



INTENTION RECOGNITION TOOL

Georgios Alaveras

Christos Christodoulou

Luciano Del Corro

David Landsberg

Panayiotis Papakyriakou

Shafeen Tejani

Supervisor: Dr. Fariba Sadri

OUTLINE

- ◆ Introduction
- ◆ Specifications
- ◆ Motivating Example
- ◆ System Design
- ◆ Methodology
- ◆ Implementation
- ◆ Demonstration



2

INTRODUCTION

◆ What is **Intention Recognition**?

- ▶ Predicting an agent's intentions by analysing the actions performed.

◆ What are the **Potential Applications**?

- ▶ Ambient intelligence & assisted living
- ▶ Intrusion & terrorism detection



INTRODUCTION

◆ Probabilistic approaches

- ▶ Bayesian Networks
- ▶ Markov Models

◆ Logic-Based approaches

- ▶ Abductive
- ▶ Causal Theories



INTRODUCTION

◆ Probabilistic approaches

- ▶ Bayesian Networks
- ▶ Markov Models

◆ Logic-Based approaches

- ▶ Abductive
- ▶ Causal Theories

INTRODUCTION

- ◆ 3 main **components** of an intention recognition system
 - ▶ Set of intentions to choose from
 - ▶ Representation of actions and effects
 - ▶ Sequence of observed actions

INTRODUCTION

- ◆ 3 main **components** of an intention recognition system
 - ▶ Set of intentions to choose from
 - ▶ **Representation of actions and effects**
 - ▶ Sequence of observed actions

