Extensions of the BDI Architecture as Reusable Solutions to Ease the Development of Sophisticated BDI Agents

Ingrid Nunes
Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, Brazil

WESAAC 2019 – May 4, 2019 – Florianópolis, BR
BDI Architecture

• Belief-Desire-Intention (BDI)
  • Model of agency with roots in practical reasoning and intentional systems
    • Set of concepts for thinking about and building agents
  • Behavior arises due to the agent
    • Committing to some of its desires
    • Selecting actions that achieve its intentions given its beliefs
Abstract Rational Architecture

• Rao and Georgeff (1995) proposed an abstract interpreter for agents

• Key: the notion of events
  • the inputs of the system
  • both externals (from the environment) and internal (within the system)

Algorithm 1 BDI Rational Agent Interpreter

1: initialize_state()
2: while ¬quit do
3:  options ← option_generator(event_queue, B, G, I);
4:  selected_options ← deliberate(options, B, G, I);
5:  update_intentions(selected_options, I);
6:  execute(I);
7:  drop_successful_attitudes(B, G, I);
8:  drop_impossible_attitudes(B, G, I);
9:  end while
BDI Architecture

Goal vs. Plans

Objects vs. BDI Agents

CLASS
- att1
- att2
+ met1()
+ met2()

AGENT
- belief1
- belief2
+ goal1
+ goal2
- plan1A()
- plan2A()
- plan2B()
BDI4JADE is an agent platform that implements the BDI (belief-desire-intention) architecture. It consists of a BDI layer implemented on top of JADE.

BDI4JADE leverages all the features provided by JADE and reuses it as much as possible. Other highlights of our JADE extension, besides providing BDI abstractions and the reasoning cycle, include:

- **Use of Capabilities** – agents aggregate a set of capabilities, which are a collection of be-
public class PingPongCapability extends Capability {

    @GoalOwner(capability = PingPongCapability.class)
    public static class PingGoal implements Goal {
        private static final long serialVersionUID = -7733145369836002329L;
    }

    private static final long serialVersionUID = -4800805796961540570L;

    @Belief
    String neighbour;

    @bd14jade.annotation.Plan
    private Plan pingPlan = new DefaultPlan(PingGoal.class, PingPlanBody.class);

    @Belief
    Integer pingTimes;

    @bd14jade.annotation.Plan
    private Plan pongPlan = new DefaultPlan(MessageTemplate.MatchContent(PingPlanBody.MSG_CONTENT), PongPlanBody.class);

    public PingPongCapability(String neighbour, int pingTimes) {
        this.neighbour = neighbour;
        this.pingTimes = pingTimes;
    }
}
public class PongPlanBody extends AbstractPlanBody {

    private static final Log log = LogFactory.getLog(PongPlanBody.class);
    public static final String MSG_CONTENT = "pong";
    private static final long serialVersionUID = -3352874506241004611L;

    private ACLMessage pingMsg;

    @Override
    public void action() {
        log.info("Ping received from agent " + pingMsg.getSender().getName() + ",!");
        ACLMessage reply = pingMsg.createReply();
        reply.setContent(MSG_CONTENT);
        this.myAgent.send(reply);
        log.info("Pong sent to agent " + pingMsg.getSender().getName() + ",!");
        setEndState(EndState.SUCCESSFUL);
    }

    @Parameter(direction = Direction.IN)
    public void setMessage(ACLMessage pingMsg) {
        this.pingMsg = pingMsg;
    }
}

04/05/2019
Ingrid Nunes <ingridnunes@inf.ufrgs.br>
public PingPongAction() {
    super.putValue(Action.NAME, "Ping Pong Agents");
    this.agent1 = new SingleCapabilityAgent(new PingPongCapability(AGENT_2, 2));
    this.agent2 = new SingleCapabilityAgent(new PingPongCapability(AGENT_1, 1));
}

@Override
public void actionPerformed(ActionEvent e) {
    this.agent1.addGoal(new PingPongCapability.PingGoal());
    this.agent2.addGoal(new PingPongCapability.PingGoal());
}
BDI Architecture

- **Flexible** architecture to implement cognitive agents
- Separation of **Goals** and **Plans**
- **Customisation Functions**
  - Key issue

- Existing techniques
  - Direct use by mainstream software developers is still problematic
    - Expert knowledge is still required to develop BDI agents
    - Real barrier to the large-scale adoption of this kind of agent technology

Ingrid Nunes <ingridnunes@inf.ufrgs.br>
Agenda

1. Network Function Virtualisation Orchestration with BDI Agents
2. Extensions to the BDI Architecture
   A. Plan Selection based on Multi-attribute Utility Theory (MAUT)
   B. Learning-based Plan Selection
3. Automated Management of Remediation Actions
NFV Orchestration with BDI Agents

Extensions of the BDI Architecture as Reusable Solutions to Ease the Development of Sophisticated BDI Agents
NFV Orchestration with BDI Agents

- Network Functions Virtualisation (NFV)
  - Decouples network services from physical devices (middleboxes) through virtualization
  - Middleboxes can be virtualised into cheap and easily deployable virtual machines
NFV Orchestration with BDI Agents

• NFV orchestration problems
  • Selection
    • Which VNFs are used and their ordering
  • Placement
    • Where in the physical network the VNFs will be instantiated
  • Chaining
    • Which virtual paths connect the placed VNFs
NFV Orchestration with BDI Agents

• Research Problem
  • How to address the selection, placement, and chaining problems using a decentralised approach solution?

