

IG-JADE-PKSLib

An Agent-Based Framework for Advanced Web Service Composition and Provisioning

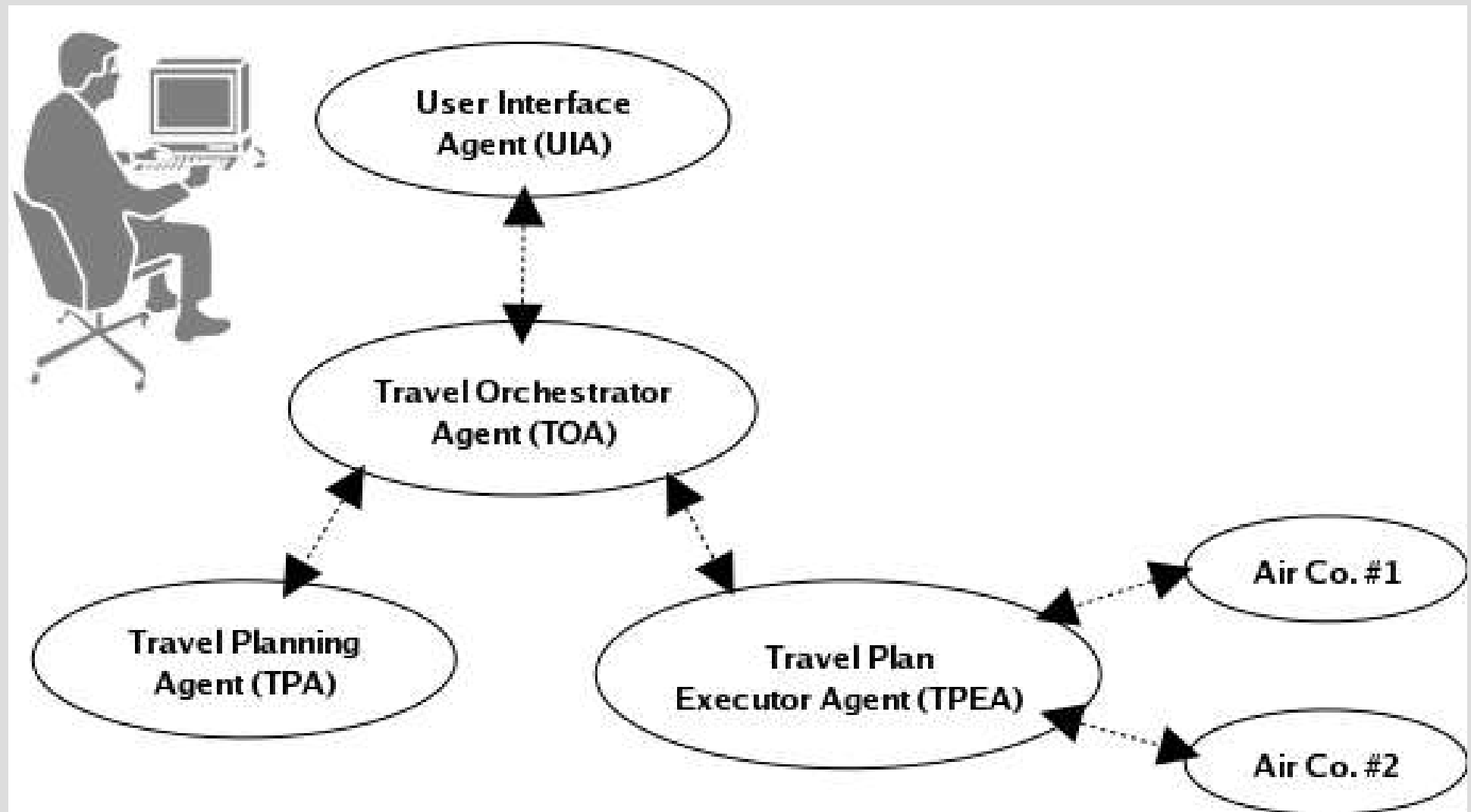
Erick Martínez & Yves Lespérance

Department of Computer Science
York University
Toronto, Canada

Motivation

- Next generation Web Services
 - Semantic Web
 - Reasoning / Planning
 - Automation
- Agent-oriented toolkit for advanced MAS / WSC and provisioning
- Previous results (knowledge-based planning)

Air Travel MAS Example (WSC & Provisioning)



IG-JADE-PKSLib Components

- **IndiGolog**: high-level model-based programming language based on the situation calculus and good for modeling dynamic domains.
- **JADE**: open source, Java-based, FIPA-compliant software framework for developing MAS
- **PKS**: efficient knowledge-based planning system, supports incomplete information and sensing actions

