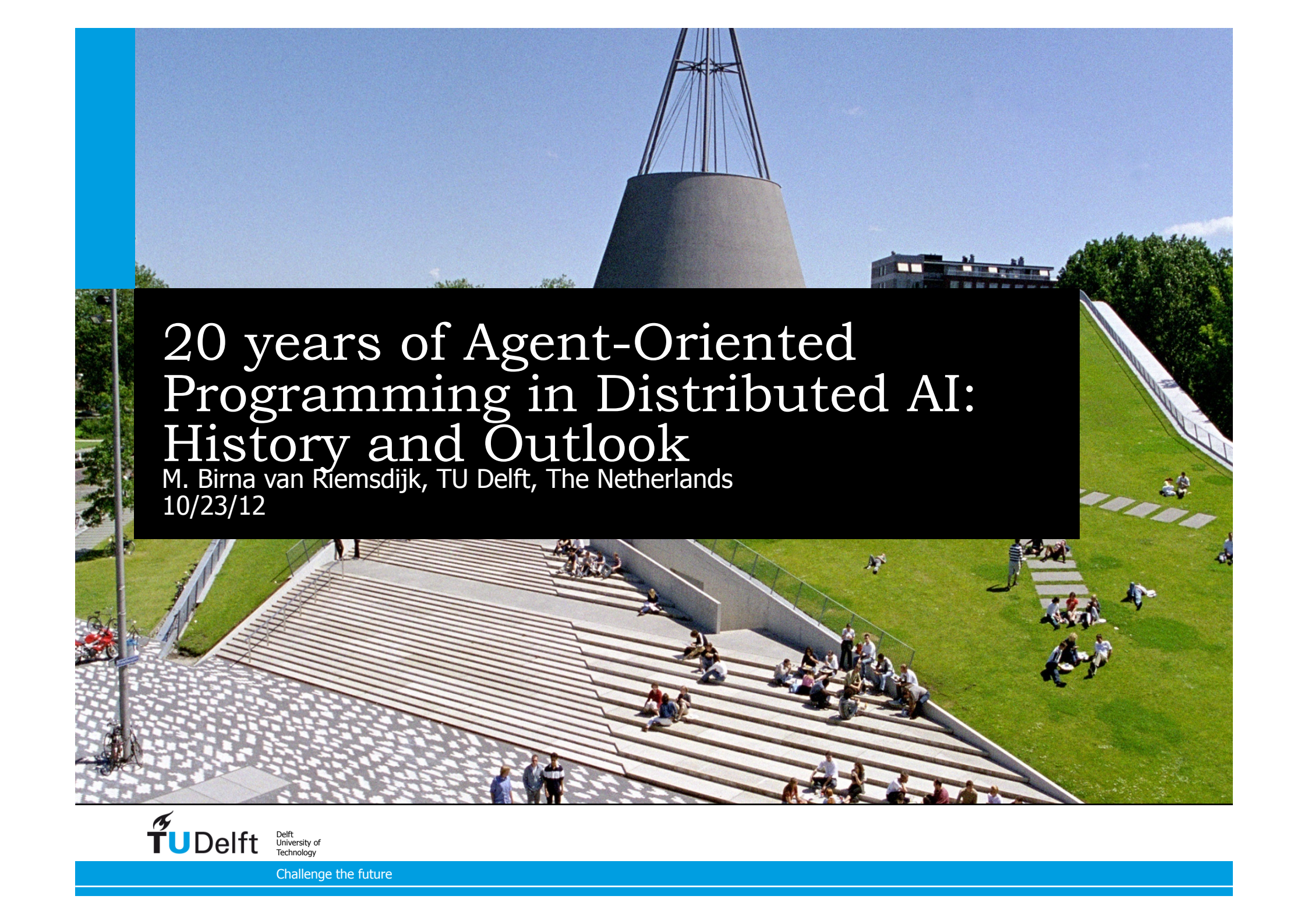




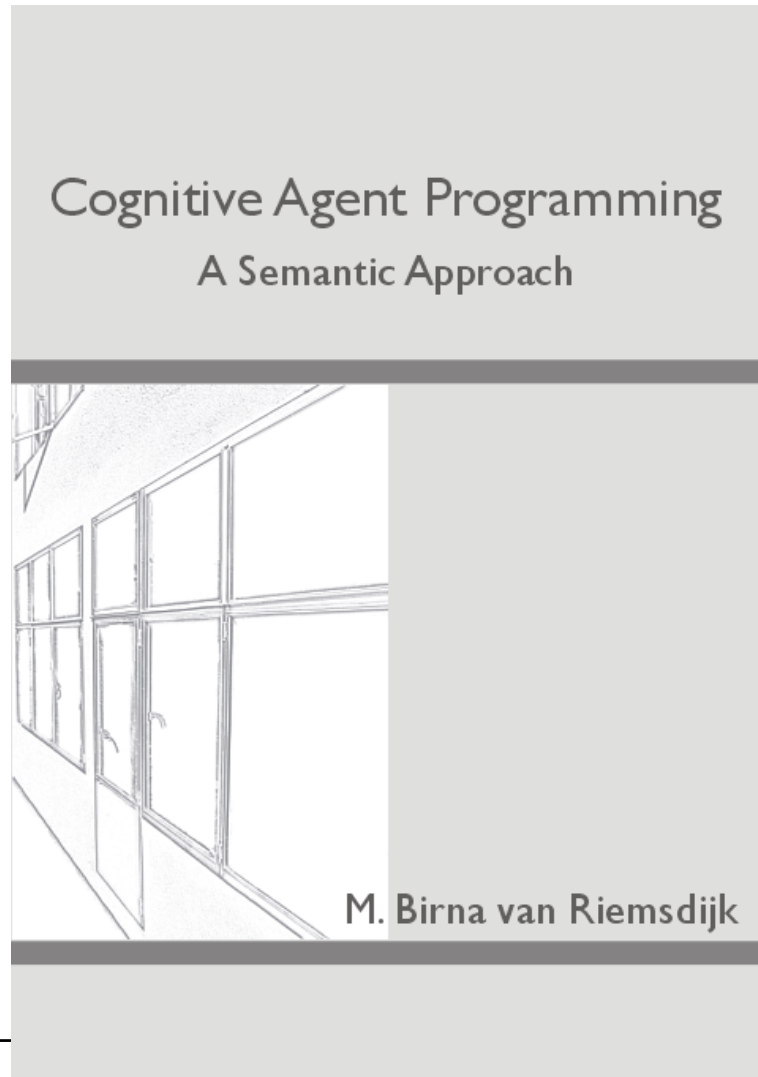
20 years of Agent-Oriented Programming in Distributed AI: History and Outlook

M. Birna van Riemsdijk, TU Delft, The Netherlands
10/23/12

1.

My Background

PhD 2002-2006



Utrecht University
Intelligent Systems

John-Jules Meyer
Frank de Boer
Mehdi Dastani

Postdoc 2006-2008

Ludwig Maximilians Universitaet Muenchen
Programming & Software Engineering

prof. dr. Martin Wirsing



Assistant Professor TU Delft 2008-...

Interactive Intelligence (II) section



Agent programming@II: theory & practice

- GOAL agent programming language (Hindriks)
 - Formal semantics, verification logic, model checking
 - Empirical software engineering for agent programming
 - Organization-aware agents & shared mental models



2.