• Proposed Solution
  • Decentralised NFV architecture that uses the BDI agents
  • Auction protocol to attack the selection, placement and chaining problems
  • Bidding heuristic
  • Implementation of the auction protocol and the bidding heuristic in a testbed
ETSI NFV Architecture
ETSI NFV Architecture
Capabilities and their Relationships

- Original Relationship
  - Inclusion

- Our Extensions
  - Association
  - Composition
  - Generalisation

Legend:
- Agent
- Capability
- Association
- Composition
- Inheritance
- Aggregation

Motorcycle
- <<external>> drive(x,y)
- achieveSpeed(s)
- turn(dir)
- speed
- location
- Ride

Car
- <<external>> findRoute(x,y)
- maps congestionZones
- preferences
- MainRoute
- FastestRoute
- ShortestRoute

GearController
- <<external>> MonitorGear
- ChangeGear
- currentGear

MotoDriver
- Accelerate
- Break
- RotateWheel

CarDriver
- Accelerate
- Break
- RotateSWheel
BDI Agents for NFV Orchestration

- Reasoning Cycle
  - The BDI reasoning cycle
• Reasoning Cycle
  • The BDI reasoning cycle

• Capabilities
  • BDI agent modules
  • Cluster a set of beliefs and plans that together are able to handle events or achieve goals
  • Modularises a particular functional behaviour
    • Enables reuse
BDI Agents for NFV Orchestration

• Reasoning Cycle
  • The BDI reasoning cycle

• Auction Capability
  • Agents negotiate through auctions to attack the selection, placement and chaining problems

Auction algorithms are simple mechanisms used to reach agreements on the allocation of scarce resources

BDI Agent
  - Bid Function
  - Reasoning Cycle
  - Auction Capability
  - Orchestration Capability

FIPA Contract Net Protocol
  - Cap.
  - Cap.
  - ...
  - Cap.

VNF VNF VNF
• Auction Process
  • NFV-A protocol
  • The bid with the **lowest cost**, considering the auctioneer’s preferences, **wins** the auction
BDI Agents for NFV Orchestration

- Reasoning Cycle
  - The BDI reasoning cycle

- Auction Capability
  - Agents negotiate through auctions to attack the selection, placement and chaining problems

- Bid Function
  - Bidding heuristic
BDI Agents for NFV Orchestration

- **Reasoning Cycle**
  - The BDI reasoning cycle

- **Auction Capability**
  - Agents negotiate through auctions to attack the selection, placement and chaining problems

- **Bid Function**
  - Bidding heuristic

- **Orchestrator Capability**
  - Download VNFs, allocate resources and create virtual paths
Implementation and Evaluation

• **BDI4JADE**
  • Allow BDI agents to be implemented in pure Java

• **Containernet**
  • Fork of Mininet that provides OS-level virtualisation

• **Agent-to-VNF communication**
  • Each VNF is controlled by a Python script that receives/makes RESTful calls from/to agents
BDI Agents to Combat a DDoS Attack

• Belief Revision
  • Detecting when the network is being over used
    • usage(link) > threshold → belief (overUsage(link))
    • usage(link) < threshold → belief (overUsage(link))
  • It is unknown that this abnormal behaviour is benign (not malicious)
    • overUsage(link) → belief (~regularUsage(link))
BDI Agents to Combat a DDoS Attack

- Goal Generation and Deliberation
  - **Protect** the network and find out if there is an **attack**
    - belief(overUsage(link)) → goal(attackPrevented(link)) ∧ goal(? regularUsage(link))
  - If there is a particular **IP** that is anomalous, achieve **similar** goals
    - belief(anomalous(ip)) → goal(restricted(ip)) ∧ goal(? benign(ip))
  - If a **flow** is identified as a threat, **respond** this
    - belief(threat(flow)) → goal(threatResponded(flow))
• Goal Generation and Deliberation
  • Ideally, the network should be fully operational
    • belief(\neg fullyOperational(link)) \land belief(regularUsage(link)) \rightarrow goal(fullyOperational(link))
    • belief(restricted(ip)) \land belief(\neg anomalous(ip)) \rightarrow goal(\neg restricted(ip))
## BDI Agents to Combat a DDoS Attack

<table>
<thead>
<tr>
<th>Condition</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>overUsage(link)</td>
<td>p</td>
</tr>
<tr>
<td>regularUsage(link)</td>
<td>~</td>
</tr>
<tr>
<td>fullyOperational(link)</td>
<td></td>
</tr>
<tr>
<td>attackPrevented(link)</td>
<td></td>
</tr>
<tr>
<td>anomalous(ip)</td>
<td></td>
</tr>
<tr>
<td>restricted(ip)</td>
<td></td>
</tr>
<tr>
<td>ipRateLimited(ip)</td>
<td></td>
</tr>
<tr>
<td>benign(ip)</td>
<td></td>
</tr>
<tr>
<td>flowRecord(ip)</td>
<td></td>
</tr>
<tr>
<td>threat(flow)</td>
<td></td>
</tr>
<tr>
<td>flowRateLimited(flow)</td>
<td></td>
</tr>
<tr>
<td>threatResponded(flow)</td>
<td></td>
</tr>
</tbody>
</table>
### BDI Agents to Combat a DDoS Attack

<table>
<thead>
<tr>
<th>Event</th>
<th>LLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>overUsage(link)</td>
<td>p</td>
</tr>
<tr>
<td>regularUsage(link)</td>
<td>~</td>
</tr>
<tr>
<td>fullyOperational(link)</td>
<td>¬</td>
</tr>
<tr>
<td>attackPrevented(link)</td>
<td>p</td>
</tr>
<tr>
<td>anomalous(ip)</td>
<td></td>
</tr>
<tr>
<td>restricted(ip)</td>
<td></td>
</tr>
<tr>
<td>ipRateLimited(ip)</td>
<td></td>
</tr>
<tr>
<td>benign(ip)</td>
<td></td>
</tr>
<tr>
<td>flowRecord(ip)</td>
<td></td>
</tr>
<tr>
<td>threat(flow)</td>
<td></td>
</tr>
<tr>
<td>flowRateLimited(flow)</td>
<td></td>
</tr>
<tr>
<td>threatResponded(flow)</td>
<td></td>
</tr>
</tbody>
</table>

