Propositional Planning in BDI Agents

Felipe Rech Meneguzzi
Avelino Francisco Zorzo
Michael da Costa Móra

PUCRS – Porto Alegre – Brazil

Agenda

- Motivation
- Objectives
- X-BDI
- Prototype
- Experiments

Motivation

- Agent Design Problem
  - Given an environment and a set of goals, determine if an agent is capable of accomplishing them
- BDI Model
  - Mental States describe behaviour
  - One of the most widely studied model
- Means-ends reasoning
  - Plan library
  - Planning at runtime

Motivation

- Propositional Planning
  - PSPACE Complexity for the general case
  - Advances in algorithms (e.g. Graph-based)
- Verify the possibility to map BDI means-end reasoning into a propositional planning problem

Objectives

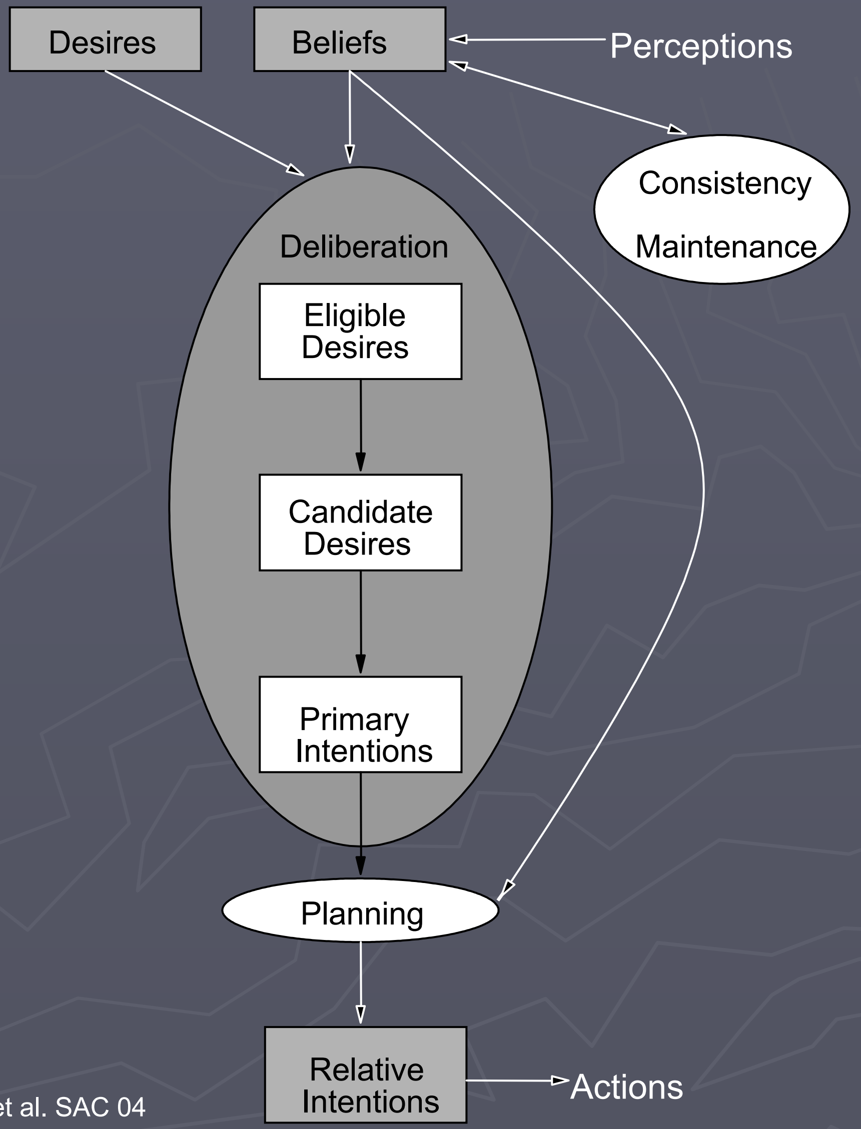
- Mapping BDI Model → Planning
- Modify BDI tools so as to use external planning algorithms
  - Include a propositional planning algorithm within one such tool
  - Verify the results of the interaction between the BDI tool and the planning algorithm

Related Work

X-BDI

X-BDI

- Described using Extended Logic Programming (ELP)
- Agent Description is directly executable
- ELP provides non-monotonic reasoning mechanisms



