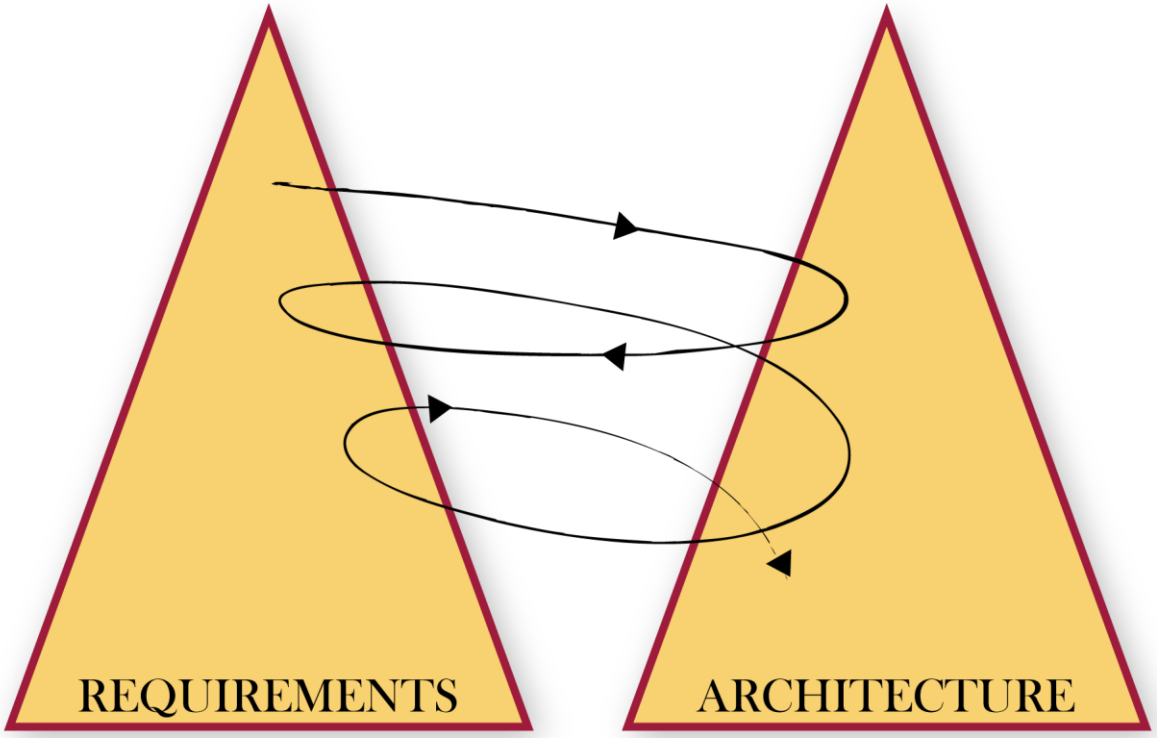
- www.cin.ufpe.br/~jhcp

(1) Requirements and Architecture

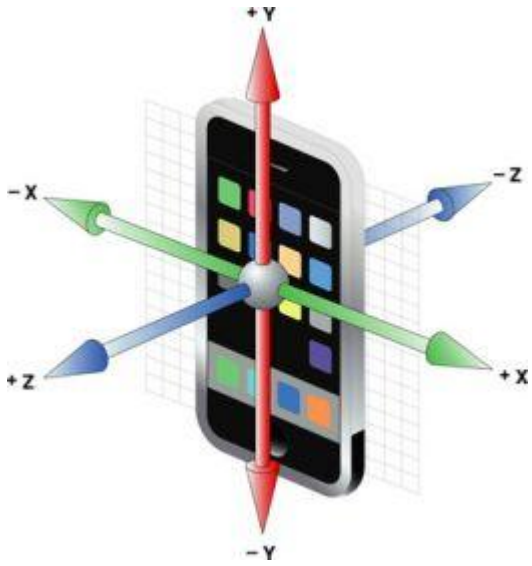

Requirements
Problem
User/client needs
Domain characteristics
Subjective
Architecture
Solution
Developer concerns
Technological constraints
Objective


Requirements
Architecture

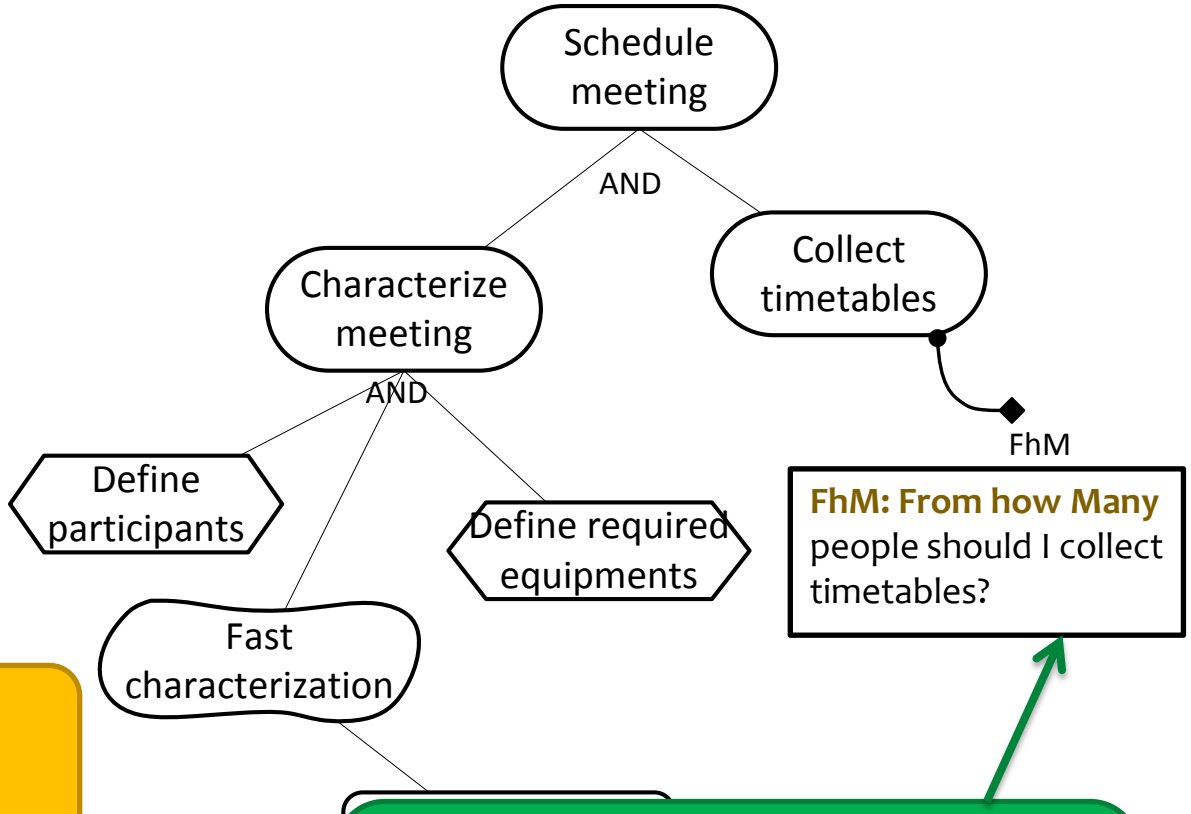
Twin Peaks



The architecture can influence the requirements



So, remember Zanshin?



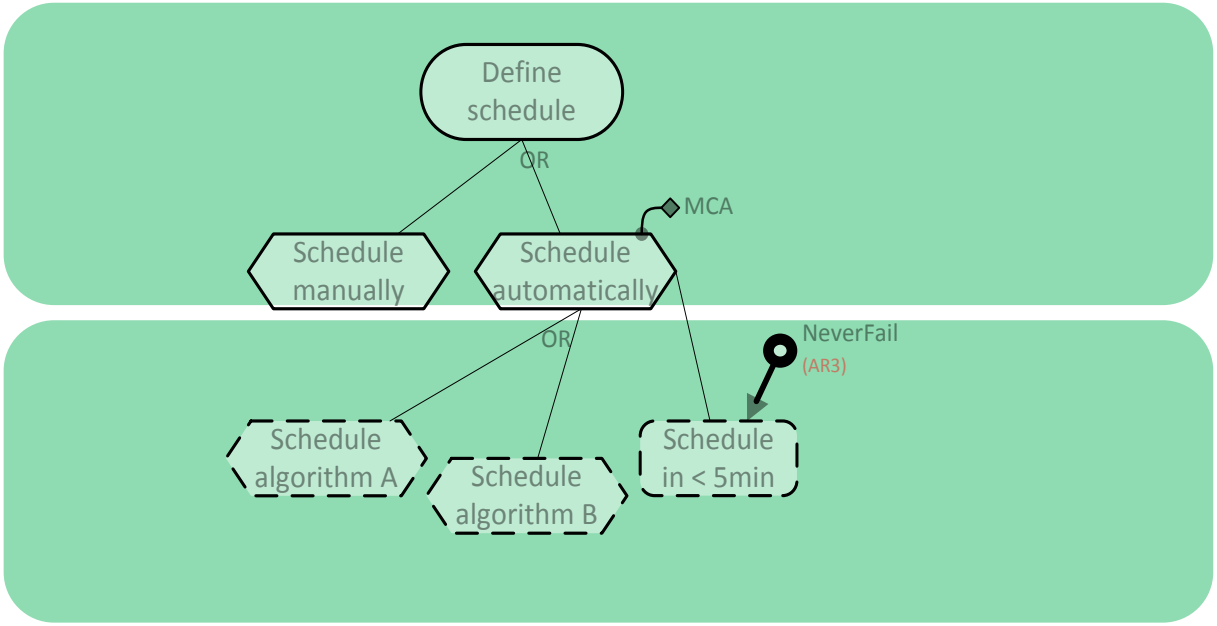
Awareness Requirements: stuff you monitor