Logical Formalism: **The Event Calculus**

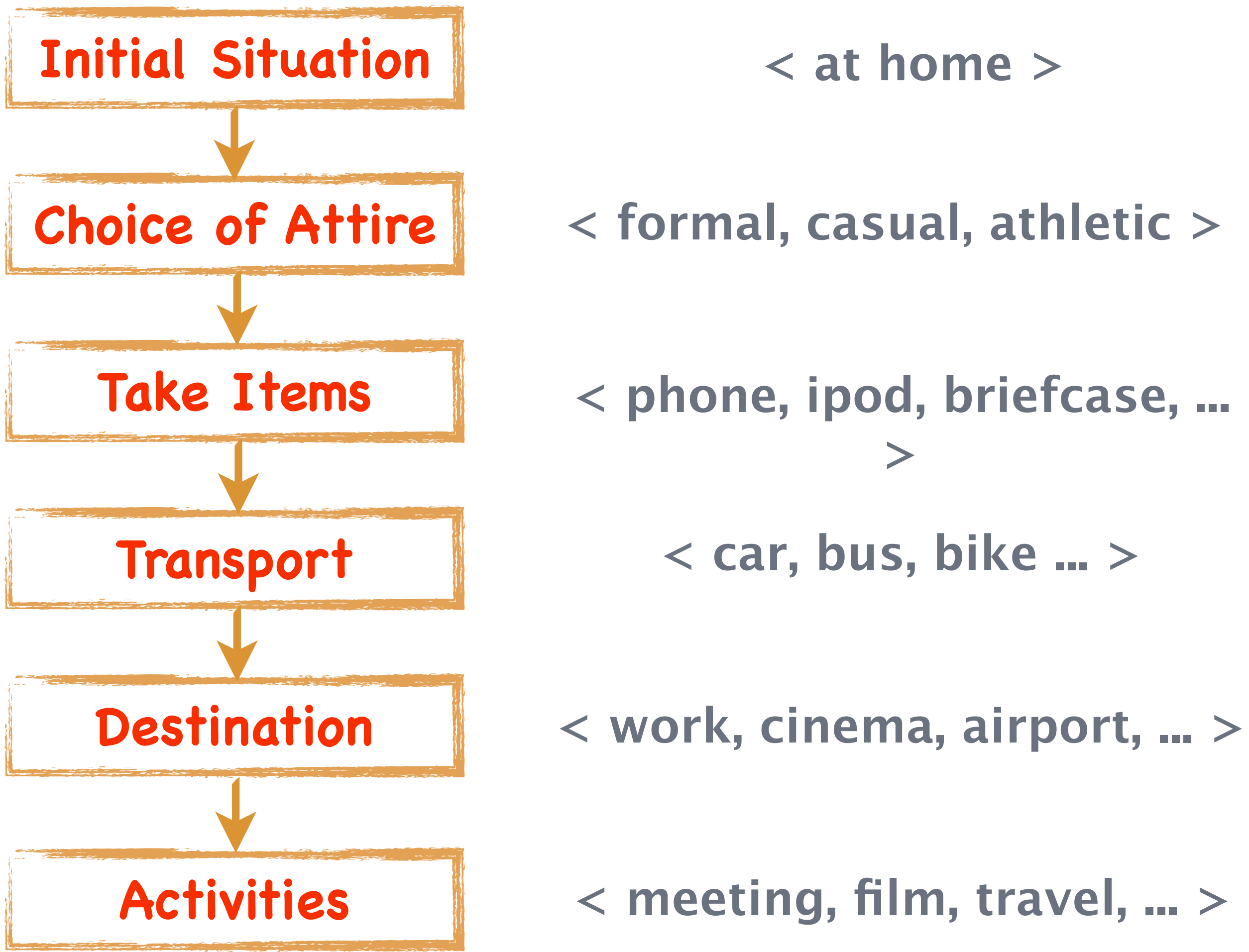
SPECIFICATIONS

- ◆ Intention Recognition using the **Event Calculus**
- ◆ Incorporate **time** and action **durations**
- ◆ Users can define their own **scenarios**
- ◆ User specifies **observed actions**, and intentions are predicted