IndiGoLog (IG)

(De Giacomo & Levesque, 1999)

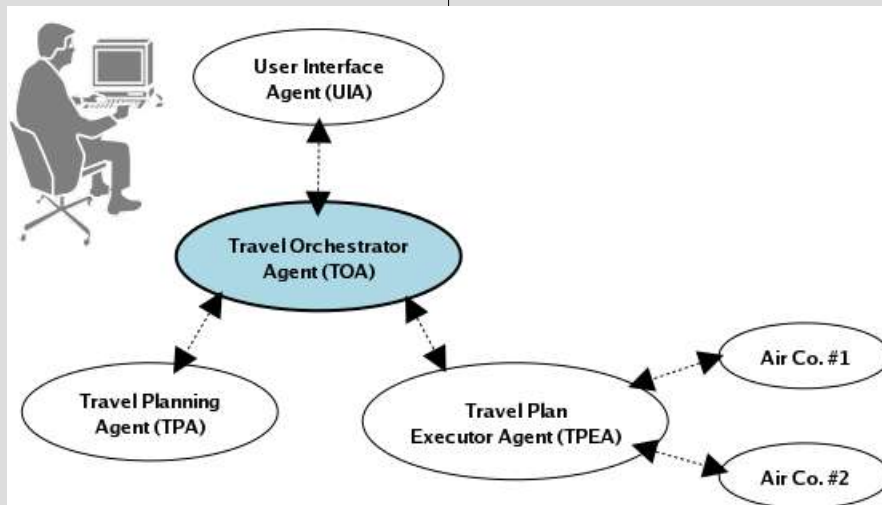
- Supports:
 - Reasoning about action (e.g., diagnosis)
 - Exogenous events handling,
 - Sensing,
 - Planning / Re-planning,
 - Plan execution,
 - Execution monitoring
- Limitations:
 - mainly intended for designing individual autonomous agents

IG Agent Specification

Has two parts:

- Declarative specification of the domain and its dynamics in the situation calculus:
 - Axioms describing the initial situation
 - Action precondition axioms
 - Successor state axioms
 - Sensed fluent axioms
 - Unique names axioms for the primitive actions
 - Some foundational, domain independent axioms
- Procedural description of the agent behaviour in the IG process language
 - Rich programming language, with support for non-determinism, concurrency, interrupts, search, etc.

IG Travel Orchestrator Agent Example



```
proc orchControl
```

```
 $\pi$  id, pid, xid .
```

```
?(agent_ID = id),
```

```
?(plan_ID = pid),
```

```
?(exec_ID = xid),
```

```
/* Three Concurrent Processes */
```

```
handleRequest(id) ||
```

```
handlePlanning(pid) ||
```

