Human-Inspired Metaphors for Software: What, Why and How?

Professor Michael Winikoff
IPL, 16th Sept. 2014

"I don't know. I'd never been that brief." – Alan Kay
(when asked if he could give a two-hour lecture on the FLEX machine)


Dole
PEEL, PLAY & WIN!
Download the free app
스티커 속의 코드를 통해
다양한 Dole앱을 무료로 즐기고
경품도 받으세요.
ドールのお楽しみアプリを
今すぐダウンロードしよう！
(無料)


Dole
PEEL, PLAY & WIN!
Download the free app
스티커 속의 코드를 통해
다양한 Dole앱을 무료로 즐기고
경품도 받으세요.
ドールのお楽しみアプリを
今すぐダウンロードしよう！
(無料)

Software is everywhere!

Summary

- Software is everywhere
- Software is based on a metaphor
- The choice of metaphor matters
- Traditional “sequence of instructions” metaphor has limitations
- Other metaphors? Draw inspiration from humans & human society!
- Key questions: what metaphors? how realise in software? how guide design?
- (unrelated) On technology, universities and disruption

```
  if (func.charAt(0) != '_') {
  // ie the goal is not a system predicate
    // if there is an alternative clause push
choicepoint:
    if (goal.numclauses > clausenum)
      stack.push(new ChoicePoint(clausenum +1, goal));

    if (clausenum > goal.numclauses) {
      clause = failgoal;
      IO.diagnostic(func + "/" + arity + " undefined!");
      clause = new TermList(new Term("fail",0), goal);
      clause.resolve(db);
    } else
      clause = goal.definer[clausenum];

    clausenum = 1; // reset ...
    // check unification ...
    vars = new Term[300]; // arbitrary limit
    if
(clause.term.refresh(vars).unify(goal.term,stack)) {
```

```
  if (func.charAt(0) != '_') {
  // ie the goal is not a system predicate
    // if there is an alternative clause push
choicepoint:
    if (goal.numclauses > clausenum)
      stack.push(new ChoicePoint(clausenum +1, goal));

    if (clausenum > goal.numclauses) {
      clause = failgoal;
      IO.diagnostic(func + "/" + arity + " undefined!");
      clause = new TermList(new Term("fail",0), goal);
      clause.resolve(db);
    } else
      clause = goal.definer[clausenum];

    clausenum = 1; // reset ...
    // check unification ...
    vars = new Term[300]; // arbitrary limit
    if
(clause.term.refresh(vars).unify(goal.term,stack)) {
```

What’s the problem?

Natural usage requires
anticipating possible
sequences and paths ...

... which works really poorly if

- things can and do go wrong
- there are multiple ways of achieving desired outcomes
- you’re dealing with humans and need flexibility (e.g. business processes)

What's the problem?

Natural usage requires **anticipating** possible sequences and paths ...

... which works really poorly if

- things can and do go wrong
- there are multiple ways of achieving desired outcomes
- you're dealing with humans and need flexibility (e.g. business processes)

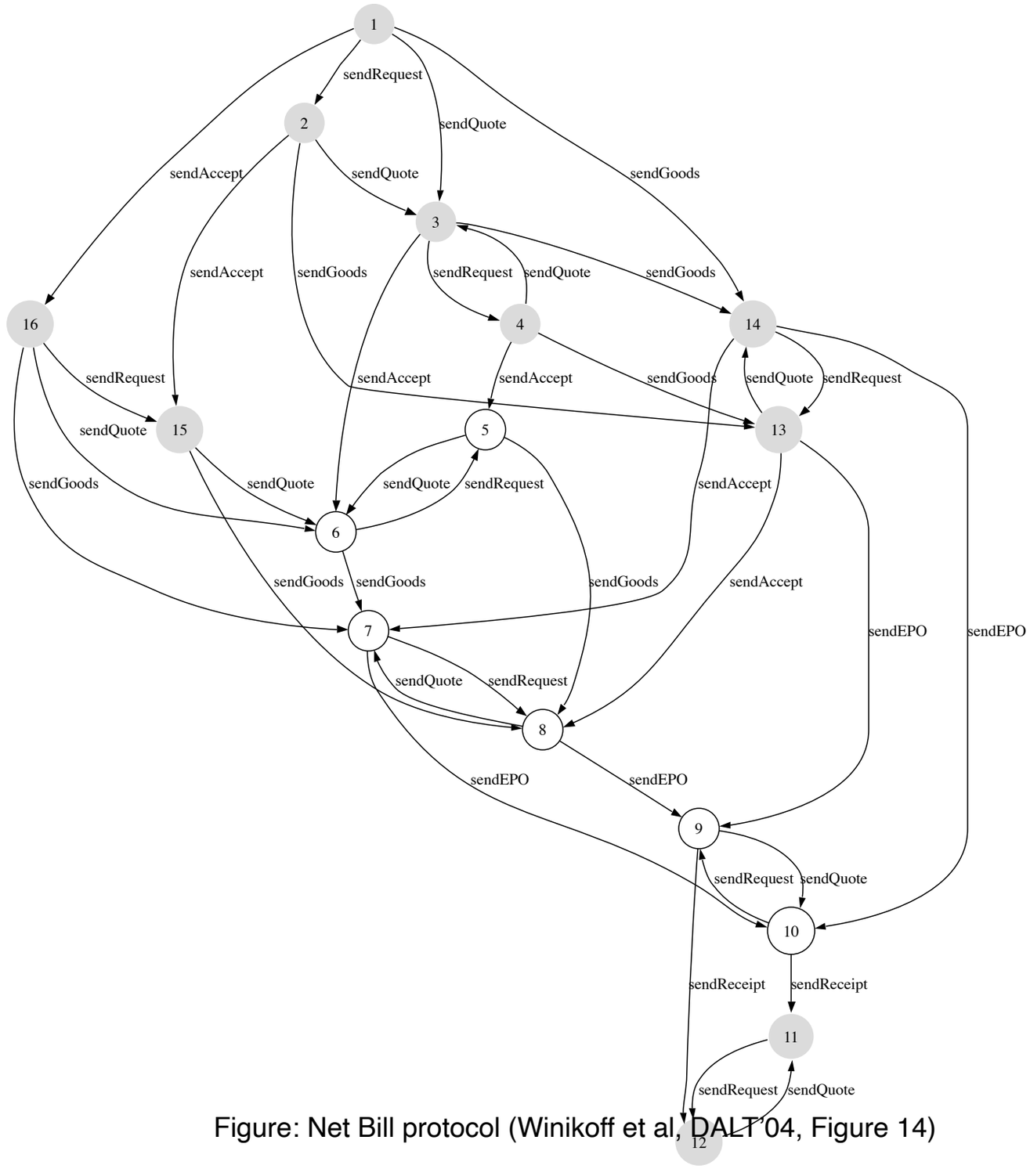
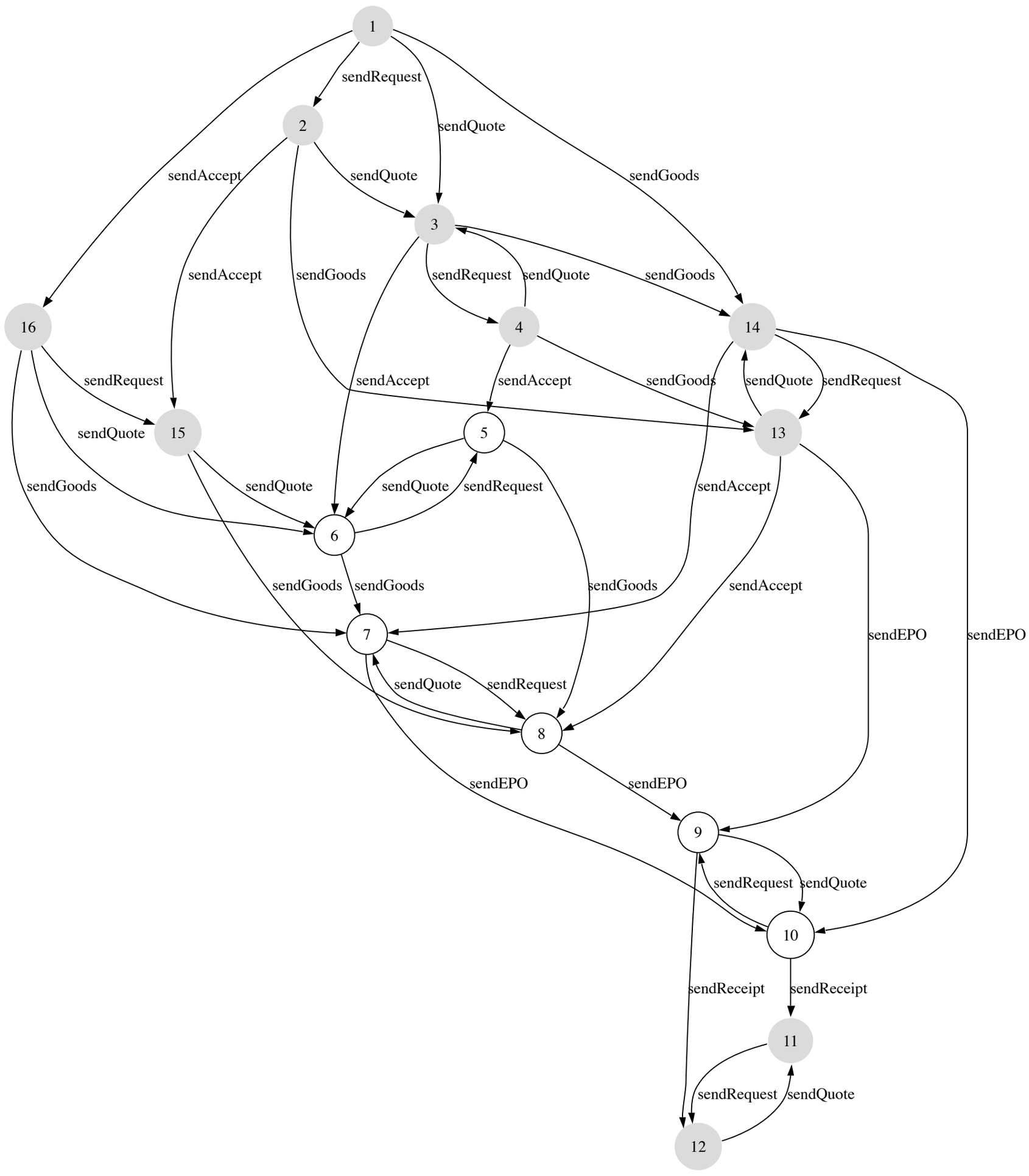


Figure: Net Bill protocol (Winikoff et al, DALT'04, Figure 14)


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
sendRequest
sendQuote
sendAccept
sendGoods
sendEPO