**Plan:** LimitLinkRate (LLR)

**Goal:** attackPrevented(link)

**Context:** overUsage(link)

**Actions:**
- limit(link, rate)
- belief(¬fullyOperational(link))
- belief(attackPrevented(link))

---

04/05/2019

Ingrid Nunes <ingridnunes@inf.ufrgs.br>
## BDI Agents to Combat a DDoS Attack

<table>
<thead>
<tr>
<th></th>
<th>LLR</th>
<th>ALS</th>
</tr>
</thead>
<tbody>
<tr>
<td>overUsage(link)</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>regularUsage(link)</td>
<td>~</td>
<td>~</td>
</tr>
<tr>
<td>fullyOperational(link)</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>attackPrevented(link)</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>anomalous(ip)</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>restricted(ip)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ipRateLimited(ip)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>benign(ip)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>flowRecord(ip)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>threat(flow)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>flowRateLimited(flow)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>threatResponded(flow)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Plan:** AnalyseLinkStatistics (ALS)
**Goal:** ?regularUsage(link)
**Context:** -
**Actions:**

```plaintext
// Analyse link and detect outliers
∀ip.(outlier(ip)) → belief(anomalous(ip)) ∧ belief(~ benign(ip))
∃ip.(anomalous(ip)) → belief(~regularUsage(link))
♯ip.(anomalous(ip)) → belief(regularUsage(link))
```

---

04/05/2019

Ingrid Nunes <ingridnunes@inf.ufrgs.br>
BDI Agents to Combat a DDoS Attack

<table>
<thead>
<tr>
<th></th>
<th>LLR</th>
<th>ALS</th>
<th>LIPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>overUsage(link)</td>
<td>p</td>
<td>p</td>
<td>¬*</td>
</tr>
<tr>
<td>regularUsage(link)</td>
<td>~</td>
<td>~</td>
<td>¬</td>
</tr>
<tr>
<td>fullyOperational(link)</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>attackPrevented(link)</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>anomalous(ip)</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>restricted(ip)</td>
<td></td>
<td></td>
<td>p</td>
</tr>
<tr>
<td>ipRateLimited(ip)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>benign(ip)</td>
<td>~</td>
<td></td>
<td></td>
</tr>
<tr>
<td>flowRecord(ip)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>threat(flow)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flowRateLimited(flow)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>threatResponded(flow)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Plan: LimitIPRate (LIPR)
Goal: restricted(ip)
Context: anomalous(ip)
Actions:
  limit(ip, rate)
  belief(ipRateLimited(ip))
  belief(restricted(ip))

\[ \exists ip'. (anomalous(ip') \land \neg restricted(ip')) \rightarrow regularUsage(link) \]
BDI Agents to Combat a DDoS Attack

<table>
<thead>
<tr>
<th></th>
<th>LLR</th>
<th>ALS</th>
<th>LIPR</th>
<th>RLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>overUsage(link)</td>
<td>p</td>
<td>p</td>
<td>¬*</td>
<td>¬</td>
</tr>
<tr>
<td>regularUsage(link)</td>
<td>~</td>
<td>~</td>
<td>¬</td>
<td>p</td>
</tr>
<tr>
<td>fullyOperational(link)</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>p</td>
</tr>
<tr>
<td>attackPrevented(link)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>~</td>
</tr>
<tr>
<td>anomalous(ip)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>restricted(ip)</td>
<td>p</td>
<td>p</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>ipRateLimited(ip)</td>
<td>p</td>
<td>p</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>benign(ip)</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>flowRecord(ip)</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>threat(flow)</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>flowRateLimited(flow)</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>threatResponded(flow)</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
</tr>
</tbody>
</table>

Plan: RestoreLinkRate (RLR)
Goal: fullyOperational(link)
Context: regularUsage(link)
Actions:
  unlimit(link)
  belief(fullyOperational(link))
  belief(¬ attackPrevented(link))

04/05/2019  Ingrid Nunes <ingridnunes@inf.ufrgs.br>
### BDI Agents to Combat a DDoS Attack

<table>
<thead>
<tr>
<th></th>
<th>LLR</th>
<th>ALS</th>
<th>LIPR</th>
<th>RLR</th>
<th>AIPF(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>overUsage(link)</td>
<td>p</td>
<td>p</td>
<td>¬*</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>regularUsage(link)</td>
<td>~</td>
<td>~</td>
<td>¬</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>fullyOperational(link)</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>attackPrevented(link)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>~</td>
<td>~</td>
</tr>
<tr>
<td>anomalous(ip)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>restricted(ip)</td>
<td>p</td>
<td>p</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>ipRateLimited(ip)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>benign(ip)</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>flowRecord(ip)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>threat(flow)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flowRateLimited(flow)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>threatResponded(flow)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Plan:** AnalyzeIPFlows (AIPF)
**Goal:** ?benign(ip)
**Context:** anomalous(ip)
**Actions:**
- `goal(? flowRecord(ip))`
  - `∀flow. (malicious(flow)) → belief(threat(flow))`
  - `∃flow. (threat(flow) ∧ srcIP(flow) = ip) → belief(¬benign(ip))`
  - `∃flow. (threat(flow) ∧ srcIP(flow) = ip) → belief(benign(ip))`
### BDI Agents to Combat a DDoS Attack

<table>
<thead>
<tr>
<th>State</th>
<th>LLR</th>
<th>ALS</th>
<th>LIPR</th>
<th>RLR</th>
<th>AIPF(1)</th>
<th>RF</th>
</tr>
</thead>
<tbody>
<tr>
<td>overUsage(link)</td>
<td>p</td>
<td>p</td>
<td>¬*</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>regularUsage(link)</td>
<td>~</td>
<td>~</td>
<td>¬</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>fullyOperational(link)</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>attackPrevented(link)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>~</td>
<td>~</td>
<td>~</td>
</tr>
<tr>
<td>anomalous(ip)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>restricted(ip)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>ipRateLimited(ip)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>benign(ip)</td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>~</td>
<td></td>
</tr>
<tr>
<td>flowRecord(ip)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
</tr>
<tr>
<td>threat(flow)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flowRateLimited(flow)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>threatResponded(flow)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Plan:** RecordFlow (RF)

**Goal:** flowRecord(ip)

**Context:**

**Actions:**
- recordFlow(ip)
- belief(flowRecord(ip))
## BDI Agents to Combat a DDoS Attack

<table>
<thead>
<tr>
<th>State</th>
<th>LLR</th>
<th>ALS</th>
<th>LIPR</th>
<th>RLR</th>
<th>AIPF(1)</th>
<th>RF</th>
<th>AIPF(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>overUsage(link)</td>
<td>p</td>
<td>p</td>
<td>¬*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>regularUsage(link)</td>
<td>~</td>
<td>~</td>
<td>¬</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fullyOperational(link)</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>attackPrevented(link)</td>
<td>p</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>anomalous(ip)</td>
<td></td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>restricted(ip)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ipRateLimited(ip)</td>
<td></td>
<td></td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>benign(ip)</td>
<td></td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>¬</td>
<td></td>
</tr>
<tr>
<td>flowRecord(ip)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>threat(flow)</td>
<td></td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flowRateLimited(flow)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>threatResponded(flow)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Plan: AnalyseIPFlows (AIPF)
- **Goal: ?benign(ip)**
- **Context: anomalous(ip)**