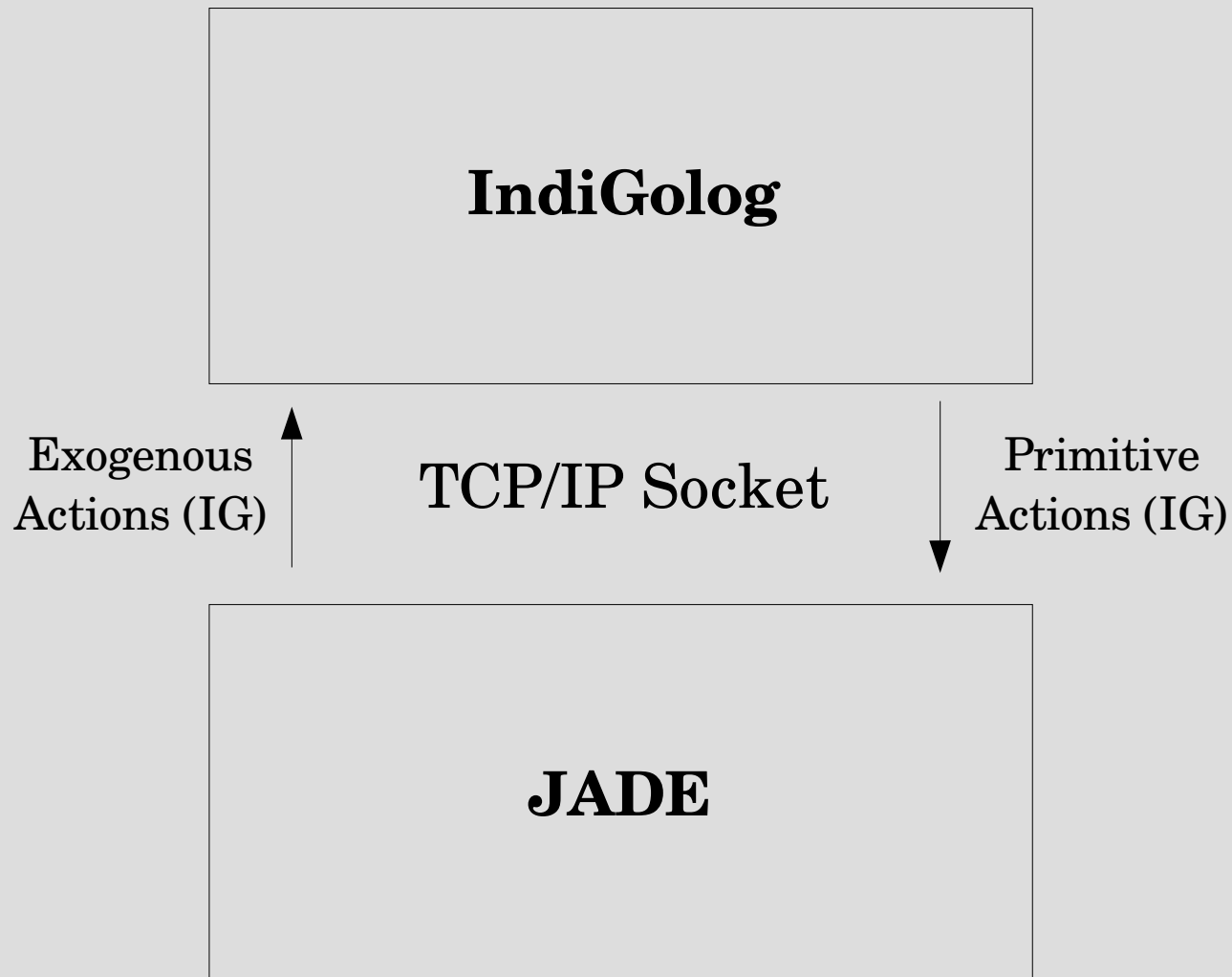
```
handleExecution(xid)
```

```
endProc
```

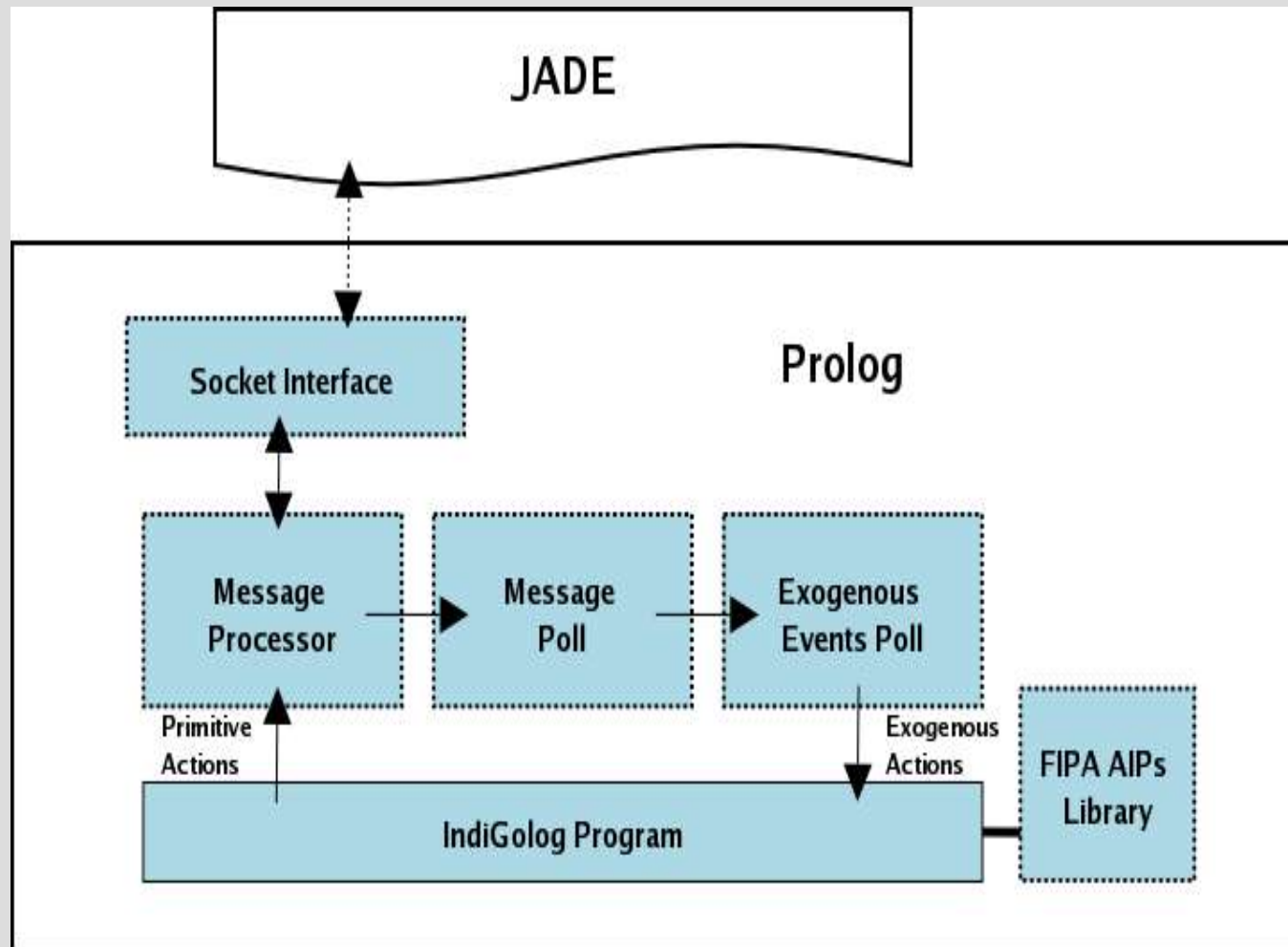
Java Agent DEvelopment Framework (JADE) (TILAB/AOT Labs)