Six step process

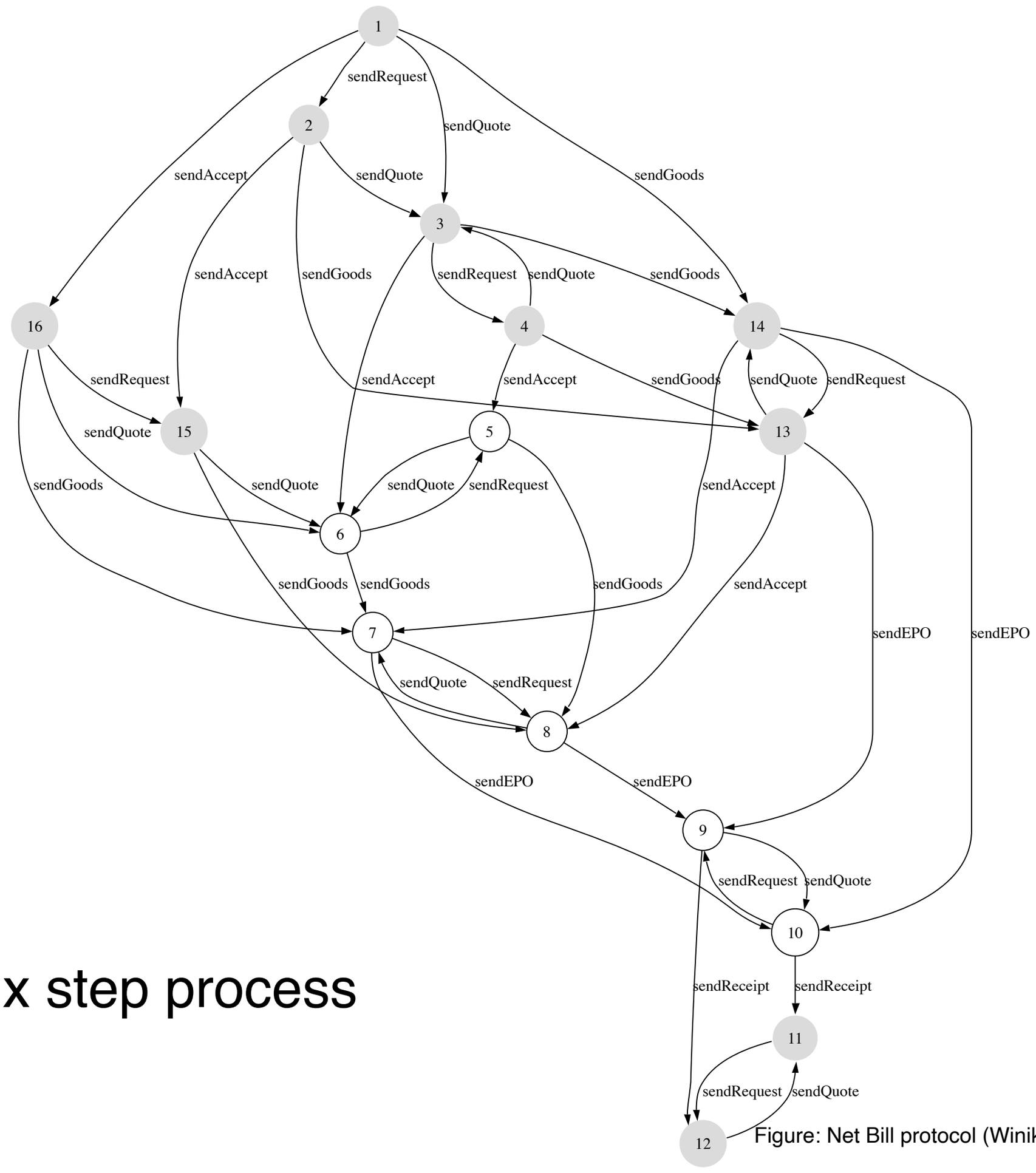
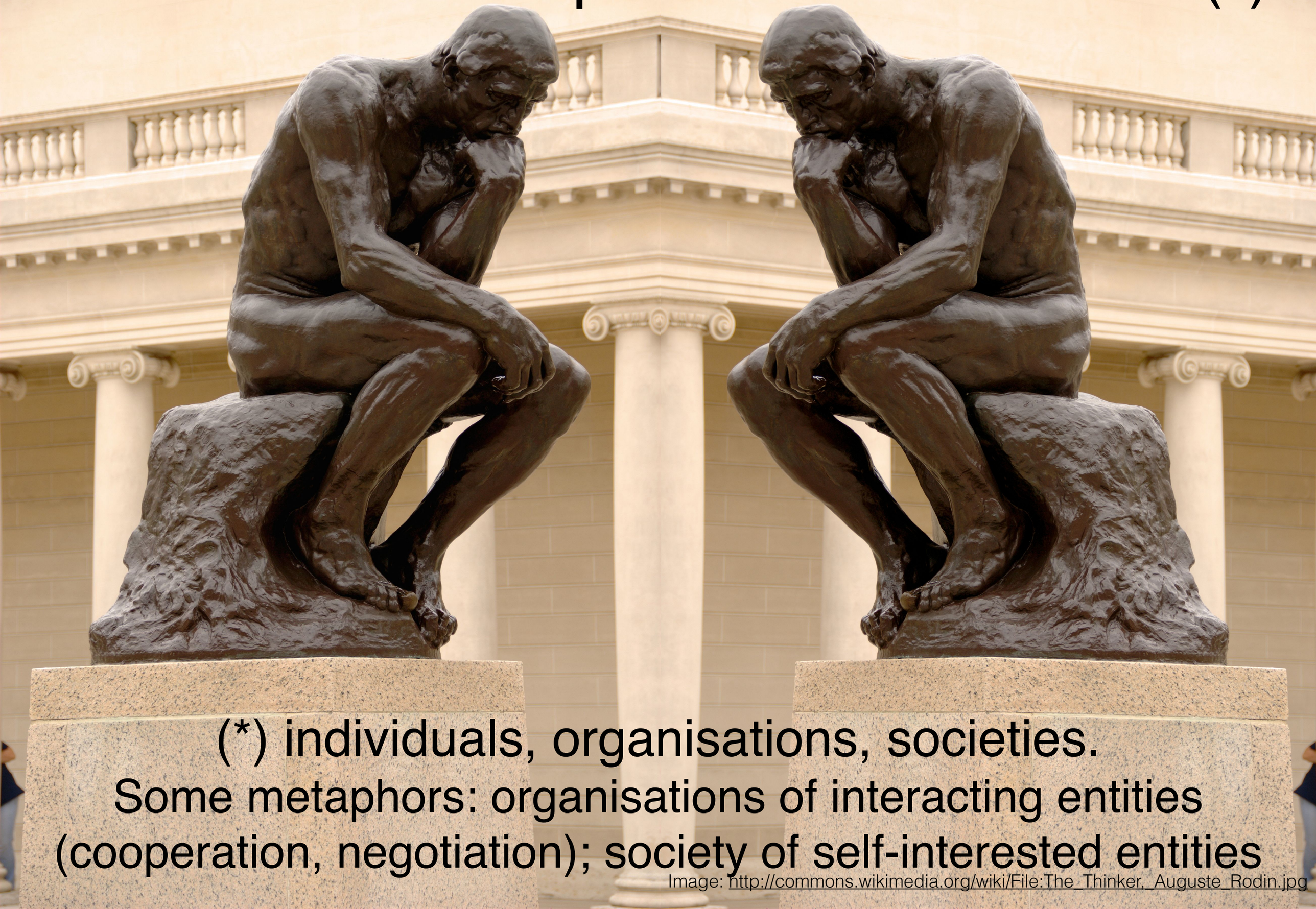


Figure: Net Bill protocol (Winikoff et al, DALT'04, Figure 14)

Solution: draw inspiration from humans(*)



(*) individuals, organisations, societies.

Some metaphors: organisations of interacting entities (cooperation, negotiation); society of self-interested entities

Image: http://commons.wikimedia.org/wiki/File:The_Thinker,_Auguste_Rodin.jpg

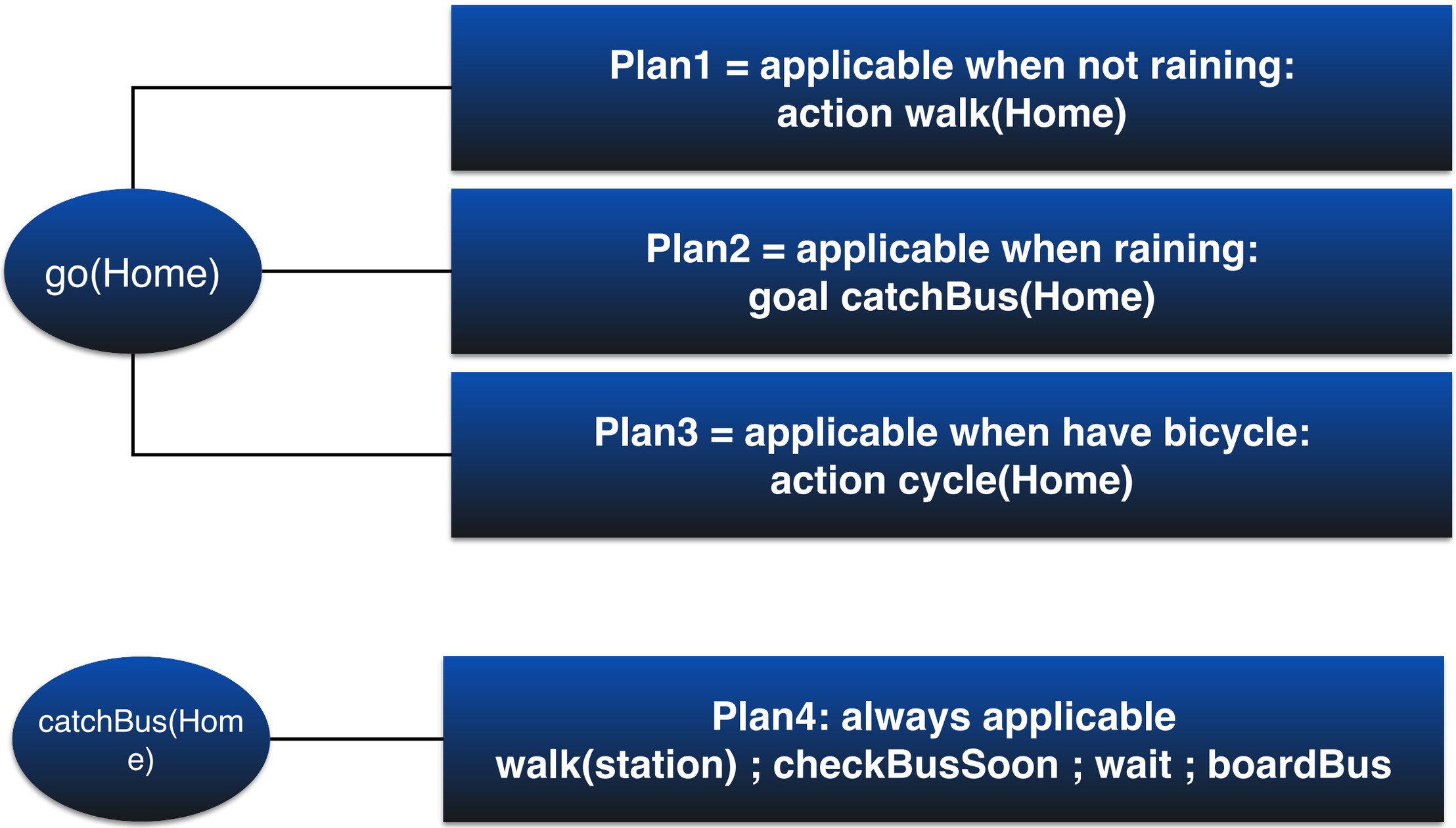
One Possible Metaphor: Goals and Plans

- Software entity (“Agent”) has a collection of active goals and a library of plans

- Goals are **persistent** - if one plan fails, try another!

- Each plan includes: what goal it can be used to achieve (“**why**”), **when** the plan can be used, and **what** the plan does.

Example


go(Home)
Plan1 = applicable when not raining:
action walk(Home)
Plan2 = applicable when raining:
goal catchBus(Home)
Plan3 = applicable when have bicycle:
action cycle(Home)
catchBus(Hom
e)
Plan4: always applicable
walk(station) ; checkBusSoon ; wait ; boardBus

Example

go(Home)

Plan1 = applicable when not raining:
action walk(Home)

Plan2 = applicable when raining:
goal catchBus(Home)

Plan3 = applicable when have bicycle:
action cycle(Home)

catchBus(Home)

Plan4: always applicable
walk(station) ; checkBusSoon ; wait ; boardBus

Example

go(Home)

? **Plan1 = applicable when not raining:**
action walk(Home)

? **Plan2 = applicable when raining:**
goal catchBus(Home)

? **Plan3 = applicable when have bicycle:**
action cycle(Home)

catchBus(Home)

Plan4: always applicable
walk(station) ; checkBusSoon ; wait ; boardBus

Example

go(Home)

? **Plan1 = applicable when not raining: action walk(Home)** ✗

? **Plan2 = applicable when raining: goal catchBus(Home)** ✓

? **Plan3 = applicable when have bicycle: action cycle(Home)** ✓

catchBus(Home)

Plan4: always applicable
walk(station) ; checkBusSoon ; wait ; boardBus

Example

go(Home)

?

Plan1 = applicable when not raining:
action walk(Home)

X

?

Plan2 = applicable when raining:
goal catchBus(Home)

?

Plan3 = applicable when have bicycle:
action cycle(Home)

catchBus(Home)

Plan4: always applicable
walk(station) ; checkBusSoon ; wait ; boardBus

Example

go(Home)

Plan1 = applicable when not raining:
action walk(Home)

X

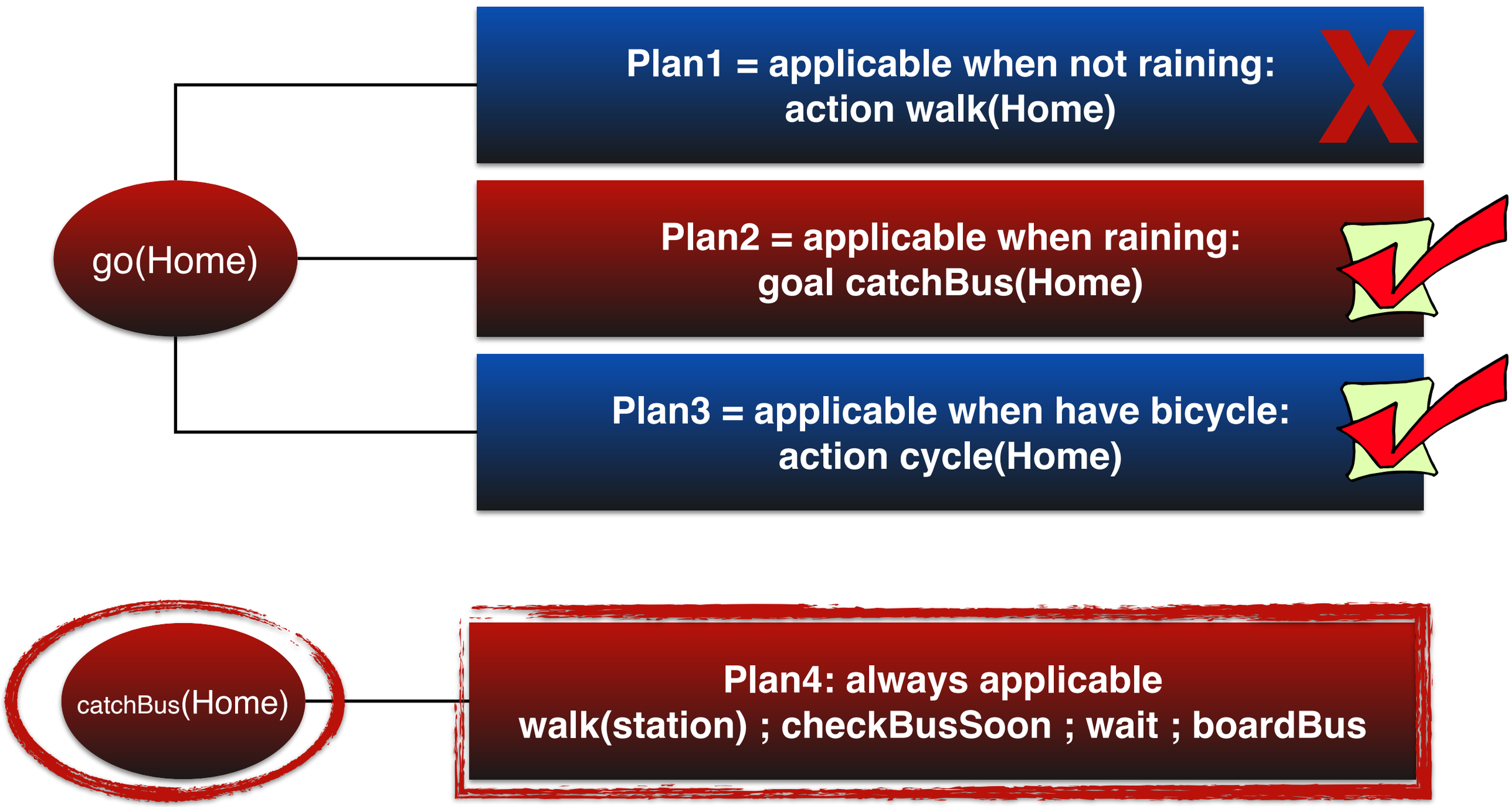
Plan2 = applicable when raining:
goal catchBus(Home)