**Actions:**
- goal(?flowRecord(ip))
- \( \forall \text{flow}.(\text{malicious}(\text{flow}) \rightarrow \text{belief}(\text{threat}(\text{flow}))) \)
- \( \exists \text{flow}.(\text{threat}(\text{flow}) \land \text{srcIP}(\text{flow}) = ip) \rightarrow \text{belief}(\neg \text{benign}(ip)) \)
- \( \neg \exists \text{flow}.(\text{threat}(\text{flow}) \land \text{srcIP}(\text{flow}) = ip) \rightarrow \text{belief}(\text{benign}(ip)) \)
# BDI Agents to Combat a DDoS Attack

<table>
<thead>
<tr>
<th>Condition</th>
<th>LLR</th>
<th>ALS</th>
<th>LIPR</th>
<th>RLR</th>
<th>AIPF(1)</th>
<th>RF</th>
<th>AIPF(2)</th>
<th>LFR</th>
</tr>
</thead>
<tbody>
<tr>
<td>overUsage(link)</td>
<td>p</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>regularUsage(link)</td>
<td>~</td>
<td>~</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fullyOperational(link)</td>
<td>¬</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>attackPrevented(link)</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>anomalous(ip)</td>
<td></td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>restricted(ip)</td>
<td></td>
<td></td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>ipRateLimited(ip)</td>
<td></td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td></td>
<td>p</td>
</tr>
<tr>
<td>benign(ip)</td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>¬</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>flowRecord(ip)</td>
<td></td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>¬</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>threat(flow)</td>
<td></td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>~</td>
<td></td>
</tr>
<tr>
<td>flowRateLimited(flow)</td>
<td></td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
</tr>
<tr>
<td>threatResponded(flow)</td>
<td></td>
<td></td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
</tr>
</tbody>
</table>

**Plan:** LimitFlowRate (LFR)

**Goal:** threatResponded(flow)

**Context:** threat(flow)

**Actions:**

- `limit(flow, rate)`
- `belief(flowRateLimited(flow))`
- `belief(threatResponded(flow))`
- `belief(\neg threat(flow))`

\[ \forall f. (\text{threat}(f) \land srcIP(flow) = srcIP(f)) \rightarrow belief(benign(ip)) \]
# BDI Agents to Combat a DDoS Attack

<table>
<thead>
<tr>
<th></th>
<th>LLR</th>
<th>ALS</th>
<th>LIPR</th>
<th>RLR</th>
<th>AIPF(1)</th>
<th>RF</th>
<th>AIPF(2)</th>
<th>LFR</th>
<th>RIPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>overUsage(link)</td>
<td>p</td>
<td>p</td>
<td>¬*</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>regularUsage(link)</td>
<td>~</td>
<td>~</td>
<td>¬</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>fullyOperational(link)</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>attackPrevented(link)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>p</td>
</tr>
<tr>
<td>anomalous(ip)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>~</td>
</tr>
<tr>
<td>restricted(ip)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>¬</td>
</tr>
<tr>
<td>ipRateLimited(ip)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>¬</td>
</tr>
<tr>
<td>benign(ip)</td>
<td>~</td>
<td>¬</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>flowRecord(ip)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>threat(flow)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
</tbody>
</table>

**Plan:** RestoreIPRate (RIPR)

**Goal:** ¬restricted(ip)

**Context:** benign(ip) ∧

  *ipRateLimited(ip)*

**Actions:**

- permit(ip)
- belief(¬ipRateLimited(ip))
- belief(¬restricted(ip))
- belief(¬anomalous(ip))

**threatResponded(flow):**

<table>
<thead>
<tr>
<th></th>
<th>LLR</th>
<th>ALS</th>
<th>LIPR</th>
<th>RLR</th>
<th>AIPF(1)</th>
<th>RF</th>
<th>AIPF(2)</th>
<th>LFR</th>
<th>RIPR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
</tbody>
</table>

04/05/2019

Ingrid Nunes <ingridnunes@inf.ufrgs.br>
# BDI Agents to Combat a DDoS Attack

<table>
<thead>
<tr>
<th>Condition</th>
<th>LLR</th>
<th>ALS</th>
<th>LIPR</th>
<th>RLR</th>
<th>AIPF(1)</th>
<th>RF</th>
<th>AIPF(2)</th>
<th>LFR</th>
<th>RIPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>overUsage(link)</td>
<td>p</td>
<td>p</td>
<td>¬*</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
</tr>
<tr>
<td>regularUsage(link)</td>
<td>~</td>
<td>~</td>
<td>¬</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>fullyOperational(link)</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>attackPrevented(link)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>p</td>
</tr>
<tr>
<td>anomalous(ip)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>~</td>
</tr>
<tr>
<td>restricted(ip)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>¬</td>
</tr>
<tr>
<td>ipRateLimited(ip)</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>¬</td>
</tr>
<tr>
<td>benign(ip)</td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>~</td>
<td>¬</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>flowRecord(ip)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>threat(flow)</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
<td>~</td>
<td>~</td>
<td></td>
</tr>
<tr>
<td>flowRateLimited(flow)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>threatResponded(flow)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
<td>p</td>
<td></td>
</tr>
</tbody>
</table>
BDI Agents to Combat a DDoS Attack