Control Variable: stuff you can change to improve the results

Now with architectural design elements

Requirements

Architectural Design



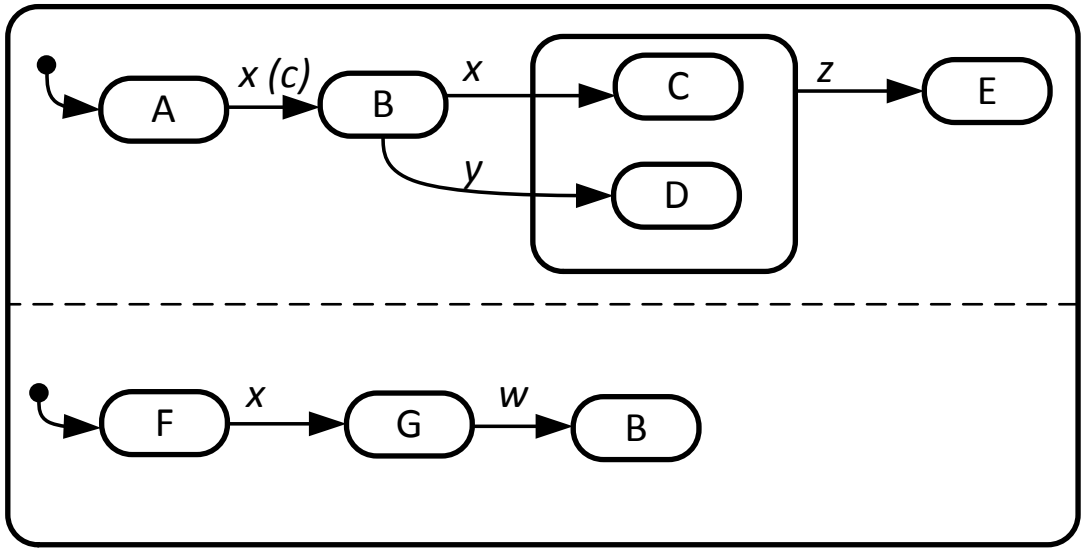
Bottom line

- By including architectural elements in the goal model we can borrow the i* infrastructure, such as reasoning for
  - Feedback-loop for adaptation
  - Context annotations
  - Preferences
  - Uncertainty
  - ...
- But we still need other kinds of models

(2)