Plan3 = applicable when have bicycle:
action cycle(Home)

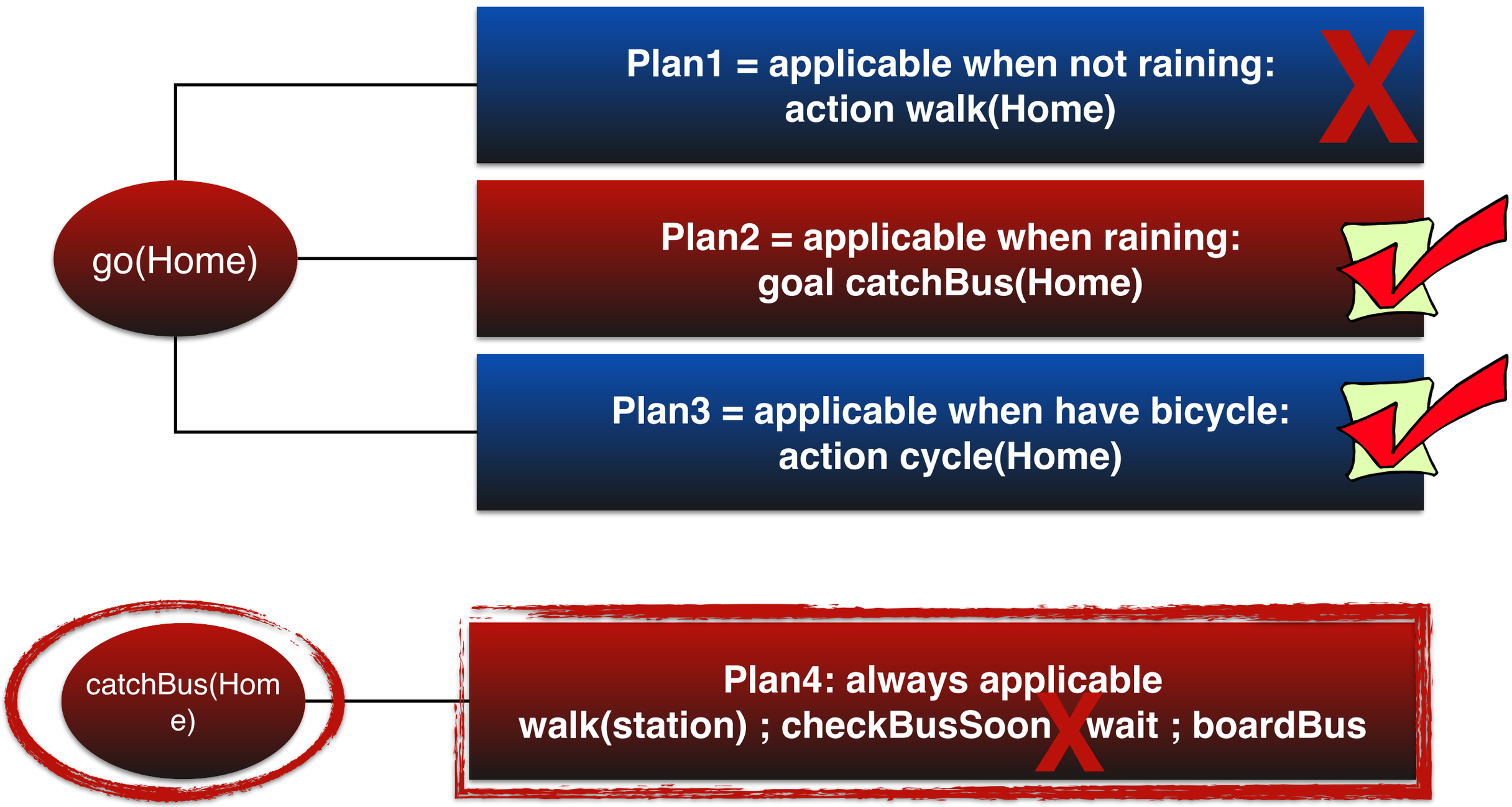
catchBus(Home)

Plan4: always applicable
walk(station) ; checkBusSoon ; wait ; boardBus

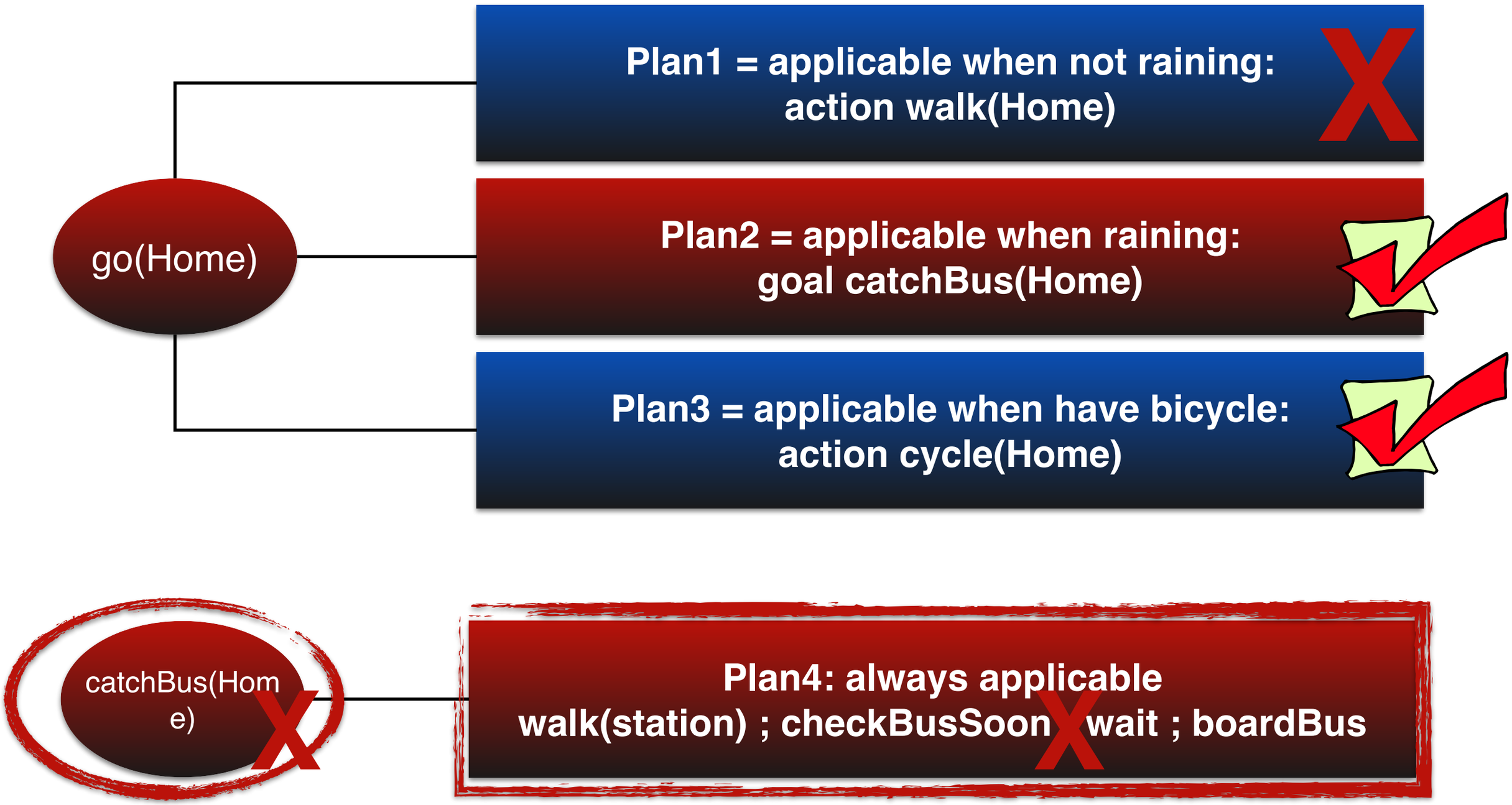
Example


Plan1 = applicable when not raining:
action walk(Home)
X
go(Home)
Plan2 = applicable when raining:
goal catchBus(Home)
Plan3 = applicable when have bicycle:
action cycle(Home)
catchBus(Home)
Plan4: always applicable
walk(station) ; checkBusSoon ; wait ; boardBus

Example


Plan1 = applicable when not raining:
action walk(Home)
X
go(Home)
Plan2 = applicable when raining:
goal catchBus(Home)
Plan3 = applicable when have bicycle:
action cycle(Home)
catchBus(Home)
Plan4: always applicable
walk(station) ; checkBusSoon X wait ; boardBus

Example

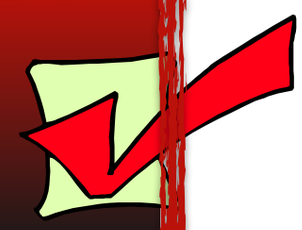

Plan1 = applicable when not raining:
action walk(Home)
X
go(Home)
Plan2 = applicable when raining:
goal catchBus(Home)
Plan3 = applicable when have bicycle:
action cycle(Home)
catchBus(Hom
e)
X
Plan4: always applicable
walk(station) ; checkBusSoon
X
wait ; boardBus

Example

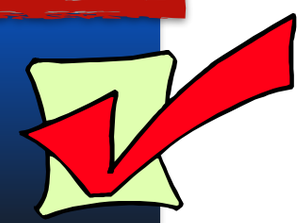
go(Home)

Plan1 = applicable when not raining:
action walk(Home) X

Plan2 = applicable when raining:
goal catchBus(Home) X



Plan3 = applicable when have bicycle:
action cycle(Home)



catchBus(Home)

Plan4: always applicable
walk(station) ; checkBusSoon ; wait ; boardBus X

Example

go(Home)

?

Plan1 = applicable when not raining:
action walk(Home)

Plan2 = applicable when raining:
goal catchBus(Home)

?

Plan3 = applicable when have bicycle:
action cycle(Home)

catchBus(Home)

Plan4: always applicable
walk(station) ; checkBusSoon ; wait ; boardBus

Example

go(Home)

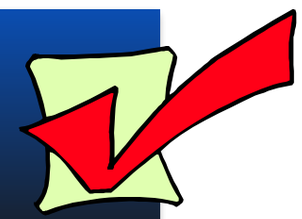
?

Plan1 = applicable when not raining:
action walk(Home)

Plan2 = applicable when raining:
goal catchBus(Home)

?

Plan3 = applicable when have bicycle:
action cycle(Home)



catchBus(Home)

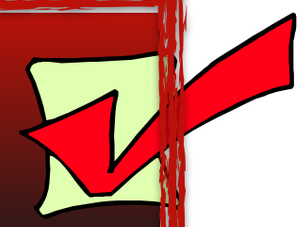
Plan4: always applicable
walk(station) ; checkBusSoon ; wait ; boardBus

Example

go(Home)

?

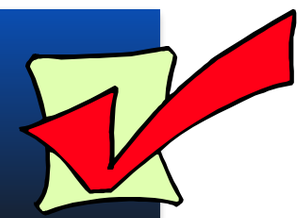
Plan1 = applicable when not raining:
action walk(Home)



Plan2 = applicable when raining:
goal catchBus(Home)

?