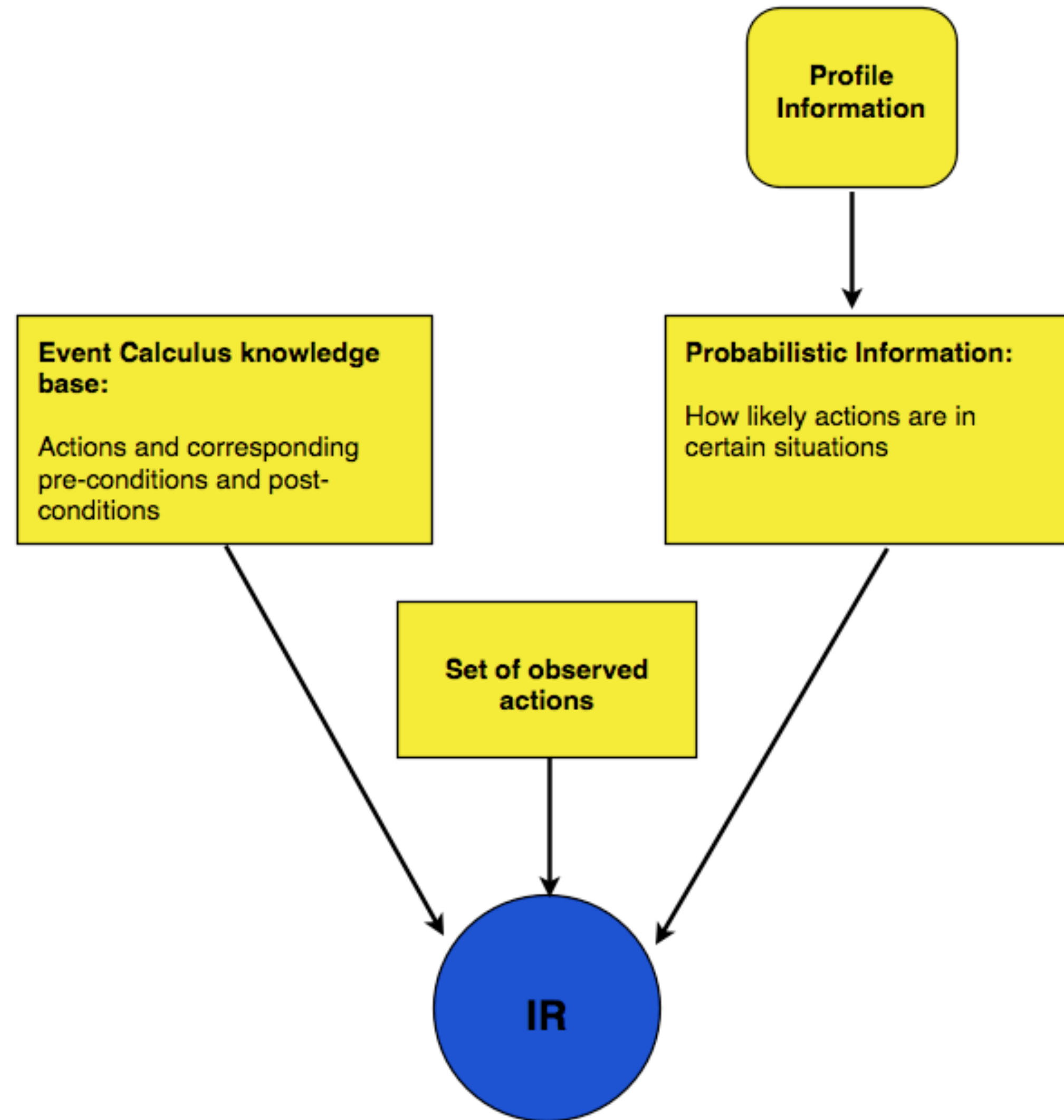
SPECIFICATIONS

- ◆ **Ordering preference** of intentions
- ◆ Add some **probabilistic** information
- ◆ Users can specify **profile** information
- ◆ **Graphical User Interface**
 - ▶ Visual display of intentions
 - ▶ Action paths