From Goals to Statecharts

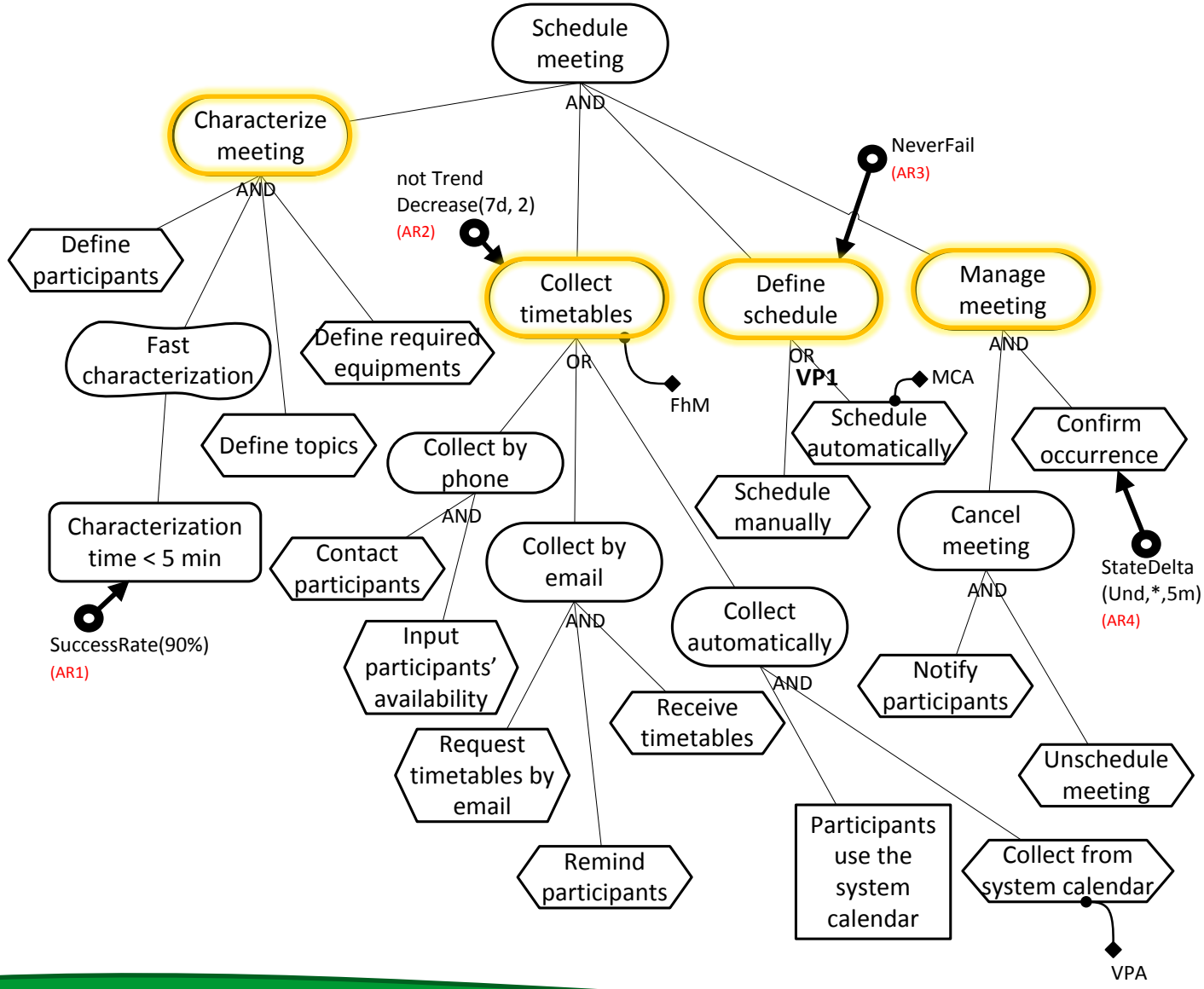
Statecharts



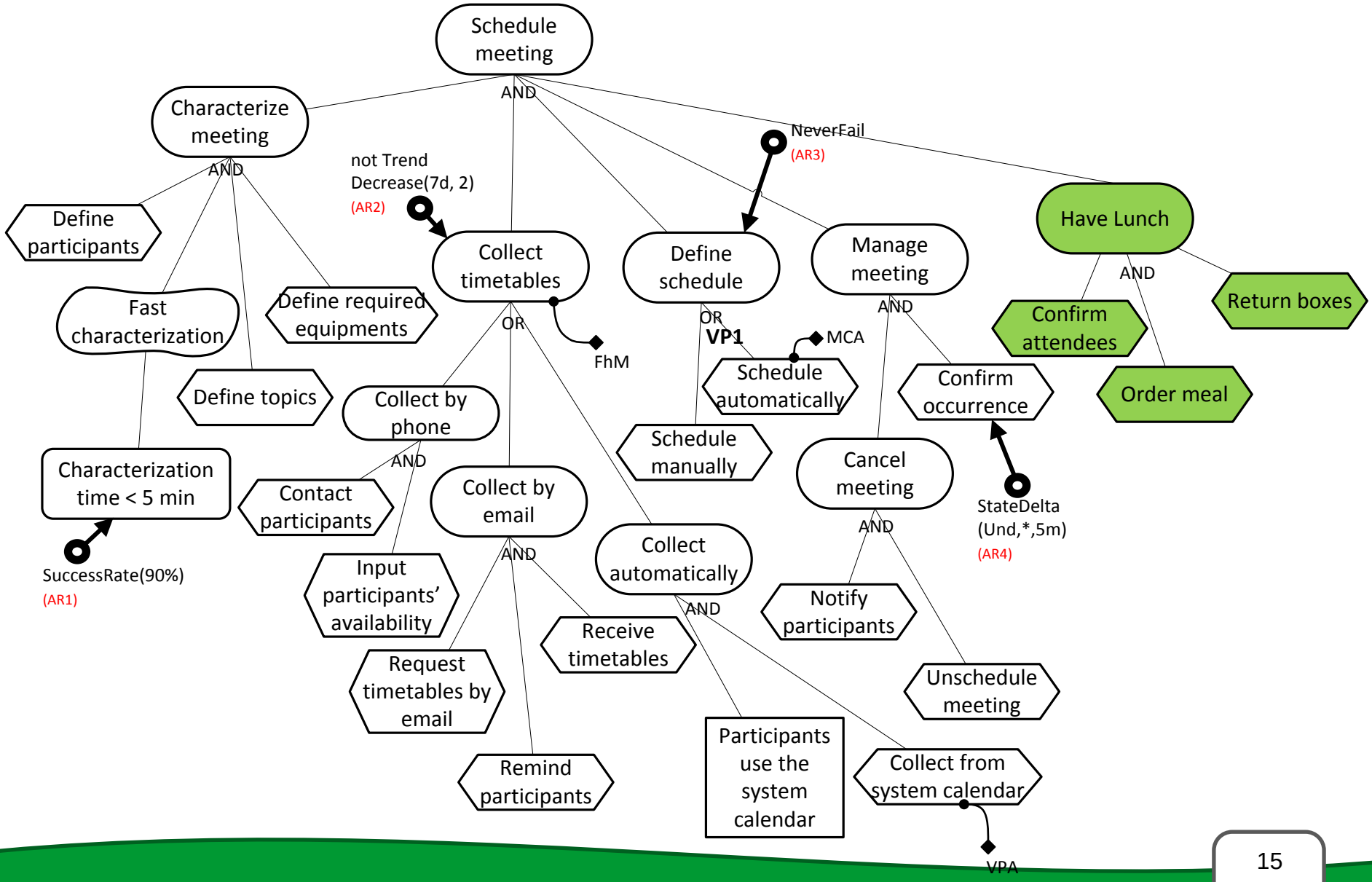
Outcome of the process

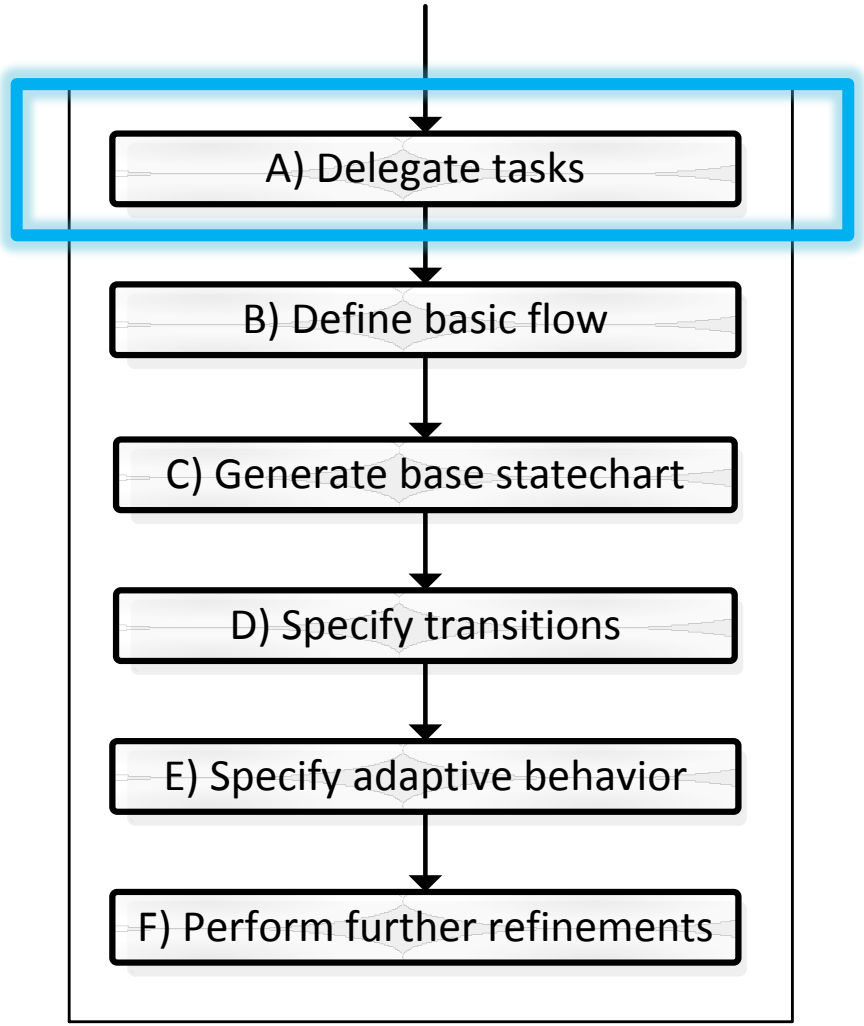
- Statechart with
  - Control Variables
  - Actions communicating with the Zanshin framework
- Plus
  - new Awareness Requirements
  - new Control Variables