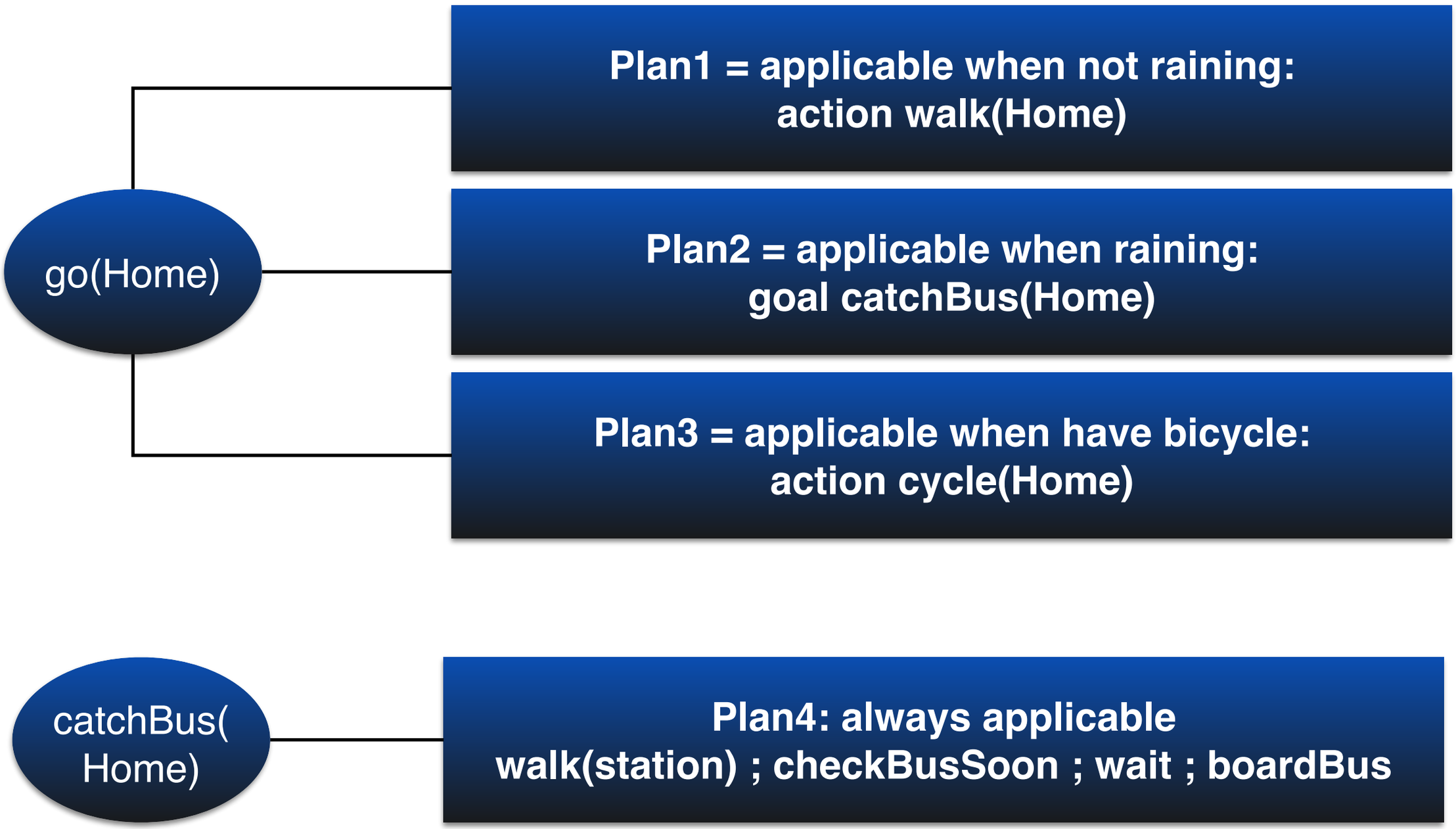
Plan3 = applicable when have bicycle:
action cycle(Home)



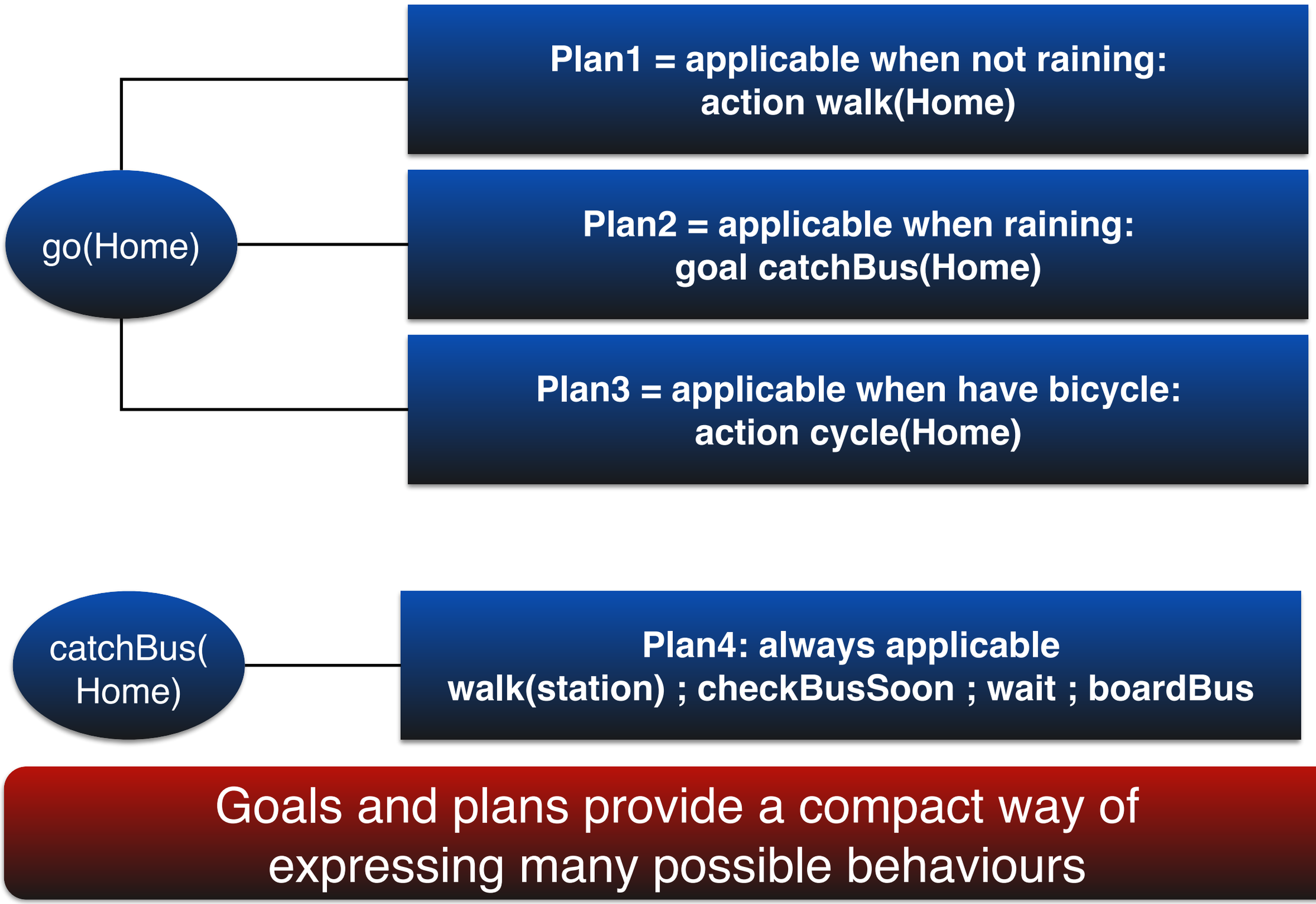
catchBus(Home)

Plan4: always applicable
walk(station) ; checkBusSoon ; wait ; boardBus

Example


go(Home)
Plan1 = applicable when not raining:
action walk(Home)
Plan2 = applicable when raining:
goal catchBus(Home)
Plan3 = applicable when have bicycle:
action cycle(Home)
catchBus(
Home)
Plan4: always applicable
walk(station) ; checkBusSoon ; wait ; boardBus

Example



Goals and plans provide a compact way of expressing many possible behaviours

"Based on this analysis, agent-oriented methodology, architecture, and development delivered a 368% improvement on overall project productivity. (2.11 FP Actual/0.45 FP Expected)"

"In a wide range of complex business applications, we show that the use of BDI [belief-desire-intention] technology incorporated within an enterprise-level architecture can improve overall developer productivity by an average [of] 350%. For java coding alone, the increase in productivity was over 500%" (emphasis added)

Steve S. Benfield, Jim Hendrickson, and Daniel Galanti. *Making a strong business case for multiagent technology*. In Peter Stone and Gerhard Weiss, editors, *Autonomous Agents and Multi-Agent Systems (AAMAS 2006)*, pages 10-15.

Industries include finance, insurance, logistics, energy, IT infrastructure.

Agentis

Some Applications of Human-Inspired Software ("Agents")

- Distribution
- Adaptability
- Flexibility
- Robustness
- Autonomy
- Complexity

Human-Inspired Metaphors for Software: What, Why and How?

- Software is everywhere
- Software is based on a metaphor
- The choice of metaphor matters
- Traditional “sequence of instructions” metaphor has limitations
- Other metaphors? Can draw inspiration from humans & human society!
- Key questions: what metaphors? how realise in software? how guide design?
- (unrelated) On technology and universities …

Human-Inspired Metaphors for Software: ~~What, Why and~~ How?

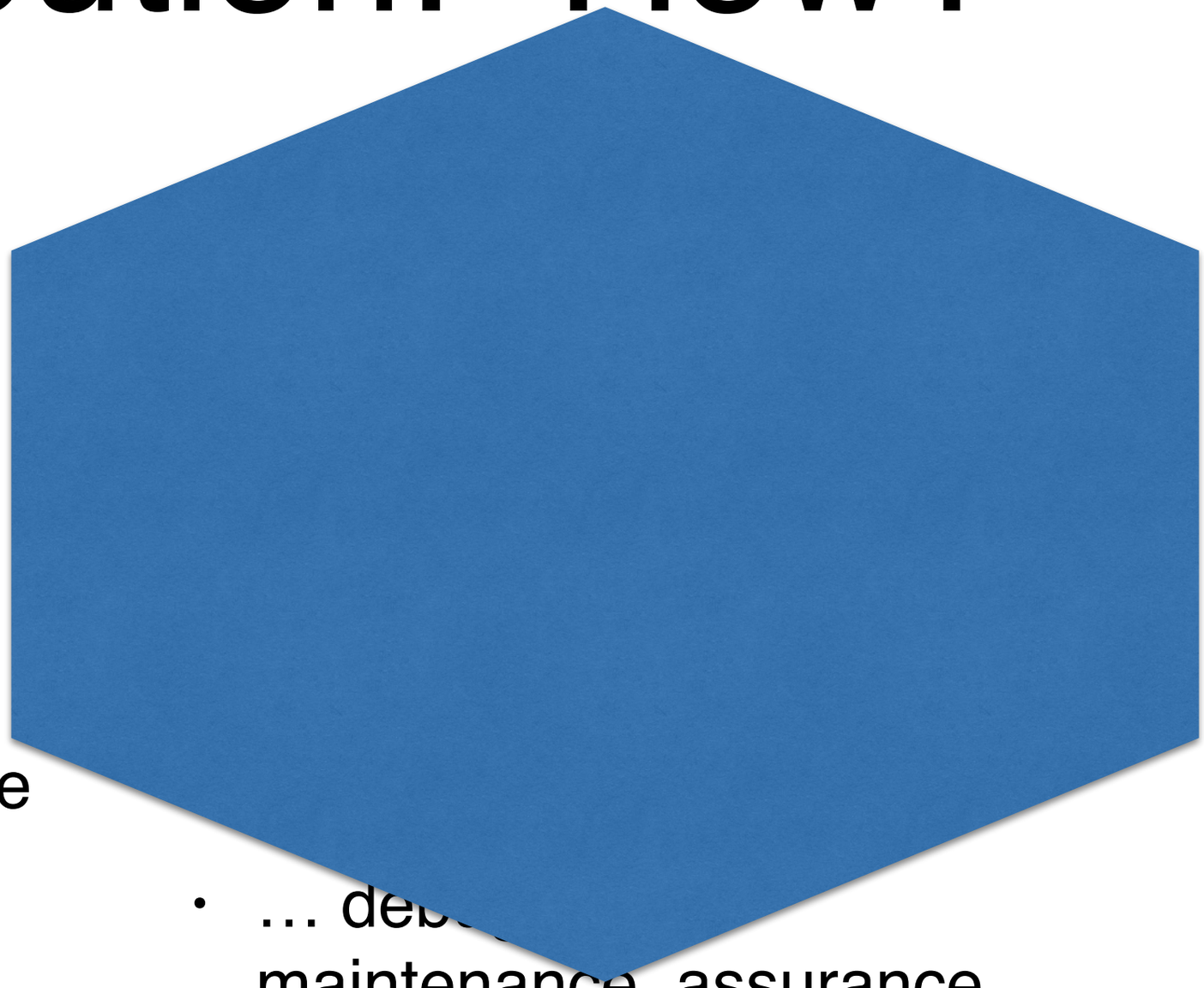
- ~~Software is everywhere~~
- ~~Software is based on a metaphor~~
- ~~The choice of metaphor matters~~
- ~~Traditional “sequence of instructions” metaphor has limitations~~
- ~~Other metaphors? Can draw inspiration from humans & human society!~~
- ~~Key questions: what metaphors?~~ how realise in software? how guide design?
- (unrelated) On technology and universities …

My contribution: “How?”

- **Concepts** - what is the metaphor?
- **Goals** - properties, life-cycle, behaviour, interaction …
- **Methodology**: how guide programmers to design software?
- Comparison of methodologies
- Designing flexible interactions
- Quantifying behaviour space size
- … debugging, maintenance, assurance …
- Industry applications

My contribution: “How?”

- **Concepts** - what is the metaphor?
- **Goals** - properties, life-cycle, behaviour, interaction …
- **Methodology**: how guide programmers to design software?