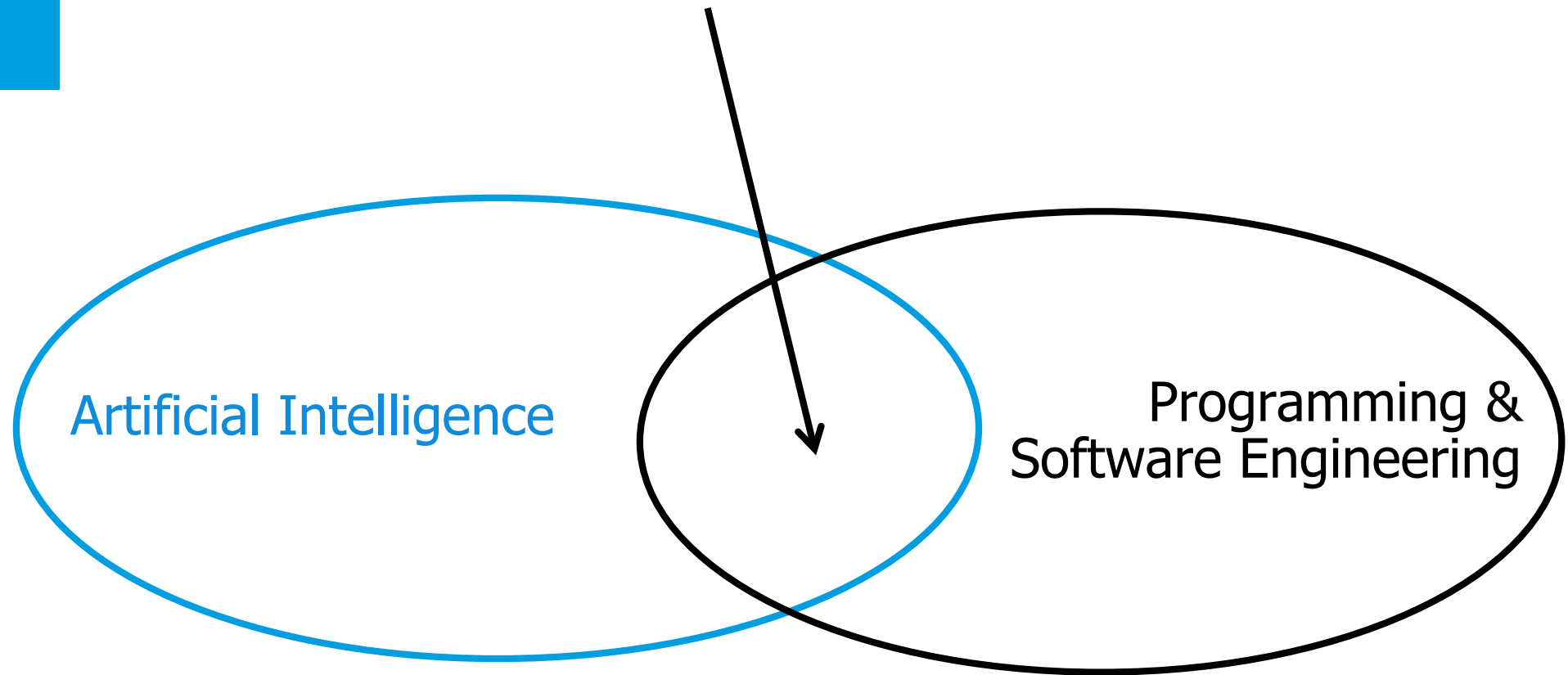
Introduction

What is an agent?

Ehm...

An agent is an **encapsulated** computer system that is **situated** in some environment and that is capable of **flexibel autonomous action** in that environment in order to meet its **design objectives**. (Jennings, 1999)

What is agent-oriented programming?



Agent-oriented programming

- Which **concepts** to use for programming autonomous agents and multi-agent systems?
- How to translate these concepts to **language constructs and abstractions**?
- How to **use** these languages?



The Intentional Stance (1987)

Daniell C. Dennett, Philosopher

Predicting & explaining the behavior of an object

- Physical stance: physics & chemistry
- Design stance: biology & engineering
- Intentional stance: software & minds



The Intentional Stance (1987)

Daniell C. Dennett, Philosopher

Here is how it works: first you decide to treat the object whose behavior is to be predicted as a **rational agent**; then you figure out what **beliefs** that agent ought to have, given its place in the world and its purpose. Then you figure out what **desires** it ought to have, on the same considerations, and finally you **predict** that this rational agent will **act to further its goals in the light of its beliefs**. A little practical reasoning from the chosen set of beliefs and desires will in most instances yield a decision about what the agent ought to do; that is what you predict the agent will do.”

=> seeing agents as intentional beings, acting according to beliefs and desires

“A little practical reasoning...”



What's the missing link?



Intention!

Michael E. Bratman, Philosopher (1987)

Belief-Desire-Intention (BDI) Logics

- Intention is Choice with Commitment –
Cohen and Levesque (1990)

2006 Winner of IFAAMAS Influential Paper Award

- Modeling Rational Agents within a BDI-Architecture –
Rao and Georgeff (1991)

2007 Winner of IFAAMAS Influential Paper Award



What does this have to do with
programming???