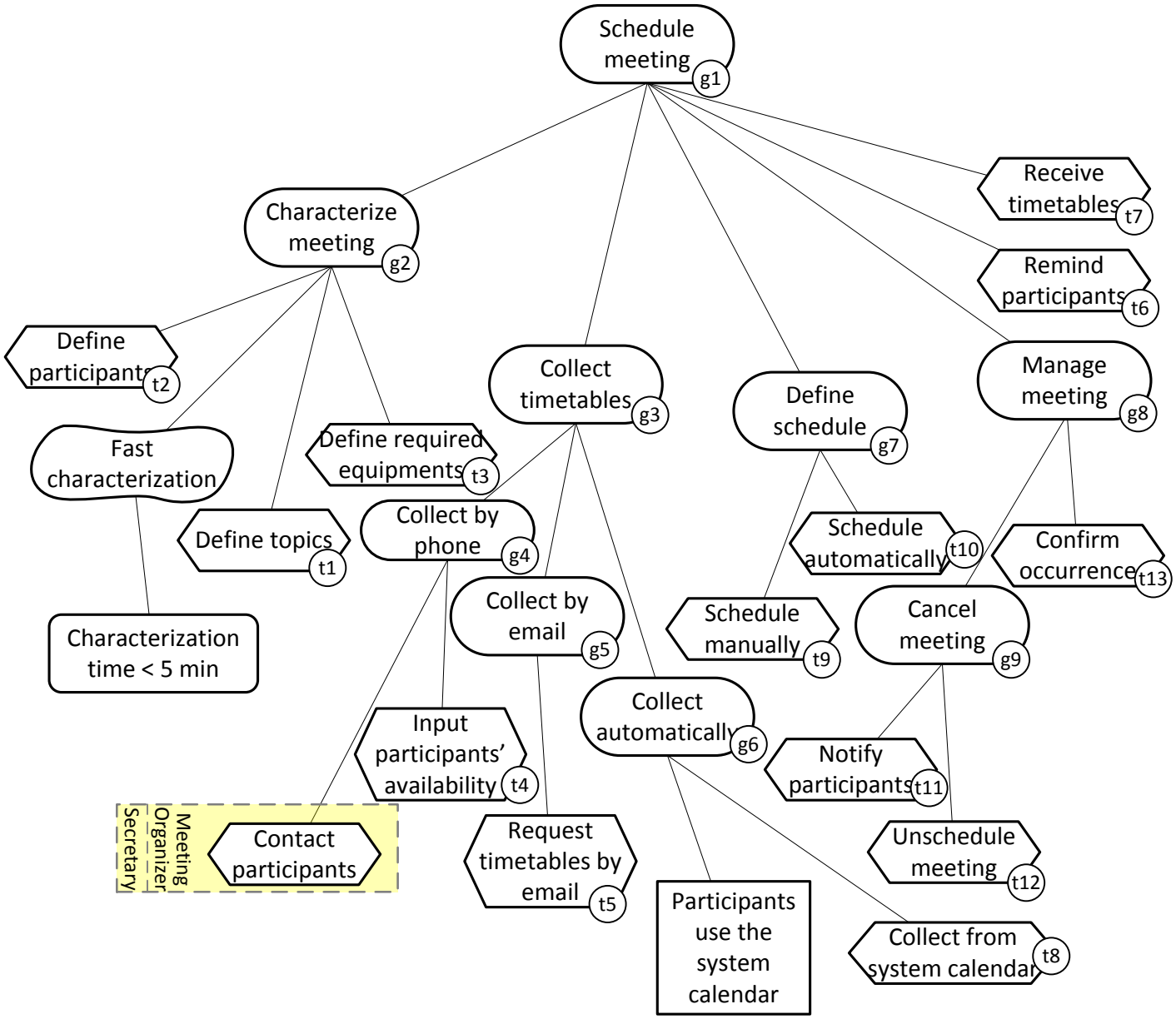
Running example

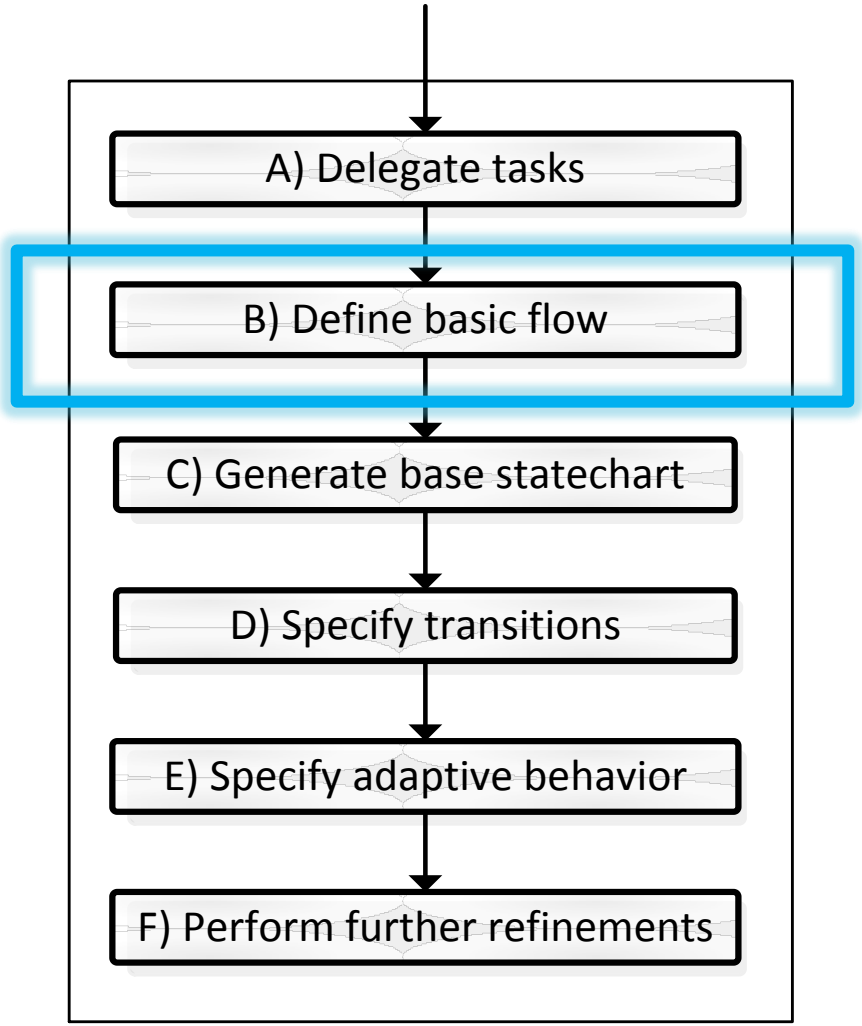


Running example v.2




A) Delegate tasks
B) Define basic flow
C) Generate base statechart
D) Specify transitions
E) Specify adaptive behavior
F) Perform further refinements


Schedule meeting
g1
Characterize meeting
g2
Collect timetables
g3
Define schedule
g7
Manage meeting
g8
Receive timetables
t7
Remind participants
t6
Define participants
t2
Fast characterization
Define required equipments
t3
Define topics
t1
Collect by phone
g4
Collect by email
g5
Collect automatically
g6
Schedule automatically
t10
Confirm occurrence
t13
Schedule manually
t9
Cancel meeting
g9
Characterization time < 5 min
Input participants' availability
t4
Notify participants
t11
Meeting Organizer
Secretary
Contact participants
Request timetables by email
t5
Unschedule meeting
t12
Participants use the system calendar
Collect from system calendar
t8

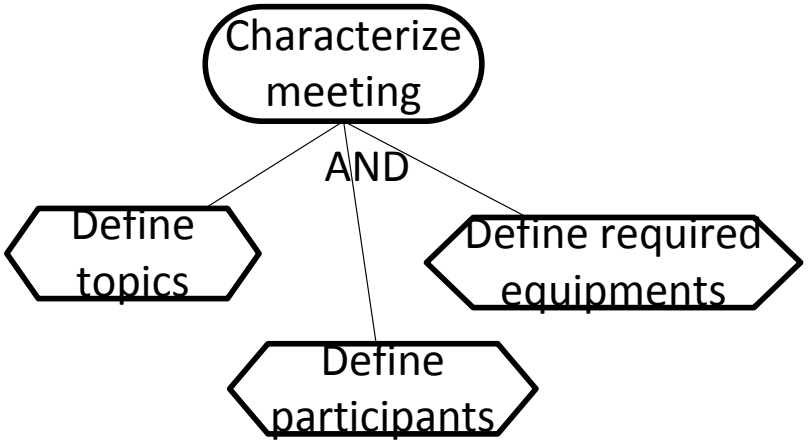

A) Delegate tasks
B) Define basic flow
C) Generate base statechart
D) Specify transitions
E) Specify adaptive behavior
F) Perform further refinements

Flow Expressions

(A B (C | D) E F G) o (H*)*

- A B ➔ B after A
- A | B ➔ A or B
- A* ➔ A zero or more times
- A+ ➔ what we wish for in our courses

  and A one or more times
- A? ➔ A is optional
- A o B ➔ A and B concurrently



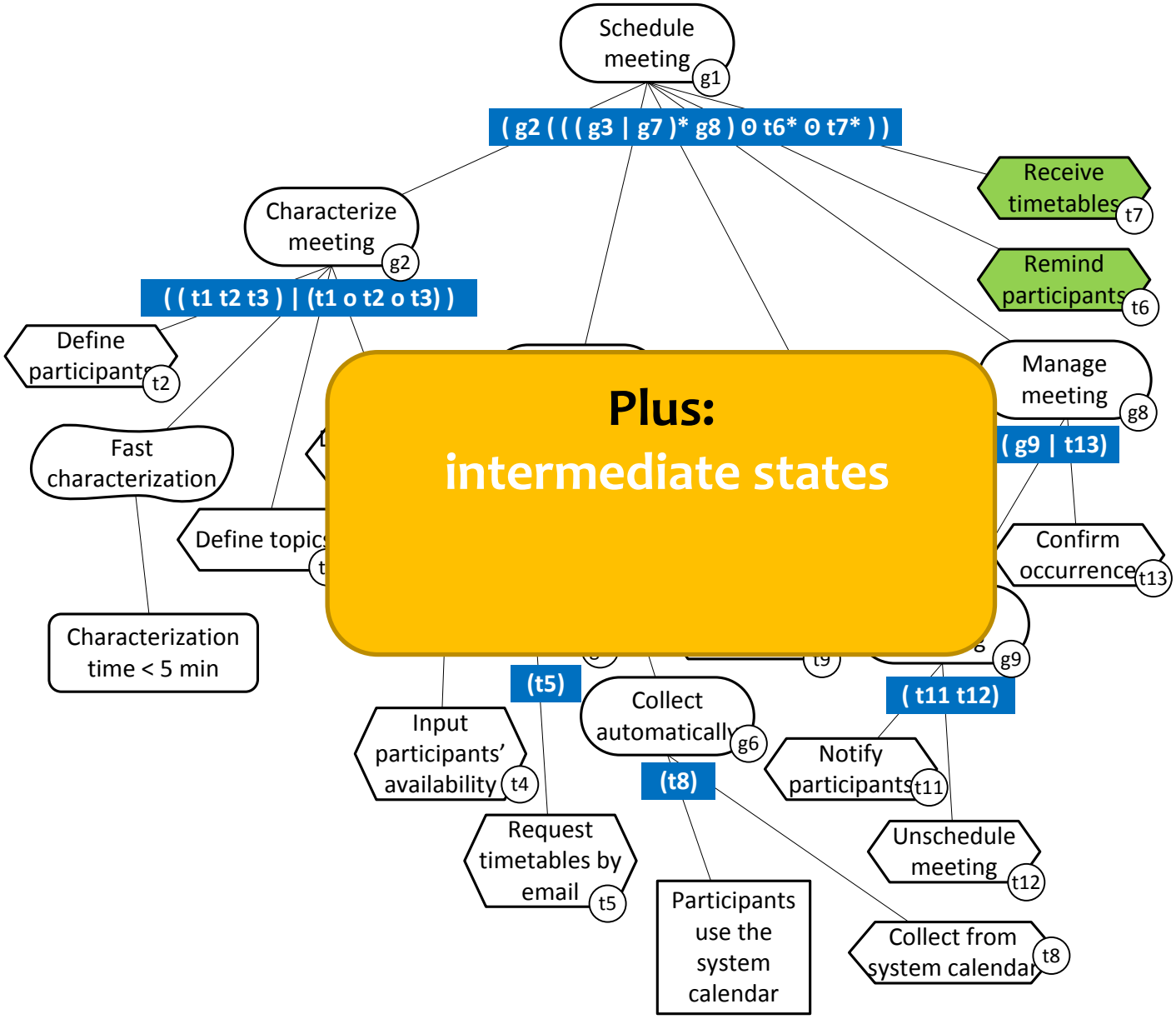
Option 1) DefineTopics DefineParticipants DefineRequiredEquipments