X-BDI – Focal Points

- Selection of Candidate Desires
  - The necessary actions in order to satisfy a desire are determined through abductive reasoning
  - This process is integrated to planning
- Planning
  - Selection of Relative Intentions
  - Abductive Planning, PSPACE

Modifications to X-BDI

Modifications to X-BDI

- Replacement of the abduction process for an external planner
- Mapping Processes
  - Mental States → Planning Problem
  - Plan → Mental States

Modifications to X-BDI

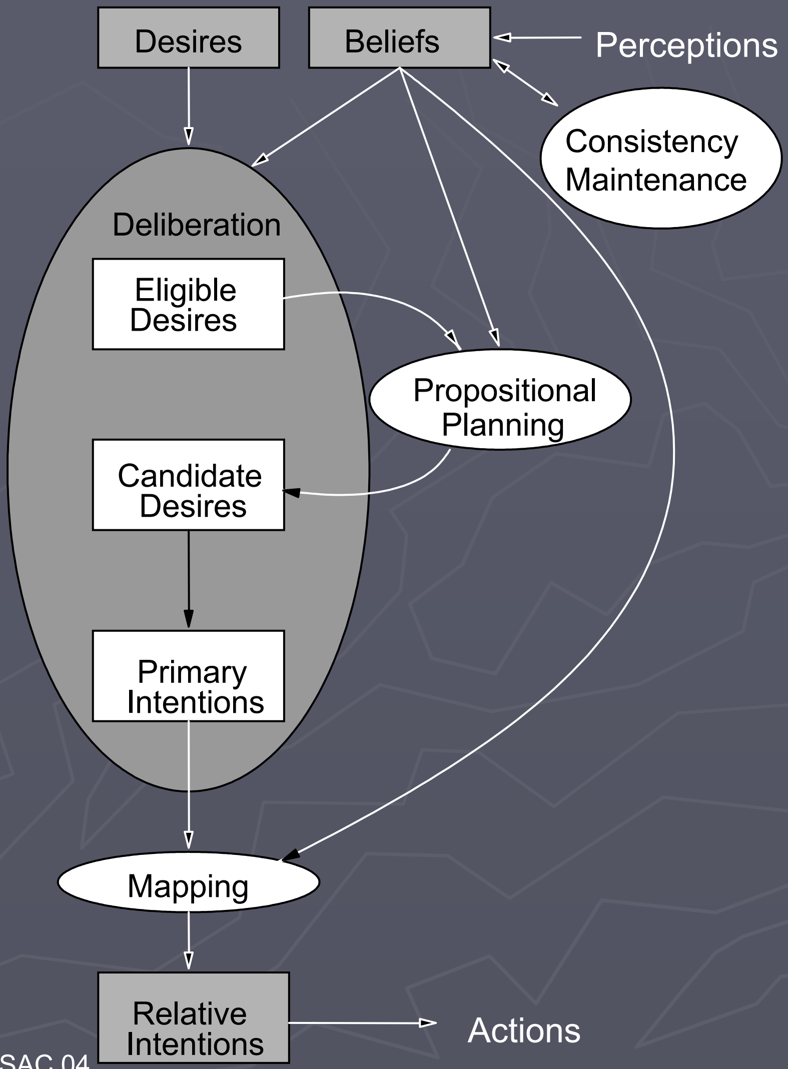
- Candidate Desires
  - Possibility of is verified by a planning function
  - $Plan(\Pi)$
    - *Δ iff exists an Δ such that it is a solution to Π*
    - *{ } otherwise*
  - Any planning function satisfying this definition can be used by the agent
- Relative Intentions
  - Are generated based on the plan calculated by the planning function

Mapping Process

- Beliefs and Eligible Desires are used to generate a propositional planning problem
    - Beliefs → Start State
    - Actions → Operators
    - Eligible Desires → Goal State
- Subsets of the Eligible Desires are sent to the planner