- Supports:
 - FIPA-compliant agent architecture
 - Agent Management System
 - Directory Facilitator
 - Agent Communication Channel
 - Full implementation of FIPA communication model
 - Agent interaction protocols (AIPs),
 - Agent Communication Language (ACL),
 - SL content language,
 - Some types of ontologies, and
 - Transport protocols
- Limitations:
 - Lack of reasoning mechanisms for agents
 - Interface to JESS, but JESS has limitations with respect to reasoning about dynamic domains (e.g., WS)

IG-JADE Interface

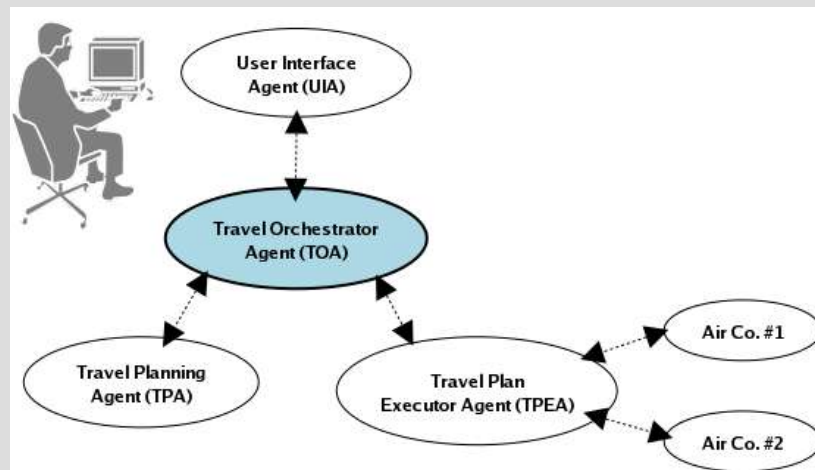


IG-JADE Interface (IG Arch.)



- **Original IG**
Limitation: intended for designing individual autonomous agents
- **Extended IG Features:**
 - FIPA ACL (message) handling
 - FIPA SL (content) handling
 - Support for FIPA Agent Interaction Protocols (AIPs)
- **Integration to MAS**
- **Asynchronous communication**
 - Proactive + reactive IG agents