- … debugging, maintenance, assurance …
- Industry applications

$$\frac{B \models c}{?c \longrightarrow true}\ ?c_t \qquad \frac{B \not\models c}{?c \longrightarrow fail}\ ?c_f \qquad \frac{}{B, +b \longrightarrow B \cup \{b\}, true}\ +b \qquad \frac{}{B, -b \longrightarrow B \setminus \{b\}, true}\ -b$$

$$\frac{}{a \longrightarrow true}\ act \qquad \frac{B \models x \in X}{P \longrightarrow_X ex(x)}\ check$$

$$\frac{(t_i{:}c_i \leftarrow b_i{:}P_i) \in \Pi \quad B \models c_i}{!e \longrightarrow (\!|b_1{:}P_1, \ldots, b_n{:}P_n|\!)}\ Ev \qquad \frac{\neg \exists b_i{:}P_i \in \Delta . B \models b_i}{(\!|\Delta|\!) \longrightarrow fail}\ Sel_f \qquad \frac{b_i{:}P_i \in \Delta \quad B \models b_i}{(\!|\Delta|\!) \longrightarrow P_i \triangleright (\!|\Delta \setminus \{b_i{:}P_i\}|\!)}\ Sel$$

$$\frac{P_1 \longrightarrow P'}{P_1;P_2 \longrightarrow P';P_2}\ ; \qquad \frac{}{true;P \longrightarrow P}\ ;_t \qquad \frac{}{fail;P \longrightarrow fail}\ ;_f \qquad \frac{}{ex(X);P \longrightarrow ex(X)}\ ;_x$$

$$\frac{P_1 \longrightarrow P'}{P_1 \| P_2 \longrightarrow P' \| P_2}\ \|_1 \qquad \frac{P_2 \longrightarrow P'}{P_1 \| P_2 \longrightarrow P_1 \| P'}\ \|_2 \qquad \frac{}{true \| P \longrightarrow P}\ \|_{t1} \qquad \frac{}{P \| true \longrightarrow P}\ \|_{t2}$$

$$\frac{}{fail \| P \longrightarrow fail}\ \|_{f1} \qquad \frac{}{P \| fail \longrightarrow fail}\ \|_{f2} \qquad \frac{}{ex(X) \| P \longrightarrow ex(X)}\ \|_{x1} \qquad \frac{}{P \| ex(X) \longrightarrow ex(X)}\ \|_{x2}$$

$$\frac{P_1 \longrightarrow P'}{P_1 \triangleright P_2 \longrightarrow P' \triangleright P_2}\ \triangleright \qquad \frac{}{true \triangleright P \longrightarrow true}\ \triangleright_t \qquad \frac{}{fail \triangleright P \longrightarrow P}\ \triangleright_f \qquad \frac{}{ex(X) \triangleright P \longrightarrow ex(X)}\ \triangleright_x$$

$$\frac{G \cup \{\mathsf{Goal}(s,f)\}, \mathsf{Goal}_P(s,P,f) \longrightarrow G', P'}{G, \mathsf{Goal}(s,P,f) \longrightarrow G', P'}\ \mathsf{G}_{copy} \qquad \frac{P \longrightarrow_{X \cup \{s,f\}} P'}{\mathsf{Goal}_P(s,P,f) \longrightarrow_X \mathsf{Goal}_P(s,P',f)}\ \mathsf{G}$$

$$\frac{Q \in \{fail, true\}}{\mathsf{Goal}_P(s,Q,f) \longrightarrow \mathsf{Goal}_P(s,P,f)}\ \mathsf{G}_{tf} \qquad \frac{}{G, \mathsf{Goal}_P(s,ex(s),f) \longrightarrow G \setminus \{\mathsf{Goal}(s,f)\}, true}\ \mathsf{G}_{x1}$$

$$\frac{}{G, \mathsf{Goal}_P(s,ex(f),f) \longrightarrow G \setminus \{\mathsf{Goal}(s,f)\}, fail}\ \mathsf{G}_{x2} \qquad \frac{y \notin \{s,f\}}{G, \mathsf{Goal}_P(s,ex(y),f) \longrightarrow G \setminus \{\mathsf{Goal}(s,f)\}, ex(y)}\ \mathsf{G}_{x3}$$

$$\frac{P \in \Gamma \quad \langle B, G, P \rangle \longrightarrow \langle B', G', P' \rangle}{\langle N, B, G, \Gamma \rangle \longrightarrow \langle N, B', G', (\Gamma \setminus \{P\}) \cup \{P'\} \rangle}\ Agent$$

$$\frac{B \models c}{?c \longrightarrow true}\ ?c_t \qquad \frac{B \not\models c}{?c \longrightarrow fail}\ ?c_f \qquad \frac{}{B, +b \longrightarrow B \cup \{b\}, true}\ +b \qquad \frac{}{B, -b \longrightarrow B \setminus \{b\}, true}\ -b$$

$$\frac{}{a \longrightarrow true}\ act \qquad \frac{B \models x \in X}{P \longrightarrow_X ex(x)}\ check$$

$$\frac{(t_i{:}c_i \leftarrow b_i{:}P_i) \in \Pi \quad B \models c_i}{!e \longrightarrow (\!|b_1{:}P_1, \ldots, b_n{:}P_n|\!)}\ Ev \qquad \frac{\neg \exists b_i{:}P_i \in \Delta . B \models b_i}{(\!|\Delta|\!) \longrightarrow fail}\ Sel_f \qquad \frac{b_i{:}P_i \in \Delta \quad B \models b_i}{(\!|\Delta|\!) \longrightarrow P_i \triangleright (\!|\Delta \setminus \{b_i{:}P_i\}|\!)}\ Sel$$

$$\frac{P_1 \longrightarrow P'}{P_1;P_2 \longrightarrow P';P_2}\ ; \qquad \frac{}{true;P \longrightarrow P}\ ;_t \qquad \frac{}{fail;P \longrightarrow fail}\ ;_f \qquad \frac{}{ex(X);P \longrightarrow ex(X)}\ ;_x$$

$$\frac{P_1 \longrightarrow P'}{P_1 \| P_2 \longrightarrow P' \| P_2}\ \|_1 \qquad \frac{P_2 \longrightarrow P'}{P_1 \| P_2 \longrightarrow P_1 \| P'}\ \|_2 \qquad \frac{}{true \| P \longrightarrow P}\ \|_{t1} \qquad \frac{}{P \| true \longrightarrow P}\ \|_{t2}$$

$$\frac{}{fail \| P \longrightarrow fail}\ \|_{f1} \qquad \frac{}{P \| fail \longrightarrow fail}\ \|_{f2} \qquad \frac{}{ex(X) \| P \longrightarrow ex(X)}\ \|_{x1} \qquad \frac{}{P \| ex(X) \longrightarrow ex(X)}\ \|_{x2}$$

$$\frac{P_1 \longrightarrow P'}{P_1 \triangleright P_2 \longrightarrow P' \triangleright P_2}\ \triangleright \qquad \frac{}{true \triangleright P \longrightarrow true}\ \triangleright_t \qquad \frac{}{fail \triangleright P \longrightarrow P}\ \triangleright_f \qquad \frac{}{ex(X) \triangleright P \longrightarrow ex(X)}\ \triangleright_x$$

$$\frac{G \cup \{\mathsf{Goal}(s,f)\}, \mathsf{Goal}_P(s,P,f) \longrightarrow G', P'}{G, \mathsf{Goal}(s,P,f) \longrightarrow G', P'}\ \mathsf{G}_{copy} \qquad \frac{P \longrightarrow_{X \cup \{s,f\}} P'}{\mathsf{Goal}_P(s,P,f) \longrightarrow_X \mathsf{Goal}_P(s,P',f)}\ \mathsf{G}$$

$$\frac{Q \in \{fail, true\}}{\mathsf{Goal}_P(s,Q,f) \longrightarrow \mathsf{Goal}_P(s,P,f)}\ \mathsf{G}_{tf} \qquad \frac{}{G, \mathsf{Goal}_P(s,ex(s),f) \longrightarrow G \setminus \{\mathsf{Goal}(s,f)\}, true}\ \mathsf{G}_{x1}$$

$$\frac{}{G, \mathsf{Goal}_P(s,ex(f),f) \longrightarrow G \setminus \{\mathsf{Goal}(s,f)\}, fail}\ \mathsf{G}_{x2} \qquad \frac{y \notin \{s,f\}}{G, \mathsf{Goal}_P(s,ex(y),f) \longrightarrow G \setminus \{\mathsf{Goal}(s,f)\}, ex(y)}\ \mathsf{G}_{x3}$$

$$\frac{P \in \Gamma \quad \langle B, G, P \rangle \longrightarrow \langle B', G', P' \rangle}{\langle N, B, G, \Gamma \rangle \longrightarrow \langle N, B', G', (\Gamma \setminus \{P\}) \cup \{P'\} \rangle}\ Agent$$

$$\frac{B \models c}{?c \longrightarrow true}\ ?c_t \qquad \frac{B \not\models c}{?c \longrightarrow fail}\ ?c_f \qquad \frac{}{B, +b \longrightarrow B \cup \{b\}, true}\ +b \qquad \frac{}{B, -b \longrightarrow B \setminus \{b\}, true}\ -b$$

$$\frac{}{a \longrightarrow true}\ act \qquad \frac{B \models x \in X}{P \longrightarrow_X ex(x)}\ check$$

$$\frac{(t_i{:}c_i \leftarrow b_i{:}P_i) \in \Pi \quad B \models c_i}{!e \longrightarrow (\!|b_1{:}P_1, \ldots, b_n{:}P_n|\!)}\ Ev \qquad \frac{\neg \exists b_i{:}P_i \in \Delta . B \models b_i}{(\!|\Delta|\!) \longrightarrow fail}\ Sel_f \qquad \frac{b_i{:}P_i \in \Delta \quad B \models b_i}{(\!|\Delta|\!) \longrightarrow P_i \triangleright (\!|\Delta \setminus \{b_i{:}P_i\}|\!)}\ Sel$$

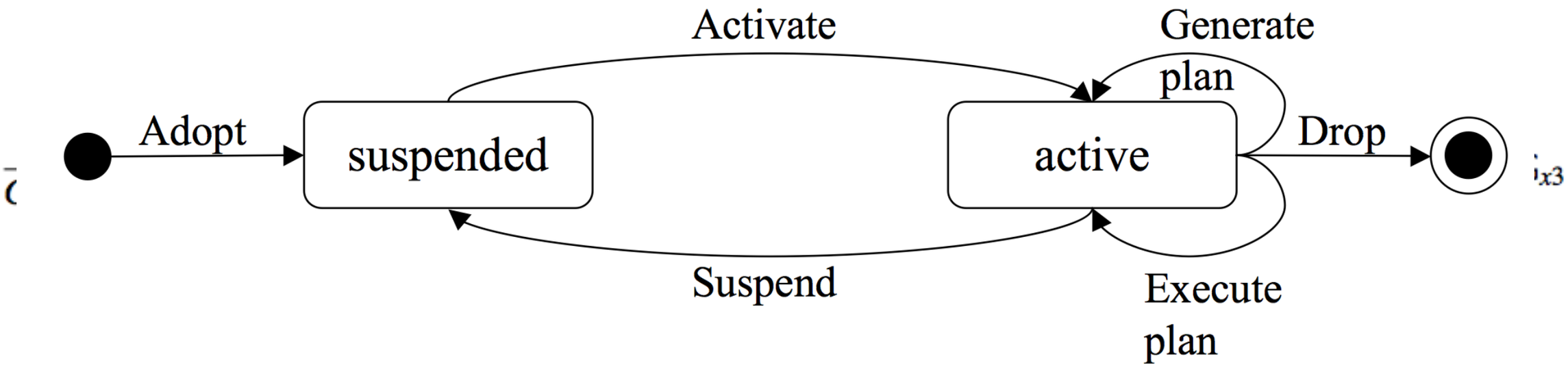
$$\frac{P_1 \longrightarrow P'}{P_1;P_2 \longrightarrow P';P_2}\ ; \qquad \frac{}{true;P \longrightarrow P}\ ;_t \qquad \frac{}{fail;P \longrightarrow fail}\ ;_f \qquad \frac{}{ex(X);P \longrightarrow ex(X)}\ ;_x$$

$$\frac{P_1 \longrightarrow P'}{P_1 \| P_2 \longrightarrow P' \| P_2}\ \|_1 \qquad \frac{P_2 \longrightarrow P'}{P_1 \| P_2 \longrightarrow P_1 \| P'}\ \|_2 \qquad \frac{}{true \| P \longrightarrow P}\ \|_{t1} \qquad \frac{}{P \| true \longrightarrow P}\ \|_{t2}$$

$$\frac{}{fail \| P \longrightarrow fail}\ \|_{f1} \qquad \frac{}{P \| fail \longrightarrow fail}\ \|_{f2} \qquad \frac{}{ex(X) \| P \longrightarrow ex(X)}\ \|_{x1} \qquad \frac{}{P \| ex(X) \longrightarrow ex(X)}\ \|_{x2}$$

$$\frac{P_1 \longrightarrow P'}{P_1 \triangleright P_2 \longrightarrow P' \triangleright P_2}\ \triangleright \qquad \frac{}{true \triangleright P \longrightarrow true}\ \triangleright_t \qquad \frac{}{fail \triangleright P \longrightarrow P}\ \triangleright_f \qquad \frac{}{ex(X) \triangleright P \longrightarrow ex(X)}\ \triangleright_x$$

$$\frac{G \cup \{\mathsf{Goal}(s,f)\}, \mathsf{Goal}_P(s,P,f) \longrightarrow G', P'}{\quad}\ \mathsf{G}_{copy} \qquad \frac{P \longrightarrow_{X \cup \{s,f\}} P'}{\quad}\ \mathsf{G}$$



The *Prometheus* Methodology

- **Methodology = Concepts + <u>Process</u> + Models/Notations + Techniques + Tool Support**