Mapping Process

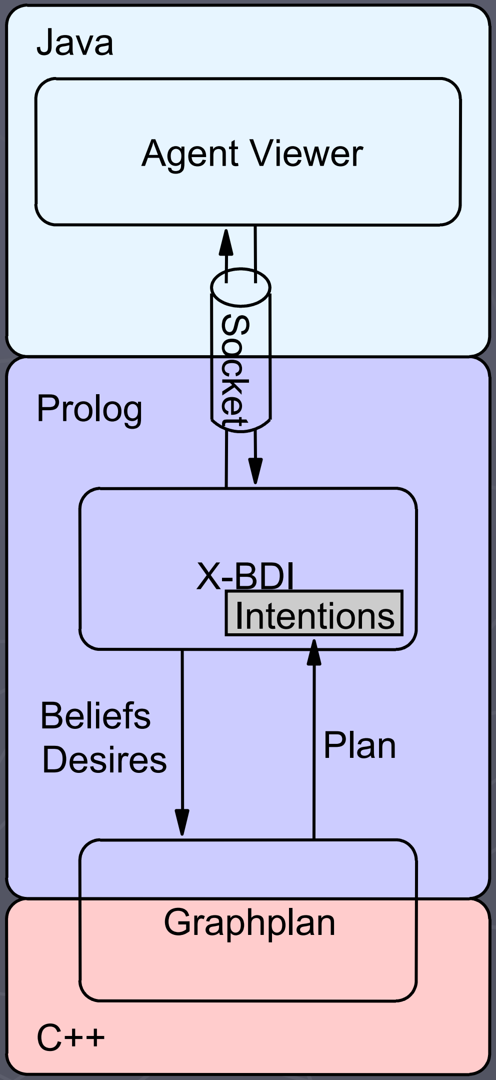
- If planning was successful an ordered set of actions is generated
- Resulting actions generate relative intentions