IG Travel Orchestrator Agent Example (Revisited)



```

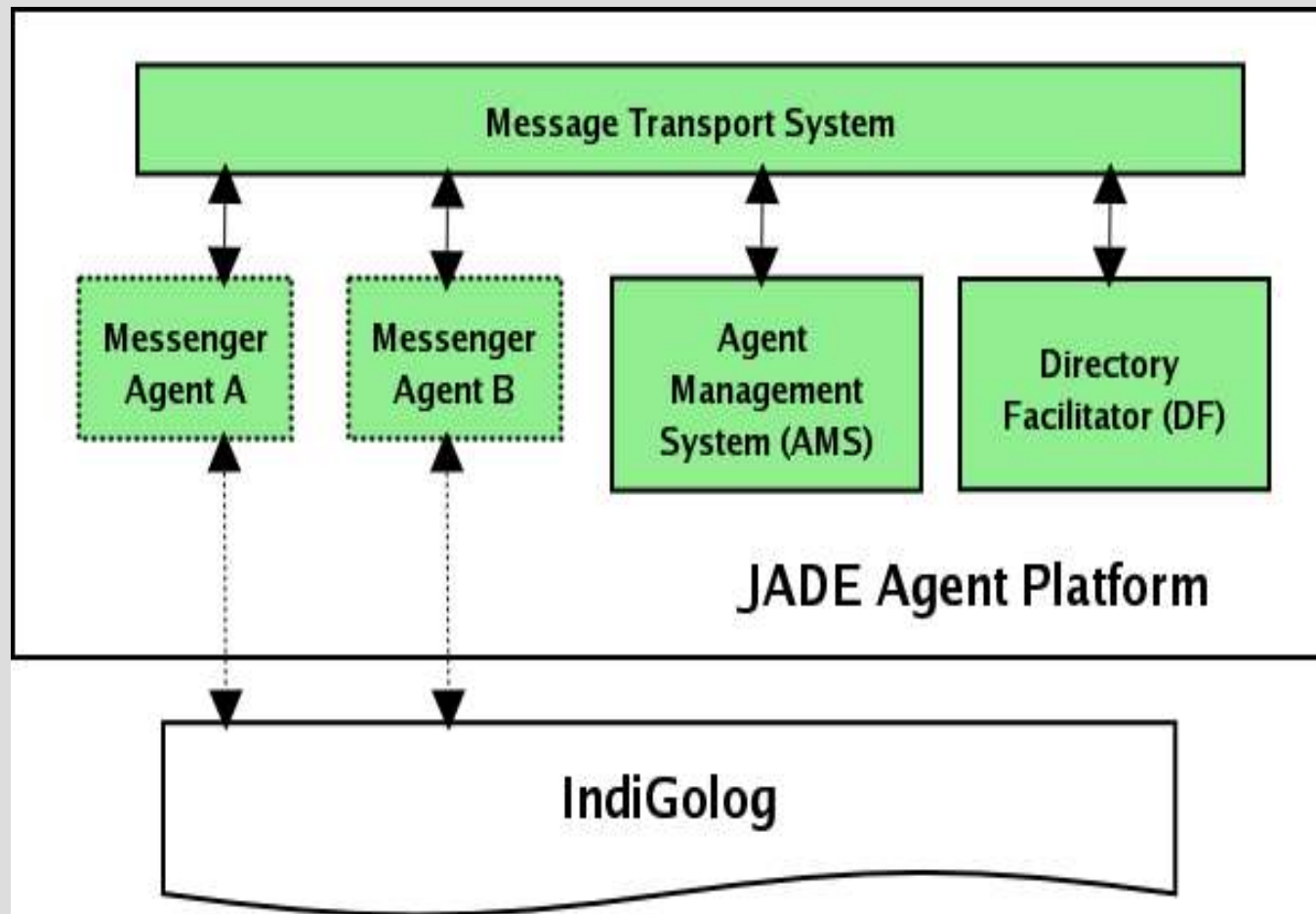
proc orchControl
   $\pi$  id, pid, xid .
    ?(agent_ID = id),
    ?(plan_ID = pid),
    ?(exec_ID = xid),
    handleRequest(id) ||
    handlePlanning(pid) ||
    handleExecution(xid)
endProc
  