?

Artificial Intelligence 60 (1993) 51–92
Elsevier

ARTINT 931



Agent-oriented programming

Yoav Shoham

*Robotics Laboratory, Computer Science Department, Stanford University, Stanford,
CA 94305, USA*

Received June 1991

Revised February 1992

AGENT-0



(IF mntlcond action)

(IF (B (t' (employee smith acme)))
 (INFORM t a (t' (employee smith acme))))).

3.

The First Decade

The First Decade: Agent Programming $\approx$ Programming with Mental Models

1990, Intention logic

1991, BDI logic

The Concepts

1993, AGENT-0

1994, GOLOG

1996, AgentSpeak(L)

1999, 3APL

The Languages

AgentSpeak(L) (Rao, 1996)

```
triggering_event ":" context "<-" body "."
```

```
+location(waste,X):location(robot,X) &  
location(bin,Y)
```

Beliefs

```
<- pick(waste);
```

```
!location(robot,Y);
```

Goal (trigger)

```
drop(waste). Goal (adopt)
```

```
+!location(robot,X):location(robot,Y) &  
 (not (X = Y)) &  
 adjacent(Y,Z) &  
 (not (location(car, Z)))  
 <- move(Y,Z);  
 +!location(robot,X).
```



Execution Cycle of AgentSpeak(L)

- Process percepts & messages: generate events, update beliefs
- Select event to handle
- Select applicable plan for event
- Generate intention
- Execute one step of an intention

GOAL Message Passing and Handling

- Sending a message

```
send(A, Proposition)
```

- Processing a received message

```
if bel(received(A, Proposition))  
then insert(Proposition) + delete(received(A, Proposition))
```

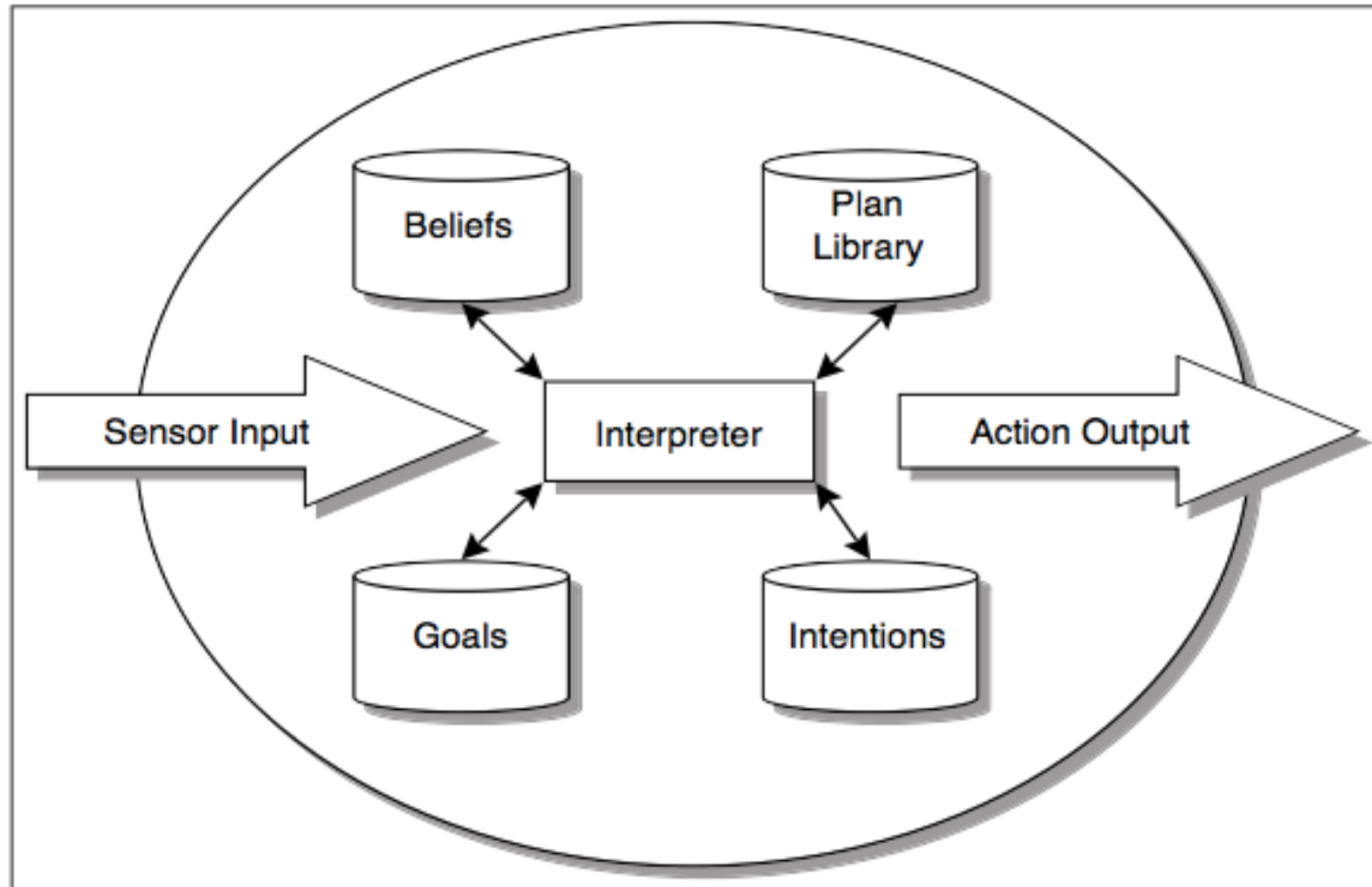
From: An Empirical Study of Patterns in Agent Programs. Koen V. Hindriks, M. Birna van Riemsdijk, and Catholijn M. Jonker, PRIMA 2010.