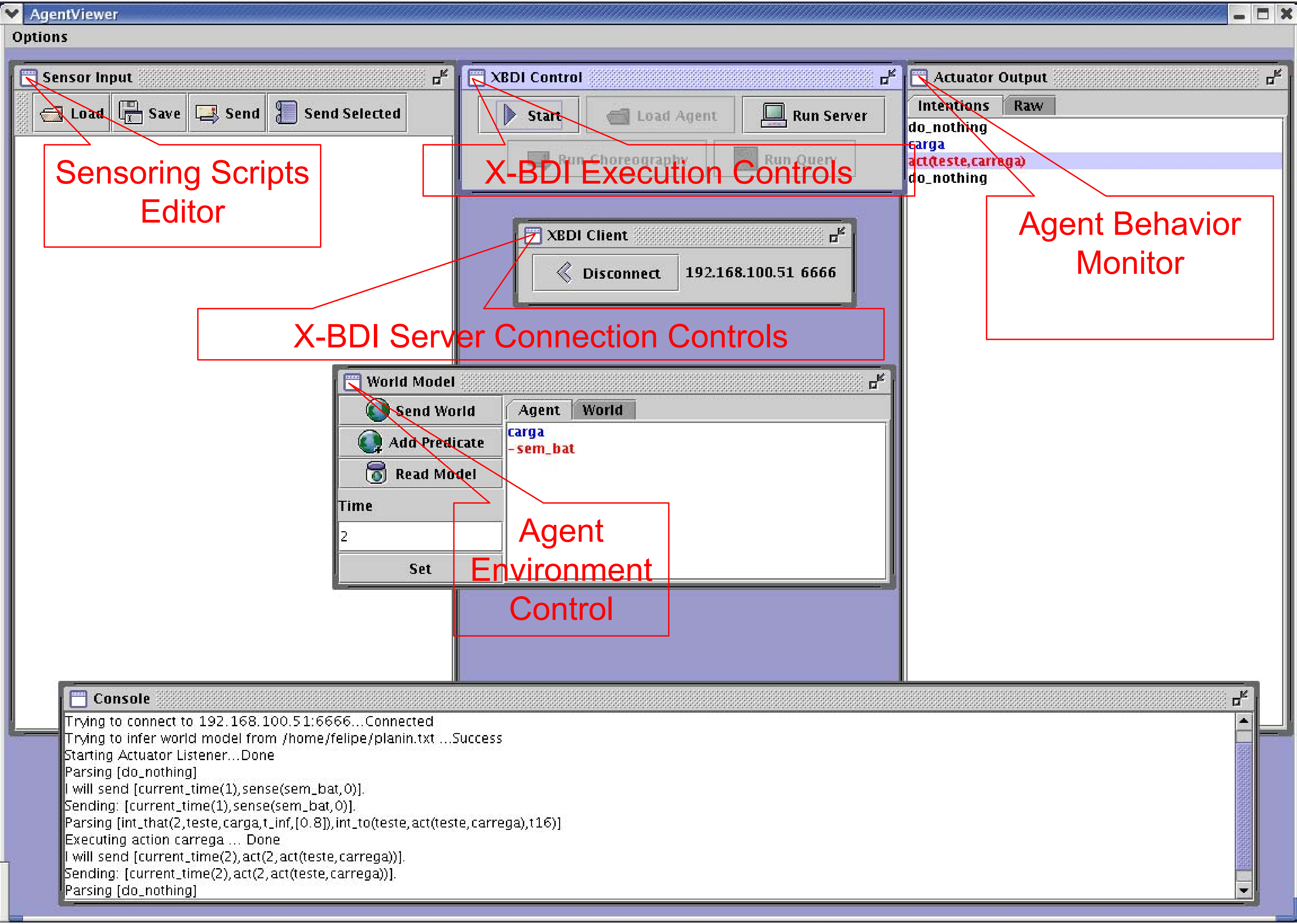
Prototype

Prototype Architecture

- Agent Description is sent to X-BDI for execution
- *AgentViewer* provides sensor input
- During the deliberation process X-BDI invokes the external planner
- Deliberation results (actions) is sent back to *AgentViewer*



Meneguzzi et al. SAC 04



Sensoring Scripts Editor

X-BDI Execution Controls

Agent Behavior Monitor

X-BDI Server Connection Controls

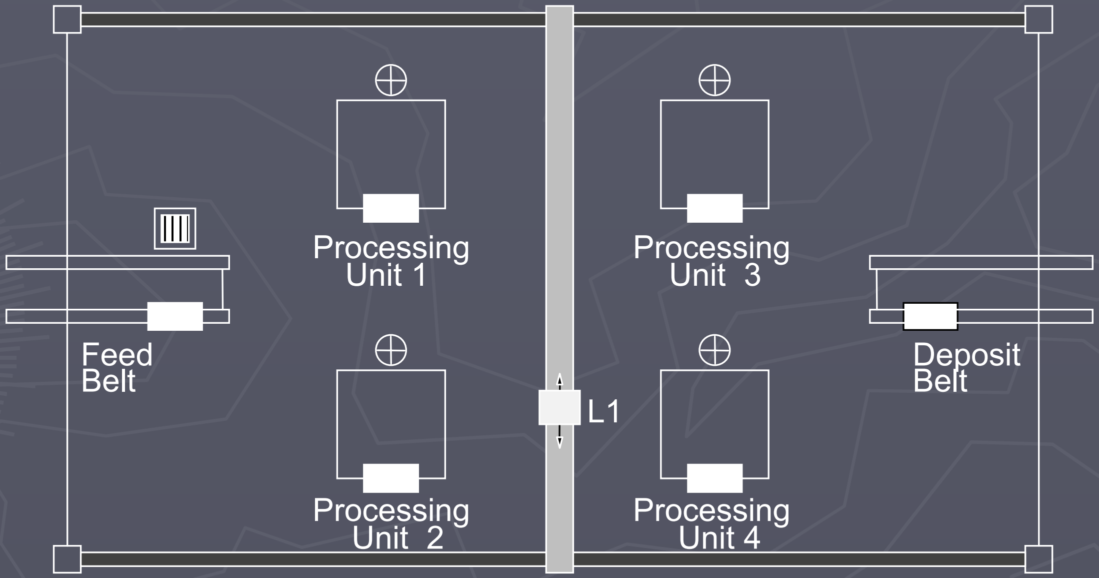
Agent Environment Control

```
Trying to connect to 192.168.100.51:6666...Connected
Trying to infer world model from /home/felipe/planin.txt ...Success
Starting Actuator Listener...Done
Parsing [do_nothing]
I will send [current_time(1),sense(sem_bat,0)].
Sending: [current_time(1),sense(sem_bat,0)].
Parsing [int_that(2,teste,carga,t_inf,[0.8]),int_to(teste,act(teste,carrega),t16)]
Executing action carrega ... Done
I will send [current_time(2),act(2,act(teste,carrega))].
Sending: [current_time(2),act(2,act(teste,carrega))].
Parsing [do_nothing]
```

