

# *Real-Time Agent Systems*

*Konstantin Vikhorev, Brian Logan, Natasha Alechina  
Agent Programming lab University of Nottingham, UK*

# Content

- Introduction
- Real-time Guarantees
- Changes to the BDI architecture
- AgentSpeak(RT): Agent Real-Time Architecture
- Future work
- Conclusion

# Introduction

- **Real-time is**
  - not about being 'fast', it is about being 'fast enough' to act on surrounded environments in well-specified way.
- **Two differences:**
  - time constraints on individual activities;
  - mechanism to meet those time constraints with acceptable predictability.
- **Architectures for highly dynamic environments:**
  - PRS, Spark, JAM have to be individually programmed

# BDI Model

- Three Components of BDI model:
  - **Beliefs** - represent the informational model of the agent's world;
  - **Desires (goals)** - represent the motivational state of the agent;
  - **Intentions** - represent the agent's choice of particular courses of action, that is, desires which the agent has chosen to pursue in the current situation.
  - *Events*: represent triggers for reactive activity by the agent.
- Reasoning cycle of a BDI agent:
  - Update beliefs;
  - Choose subset of desires;
  - Deliberate over desires and commit intentions to fulfill them.

# Real-time Guarantees

- Two types of Real-Time guarantees.
  - **Hard Real-time** – agent must process input information and produce response in a defined time or before that time.
  - **Soft Real-Time** – agent may not necessarily in all cases respond at the required time.
- ***Degree of “softness”*** of a task characterised by confidence level, i.e. probability that a task should complete successfully before its deadline

# Real-time BDI agent

- Agent's tasks are associated with a deadline and priority
- A set of tasks which can all be achieved by the deadlines is termed *feasible*.
- A maximally feasible set of tasks, which contains high priority tasks in preference to low priority tasks is defined as *priority maximal*.
- **Real-time BDI agent** – an agent, an agent which commits to and, with confidence  $\alpha$ , successfully executes a priority-maximal set of tasks by their deadlines.

# Changes to BDI model

- **Additional Information**
  - Deadlines
  - Priorities
  - Duration
- **Changes to BDI Execution cycle**
  - Scheduling algorithm

# Additional Information

## Changes in the goal and belief notions:

- External events for goals and beliefs must have a *deadline* and a *priority*.

## Changes in the plan notion:

- Plan inherits the *deadline* from top-level event.
- Plan also associated with a *duration*

# Changes to BDI reasoning

- Events are processed in priority order
  - the plans for each event are checked for a feasibility and an excitability.
- The time-limited matching of plans
  - selection stops when all events have an executable feasible plan or a user definable plan selection timeout is reached
- Effective task scheduling

# Scheduling Algorithm

- **Algorithm:**
  1. find the highest priority feasible intention;
  2. find the next most important intention which is feasible for the existing schedule;
  3. repeat 2 until no more intentions can be scheduled.
- **Earliest Deadline First execution**

# AgentSpeak(RT)

- Provide soft real-time guarantees.
  - Commits to and executes priority-maximal set of tasks with user-defined confidence level.
- AgentSpeak(RT) is an extension of AgentSpeak(L).
- Implemented on Java.
- Five components: a Belief Base, a Set of Events, a Plan Library, an Intention Structure, an Interpreter.

# Beliefs and Goals

- *Beliefs* represent the state of the agent and its environment.
- Beliefs are represented as ground atomic formulas
- A *Belief base* contains current beliefs about the world

```
lecture-notes(agents101,myNotes)  
open(library)
```

- A goal is a state the agent wish to bring about or a query to be evaluated

```
!prepare-lecture(agents101)  
?lecture-notes(agents101,Y)
```

# Events

- Changes in the agent's beliefs or the acquisition of new achievement goals give rise to **events**.
- We distinguish between
  - *internal* events generated by an agent's program
  - *external* events originated by a user or another agent
- External events may optionally specify a **deadline** and a **priority**

```
+!prepare-lecture(agents101)[50,9];
```

```
+lunch-time[40,7];
```