Agent Communication: Social Commitments (Singh, 1996)

$C(\textit{Barbara}, \textit{Alice}, \textit{paid}, \textit{deliverPhone})$

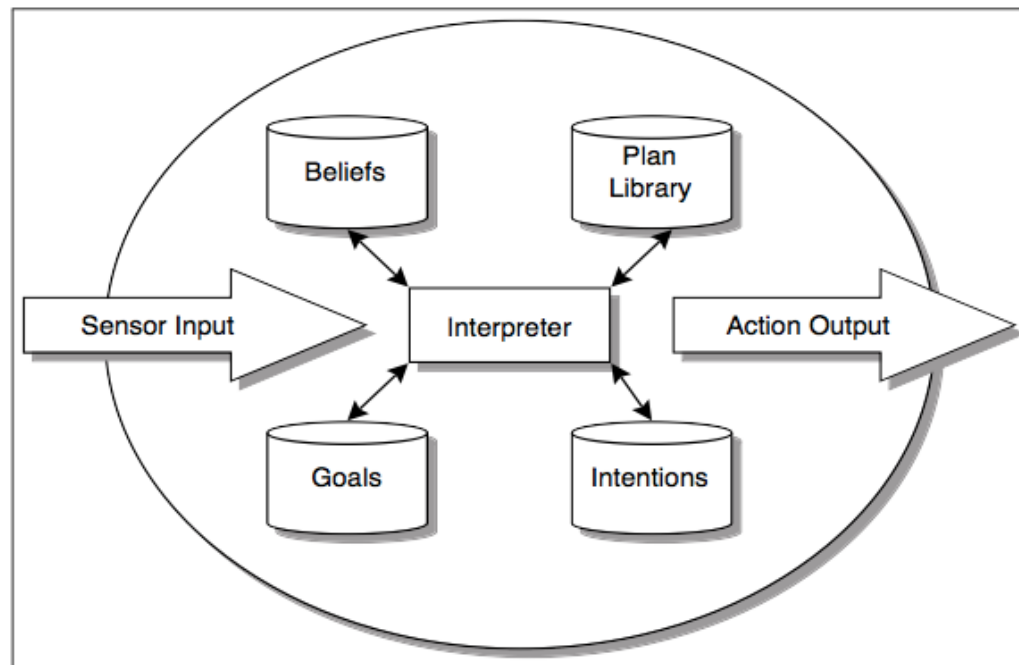
Barbara has committed to Alice that if the phone has been paid, it will be delivered

2001 - End of the first decade: where are we?



2001 - End of the first decade: are we there yet?