Experiments

BDI Production Cell

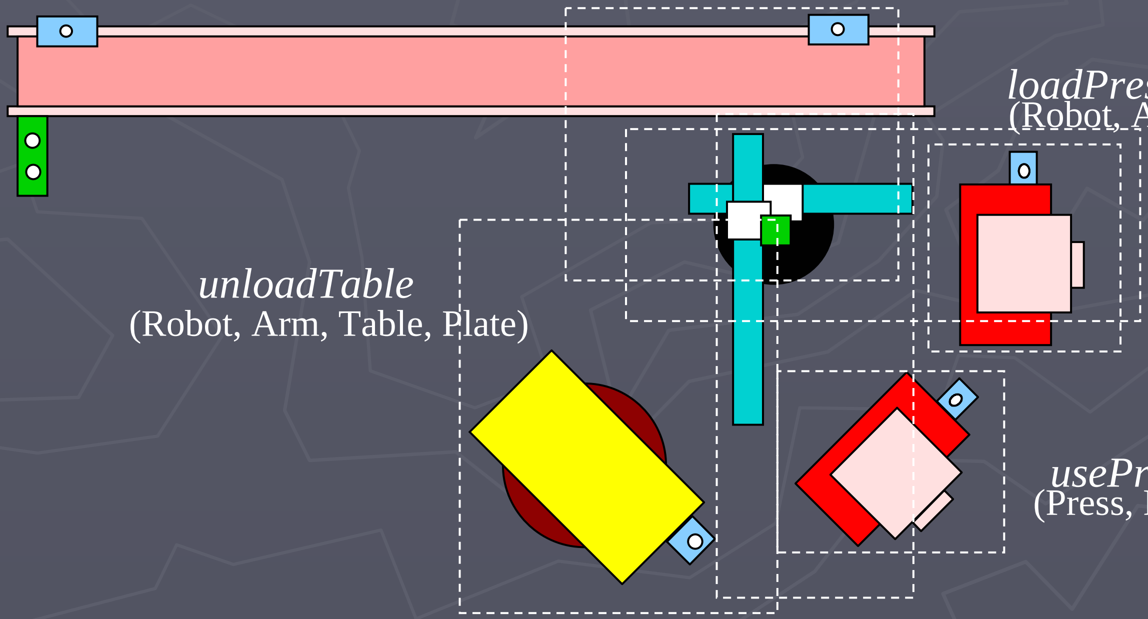
- Modeled after a real production cell
- Controlled by a BDI agent responsible for scheduling the processing of parts within the cell



Meneguzzi et al. SAC 04

Fault Injection

- Production Cell in which faults are possible
- Actions represent interactions among the components of the Cell



loadDepositBelt
(Robot, Arm, DepositBelt, Plate)

loadPress/unloadPress
(Robot, Arm, Press, Plate)

unloadTable
(Robot, Arm, Table, Plate)

usePress
(Press, Plate)

usePress
(Press, Plate)

loadPress/unloadPress
(Robot, Arm, Press, Plate)

Concluding Remarks

Conclusions

- It is possible to map means-end reasoning within X-BDI into any propositional planner
- Various modifications were necessary in order to externalize planning from the original X-BDI
- The class of problems tractable by X-BDI was augmented.

Results so far

- Mapping between BDI mental states and propositional planning problems (AAMAS'04 submitted)
- New definitions for desire possibility within X-BDI
- Tool for agent experimentation in X-BDI

Future Work

- Study regarding planning algorithm performance
- Comparison to other BDI formalisms
  - Relationship with other approaches
  - Performance
- Generalization of the mapping to other formalisms