System specification

Architectural design

Detailed design

Scenarios

System goals

Initial Role descriptors

Actions, percepts

agent grouping

agent acquaintance

Interaction diagrams

messages

shared data

Protocols

System Overview

Agent descriptors

Process

Agent Overview

Capability descriptors

Capability overview

Event descriptors

Data descriptions

Plan descriptors

Key

final design artifact

intermediate design tool

crosscheck

derives

Dynamic

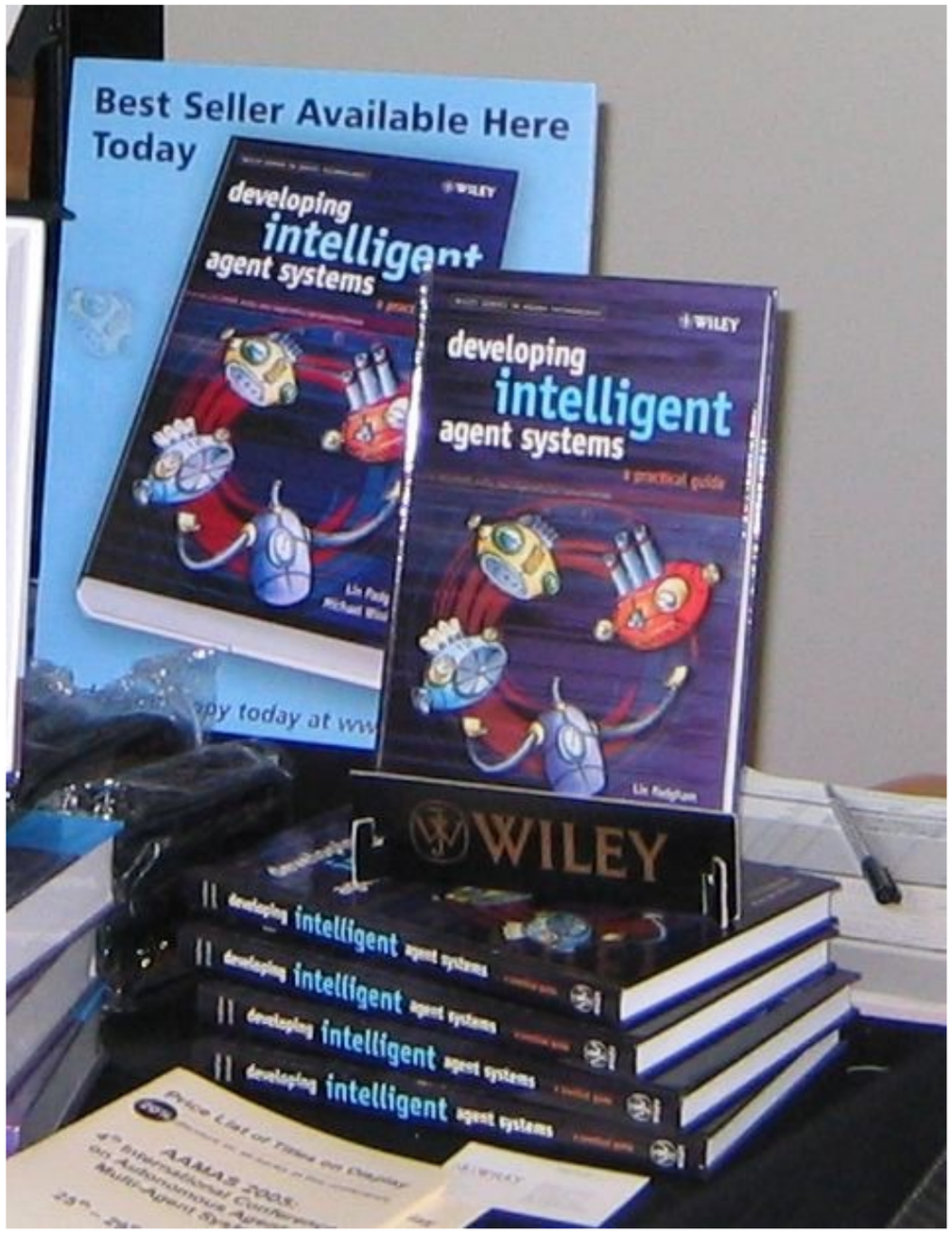
Overview

Detailed Form

Analogy: storyboarding

The *Prometheus* Methodology

- **Methodology = Concepts + Process + Models/Notations + Techniques + Tool Support**
- Detailed process and products
- Evolved out of practical industrial and pedagogic experience and refined
- Hierarchical structuring: scales to large designs



Prometheus Design Tool : 2.0

File Tools Scoping Entities View Help

Diagrams

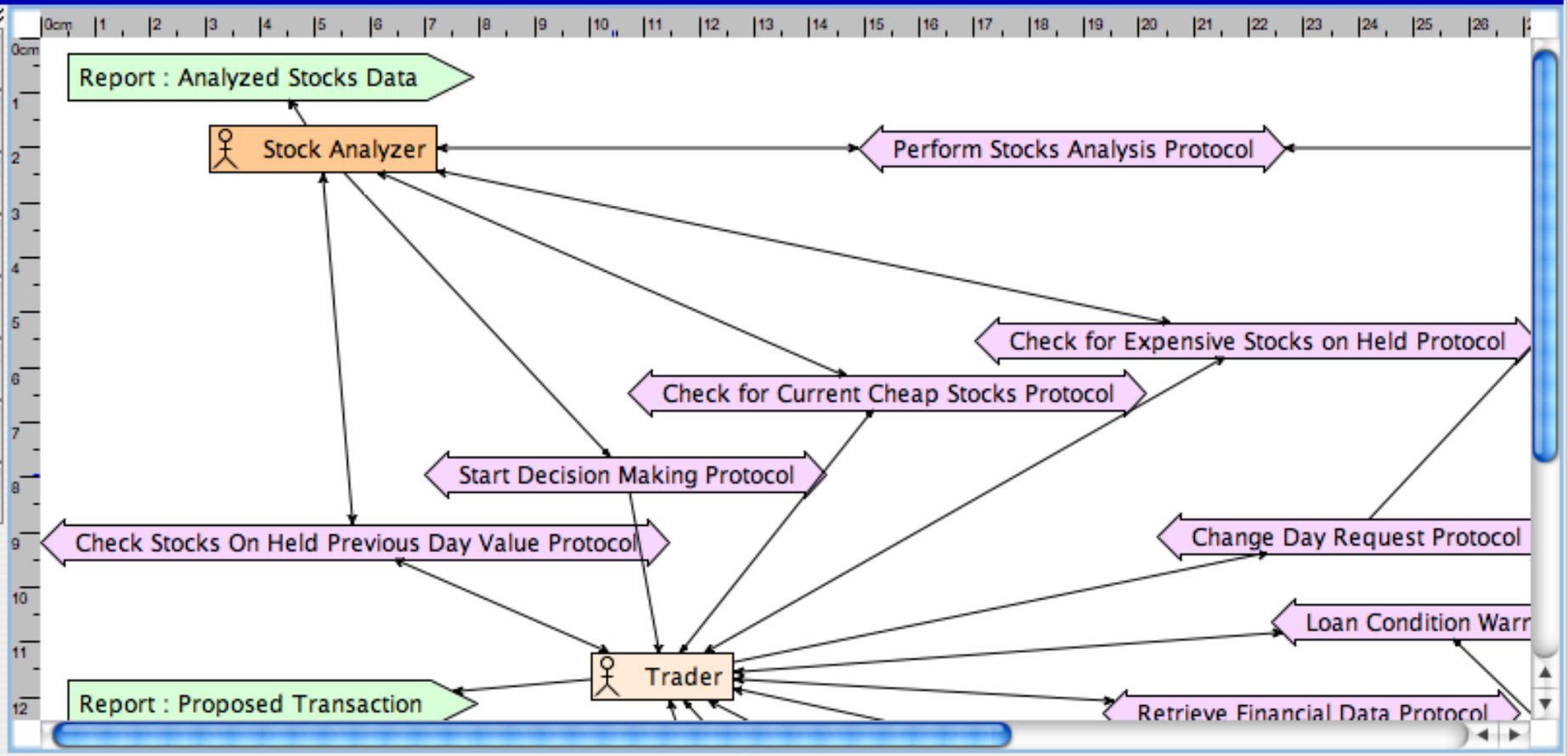
- **Stock Simulator**
  - **System Specification**
    - Stakeholders
    - Scenarios
    - Goal Overview
    - Functionalities
    - System Interface
  - **Architectural Design**
    - Data Coupling
    - Agent Acquaintance
    - System Overview
  - **Detailed Design**
    - Stock Market
    - Stock Analyzer
    - Simulator
    - Trader

Entities

System entities | Filter

- Stock Analyzer
- Stock Market
- Trader
- Change Day Capability
- Determine Stocks To Buy Capability
- Determine Stocks To Sell Capability
- Handle Loan Warning Capability
- Process Borrow Loan Transaction Cap
- Process Buy Transaction Capability
- Process Sell Transaction Capability
- Provide Analyzed Stocks Data Capabi
- Provide Assets Data

System Overview Diagram



Stock Analyzer – Descriptor

Name: Stock Analyzer

Description:

Performs all types of stock analysis and history keeping, such as determining 'cheap' and 'expensive' stocks using their price hist

Cardinality minimum: 1

Cardinality maximum: 1

Delete Set

My contribution: “How?”

- **Concepts** - what is the metaphor?
- **Goals** - properties, life-cycle, behaviour, interaction …
- **Methodology**: how guide programmers to design software?
- Comparison of methodologies
- Designing flexible interactions
- Quantifying behaviour space size
- … debugging, maintenance, assurance …
- Industry applications

Comparing Methodologies

- Lots of methodologies - compare them!
- Method: develop set of criteria, and assess each methodology
- But very subjective! … and not well informed?
- Solution: student projects + survey each methodology's creators

Concepts & Properties	MaSE	Prometheus	Tropos
Autonomy	H/M/DK	H/NA/H	H/M/M
Mental attitudes	L/M/H	H/M/H	H
Proactive	H/M/H	H/M/DK	H
Reactive	M	H/M/DK	H/L/DK
Concurrency	H/M/H	M/L/DK	H/M/H
Teamwork	H/M/H	N/L/NA	H/H/M
Protocols	H	M/H/M	NA/M/M
Situated	M/L/H	H	H
Clear concepts	SA/A/A	A/A/DA	SA/A/N
Concepts overloaded	A/N/SA	N	SDA/N/DA
Agent-oriented	SA/A/A	SA	SA/A/SA
Modelling & Notation			
Static+Dynamic	SA/A/A	SA/A/A	N/A/A
Syntax defined	A/A/SA	SA/A/A	SA/N/A
Semantics defined	A/SA/SA	A	SA/A/A
Clear notation	A	SA/A/A	SA/A/N
Easy to use	SA/A/A	A/N/A	SA/A/N
Easy to learn	N/N/A	SA/NA/SA	SA/N/A
Different views	N/N/A	A/A/SA	SA/A/N
Language adequate & expressive	SA/N/N	A	SA/A/N
Traceability	A/SA/SA	A	A/N/A
Consistency check	SA/A/SA	SA/A/A	-/A/DA
Refinement	SA/A/A	SA	SA/A/DA
Modularity	SA/A/A	SA/SA/A	SA/A/N
Reuse	N/SA/A	N/A/N	-/A/DA
Hierarchical modelling	N/A/A	SA/A/A	SA/A/DA
Process			
Requirements	SPEH	SPEH	SPE
Architectural design	SPEH	SPEH	SPE
Detailed design	SPEH	SPEH	SPE
Implementation	SEH/SPE/S	SPEH/S/n	SE/SPE/SPEH
Testing & Debugging	SPE/n/n	SPEH/S/n	n
Deployment	SE/SPE/SPEH	n	n
Maintenance	n/SPE/n	n	n
Pragmatics			
Quality	N/DA/A	A/N/N	DA/A/-
Cost estimation	-/DA/SA	DA/DA/N	DA/N/-
Management decision	-/DA/SA	SDA/N/-	SA/A/-
apps	21+	6-20	1-5
Real apps	no	no	no
Used by non-creators	yes	yes	yes/no/no
Domain specific	no	no	yes/no/no
Scalable	-/N/N	N/A/N	N/N/-
Distributed	-/SA/SA	SA/A/N	N/A/-

Designing Flexible Interactions

	Prometheus								Hermes						
Participant	01	04	06	12	13	14	15	Σ	02	03	05	07	08	10	Σ
Step															
1	✓	✓	✓	✓	✓	✓	✓	7	✓	✓	✓	✓	✓	✓	6
2	✓	✓	✓	✓	✓	✓	✓	7	✓	✓	✓	✓	✓	✓	6
3	✓	✓	✓	✓	✓	✓	✓	7	✓	✓	✓	✓	✓	✓	6
4	✓	✓	✗	✓	✗	✓	✓	5	✓	✓	✓	✓	✓	✓	6
5	✓	✓	✓	✓	✓	✓	✓	7	✓	✓	✓	✓	✓	✓	6
6	✓	✓	✗	✓	✗	✓	✓	5	✓	✓	✓	✓	✓	✓	6
7	✓	✓	✗	✓	✓	✓	✓	6	✓	✓	✓	✓	✓	✓	6
8	✗	✓	✓	✓	✓	✓	✓	6	✓	✓	✓	✓	✓	✓	6
9	✓	✓	✓	✓	✓	✓	✓	7	✓	✓	✓	✓	✓	✓	6
10	✓	✓	✓	✓	✓	✓	✓	7	✓	✓	✓	✓	✓	✓	6
11	✓	✓	✓	✓	✓	✗	✓	6	✓	✓	✓	✓	✓	✓	6
12	✓	✓	✓	✓	✗	✗	✓	5	✓	✓	✓	✓	✓	✓	6
13	✓	✓	✓	✓	✓	✗	✓	6	✓	✓	✓	✓	✓	✓	6
14	✓	✓	✗	✓	✓	✗	✓	5	✓	✓	✓	✓	✓	✓	6
Total	13	14	10	14	11	10	14	–	14	14	14	14	14	14	–

Christopher Cheong and Michael Winikoff. **A Comparison of Two Agent Interaction Design Approaches**. Multiagent and Grid Systems (an international journal), volume 9, pages 1-44, 2013.