Option 2) DefineTopics ʘ DefineParticipants ʘ DefineRequiredEquipments

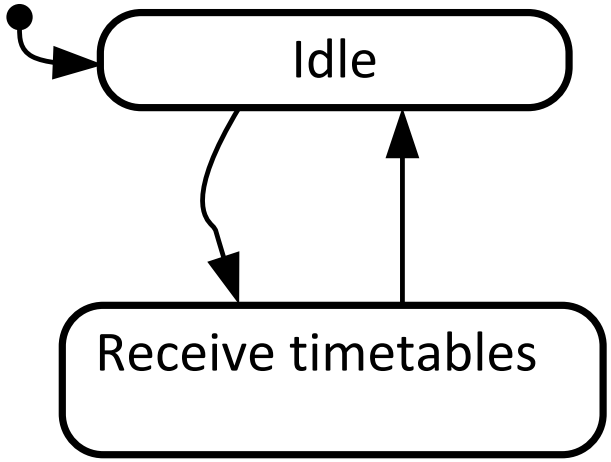
Option 3) DefineParticipants DefineTopics**?** DefineRequiredEquipments**?**

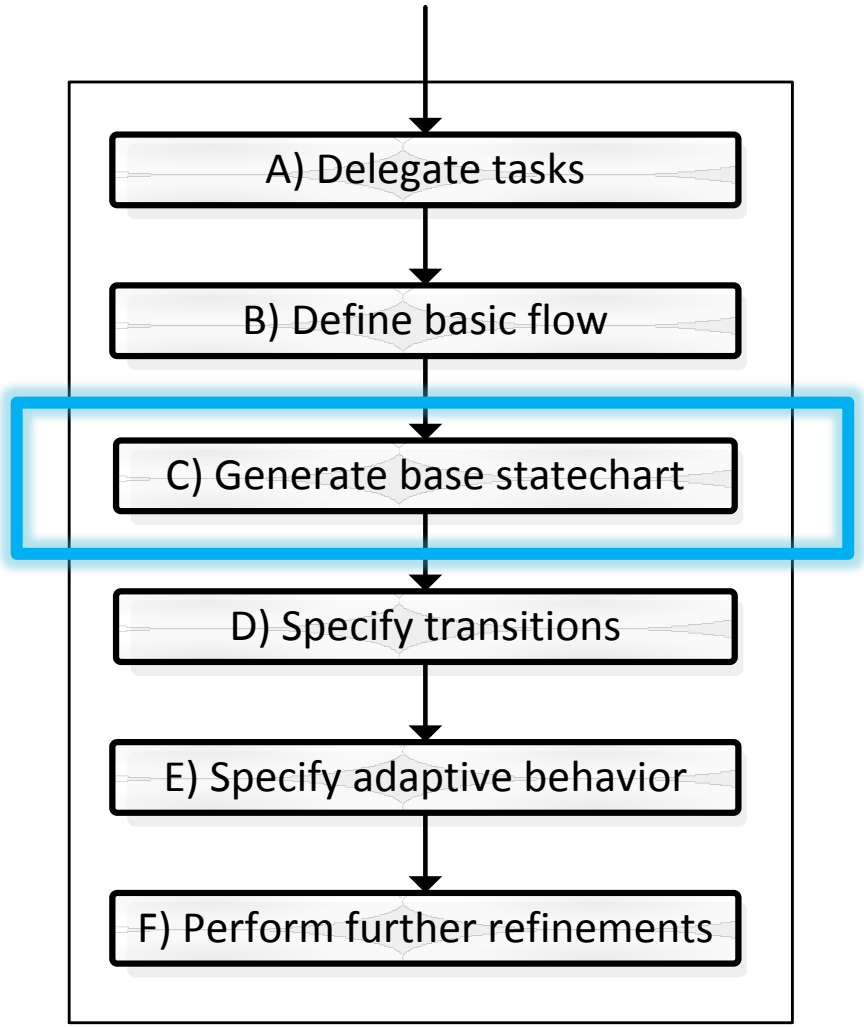
Option 4) (DefineParticipants DefineTopics DefineRequiredEquipments**) |**
(DefineTopics ʘ DefineParticipants ʘ DefineRequiredEquipments**)**

Alternative behaviors


Schedule meeting
g1
(g2 (((g3 | g7)* g8) ʘ t6* ʘ t7*))
Receive timetables
t7
Characterize meeting
g2
Remind participants
t6
((t1 t2 t3) | (t1 o t2 o t3))
Define participants
t2
Manage meeting
g8
Plus:
intermediate states
Fast characterization
(g9 | t13)
Define topics
Confirm occurrence
t13
Characterization time < 5 min
g9
(t5)
Collect automatically
g6
(t11 t12)
Input participants' availability
t4
Notify participants
t11
(t8)
Request timetables by email
t5
Unschedule meeting
t12
Participants use the system calendar
Collect from system calendar
t8

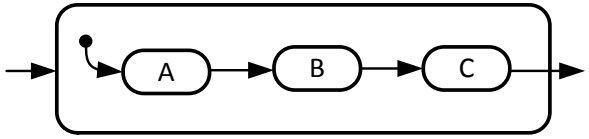
Intermediate states



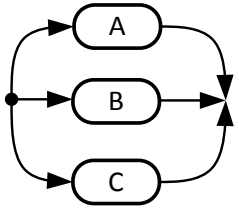

A) Delegate tasks
B) Define basic flow
C) Generate base statechart
D) Specify transitions
E) Specify adaptive behavior
F) Perform further refinements

Transformation patterns

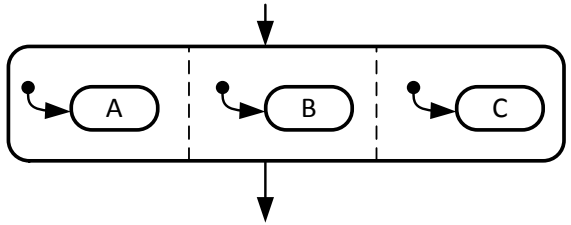
Sequential tasks:
ABC



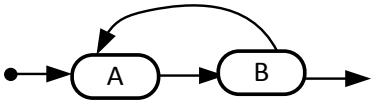
Alternative tasks
(triggered):
A | B | C