• Simulation Settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of servers</td>
<td>1 (HTTP and Streaming)</td>
</tr>
<tr>
<td>Number of non-malicious hosts</td>
<td>50</td>
</tr>
<tr>
<td>Traffic profile</td>
<td>Video: 17%, Web: 83%</td>
</tr>
<tr>
<td>Number of malicious hosts</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VNF</th>
<th>CPU</th>
<th>Memory</th>
<th>Disk</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link Monitor</td>
<td>1</td>
<td>1 GB</td>
<td>2 GB</td>
<td>-</td>
</tr>
<tr>
<td>Rate Limiter</td>
<td>1</td>
<td>1 GB</td>
<td>2 GB</td>
<td>-</td>
</tr>
<tr>
<td>Anomaly Detector</td>
<td>1</td>
<td>2 GB</td>
<td>10 GB</td>
<td>2 MB/s</td>
</tr>
<tr>
<td>Classifier</td>
<td>2</td>
<td>4 GB</td>
<td>10 GB</td>
<td>1 MB/s</td>
</tr>
<tr>
<td>Load Balancer</td>
<td>1</td>
<td>1 GB</td>
<td>2 GB</td>
<td>-</td>
</tr>
</tbody>
</table>

Diagram showing simulation settings with various NFVI-PoPs and VNFs.
BDI Agents to Combat a DDoS Attack

- 1: linkRateLimited(link)
BDI Agents to Combat a DDoS Attack

### Goal

<table>
<thead>
<tr>
<th>linkRateLimited(link)</th>
<th>$p^f_T$</th>
<th>$p^f_C$</th>
<th>$p^f_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.8</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

### Table

<table>
<thead>
<tr>
<th>Agent</th>
<th>$T$</th>
<th>$C$</th>
<th>$D$</th>
<th>Bid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50.00</td>
<td>16.88</td>
<td>25.84</td>
<td>0.50</td>
</tr>
<tr>
<td>2</td>
<td>55.56</td>
<td>26.91</td>
<td>39.63</td>
<td>0.54</td>
</tr>
<tr>
<td>3</td>
<td>100.00</td>
<td>16.87</td>
<td>33.12</td>
<td>0.88</td>
</tr>
<tr>
<td>4</td>
<td>100.00</td>
<td>4.88</td>
<td>18.01</td>
<td>0.87</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>5.65</td>
<td>13.22</td>
<td>0.09</td>
</tr>
<tr>
<td>6</td>
<td>62.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.60</td>
</tr>
</tbody>
</table>

### Diagram

- **NFVI-PoP 1**
  - CPUs: 6
  - Memory: 6 GB
  - Disk: 700 GB

- **NFVI-PoP 2**
  - CPUs: 4
  - Memory: 4 GB
  - Disk: 1000 GB

- **NFVI-PoP 3**
  - CPUs: 4
  - Memory: 4 GB
  - Disk: 1000 GB

- **NFVI-PoP 4**
  - CPUs: 2
  - Memory: 6 GB
  - Disk: 1000 GB

- **NFVI-PoP 5**
  - CPUs: 3/4
  - Memory: 4/6 GB
  - Disk: 398/400 GB

- **NFVI-PoP 6**
  - CPUs: 1/2
  - Memory: 1/4 GB
  - Disk: 498/500 GB

- **VNF Repository**

- **Rate Limiter**
  - CPUs: 1
  - Memory: 1 GB
  - Disk: 2 GB

- **Link Monitor**
  - CPUs: 1
  - Memory: 1 GB
  - Disk: 2 GB

- **HTTP and Video**

- **Internet**

- **B: 10 MB/s, L: 6 ms**
  - B: 8 MB/s
  - L: 9 ms
BDI Agents to Combat a DDoS Attack

- 1: linkRateLimited(link)
- 2: ?anomalousUsage(link)
BDI Agents to Combat a DDoS Attack

- 1: linkRateLimited(link)
- 2: ?anomalousUsage(link)
- 3: ipRateLimited(ip)
BDI Agents to Combat a DDoS Attack

- 1: linkRateLimited(link)
- 2: ?anomalousUsage(link)
- 3: ipRateLimited(ip)
- 4: splitTrafficTowardsIp(ip)
BDI Agents to Combat a DDoS Attack

- 1: linkRateLimited(link)
- 2: ?anomalousUsage(link)
- 3: ipRateLimited(ip)
- 4: splitTrafficTowardsIp(ip)
- 5a: flowRecord(ip)
- 5b: flowRecord(ip)

04/05/2019
Ingrid Nunes <ingridnunes@inf.ufrgs.br>
BDI Agents to Combat a DDoS Attack

<table>
<thead>
<tr>
<th>Goal</th>
<th>$p_f_T$</th>
<th>$p_f_C$</th>
<th>$p_f_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>flowRecord(ip)</td>
<td>0.0</td>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Agent</th>
<th>$T$</th>
<th>$C$</th>
<th>$D$</th>
<th>Bid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50.00</td>
<td>0.40</td>
<td>11.04</td>
<td>0.80</td>
</tr>
<tr>
<td>2</td>
<td>55.56</td>
<td>19.62</td>
<td>13.06</td>
<td>0.92</td>
</tr>
<tr>
<td>3</td>
<td>100.00</td>
<td>24.61</td>
<td>52.69</td>
<td>0.20</td>
</tr>
<tr>
<td>4</td>
<td>100.00</td>
<td>14.61</td>
<td>39.15</td>
<td>0.38</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
**BDI Agents to Combat a DDoS Attack**

- 1: linkRateLimited(link)
- 2: anomalousUsage(link)
- 3: ipRateLimited(ip)
- 4: splitTrafficTowardsIp(ip)
- 5a: flowRecord(ip)
- 5b: flowRecord(ip)
- 6 & 7: flowRateLimited(flow) x 8

---

04/05/2019

Ingrid Nunes <ingridnunes@inf.ufrgs.br>
Extensions to the BDI Architecture

Extensions of the BDI Architecture as Reusable Solutions to Ease the Development of Sophisticated BDI Agents
• Problem
  • How can a BDI agent select the best plan to achieve a goal based on its current preferences and the uncertainty of plan outcomes?
  • To be used by mainstream software developers
Model-driven Development

- Allows to **concentrate** in what is **important**
- May **remove** important **implementation details** in order to increase the productivity we need
  - Automation
  - Reuse
- Focus on business not code
- Key to building **complex software**
Extensions to the BDI Architecture

• **Problem**
  - How can a BDI agent select the best plan to achieve a goal based on its current preferences and the uncertainty of plan outcomes?
    - To be used by mainstream software developers