An agent is an **encapsulated** computer system that is **situated** in some environment and that is capable of **flexibel autonomous action** in that environment in order to meet its **design objectives**. (Jennings, 1999)



2001 - End of the first decade: are we there yet?

- which **knowledge representation** language to use?
- how **flexible** are these agents, really?
- where is the **environment**?
- how to **coordinate** a collection of autonomous agents?
- how to **specify and verify** multi-agent systems?
- how to **use** and **support use** of the languages in practice?

3.

The Second Decade

Knowledge Representation (1)

```
agent PingAgent extends Agent {
```

- Jack: extension of Java
- 1999
- Busetta et al.

```
#handles event PingEvent;
```

```
#uses plan PingPlan;
```

```
#posts event PingEvent pev;
```

```
void ping (String other)
```

```
{
```

```
    send(other, pev.ping(1));
```

```
}
```

```
}
```

From: JACK Intelligent Agents - Components for
Intelligent Agents in Java. Paolo Busetta, Ralph
Rönnquist, Andrew Hodgson, and Andrew Lucas,
Agentlink 1999.

Knowledge Representation (2)

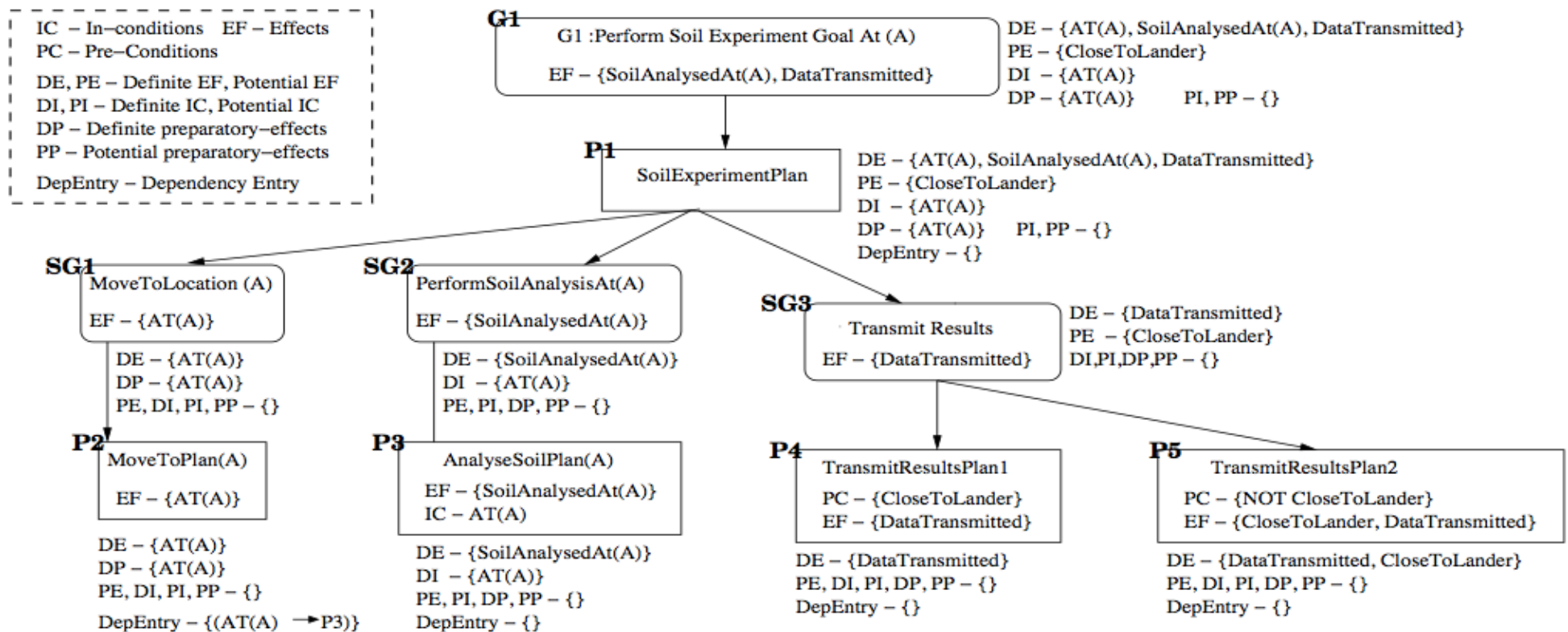
- Jadex: XML & Java
- 2003
- Braubach & Pokahr

```
<performgoal name="performpatrol" redo="true" exclude="never">  
  <contextcondition>  
    !$beliefbase.is_loading && !$beliefbase.daytime  
  </contextcondition>  
</performgoal>
```

From: Goal Representation for BDI Agent Systems. Lars Braubach, Alexander Pokahr, Daniel Moldt, and Winfried Lamersdorf, ProMAS 2004.

