# Interpreter for Asbru Light

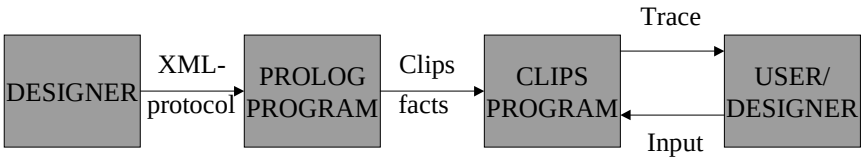
Tibor Bosse

## Overview

- General behaviour
- The Prolog program
- The Clips program
- Representation of plans
- Asbru Light
- Evaluation
- Demo

# General behaviour

- Intended users: Asbru protocol designers
- Two parts: Prolog & Clips




# The Prolog program

- Input: XML-protocol
- Output: file of Clips facts
  (representing the protocol)
- Built-in predicate for parsing

# The Clips program

- Input: file of Clips facts
- Output: trace
- Plans have states
- Plans activate each other
- Constant monitoring loop:
  - execution of plans
  - state transitions of plans
  - evaluation of logical expressions

# Representation of plans (1)

**PLAN Plan1**
PLAN BODY
**do** parallel
plan1a
plan1b

---

(plan (name plan1) (type do_parallel)
(subplans plan1a plan1b) (state ignored))

# Representation of plans (2)

Creation of fictive plans

(plan (name Check-for-jaundice-after-2-weeks) (type do_sequentially) (subplans Check-for-jaundice-after-2-weeks_1 Check-for-jaundice-after-2-weeks_2))

(plan (name Check-for-jaundice-after-2-weeks_1) (type do_any_order) (subplans …)

(plan (name Check-for-jaundice-after-2-weeks_2) (type if_then_else) (subplans …)

# Asbru Light

- Plans: all control structures + waiting strategy
- Cyclical plan: partly
- Single plan steps: partly
- Conditions: partly
- Propagation: default
- Intentions: only parsed
- Time annotations: only in conditions
- Domain definitions: yes

## Evaluation

- Useful piece of software
- But: focused on Jaundice protocol
- User friendly?

## Demo...