• **Our model-driven approaches**
  - Agents modelled based on an extended BDI model
  - Automatic code generation of agents able to select plans

• **Contributions**
  - **Meta-models** that extends the BDI model
  - **Algorithms to select plans**
  - **Transformation** of instances of our meta-model into source code
Plan Selection based on MAUT

Agent

- \( << B, G, SG, P, Pref >> \)
  - B: beliefs
  - G: goals
  - SG: softgoals
  - P: plans
  - Pref: preference function

Plan

- \( << G', C, D, Body >> \)
  - G': subset of goals
  - C: contributions \( << sg, prob, val >> \)
    - sg: softgoal
    - prob: probability
      - \([0, 1]\)
    - val: contribution value
      - \([0, 1]\)
  - D: dependencies
    - And Plan-Goal Dependency
    - Or Plan-Goal Dependency
  - Body

Softgoal (inspired by Tropos) is a broad agent objective, which is not achieved by a plan but is more or less satisfied due to the effect produced by agent actions that are part of plans.
Plan Selection based on MAUT

Preferences
Softgoal X: 0.6
Softgoal Y: 0.4

Goal A
- Plan A1
  - Goal B1
    - Plan B11
    - Plan B12
  - Goal B2
    - Plan B2
  - Goal C1
  - Goal C2
    - Plan C21
    - Plan C22
- Plan A2
  - Goal B1
    - Plan B11
    - Plan B12
  - Goal B2
    - Plan B2
  - Goal C1
  - Goal C2
    - Plan C21
    - Plan C22
- Plan 3
  - Goal B1
    - Plan B11
    - Plan B12
  - Goal B2
    - Plan B2
  - Goal C1
  - Goal C2
    - Plan C21
    - Plan C22

Softgoal X
- [0.3, 0.4; 0.7, 0.8]
- [0.2, 0.55; 0.8, 0.9]
- [0.45, 0.9; 0.55, 0.4]
- [1.0, 0.45]
- [0.9, 0.1; 0.1, 1.0]
- [1.0, 0.5]
- [0.25, 0.75; 0.75, 0.6]
- 0.55
- 0.45
Plan Selection based on MAUT

• **Contributions** for softgoals may be derived from dependencies

• Each plan has
  
  • **Expected contribution** for each softgoal
    \[
    EC(p, sg) = \sum_{c_i \in C | c_i[sg] = sg} c_i[\text{prob}] \times c_i[\text{value}]
    \]
  
  • **Plan Utility**
    \[
    PU(p, Pref, SG) = \sum_{sg \in SG} Pref(sg) \times \text{utility}(sg)
    \]

• Selected plan
  
  • **Maximum** utility

---

**Algorithm 1: SelectPlan(SG, Pref, P)**

1. selectedP ← null;
2. maxContrib ← null;
3. foreach p ∈ P do
   4. contrib ← 0;
   5. foreach sg ∈ SG do
      6. possibleContribs ← Contribution(p, sg);
      7. expectedContrib ← 0;
      8. foreach (prob, value) ∈ possibleContribs do
         9. expectedContrib ← expectedContrib + prob × value;
         10. contrib ← contrib + Pref(sg) × expectedContrib;
      11. if selectedP = null ∨ maxContrib < contrib then
         12. selectedP ← p;
         13. maxContrib ← contrib;
   14. return selectedPlan;
Model-to-Code Transformation

- Extension of **BDI4JADE**
- Use of the **Xpand2** template language
  - Part of the Model to Text (M2T) project of Eclipse

---

04/05/2019
Ingrid Nunes <ingridnunes@inf.ufrgs.br>
Model-to-Code Transformation
Model-to-Code Transformation

Metamodel

Template

```java
public class Plan extends SimplePlan {
    public Plan(String name) {
        super(name, this.nameBody.getClass());
        Map<Softgoal, List<PlanContribution>> contributions =
            (Map<Softgoal, List<PlanContribution>>) getMetaData();
        List<PlanContribution> sgContributions = null;

        // Set up contributions
        <![CDATA[
        for (PlanContribution contribution : contributions) {
            // Add contribution
            contribution.addPlan(Plan, new PlanDependency());
        }
        ]]>
    }
```
04/05/2019

Ingrid Nunes <ingridnunes@inf.ufrgs.br>
Evaluation

• Scenario
  • Transportation
    • Bike
    • Bus
    • Car
    • Motorcycle
  • Softgoals
    • Maximise Safety
    • Maximise Security
    • Maximise Performance
    • Minimise Cost
    • Maximise Comfort
  • Probabilities associated with
    • Time taken
    • Crashing
    • Being robbed

• Experiment
  • Randomly generate preferences
  • Randomly generate a scenario
  • Compute satisfaction for each transportation type
  • Select a plan using a plan selector
  • Store the satisfaction according to the selected plan
### Evaluation

<table>
<thead>
<tr>
<th>Softgoal</th>
<th>Bicycle</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prob</td>
<td>Val</td>
<td>Prob</td>
<td>Val</td>
<td>Prob</td>
<td>Val</td>
<td>Prob</td>
<td>Val</td>
</tr>
<tr>
<td>Safety</td>
<td>0.05</td>
<td>0.00</td>
<td>0.95</td>
<td>1.00</td>
<td>0.20</td>
<td>0.00</td>
<td>0.80</td>
<td>1.00</td>
</tr>
<tr>
<td>Security</td>
<td>0.20</td>
<td>0.00</td>
<td>0.80</td>
<td>1.00</td>
<td>0.05</td>
<td>0.00</td>
<td>0.475</td>
<td>0.00</td>
</tr>
<tr>
<td>Performance</td>
<td>0.05</td>
<td>0.00</td>
<td>0.475</td>
<td>0.00</td>
<td>0.475</td>
<td>0.00</td>
<td>0.475</td>
<td>0.04</td>
</tr>
<tr>
<td>Cost</td>
<td>0.05</td>
<td>0.00</td>
<td>0.95</td>
<td>0.90</td>
<td>1.00</td>
<td>0.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comfort</td>
<td>1.00</td>
<td>0.20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Probability of Being Robbed**
- **Probability of Crashing**
Evaluation: Results