```

```

proc handleRequest(id)
  /* FIPA AIP (Request-participant) */
  initRequestParticipantRole(id)
endProc

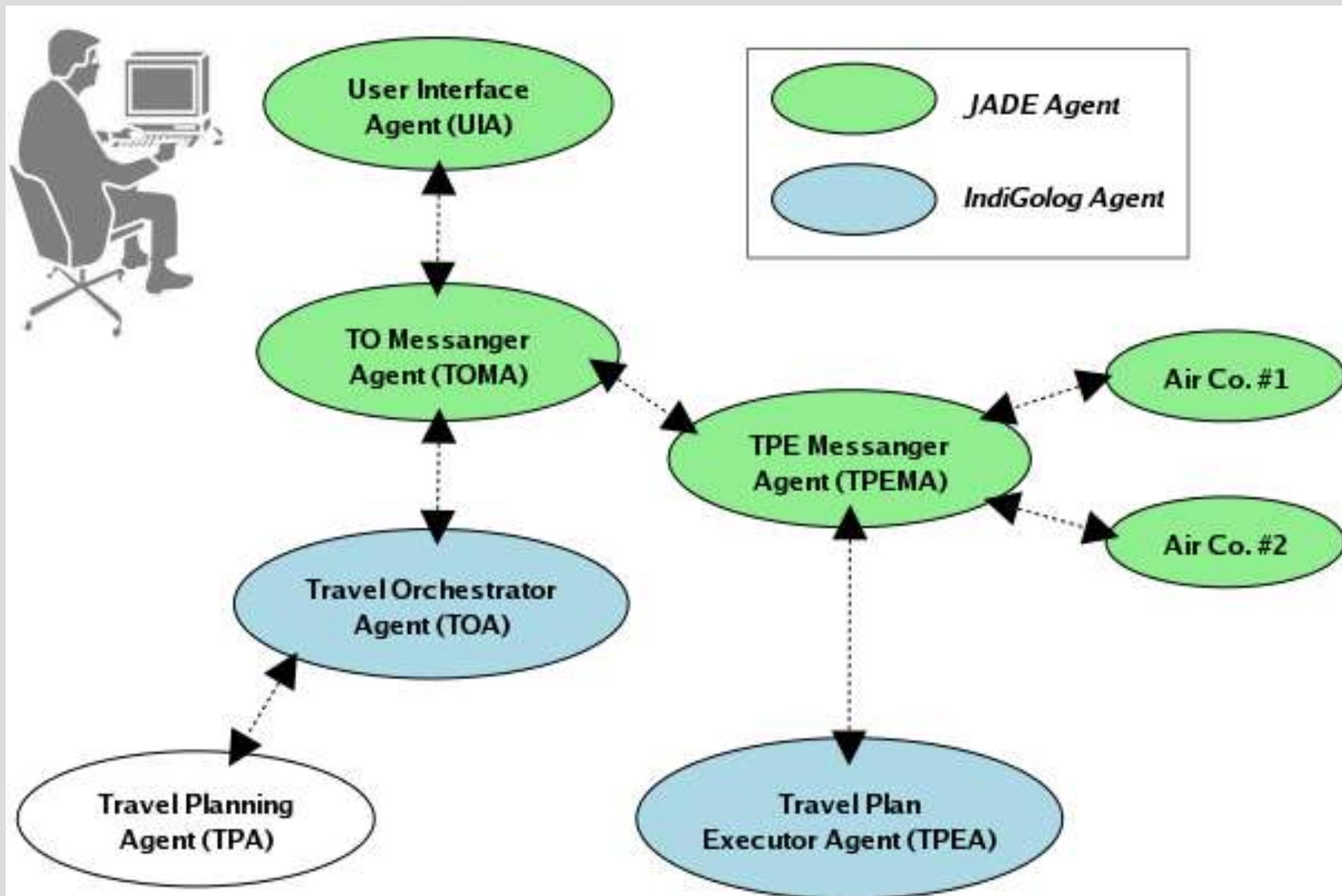
proc handlePlanning(pid)
  <cid: state_planning(pid, cid) = ready
  /* FIPA AIP (Request-initiator) */
  → initRequestInitiatorRole(id)>
  >>
  <true → noOp>
endProc
  
```

IG-JADE Interface (JADE Arch.)



- **Original JADE**
Limitation: lack of reasoning mechanisms for agents (dynamical domains)
- **Extended JADE Features:**
 - Socket interface to IG
- **Messenger is a full-fledged JADE agent**
 - ACL handling
 - Translator/mailbox
- **Extended capabilities to do reasoning on dynamical domains**

Air Travel MAS Example (Revisited)



Automated Web Service Composition as Planning

- **WSC Problem:** given a set of Web services and some user defined task or goal to be achieved, automatically find a composition of the available services to accomplish the task.
- **WSC as a Planning Problem:**
 - Predefined available services as the building blocks of a plan
 - Many WS actions involve sensing
 - Large search space, incomplete information in the initial state

Planning with Knowledge and Sensing (PKS)

- Supports:
 - Planning under conditions of incomplete knowledge
 - Sensing actions,
 - Parameterized conditional plans,
 - Planning at the knowledge level (without considering all the different ways the physical world can be configured),
 - Efficient, sound, forward chaining inference algorithm
- Limitations:
 - Only limited forms of incomplete knowledge can be represented
 - Inference algorithm is efficient but incomplete

WSC in PKS

(Martínez & Lespérance, 2004)

- PKS primitive actions correspond to WS
 - Knowledge-producing actions \leftrightarrow information gathering WS
 - Physical actions \leftrightarrow world-altering WS
- New WS becomes available \rightarrow add new primitive action to domain specification