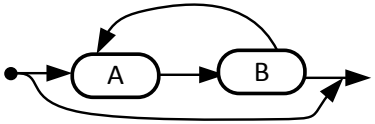
Concurrent tasks:
A ʘ B ʘ C



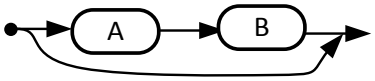
One or more
executions: (AB)+



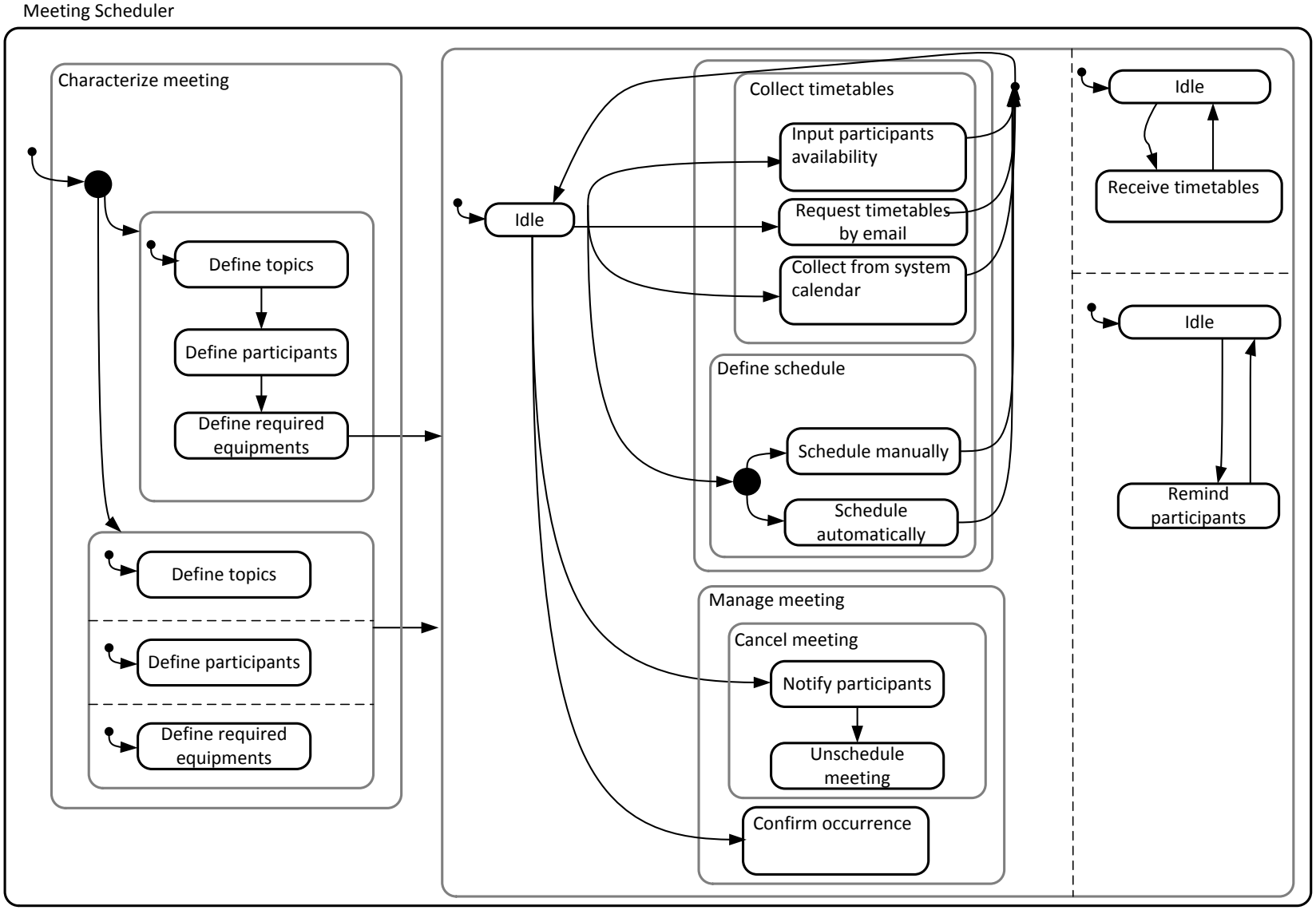
Zero or more
executions: (AB)*

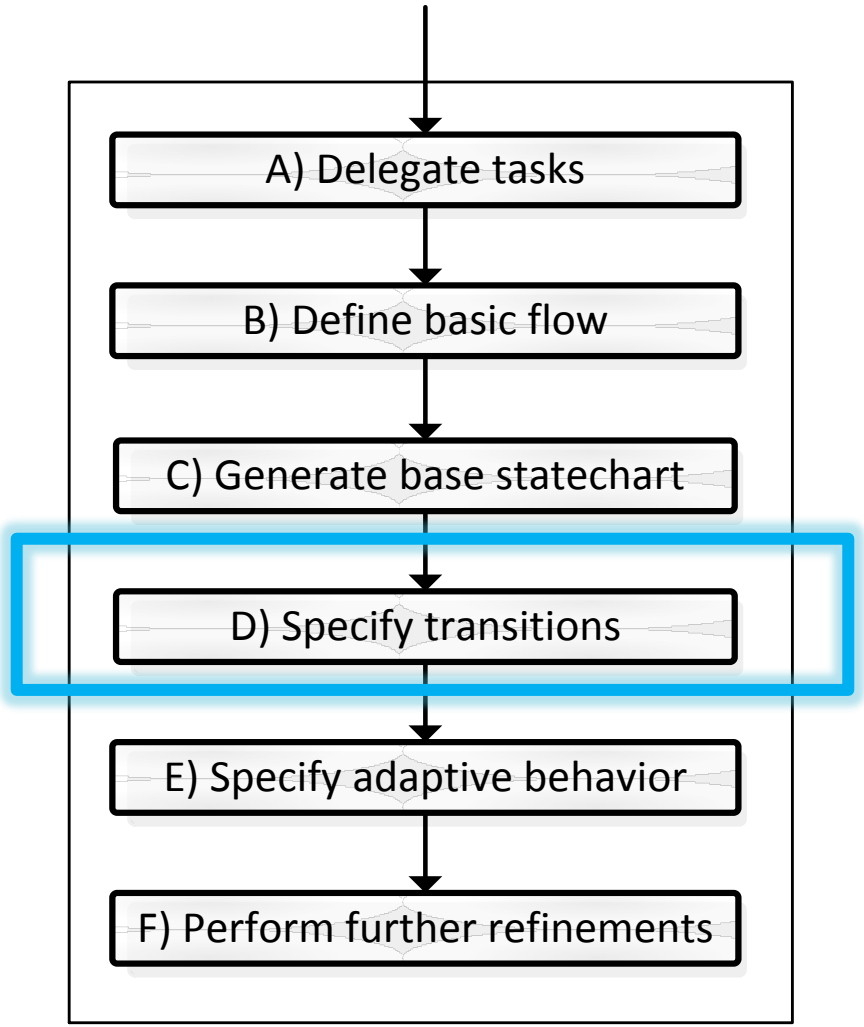


Optional
execution: (AB)?



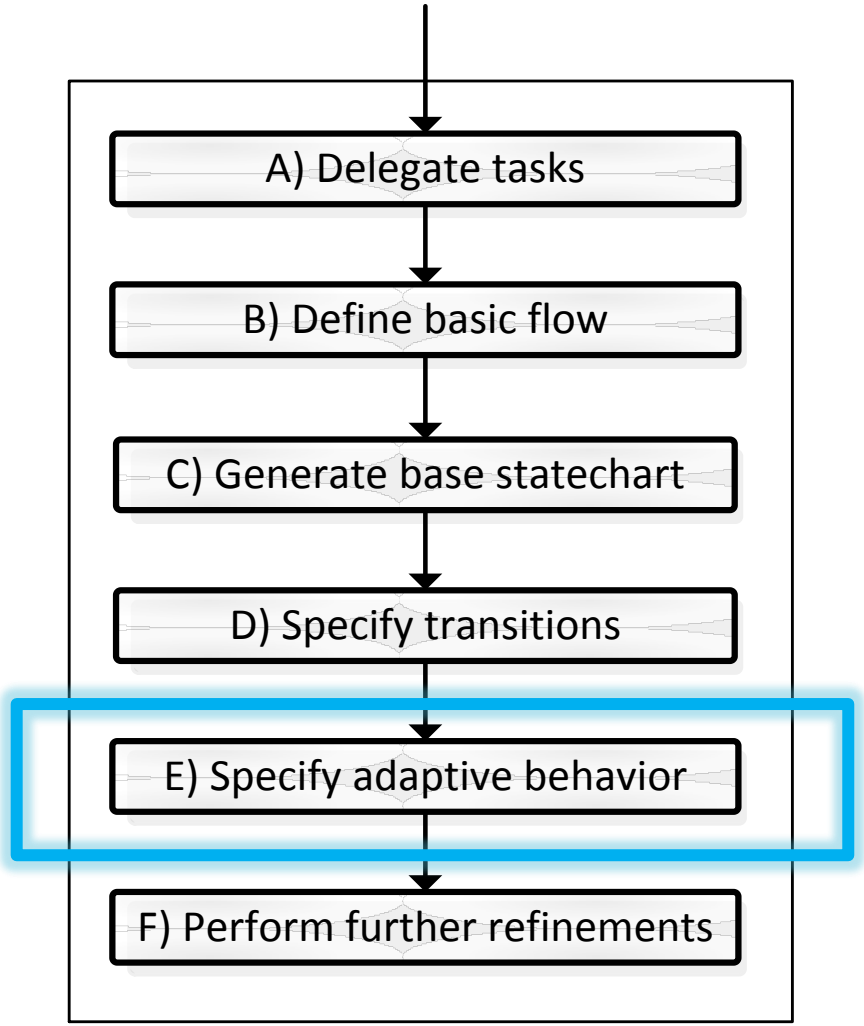
Generated statechart




A) Delegate tasks
B) Define basic flow
C) Generate base statechart
D) Specify transitions
E) Specify adaptive behavior
F) Perform further refinements