One-way ANOVA with Post-hoc Tukey’s HSD tests → Significant differences

M = 0.57 0.63 0.64 0.58
n = 5000

Plan Selector

04/05/2019
Ingrid Nunes <ingridnunes@inf.ufrgs.br>
Evaluation: Results

- **Best results**
- **Limitations**
  - Representation of dependent probabilities
  - Plans have the same set of contributions
  - Quantitative values for preferences
Learning-based Plan Selection

- Several plan-selection approaches have been proposed in this context
  - Learning the context in which a plan possibly fails
  - Identifying the plan that best matches agent's preferences
- They do not consider
  - Uncertainty involving plan executions
  - Rely on information that is hard to elicit at development time

How to select a plan considering the context in which an agent is inserted, taking into account agent's preferences and the uncertainty of plan executions?
Learning-based Plan Selection

• Agents learn which are the expected plan outcomes according to particular contexts
  • It makes them able to predict plan outcomes based on the current context

• Information used to select a plan to achieve a goal
  • Taking into account preferences that are expressed over agent's softgoals

• Approach components
  • A meta-model
  • A technique to learn and predict plan outcomes
Learning-based Plan Selection

* Implemented as an extension of the BDI4JADE platform
Learning-based Plan Selection

- Describes how to use the information provided by our meta-model in order to properly select a plan in a given context.
- In an initial learning stage, plans are selected randomly.
  - Information collected and stored from plan executions.
  - Threshold.

<table>
<thead>
<tr>
<th>Execution</th>
<th>Truck Conditions</th>
<th>Traffic Conditions</th>
<th>Road Conditions</th>
<th>Time Taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.591</td>
<td>0.903</td>
<td>0.096</td>
<td>2.35</td>
</tr>
<tr>
<td>2</td>
<td>0.858</td>
<td>0.987</td>
<td>0.419</td>
<td>2.5</td>
</tr>
<tr>
<td>3</td>
<td>0.495</td>
<td>0.430</td>
<td>0.677</td>
<td>1.725</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Learning-based Plan Selection

- Use of existing **machine learning algorithms** to predict outcomes values
  - **Linear regression**
Learning-based Plan Selection

- Predicted outcomes are **converted** to an agent **utility**
- Utilities are **combined**
Learning-based Plan Selection

• Plan with the **highest predicted utility** is selected
Learning-based Plan Selection

<table>
<thead>
<tr>
<th>Plan</th>
<th>Softgoal</th>
<th>Influence Factor</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>AirplanePlan</td>
<td>minCosts</td>
<td>AirplaneConditions, WeatherConditions, Distance</td>
<td>FuelConsumption</td>
</tr>
<tr>
<td></td>
<td>maxPerformance</td>
<td>AirplaneConditions, WeatherConditions</td>
<td>TimeTaken</td>
</tr>
<tr>
<td></td>
<td>maxReliability</td>
<td>AirplaneConditions, AccidentProbability, ChanceOfTheft</td>
<td>LoadIntegrity</td>
</tr>
<tr>
<td>ShipPlan</td>
<td>minCosts</td>
<td>ShipConditions, WeatherConditions, Distance</td>
<td>FuelConsumption</td>
</tr>
<tr>
<td></td>
<td>maxPerformance</td>
<td>ShipConditions, WeatherConditions, SeaConditions, HarborConditions</td>
<td>TimeTaken</td>
</tr>
<tr>
<td></td>
<td>maxReliability</td>
<td>ShipConditions, AccidentProbability, ChanceOfTheft</td>
<td>LoadIntegrity</td>
</tr>
<tr>
<td>TrainPlan</td>
<td>minCosts</td>
<td>TrainConditions, WeatherConditions, Distance</td>
<td>FuelConsumption</td>
</tr>
<tr>
<td></td>
<td>maxPerformance</td>
<td>TrainConditions, TrafficConditions, RailroadConditions</td>
<td>TimeTaken</td>
</tr>
<tr>
<td></td>
<td>maxReliability</td>
<td>TrainConditions, AccidentProbability, ChanceOfTheft</td>
<td>LoadIntegrity</td>
</tr>
<tr>
<td>TruckPlan</td>
<td>minCosts</td>
<td>TrackConditions, TrafficConditions, Distance</td>
<td>FuelConsumption</td>
</tr>
<tr>
<td></td>
<td>maxPerformance</td>
<td>TrackConditions, TrafficConditions, RoadConditions</td>
<td>TimeTaken</td>
</tr>
<tr>
<td></td>
<td>maxReliability</td>
<td>TrackConditions, AccidentProbability, ChanceOfTheft</td>
<td>LoadIntegrity</td>
</tr>
</tbody>
</table>
Learning-based Plan Selection

- Measurement of the agent satisfaction produced using different plan selection strategies
  - Random-based (RAN), LB-1, LB-100, LB-500 and LB-1000
- Threshold of plan executions: 50
- Number of iterations: 5000

<table>
<thead>
<tr>
<th>Plan Selector</th>
<th>M</th>
<th>SD</th>
<th>Min</th>
<th>Max</th>
<th>Cum Sat</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB-1</td>
<td>0.6812</td>
<td>0.091</td>
<td>0.219</td>
<td>0.961</td>
<td>3406.04</td>
</tr>
<tr>
<td>LB-100</td>
<td>0.6815</td>
<td>0.091</td>
<td><strong>0.262</strong></td>
<td>0.968</td>
<td>3407.99</td>
</tr>
<tr>
<td>LB-500</td>
<td>0.680</td>
<td>0.092</td>
<td>0.214</td>
<td><strong>0.982</strong></td>
<td>3402.68</td>
</tr>
<tr>
<td>LB-1000</td>
<td><strong>0.6816</strong></td>
<td>0.090</td>
<td>0.138</td>
<td>0.969</td>
<td><strong>3408.00</strong></td>
</tr>
<tr>
<td>RAN</td>
<td>0.597</td>
<td><strong>0.114</strong></td>
<td>0.102</td>
<td>0.941</td>
<td>2988.98</td>
</tr>
</tbody>
</table>
Learning-based Plan Selection
The Sam Tool

• Tool to support the design and implementation of software agents with learning capabilities

• Implemented as a plug-in for Eclipse

• Technologies
  • Graphiti – graphical modelling
  • Xpand – code generation
Automated Management of Remediation Actions