PKS Spec. Example

(Air Travel Domain)

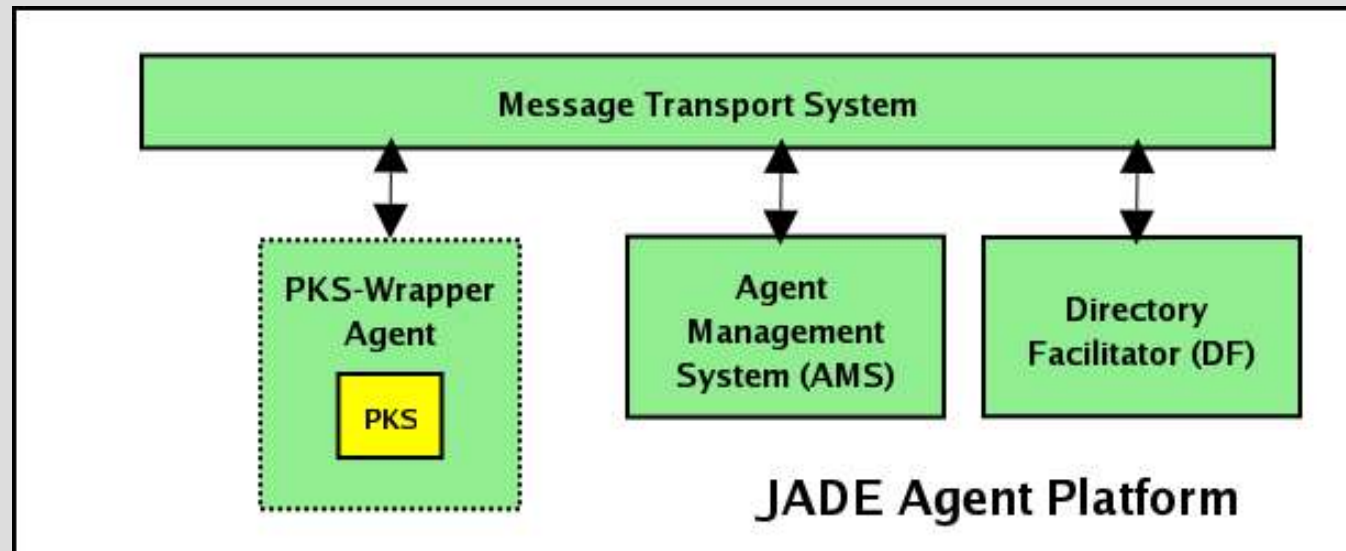
Action	Precondition	Effects
<i>findRFlight(x)</i>	$K(\text{airCo}(x))$ $K(\text{indFindRFlight}(x))$ $K(\text{desFindRFlight}(x))$	$\text{add}(Kw, \text{flightExists}(x))$ $\text{add}(Kf, \neg \text{indFindRFlight}(x))$
<i>bookFlight(x)</i>	$K(\text{airCo}(x))$ $K(\text{availFlight}(x))$ $K(\text{indBookFlight}(x))$ $K(\text{desBookFlight}(x))$	$\text{add}(Kf, \text{bookedFlight}(x))$ $\text{del}(Kf, \text{availFlight}(x))$ $\text{add}(Kf, \neg \text{indBookFlight}(x))$

Domain specific update rules (DSUR)

$$K(\text{airCo}(x)) \wedge \neg Kv(\text{flightNum}(x)) \wedge K(\text{flightExists}(x)) \Rightarrow \text{add}(Kv, \text{flightNum}(x))$$