Specify transitions

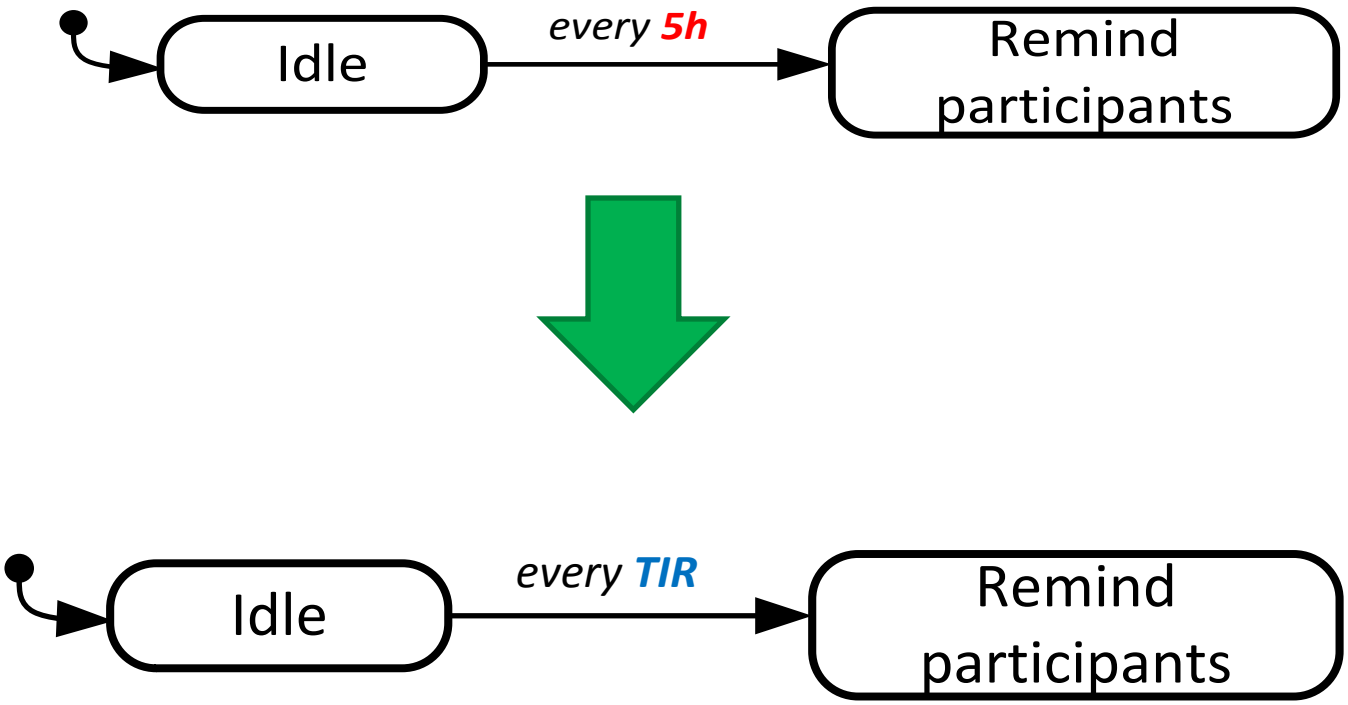
- Triggers
- Conditions (guards)
- Actions

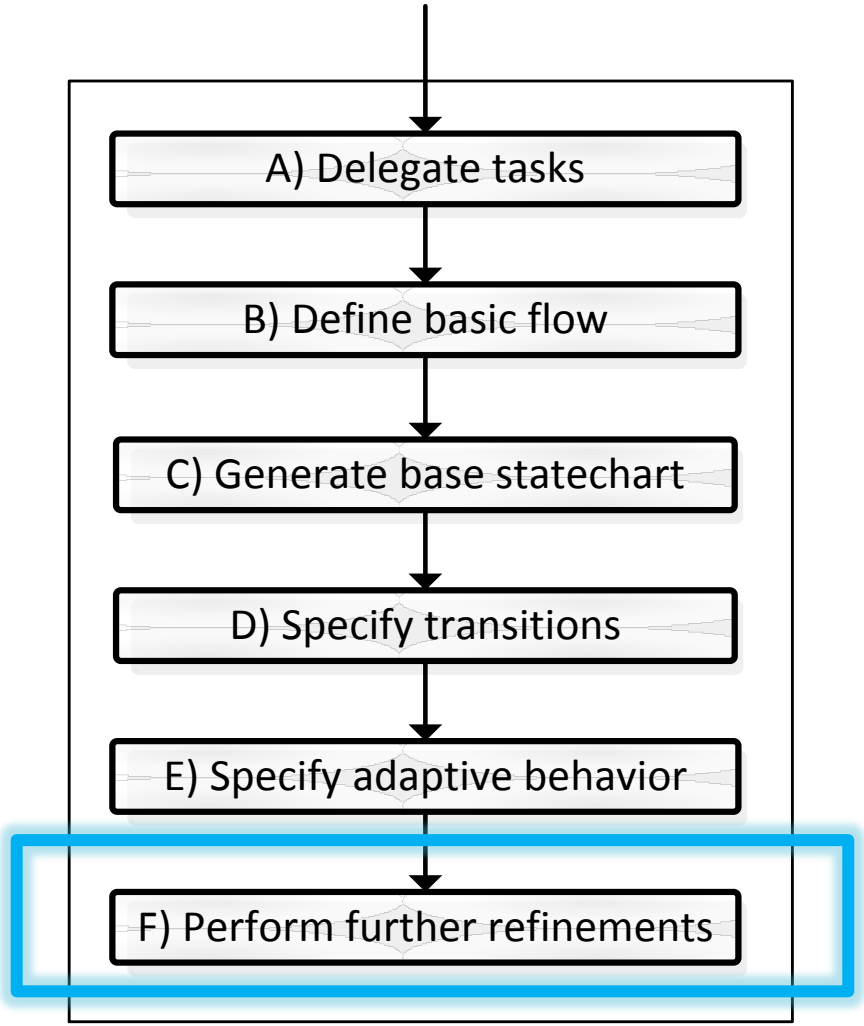

A) Delegate tasks
B) Define basic flow
C) Generate base statechart
D) Specify transitions
E) Specify adaptive behavior
F) Perform further refinements

Specify Adaptive Behavior

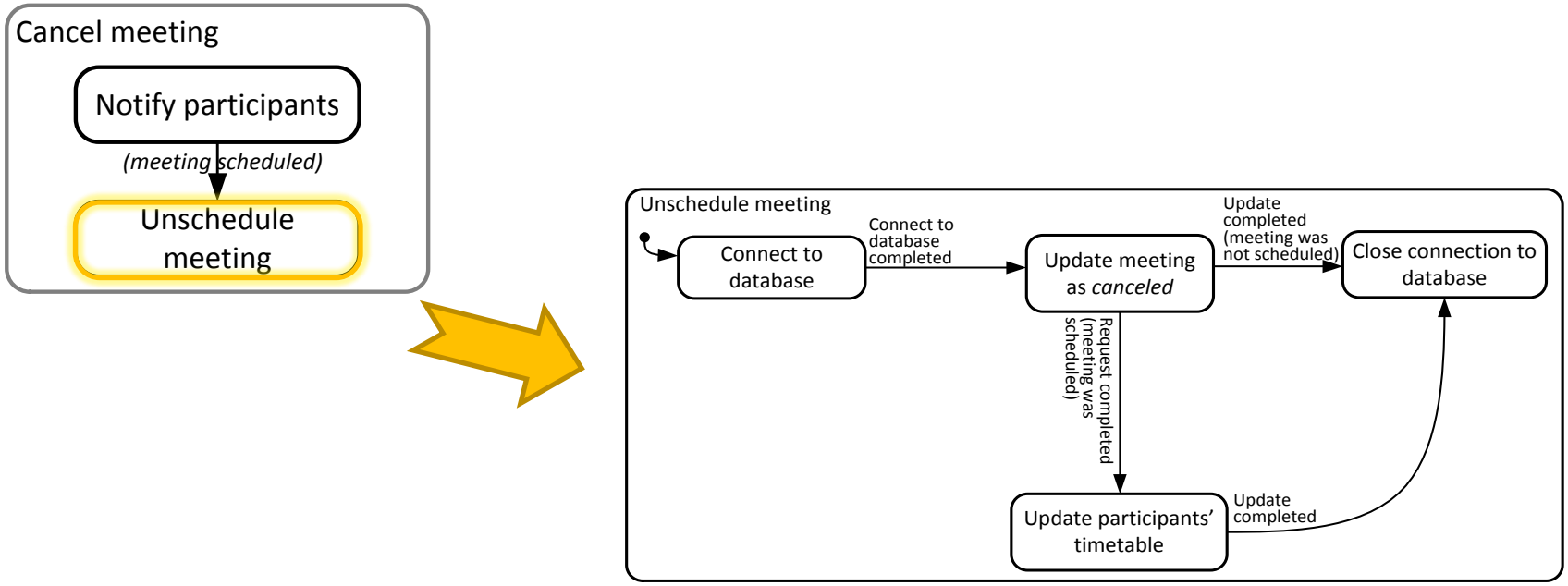
- Additional AwReqs
- Additional Control Variables
  - ***CSC*** *– Characterize in Sequence or Concurrently*
  - ***TIR*** *– Time Interval between Reminders*
  - ***ScA*** *– Scheduling Algorithm*
- Actions for Zanshin
- Receive instructions from Zanshin