Extensions of the BDI Architecture as Reusable Solutions to Ease the Development of Sophisticated BDI Agents
From standalone to connected pieces of software

• Software components depend on one another

• Key differences from the past
  • Dynamic environment
  • Little can be assumed from the behaviour of other components
  • Intensive communication based on the internet
Automated Management of Remediation Actions

• Remediation Action
  • An action that mitigates the consequences/effects of a problem

• Why?
  • Causes of the problem are unknown
  • Addressing the causes takes too long

• Example

```
notify_load(name, rate, link)
setThreshold(t)
notify_detection(IPAddress)
limit(link, rate)
start(link)
classify(flow)
limit(flow, rate)
notify_new_record(flow)
notify_classification(label, flow)
start(IPAddress, samplingRate, length)
notify_detection(IPAddress)
```

04/05/2019

Ingrid Nunes <ingridnunes@inf.ufrgs.br>
How to **automate** in a **domain-independent way** the management of **remediation actions** and long-term problem resolution so that systems with remedial behaviour can **remain operational** and **recover from abnormal situations** without compromising their performance?
Automated Management of Remediation Actions

• Goal
  • $\sim$water_floor
  • min time

• Plans
  • place_bucket
    • Post-condition: $\sim$water_floor
    • Resources: 1 min
  • repair_ceiling
    • Pre-condition: have_tools
    • Post-conditions:
      • $\sim$water_floor
      • ceiling_ok
    • Resources: 60 min
Automated Management of Remediation Actions
Automated Management of Remediation Actions

1. Constrained Goals/Plan Selection
   - Goal metadata to indicate restrictions of how goals can be achieved
     - constraints + utility-function

2. Cause-effect Knowledge Model
   - Used to identify possible problem causes

3. Goal Generation
   - Identify problem causes
   - Address identified causes
Automated Management of Remediation Actions

1. Build cause-effect status
   - [no cause-effect status]
   - [otherwise]

2. Update status of cause factors

3. Evaluate effect goal

4. Evaluate cause
   - [cause factor goals unfinished]
   - [cause factor goals finished]

5. Generate cause factor test goals
   - [known cause]
   - [cause not found]
   - [unknown cause]

6. Generate cause factor achievement goals

7. Update cause-effect problem end state

Fact
Mandatory cause
Optional cause
Alternative cause

m
[m1...1]
a
o
o'
m'
a'
e'
e'

04/05/2019
Ingrid Nunes <ingridnunes@inf.ufrgs.br>
Automated Management of Remediation Actions

LimitLink
- Time = 5
- Network Availability = 0.6
- Vulnerability = 0.5

FindLimitIP
- Time = 30
- Network Availability = 0.8
- Vulnerability = 0.5

DoNothing
- Time = 0
- Network Availability = 1.0
- Vulnerability = 1.0

Reduction in the development effort with no impact in agent performance
Automated Management of Remediation Actions

- Reversion technique
  - Extends the BDI architecture
  - Specifies steps to be performed by the reasoning cycle
    - Goal Setup
    - Monitoring
    - Reversion Execution
Collaborations and References


• SCHARDONG, F.; NUNES, I.; SCHAEFFER FILHO, A.E. A Distributed NFV Orchestrator based on BDI Reasoning. IM 2017.

Collaborations and References

• NUNES, I.; LUCK, M. Softgoal-based Plan Selection in Model-driven BDI Agents, AAMAS 2014.


Collaborations and References


Grupo Prosoft @ UFRGS
Conclusion

BDI Agents to Combat a DDoS Attack

1. `linkRateLimited(link)`
2. `anomalousUsage(link)`
3. `ipRateLimited(ip)`
4. `splitTrafficTowardsIp(ip)`
5a. `flowRecord(ip)`
5b. `flowRecord(ip)`
6 & 7. `flowRateLimited(flow)`

The Sam Tool

- Tool to support the design and implementation of software agents with learning capabilities
- Implemented as a plug-in for Eclipse
- Technologies
  - Graphiti – graphical modelling
  - Xpand – code generation

Abstract Rational Architecture

- Rao and Georgeff (1992) proposed an abstract interpreter for agents
- Key: the notion of events
  - the inputs of the system
  - both external (from the environment) and internal (within the system)

Plan Selection based on MAUT

Plan A

- Goal A
- Preferences
  - Softgoal X: 0.6
  - Softgoal Y: 0.4
- Plans
  - Plan A1
  - Plan A2
  - Plan 3

Plan B

- Goal B1
- Preferences
  - Softgoal X: 0.43
  - Softgoal Y: 0.57
- Plans
  - Plan B11
  - Plan B12
  - Plan B2

Plan C

- Goal C1
- Preferences
  - Softgoal X: 0.5
  - Softgoal Y: 0.5
- Plans
  - Plan C1
  - Plan C2
  - Plan C21
  - Plan C22

Automated Management of Remediation

- Goals
  - ~water_floor
  - min time
- Plans
  - place_bucket
    - Post-condition: ~water_floor
    - Resources: 1min
  - repair_ceiling
    - Pre-condition: have_tools
    - Post-conditions:
      - ~water_floor
      - ceiling_ok
    - Resources: 60min

Abstract Rational Architecture

- Rao and Georgeff (1992) proposed an abstract interpreter for agents
- Key: the notion of events
  - the inputs of the system
  - both external (from the environment) and internal (within the system)
Abstract

• The Belief-Desire-Intention (BDI) architecture is widely known and largely used to develop autonomous agents. It has key components connected by a reasoning cycle that drives agent actions. This reasoning cycle includes functions that can be customised to provide agents with sophisticated behaviour such as making informed decisions about which course of actions (i.e., a plan) is more suitable considering the current environment state when multiple plans can be used. In this talk, I'll first demonstrate the potential of the BDI architecture by introducing our BDI-agent-based solution for combating DDoS attacks. Then, I'll overview our proposed approaches that extend the BDI architecture to support the development of BDI agents with improved plan selection. Finally, I'll present our latest extension to the BDI architecture that supports the construction of resilient systems by means of BDI agents that can autonomously manage remediation actions with the goal of keeping systems operational under abnormal conditions.