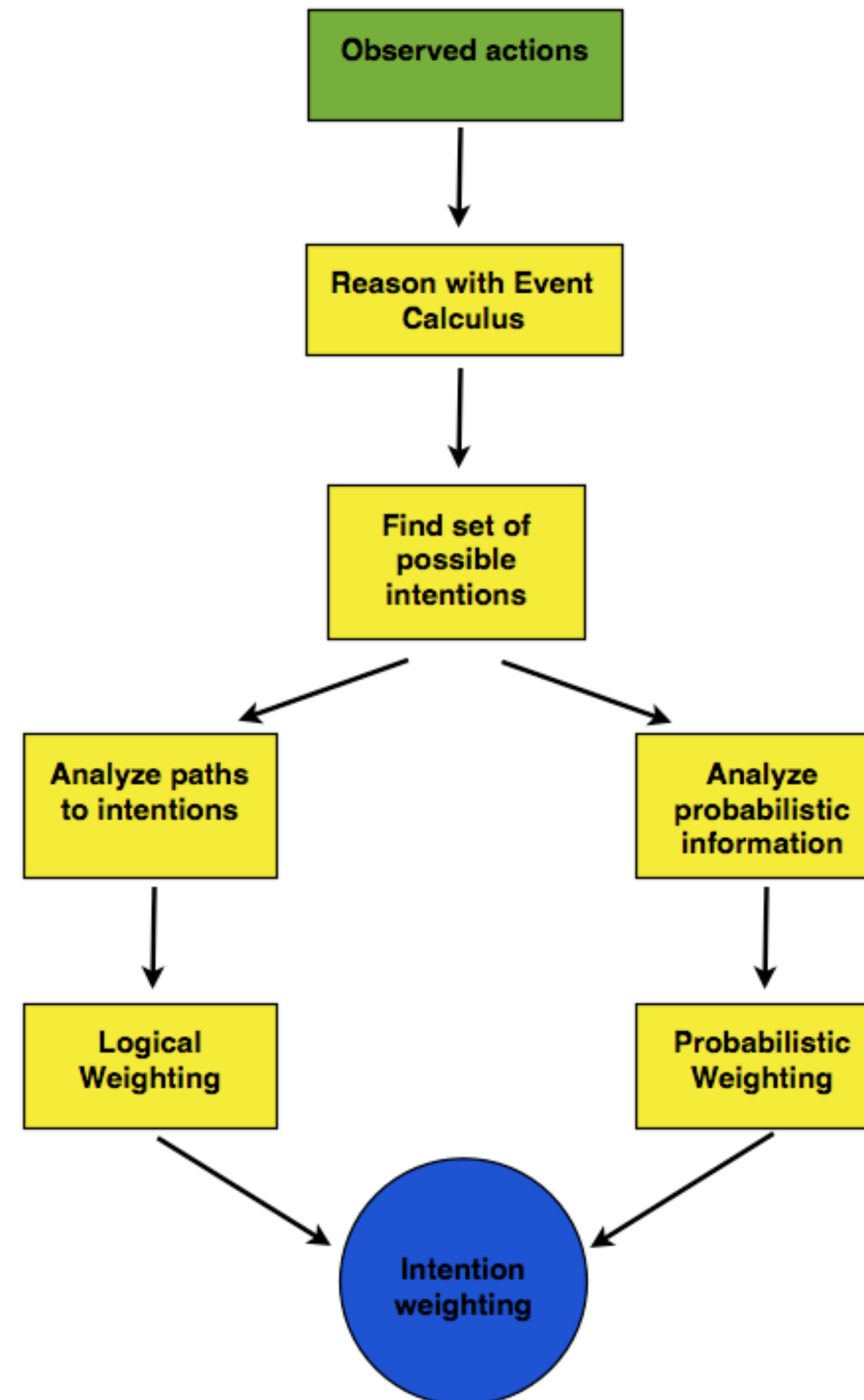
MOTIVATING EXAMPLE



SYSTEM DESIGN



SYSTEM DESIGN



METHODOLOGY – EVENT CALCULUS

◆ The Event Calculus **Predicates:**

Initiates(α, β, τ)

Terminates(α, β, τ)

Initially(β)

Precondition(α, β)

Happens(α, τ)

$\tau_1 < \tau_2$

METHODOLOGY – EVENT CALCULUS

◆ The Event Calculus **Axioms:**

$$\text{Holds}(\beta, \tau_2) \leftarrow \text{Happens}(\alpha, \tau_1) \wedge \text{Initiates}(\alpha, \beta, \tau_1) \\ \wedge \tau_1 < \tau_2 \wedge \neg \text{Clipped}(\tau_1, \beta, \tau_2)$$

$$\text{Clipped}(\tau_1, \beta, \tau_2) \leftarrow \exists \alpha, \tau (\text{Happens}(\alpha, \tau) \wedge \tau_1 < \tau < \tau_2 \\ \wedge \text{Terminates}(\alpha, \beta, \tau))$$

◆ **Ramifications:**

$$\text{Holds}(x, \tau) \leftarrow \text{Holds}(y, \tau) \wedge \text{Holds}(z, \tau) \wedge \dots$$



METHODOLOGY – SIMPLIFICATION

✦ **Simplification** of the Event Calculus:

- ▶ **Interface** for user-defined scenarios
- ▶ Easy to **understand**
- ▶ Retain most of the **expressibility** of the Event Calculus
- ▶ Users can define additional **probabilities**
- ▶ Simpler to **implement**

✦ Simplified **Notation** (Prolog Syntax):

`initially(Fluent).`

`initialTime(T).` $(0 \leq T < 24)$

`preconditions(Action,PreCs,Fluents,T).`

`postconditions(Action,PostCs,Fluents,T,D).`

`ramification(Fluent, FluentList).`

`likelihood(Action, Weight, Fluents, T).`

`diary(Action,Time,Fluents).`



METHODOLOGY – RUNNING EXAMPLE

✦ Example given in Simplified Notation

**initially(atHome)
initialTime(7).**

**preconditions(dressFormal,[atHome],_,_).
postconditions(dressFormal,[formallyDressed],__,1).**

**preconditions(goWork,[formallyDressed,inCar],_,T) ←
T > 9, T < 17.**

postconditions(goWork,[atWork,not(inCar)],__,1).

diary(goWork,9,_)



METHODOLOGY – INTENTION RECOGNITION

✦ How is the **intention recognition** carried out?