Thank You

Propositional Planning in BDI Agents

Felipe Rech Meneguzzi
`felipe@cpts.pucrs.br`

Avelino Francisco Zorzo

Michael da Costa Móra

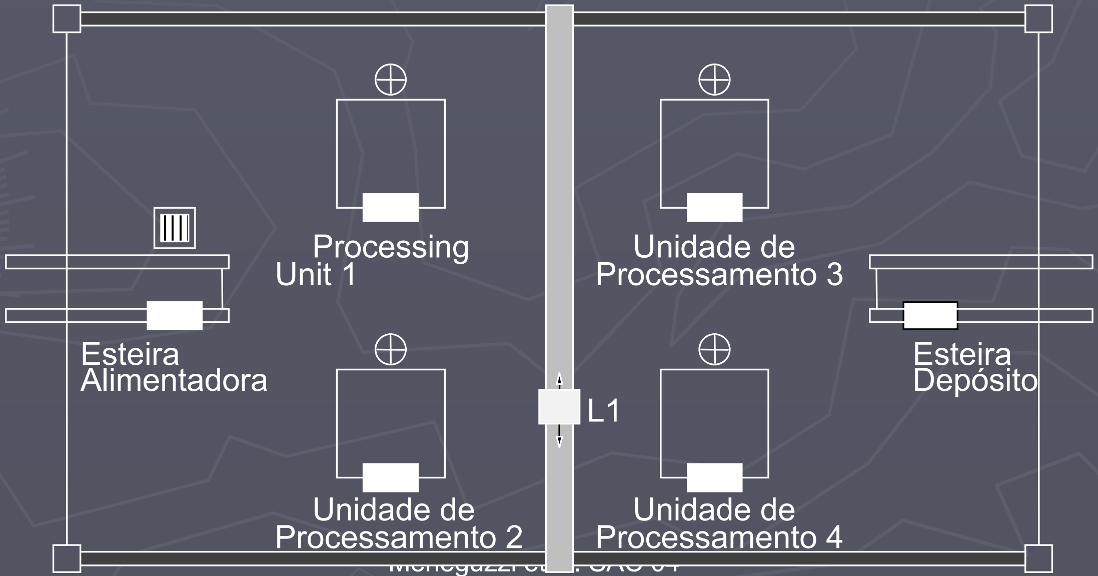
PUCRS – Porto Alegre – Brazil

Experiment

Production Cell Version 2

BDI Production Cell

- Modeled after a real production cell
- Controlled by a BDI agent responsible for scheduling the processing of parts within the cell



Agent Desires

Desire	Pre-Condition

`finished(bloc1)`	`if`
	`processed(bloc1,procUnit1)`
	`processed(bloc1,procUnit2)`
	`processed(bloc1,procUnit3)`

`processed(bloc1,procUnit1)`	`if`
	`bloc(bloc1)`

`processed(bloc1,procUnit2)`	`If`
	`bloc(bloc1)`

`processed(bloc1,procUnit3)`	`if`
	`bloc(bloc1)`

Initial Beliefs

procUnit(procUnit1)	procUnit(procUnit2)
procUnit(procUnit3)	procUnit(procUnit4)
device(procUnit1)	device(procUnit2)
device(procUnit3)	device(procUnit4)
device(depositBelt)	device(feedBelt)
empty(procUnit1)	empty(procUnit2)
empty(procUnit3)	empty(procUnit4)
empty(depositBelt)	

Processing Example

- Arrival of a new part to the cell
- Sensoring
    - `bloc(bloc1)`
    - `over(bloc1,feedBelt)`

- Eligible Desires
    - `processed(bloc1,procUnit1)`
    - `processed(bloc1,procUnit2)`
    - `processed(bloc1,procUnit3)`

Mapping

Start State	
`procUnit(procUnit1)`	`procUnit(procUnit2)`
`procUnit(procUnit3)`	`procUnit(procUnit4)`
`device(procUnit1)`	`device(procUnit2)`
`device(procUnit3)`	`device(procUnit4)`
`device(depositBelt)`	`device(feedBelt)`
`empty(procUnit1)`	`empty(procUnit2)`
`empty(procUnit3)`	`empty(procUnit4)`
`empty(depositBelt)`	`bloc(bloc1)`
`over(bloc1, feedBelt)`	

Goal State
`processed(bloc1, procUnit1)`
`processed(bloc1, procUnit2)`
`processed(bloc1, procUnit3)`

- Agent actions become STRIPS operators

Planning Outcome

- There is a plan that satisfies all of the agent’s desires

```
move(bloc1,feedBelt,procUnit2)
process(bloc1,procUnit2)
move(bloc1,procUnit2,procUnit1)
process(bloc1,procUnit1)
move(bloc1,procUnit1,procUnit3)
process(bloc1,procUnit3)
```