$$
\begin{aligned}
F^{\checkmark}_{\infty}(t,x,y,b,\lambda) &= 0 \text{ if } \lambda \leq 0 \\
F^{\times}_{\infty}(t,x,y,b,\lambda) &= 0 \text{ if } \lambda \leq 0 \\
F^{\checkmark}_{\infty}(\beta,x,y,b,\lambda) &= F^{\checkmark}_{\infty}(b[\beta],x,y,b,\lambda) \\
F^{\times}_{\infty}(\beta,x,y,b,\lambda) &= F^{\times}_{\infty}(b[\beta],x,y,b,\lambda) \\
F^{\checkmark}_{\infty}(a,x,y,b,\lambda) &= x \\
F^{\times}_{\infty}(a,x,y,b,\lambda) &= xy
\end{aligned}
$$

$$
F^{\checkmark}_{\infty}(s_1;s_2,x,y,b,\lambda)
$$

$$
= \begin{cases} F^{\checkmark}_{\infty}(s_1,x,y,b,1)\, F^{\checkmark}_{\infty}(s_2,x,y,b,\lambda-1) & \text{if } s_1 \text{ is an action} \\ F^{\checkmark}_{\infty}(s_1,x,y,b,\lambda) \overset{x\leq\lambda}{\times} F^{\checkmark}_{\infty}(s_2,x,y,b,\lambda) & \text{otherwise} \end{cases}
$$

$$
F^{\times}_{\infty}(s_1;s_2,x,y,b,\lambda)
$$

$$
= \begin{cases} F^{\times}_{\infty}(s_1,x,y,b,1)+F^{\checkmark}_{\infty}(s_1,x,y,b,1)\, F^{\times}_{\infty}(s_2,x,y,b,\lambda-1) & \text{if } s_1 \text{ is an action} \\ F^{\times}_{\infty}(s_1,x,y,b,\lambda)+F^{\checkmark}_{\infty}(s_1,x,y,b,\lambda) \overset{x\leq\lambda}{\times} F^{\times}_{\infty}(s_2,x,y,b,\lambda) & \text{otherwise} \end{cases}
$$

$$
F^{\checkmark}_{\infty}(goal(PM),x,y,b,\lambda)
$$

$$
= \sum_{p\in set(PM)} \chi_{PM}(p) F^{\checkmark}_{\infty}(p,x,y,b,\lambda) \overset{x\leq\lambda}{\times} F^{\times}_{\infty}(PM-\{p{:}1\},x,y,b,\lambda)
$$

$$
F^{\times}_{\infty}(goal(PM),x,y,b,\lambda) = |PM|! \frac{G^{\times}_{\infty}(PM,x,y,z,b,\lambda) \restriction_{power(z)=|PM|}}{z^{|PM|}}
$$

$$
\begin{aligned}
F^{\checkmark}_{\infty}(t,x,y,b,\lambda) &= 0 \text{ if } \lambda \leq 0 \\
F^{\times}_{\infty}(t,x,y,b,\lambda) &= 0 \text{ if } \lambda \leq 0 \\
F^{\checkmark}_{\infty}(\beta,x,y,b,\lambda) &= F^{\checkmark}_{\infty}(b[\beta],x,y,b,\lambda) \\
F^{\times}_{\infty}(\beta,x,y,b,\lambda) &= F^{\times}_{\infty}(b[\beta],x,y,b,\lambda) \\
F^{\checkmark}_{\infty}(a,x,y,b,\lambda) &= x \\
F^{\times}_{\infty}(a,x,y,b,\lambda) &= xy
\end{aligned}
$$

$$
F^{\checkmark}_{\infty}(s_1;s_2,x,y,b,\lambda)
$$

$$
= \begin{cases} F^{\checkmark}_{\infty}(s_1,x,y,b,1)\, F^{\checkmark}_{\infty}(s_2,x,y,b,\lambda-1) & \text{if } s_1 \text{ is an action} \\ F^{\checkmark}_{\infty}(s_1,x,y,b,\lambda) \overset{x\leq\lambda}{\times} F^{\checkmark}_{\infty}(s_2,x,y,b,\lambda) & \text{otherwise} \end{cases}
$$

$$
F^{\times}_{\infty}(s_1;s_2,x,y,b,\lambda)
$$

$$
= \begin{cases} F^{\times}_{\infty}(s_1,x,y,b,1)+F^{\checkmark}_{\infty}(s_1,x,y,b,1)\, F^{\times}_{\infty}(s_2,x,y,b,\lambda-1) & \text{if } s_1 \text{ is an action} \\ F^{\times}_{\infty}(s_1,x,y,b,\lambda)+F^{\checkmark}_{\infty}(s_1,x,y,b,\lambda) \overset{x\leq\lambda}{\times} F^{\times}_{\infty}(s_2,x,y,b,\lambda) & \text{otherwise} \end{cases}
$$

$$
F^{\checkmark}_{\infty}(goal(PM),x,y,b,\lambda)
$$

$$
= \sum_{p\in set(PM)} \chi_{PM}(p) F^{\checkmark}_{\infty}(p,x,y,b,\lambda) \overset{x\leq\lambda}{\times} F^{\times}_{\infty}(PM-\{p{:}1\},x,y,b,\lambda)
$$

$$
F^{\times}_{\infty}(goal(PM),x,y,b,\lambda) = |PM|! \frac{G^{\times}_{\infty}(PM,x,y,z,b,\lambda) \restriction_{power(z)=|PM|}}{z^{|PM|}}
$$

$$
\begin{aligned}
F^{\checkmark}_{\infty}(t,x,y,b,\lambda) &= 0 \text{ if } \lambda \leq 0 \\
F^{\times}_{\infty}(t,x,y,b,\lambda) &= 0 \text{ if } \lambda \leq 0 \\
F^{\checkmark}_{\infty}(\beta,x,y,b,\lambda) &= F^{\checkmark}_{\infty}(b[\beta],x,y,b,\lambda) \\
F^{\times}_{\infty}(\beta,x,y,b,\lambda) &= F^{\times}_{\infty}(b[\beta],x,y,b,\lambda) \\
F^{\checkmark}_{\infty}(a,x,y,b,\lambda) &= x \\
F^{\times}_{\infty}(a,x,y,b,\lambda) &= xy
\end{aligned}
$$

$$
F^{\checkmark}_{\infty}(s_1;s_2,x,y,b,\lambda)
$$

$$
= \begin{cases} F^{\checkmark}_{\infty}(s_1,x,y,b,1)\, F^{\checkmark}_{\infty}(s_2,x,y,b,\lambda-1) & \text{if } s_1 \text{ is an action} \\ F^{\checkmark}_{\infty}(s_1,x,y,b,\lambda) \overset{x\leq\lambda}{\times} F^{\checkmark}_{\infty}(s_2,x,y,b,\lambda) & \text{otherwise} \end{cases}
$$

$$
F^{\times}_{\infty}(s_1;s_2,x,y,b,\lambda)
$$

$$
= \begin{cases} F^{\times}_{\infty}(s_1,x,y,b,1)+F^{\checkmark}_{\infty}(s_1,x,y,b,1)\, F^{\times}_{\infty}(s_2,x,y,b,\lambda-1) & \text{if } s_1 \text{ is an action} \\ F^{\times}_{\infty}(s_1,x,y,b,\lambda)+F^{\checkmark}_{\infty}(s_1,x,y,b,\lambda) \overset{x\leq\lambda}{\times} F^{\times}_{\infty}(s_2,x,y,b,\lambda) & \text{otherwise} \end{cases}
$$

$$
F^{\checkmark}_{\infty}(goal(PM),x,y,b,\lambda)
$$

$$
= \sum_{p\in set(PM)} \chi_{PM}(p) F^{\checkmark}_{\infty}(p,x,y,b,\lambda) \overset{x\leq\lambda}{\times} F^{\times}_{\infty}(PM-\{p{:}1\},x,y,b,\lambda)
$$

$$
F^{\times}_{\infty}(goal(PM),x,y,b,\lambda) = |PM|! \frac{G^{\times}_{\infty}(PM,x,y,z,b,\lambda) \restriction_{power(z)=|PM|}}{z^{|PM|}}
$$

$$
\begin{aligned}
F^{\checkmark}_{\infty}(t,x,y,b,\lambda) &= 0 \text{ if } \lambda \leq 0 \\
F^{\times}_{\infty}(t,x,y,b,\lambda) &= 0 \text{ if } \lambda \leq 0 \\
F^{\checkmark}_{\infty}(\beta,x,y,b,\lambda) &= F^{\checkmark}_{\infty}(b[\beta],x,y,b,\lambda) \\
F^{\times}_{\infty}(\beta,x,y,b,\lambda) &= F^{\times}_{\infty}(b[\beta],x,y,b,\lambda) \\
F^{\checkmark}_{\infty}(a,x,y,b,\lambda) &= x \\
F^{\times}_{\infty}(a,x,y,b,\lambda) &= xy
\end{aligned}
$$

$$
F^{\checkmark}_{\infty}(s_1;s_2,x,y,b,\lambda)
$$

$$
= \begin{cases} F^{\checkmark}_{\infty}(s_1,x,y,b,1)\, F^{\checkmark}_{\infty}(s_2,x,y,b,\lambda-1) & \text{if } s_1 \text{ is an action} \\ F^{\checkmark}_{\infty}(s_1,x,y,b,\lambda) \overset{x\leq\lambda}{\times} F^{\checkmark}_{\infty}(s_2,x,y,b,\lambda) & \text{otherwise} \end{cases}
$$

$$
F^{\times}_{\infty}(s_1;s_2,x,y,b,\lambda)
$$

$$
= \begin{cases} F & \text{n action} \\ F & \text{e} \end{cases}
$$

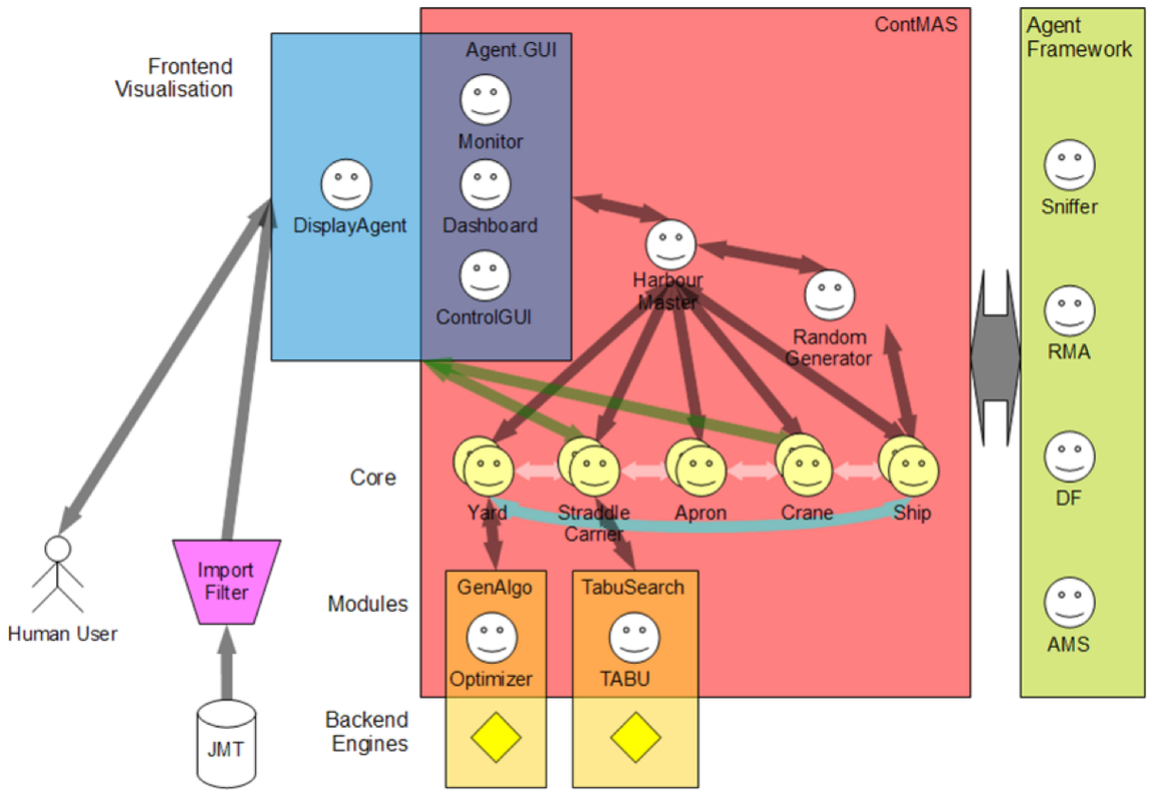
Parameters			Number of				
j	k	d	goals	plans	actions	$n^{\checkmark}(g)$	$n^{\times}(g)$
2	2	3	21	42	62 (13)	$\approx 6.33 \times 10^{12}$	$\approx 1.82 \times 10^{13}$
3	3	3	91	273	363 (25)	$\approx 1.02 \times 10^{107}$	$\approx 2.56 \times 10^{107}$
2	3	4	259	518	776 (79)	$\approx 1.82 \times 10^{157}$	$\approx 7.23 \times 10^{157}$
3	4	3	157	471	627 (41)	$\approx 3.13 \times 10^{184}$	$\approx 7.82 \times 10^{184}$