# Plans

- **Plan includes:**

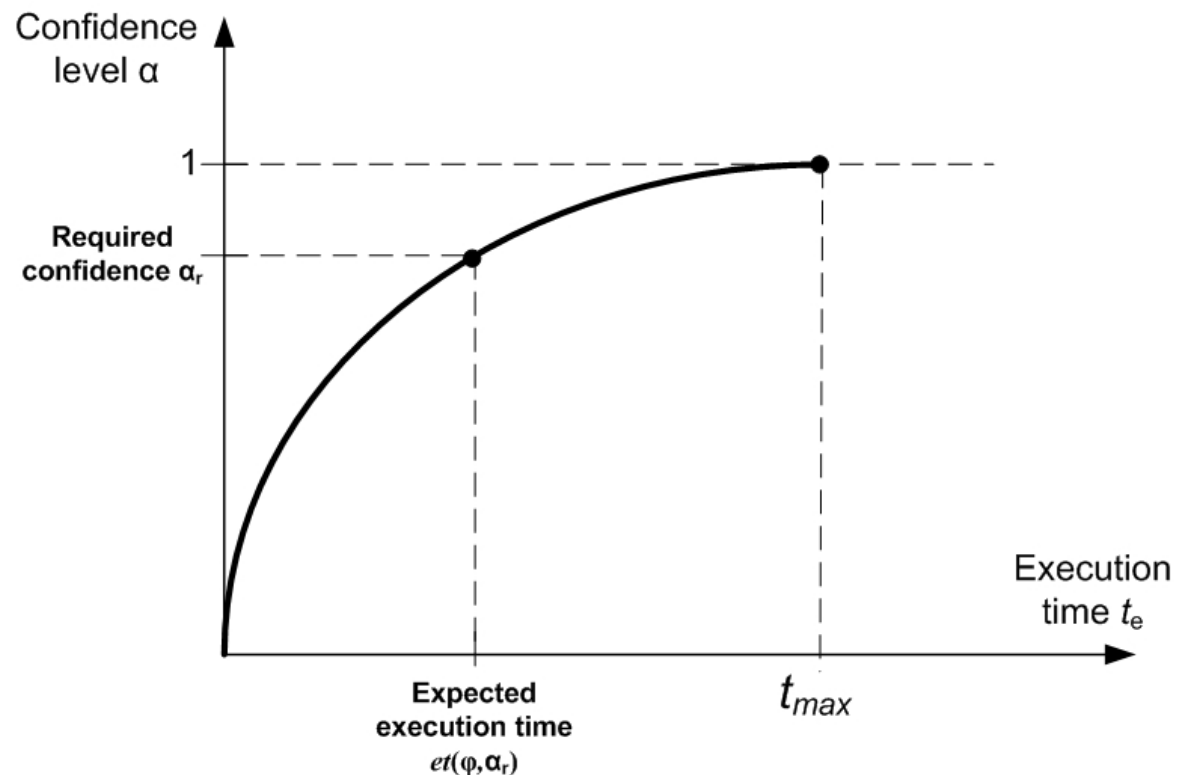
- Triggering event – used to select plan for possible execution
- Precondition – triggering condition
- Body – sequence of activities

```
+!borrow-book(B) : open(library) <-  
    goto(library);  
    !pickup(B).
```