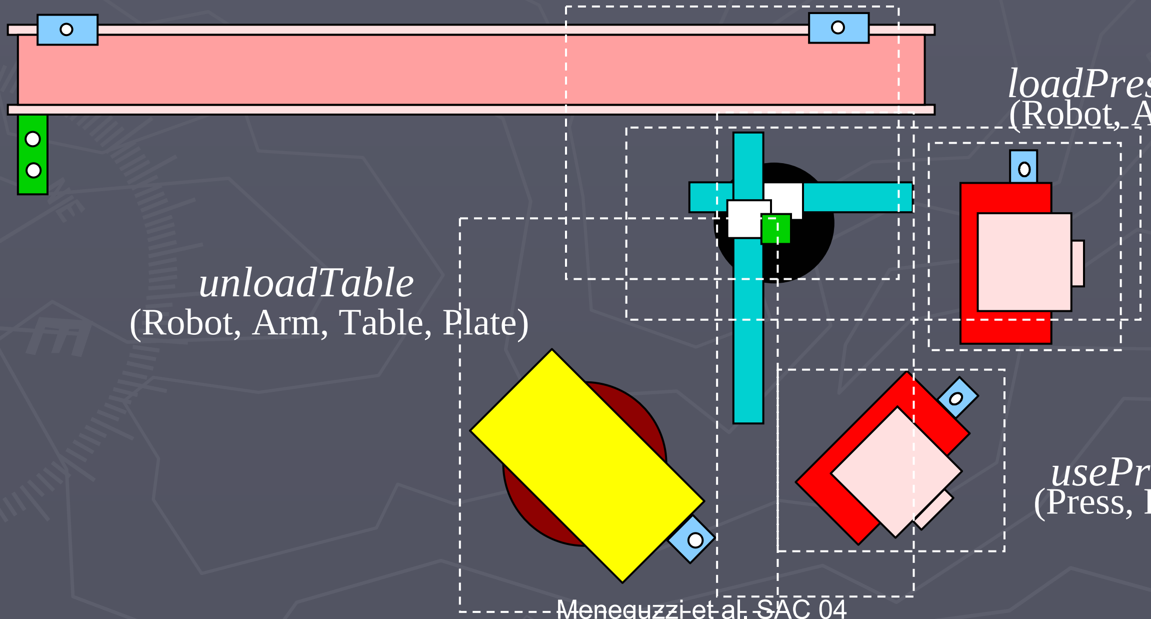
- Mapping
  - Candidate Desires → Primary Intentions
  - Plan operators become the actions within the relative intentions

Experiment

Fault Injection

Fault Injection

- Production Cell in which faults are possible
- Actions represent interactions among the components of the Cell



loadDepositBelt
(Robot, Arm, DepositBelt, Plate)

loadPress/unloadPress
(Robot, Arm, Press, Plate)

unloadTable
(Robot, Arm, Table, Plate)

usePress
(Press, Plate)

usePress
(Press, Plate)

loadPress/unloadPress
(Robot, Arm, Press, Plate)

Agent Desires

Desire	Pre-Condition

`done(P)`	`if`
	`plate(P)`

`loaded(depositBelt, P)`	`if`
	`done(P)`

Initial Beliefs

`table(table)`	`robot(robot)`
`arm(arm1,robot)`	`arm(arm2,robot)`
`empty(arm1)`	`empty(arm2)`
`press(press1)`	`press(press2)`
`empty(press1)`	`empty(press2)`
`depositBelt(depositBelt)`	`empty(depositBelt)`
`¬failed(table)`	`¬failed(robot)`
`¬failed(arm1)`	`¬failed(arm2)`
`¬failed(press1)`	`¬failed(press2)`
`¬failed(depositBelt)`	

Processing Example

- Arrival of a new metal plate
- Sensoring
  - `plate(plate1)`
  - `loaded(table,plate1)`
- Eligible Desires
  - `done(plate1)`
- Resulting Plan
  - `unloadTable(robot,arm1,table,plate1)`
  - `loadPress(robot,arm1,press1,plate1)`
  - `usePress(press1,plate1)`

Injecting a Fault

- Situation similar to the previous one
- `press1` fails
  - `failed(press1)`
- In this case, a different plan is generated
  - `unloadTable(robot,arm1,table,plate1)`
  - `loadPress(robot,arm1,press2,plate1)`
  - `usePress(press2,plate1)`