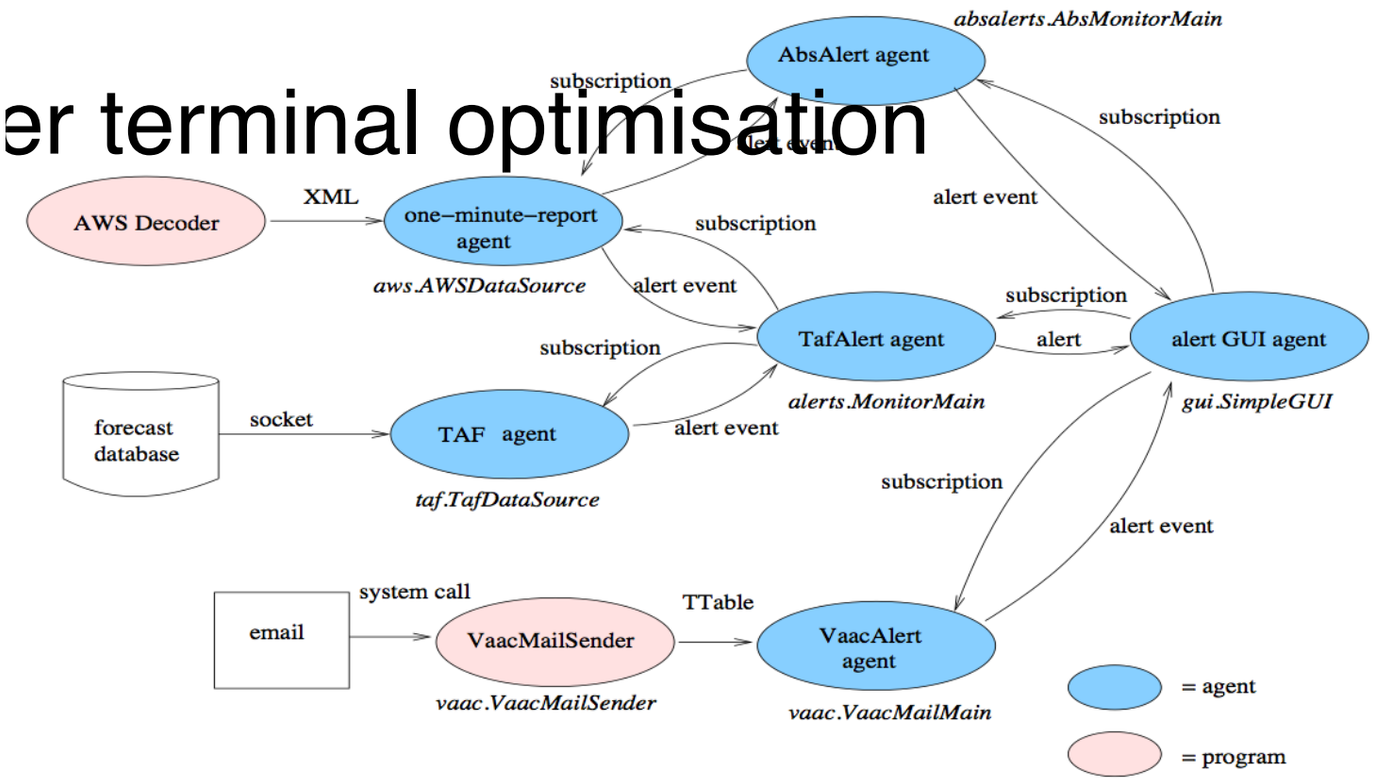
$$
F^{\checkmark}_{\infty}(goal(PM),x,y,b,\lambda) = \quad = \prod_{p \in set(PM)}
$$

$$
F^{\times}_{\infty}(goal(PM),x,y,b,\lambda) = |PM|! \frac{G^{\times}_{\infty}(PM,x,y,z,b,\lambda) \restriction_{power(z)=|PM|}}{z^{|PM|}}
$$

Industry Applications

- Agent-Oriented Software: purveyors of JACK™ agent-oriented platform - work on *Prometheus*
- Bureau of Meteorology: scalable weather alert system
- Patrick container terminal optimisation





Human-Inspired Metaphors for Software: ~~What, Why and~~ How?

- ~~Software is everywhere~~
- ~~Software is based on a metaphor~~
- ~~The choice of metaphor matters~~
- ~~Traditional “sequence of instructions” metaphor has limitations~~
- ~~Other metaphors? Can draw inspiration from humans & human society!~~
- ~~Key questions: what metaphors?~~ how realise in software? how guide design?
- (unrelated) On technology and universities …

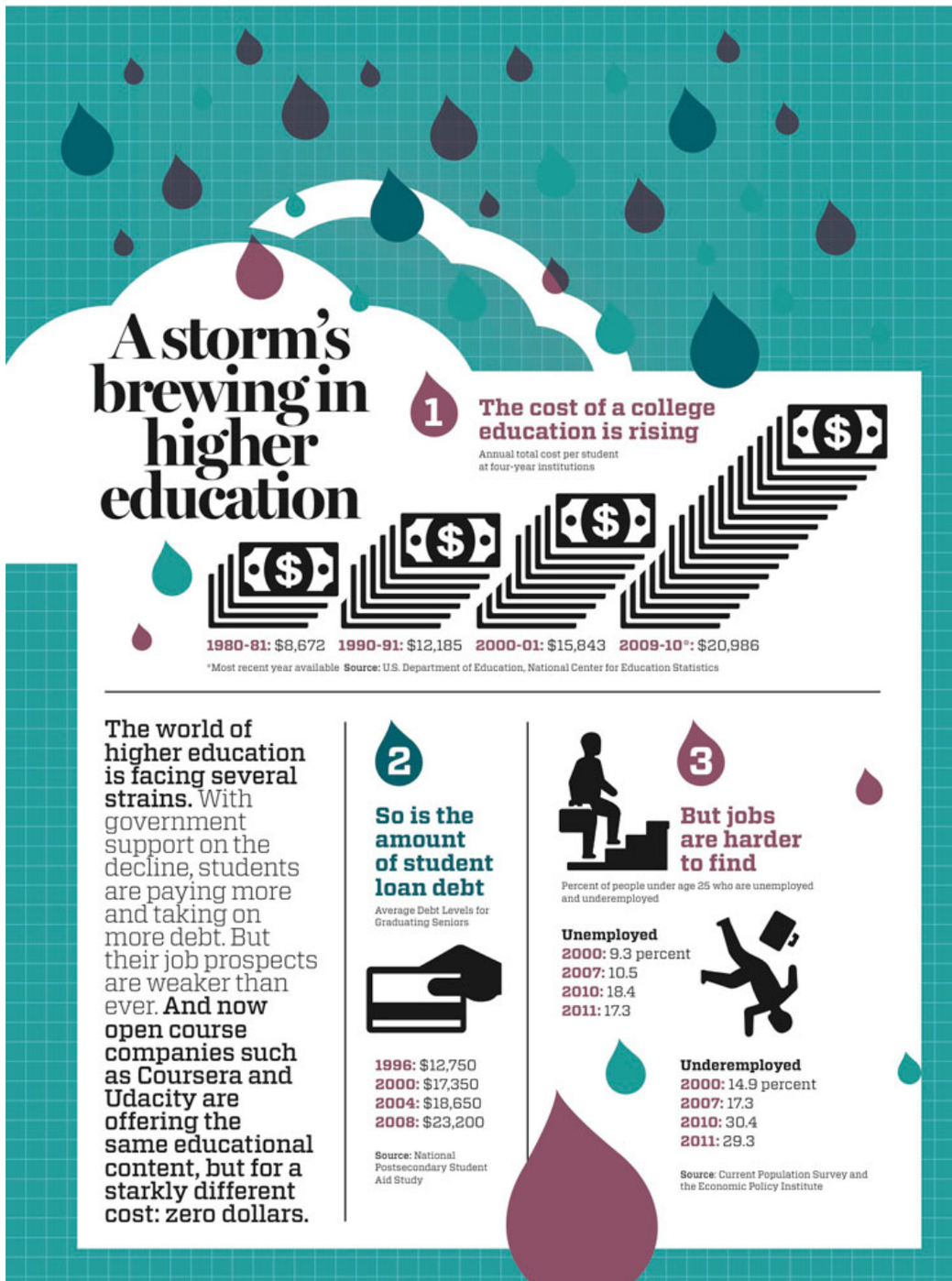
Human-Inspired Metaphors for Software: ~~What, Why and How?~~

- ~~Software is everywhere~~
- ~~Software is based on a metaphor~~
- ~~The choice of metaphor matters~~
- ~~Traditional “sequence of instructions” metaphor has limitations~~
- ~~Other metaphors? Can draw inspiration from humans & human society!~~
- ~~Key questions: what metaphors? how realise in software? how guide design?~~
- (unrelated) On technology and universities …

Technology, Universities and Disruption: some questions …

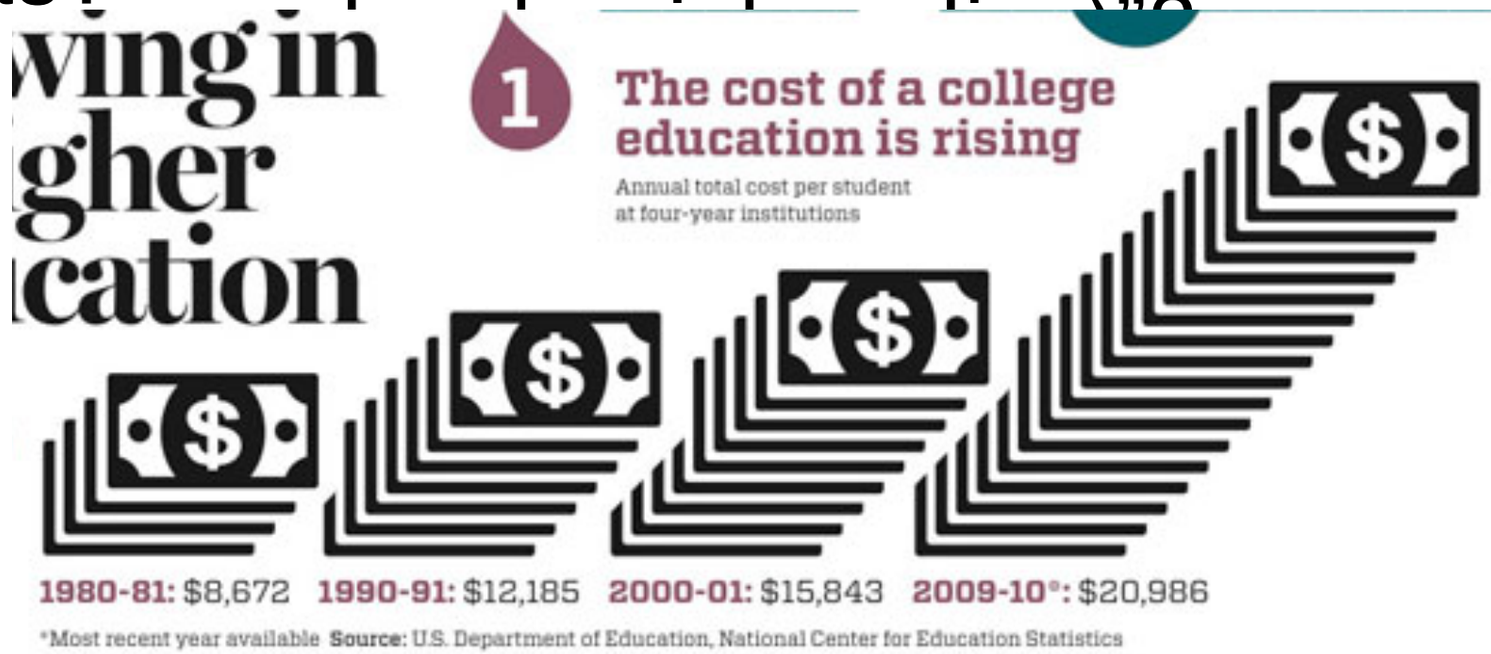
"*There's a tsunami coming.*"

"Just as technology disrupted and transformed the newspaper and music industries, **it is now poised to wreak havoc** upon another established industry: **higher education**."
http://bit.ly/MDLiJv
(cf in http://nyti.ms/II056d)



- Is is relevant to NZ, or just US?
- Is it about today's universities?
- Does it overly focus on lecturing?
- Does the proposal generalise to all students?
- ... all disciplines?
- Is the proposal financially sustainable?
- Is it just a library?
- Does it confuse "online learning (**with** teacher interaction)" with "online materials (**without** teacher interaction)"?

- Is is relevant to NZ, or just US?
- Is it about today's universities?
- Does it overly focus on lecturing?
- Does the proposal generalise to all students?
- ... all disciplines?
- Is the proposal financially sustainable?
- Is it just a library?
- Does it confuse "online learning (**with** teacher interaction)" with "online materials (**without**



- Is is relevant to NZ, or just US?
- Is it about today's universities?
- Does it overly focus on lecturing?
- Does the proposal generalise to all students?
- ... all disciplines?
- Is the proposal financially sustainable?
- Is it just a library?
- Does it confuse "online learning (**with** teacher interaction)" with "online materials (**without** teacher interaction)"?

- Is is relevant to NZ, or just US?
- Is it about today's universities?
- Does it overly focus on lecturing?
- Does the proposal generalise to all students?
- ... all disciplines?

- Is the proposal financially sustainable?
- Is it just a library?
- Does it confuse "online learning (**with** teacher interaction)" with "online materials (**without** teacher interaction)"?

MOOCs will revolutionise education because ……. free

- Is is relevant to NZ, or just US?
- Is it about today's universities?
- Does it overly focus on lecturing?
- Does the proposal generalise to all students?
- ... all disciplines?

- Is the proposal financially sustainable?
- Is it just a library?
- Does it confuse "online learning (**with** teacher interaction)" with "online materials (**without** teacher interaction)"?

Walking will revolutionise Transport because ……. free

Thank you!

(*) Not my students

Special Mentions

James Harland and Harald Søndergaard

Lin Padgham

My family

Special Mentions

James Harland and Harald Søndergaard

Lin Padgham

My family

Advertisement: Upcoming IPL by Professor Stephen Cranefield