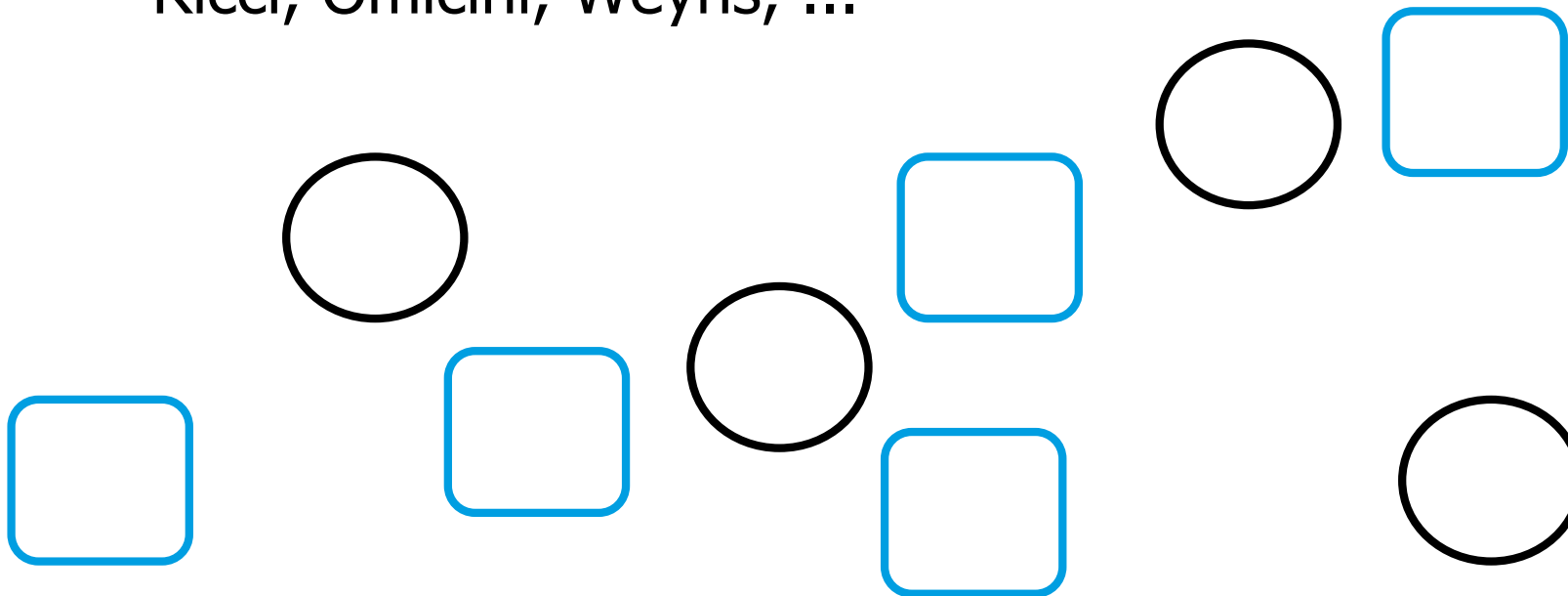
Flexibility

- Goal reasoning
- 2003
- Thangarajah, Winikoff, Dastani, van de Torre, Broersen, van Riemsdijk,...