Control Variable in a trigger

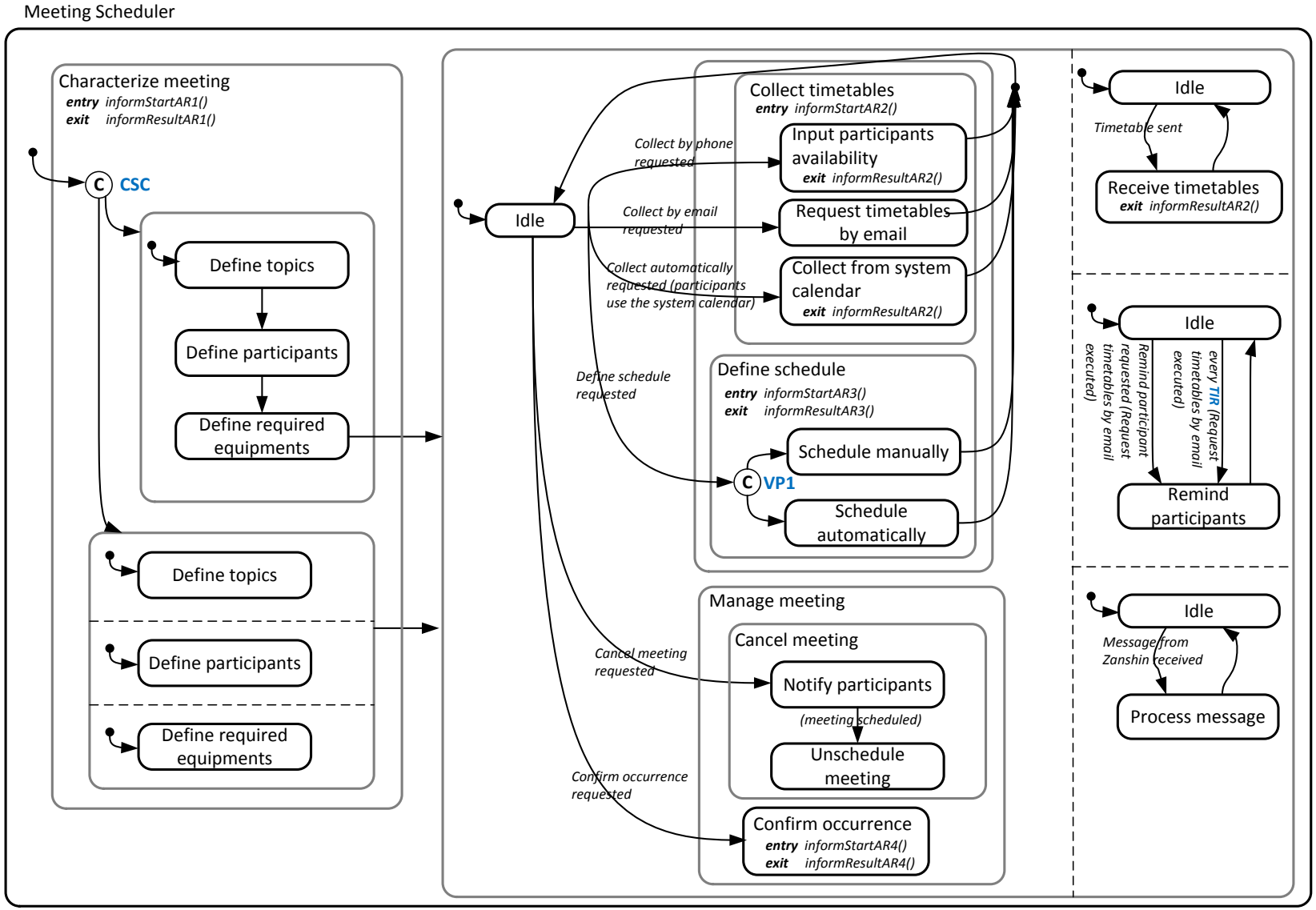



A) Delegate tasks
B) Define basic flow
C) Generate base statechart
D) Specify transitions
E) Specify adaptive behavior
F) Perform further refinements

Perform further refinements



Final statechart



(3)

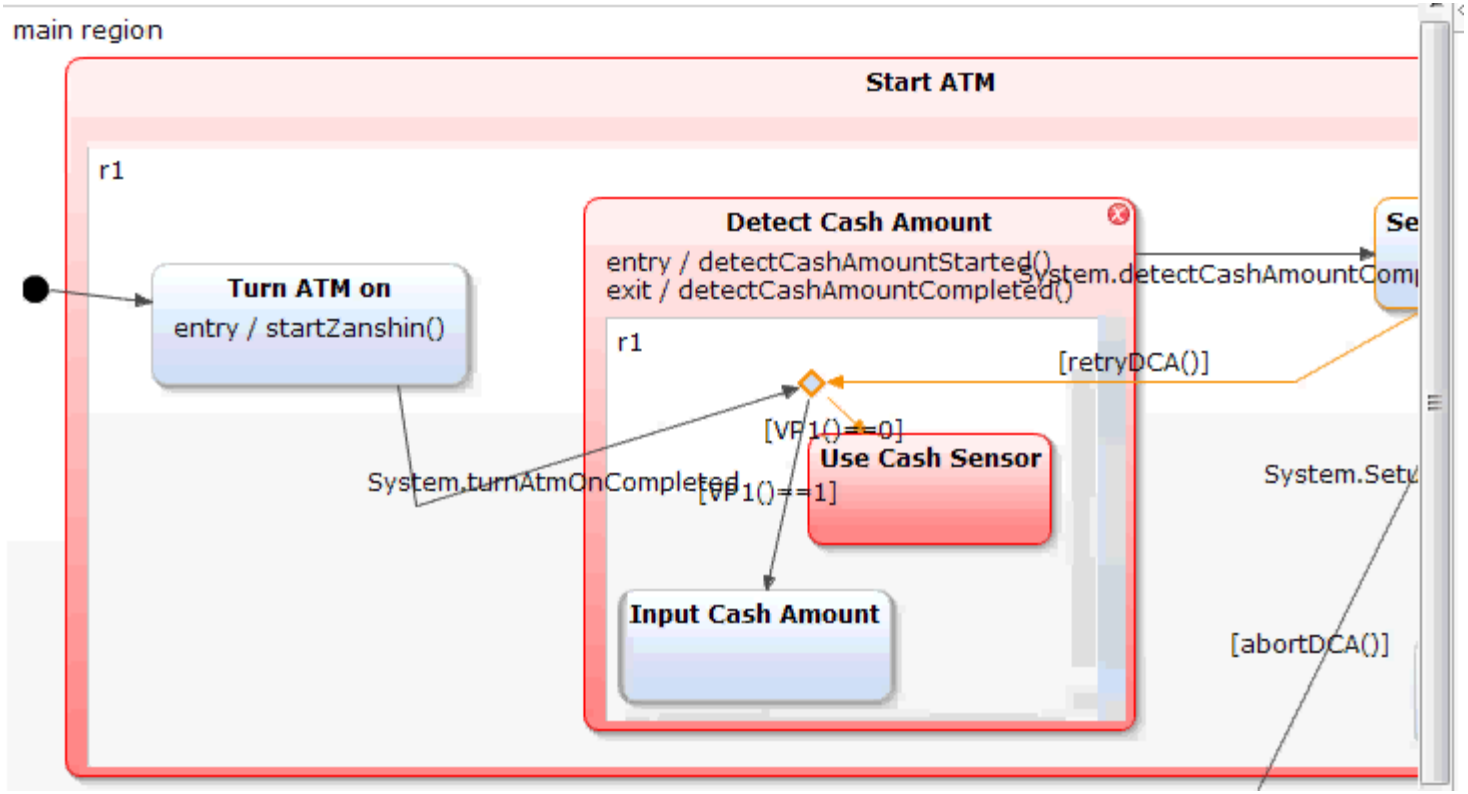
Simulation

Tools

- Yakindu Statechart Tool
  - http://statecharts.org/index.html
- Zanshin Framework
  - https://github.com/sefms-disi-unitn/Zanshin

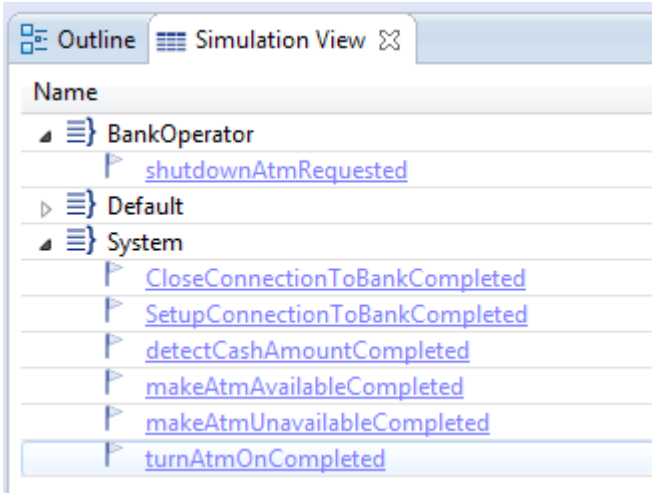
Statechart simulation

Active states are pink



Can call Java methods → Integration with Zanshin

Statechart simulation



Events can be triggered manually

Future work

- Tool support with automatic derivation
- Heuristics for defining the flow expressions
- Co-evolution of the goal and statechart models

Thanks!