1 explicit parameter: *company*

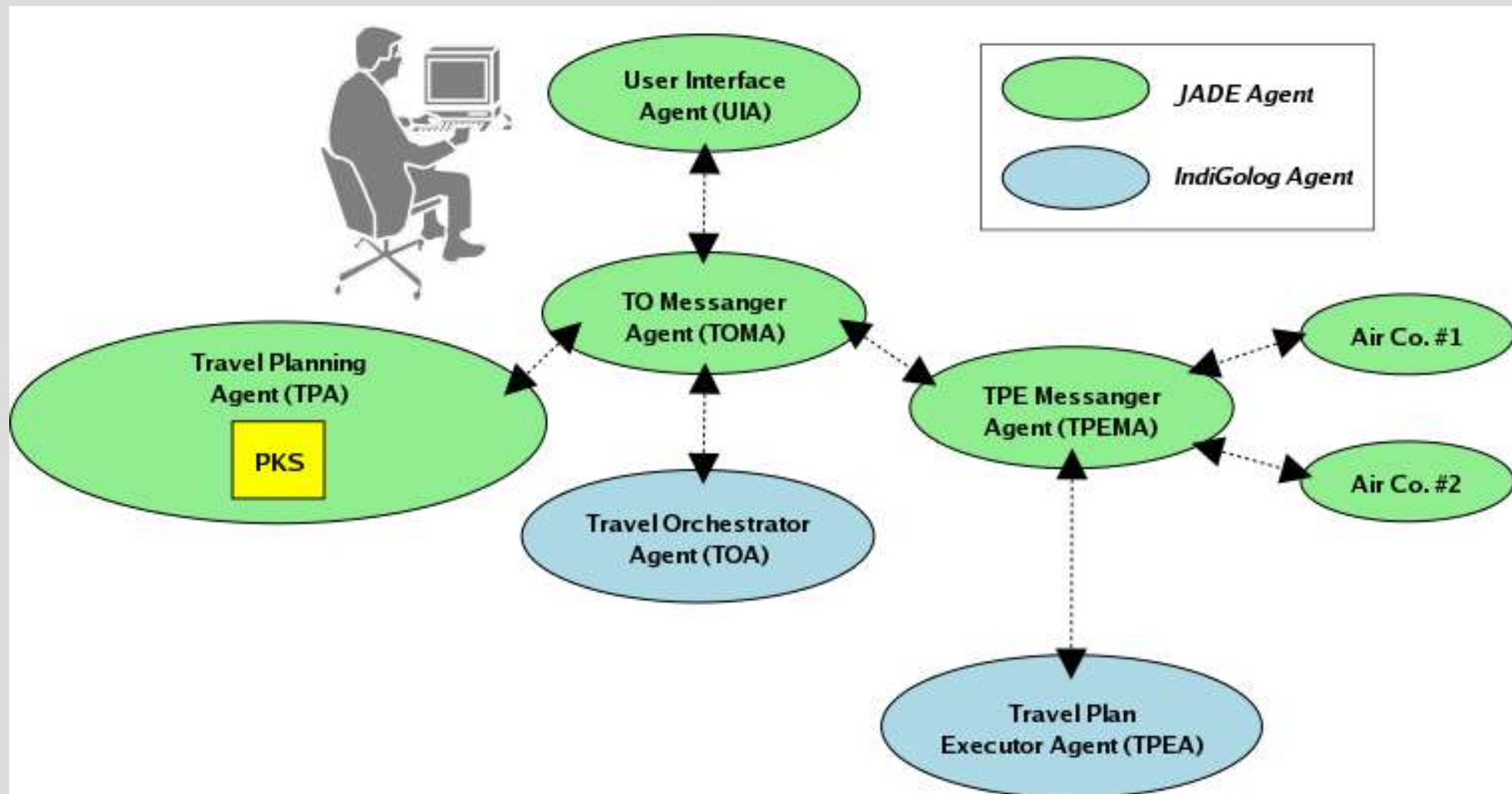
PKS-JADE Interface



- **Why PKS-JADE?**
 - Both object-oriented: JADE (Java), PKS (C++)
- **PKS-WA is a full-fledged JADE agent**
 - FIPA AIPs
 - FIPA ACL
 - FIPA SL
 - Ontology support (e.g., [pks-ontology](#))
- **Extended knowledge-based planning capabilities**

Air Travel MAS Example

(Final / *IG-JADE-PKSLib*)



Main Contributions

- Extended **IndiGolog**, added support for:
 - FIPA Agent Interaction Protocols (AIPs)
 - FIPA ACL (message) handling
 - FIPA SL (content) handling
 - Integration to MAS
- Asynchronous communication, proactive + reactive IG agent behaviours
- **IG-JADE-PKSLib**: agent-based infrastructure and toolkit, for developing inter-operable, intelligent MASs (e.g. advanced WS applications)
- Our toolkit provides *comprehensive support* for multiple features, such as: reasoning for dynamical domains, knowledge-based planning (incomplete knowledge, sensing), execution monitoring, AIPs, ontologies, etc.

Open Problems and Future Work

- Translation of OWL, DAML-S, etc. into PKS/IG specifications
- Representation of atomic services
- Customizing domain theory based on problem
 - Automatic generation (desc. goal + user pref.)
- **IG-PKS** interface (i.e., IG built-in KB planning)
 - IG passes planning calls to PKS (efficient)
 - Prolog-based reimplementations of PKS (more flexible, but less efficient)
- Experiments + case studies to validate performance and scalability of integrated framework [*Martínez & Lespérance, 2004*]
- Plan execution and contingency recovery

Thank You!