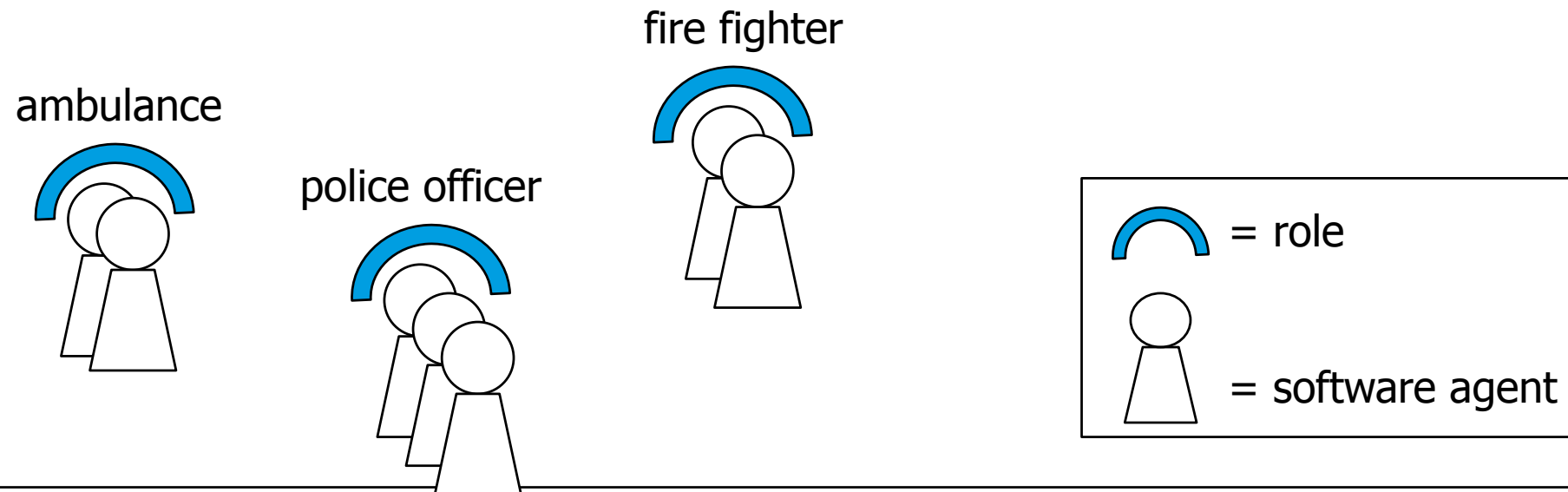
Environment

- **Environment as first class abstraction** for MAS engineering: encapsulate functionalities and services that support agent activities
- 2002
- Ricci, Omicini, Weyns, ...



Coordination

- Impose **organization** on a multi-agent system with norms, structure, task description, ...
- 2001
- Ferber, Esteva, Dignum, Boissier, Hubner, Noriega, ...



Specification & Verification

- Model checking and theorem proving
- 2002
- Bordini, Fisher, Wooldridge, Lomuscio, Meyer, Dastani, Hindriks, van Riemsdijk, ...

$$\Box(\neg(\text{Bel ag3 winner(ag3)}) \wedge \\ (\text{Des ag3 alliance(ag3, ag2)}) \Rightarrow \\ \Diamond(\text{Int ag3 alliance(ag3, ag2)}))$$

From: Model Checking AgentSpeak. Rafael H. Bordini, Michael Fisher, Carmen Pardavila, Michael Wooldridge, AAMAS 2003.

Platforms

Agent programming languages:

- Jack (commercial), Jadex, Jason/AgentSpeak, 2APL, GOAL, ...

Environment and organization:

- CArtAgO, MOISE, OperettA, AMELI, ...

4.

The Third Decade

Outlook (EMAS Dagstuhl Sem. 2012)

- Agent-oriented programming as a paradigm for development of **concurrent** and **distributed** systems
- **Tool support** for engineering MAS
- **Empirical research**, e.g., case studies
- **Integration** of agents, environment and organization (e.g., JaCaMo)
- **Agent reasoning** about organization, environment, other agents,
- Verification and **debugging**
- **Integration** with non-agent software

Websites of pictures

- http://massengale.typepad.com/venustas/2004/09/munich_new_york.html
- <http://favim.com/image/49365/>
- <http://www.crazymalc.co.nz/2007/Dec/24Dec/24Dec.htm>
- <http://305bcoaches.wordpress.com/2011/01/01/cross-roads/>
- <http://shc.stanford.edu/people/directory/michael-bratman>