- ▶ Forward projection
- ▶ Maintain a **list of fluents** which are true
- ▶ Find all possible **action sequences**
 1. Find actions whose preconditions are met
 2. Assume one of these actions happens
 3. Update fluents with postconditions
 4. Repeat this procedure recursively
- ▶ Actions at the end of sequences are **intentions**



METHODOLOGY – WEIGHTING

◆ Convex combination of:

- ▶ Path **topology** -- W_1
- ▶ User-defined **weights** -- W_2

$$Weight = \lambda W_1 + (1 - \lambda) W_2$$

$$W_1 = N \left(\frac{1}{n_1} + \frac{1}{n_2} + \dots + \frac{1}{n_k} \right)$$

- ▶ If W_2 is defined then $\lambda = 0.5$
- ▶ Otherwise $\lambda = 1$



IMPLEMENTATION

- ◆ **Split into 3 main components**
- ◆ **Event Calculus Interface**
 - ▶ **How users define scenario files**
- ◆ **Interpreter**
 - ▶ **Carries out intention recognition**
- ◆ **GUI**
 - ▶ **Visual display of intentions and weights**



20

IMPLEMENTATION – GROUP WORK

◆ Event Calculus Interface

- ▶ Luciano
- ▶ Panayiotis

◆ Interpreter

- ▶ Shafeen
- ▶ Christos

◆ GUI

- ▶ Georgios
- ▶ David

IMPLEMENTATION

- ◆ **Interpreter implemented in Prolog**
 - ▶ **Easy to carry out reasoning with logic programs**
- ◆ **GUI implemented in Java using JASPER**
 - ▶ **Created an object in Java to handle all Prolog queries and parse information**
 - ▶ **SWING library used for graphics**

IMPLEMENTATION – FUNCTIONALITY

- ◆ Users can upload a scenario file
 - ▶ Running example given as default

- ◆ At each stage
 1. Select next 'observed' action and time
 2. Click update
 3. Intentions appear as lights
 4. Click on an intention to see action paths
 5. Repeat or Stop if no more actions possible



IMPLEMENTATION – FUNCTIONALITY

The screenshot displays the 'Intention Recognition Tool' interface. At the top, it shows 'Observed Actions: dressAthletic (08:00 am)'. The main area contains a grid of 12 circular nodes, each representing an action with a percentage value. The nodes are arranged in two rows of six. The top row nodes are: 'football 0.1%', 'takeBike 14.5%', 'takeCar 14.5%', 'takePhone 6.5%', 'takeBackpack 6.5%', and 'walk 14.5%'. The bottom row nodes are: 'havePicnic 0.1%', 'takeBackpack 6.5%', 'takeBus 14.5%', 'takeLuggage 6.5%', 'takeBriefcase 6.5%', and 'takelpod 6.5%'. To the right of the grid is a control panel with a dropdown menu for 'select observed action and time:' showing 'takePhone' selected and '09:00 am' in the time field. Below the dropdown is a list of actions: 'takePhone', 'takeBackpack', 'takelpod', 'takeLuggage', 'takeBriefcase', 'takeBus', 'takeCar', and 'takeBike'. A slider labeled 'depth:' is positioned below the list, with markers at 1, 2, and 3. At the bottom of the interface, it shows 'Time: 09:00 am' and 'Fluents: isWeekday, atHome, athleticallyDressed, dressed'. An inset window titled 'Paths for Intention: takeCar' is overlaid on the left side, listing several action sequences such as 'takePhone -> takeBackpack -> takeCar'.

Observed Actions: dressAthletic (08:00 am)

Action	Percentage
football	0.1%
takeBike	14.5%
takeCar	14.5%
takePhone	6.5%
takeBackpack	6.5%
walk	14.5%
havePicnic	0.1%
takeBackpack	6.5%
takeBus	14.5%
takeLuggage	6.5%
takeBriefcase	6.5%
takelpod	6.5%

Fluents: isWeekday, atHome, athleticallyDressed, dressed

Time: 09:00 am

Paths for Intention: takeCar

- takePhone -> takeBackpack -> takeCar
- takePhone -> takelpod -> takeCar
- takePhone -> takeLuggage -> takeCar
- takePhone -> takeBriefcase -> takeCar
- takePhone -> takeCar
- takeBackpack -> takePhone -> takeCar
- takeBackpack -> takelpod -> takeCar
- takeBackpack -> takeLuggage -> takeCar
- takeBackpack -> takeBriefcase -> takeCar

NOW FOR THE DEMONSTRATION

25



THANK YOU!

ANY QUESTIONS?