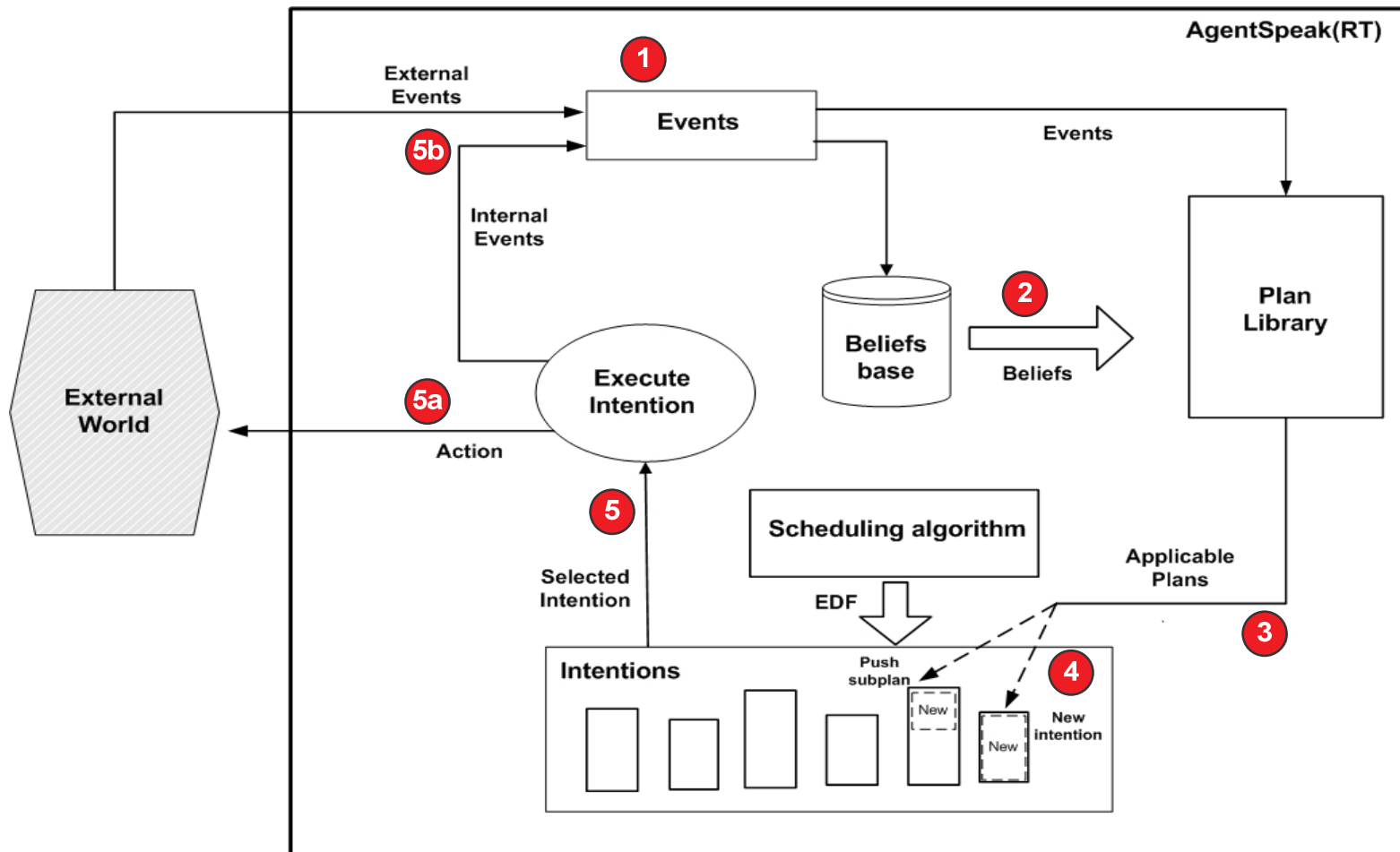
# Execution time profile

- The expected execution time for an action or plan  $\varphi$  at confidence level  $\alpha$  is given by

$$t_e = et(\varphi, \alpha_r)$$



# Interpreter



# Example(1)

## Goals:

```
+!prepare-lecture(agents101)[50,9];  
+!borrow-book(R&N)[30,2];
```

## Beliefs:

```
+lunch-time[40,7];  
+lecture-notes(agents101,myNotes)  
+open(library)
```

# Example(2)

## Plans :

```
+!prepare-lecture(X) : lecture-notes(X,Y)
    <- revise-lecture(Y).
+!borrow-book(B) : open(library) <-
    goto(library); !pickup(B).
+!pickup(X) : <-
    take(X).
+lunch-time : <-
    eat-sandwich.
```

# Example(3)

- $d_1=35min, d_2=12min, d_3=2min, d_4=20min$
- Intentions: Plan 1 (50,9), Plan 4 (40,7), Plan 2 (30,2).
- First and second intention can't be accomplished by deadlines
- Resulting schedule includes first and third intention.

# Multi-Thread Execution

- **Parallel execution of tasks:**
  - Enhanced Tasks Specifications
  - New scheduling algorithm for parallel execution of agent's tasks.
- **Task merging:**
  - merging intentions which have common parts, to allow parallel execution of an agent's tasks on the single processor.

# Conclusion

- **Applications:**
  - Robot control systems, traffic control systems, finance systems
  - Simulation of problems where data has to be processed in real-time.
- **Contribution:**
  - Analysis of the meaning of RT guarantees for AI systems.
  - Propose a new agent architecture AgentSpeak(RT), which provides a simple, predictable framework for agent developers.
  - Propose several directions for a future work.

# Questions???

Thank you