An Architecture for Universal Knowledge-based Agent (UKA)

Michal Čertický

June 19, 2009

Contributions

- Introduced a novel term - UKA (alternative approach to AOP providing wider usability of single implementation)
- Proposed solutions to problems of envir. and task universality
- Detailed architecture with needed algorithms providing a methodology for implementation

Note: UKA agent itself is composed of a lower-level **multi-agent system**, *which results in modularity and ease of modification.*

Definition of UKA

Definition

Universal Knowledge based agent (UKA) is an agent which:

1. **is usable in various environments**
2. **is usable for various tasks**
3. has knowledge base representing its knowledge about environment
4. enlarges its knowledge base by observation and reasoning
5. is capable of making plans to fulfill its goals
6. is autonomous

Environmental Universality

- Ability to perform in **various kinds of enviromnents** and with **various sets of sensors and effectors**, with modification of only a small part of program (hardware interface).
- All the **communication** with hardware (sensors and effectors) is **encapsulated** into so-called **hardware interface**, consisting of two parts of the architecture (sensory sub-agent and effector sub-agent).
- Rest of communication inside architecture is completely independent of environment (symbolic, logic-based).
- However, implementation of such hardware interface for complex environments is potentially a difficult task, since the translation of qualitative sensoric input into logic-based representation is non-trivial.

Solution by Convenient Knowledge Representation

- Communication inside UKA represented by **Extended Logic Programs** (ELP).
- All the observed world **individuals** are processed simply as **string constants** recieved as an output of hardware interface. **Reasoning algorithms don't consider any connection between those constants and any environmental entities.**
- FOL-style predicate logic syntax.
- Two negation operators ($\neg$, *not*) provide more expressive power, while keeping computational complexity bearable.
- Typical formalism for action-representation (STRIPS) needed to be modified for usage with incomplete knowledge.

Extended Logic Programs

- Best choice for resoning with **incomplete knowledge** because of **default negation**.
- "$\neg a$" expresses explicit knowledge of a not being true.
- "*not* a" expresses lack of knowledge about a being true.

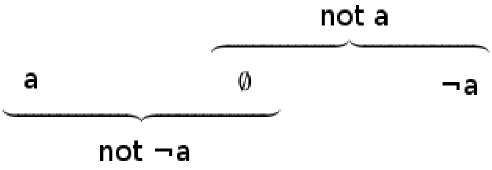


Figure: Truth values in ELP.

Extended Logic Programs - Example

- Example of observation expressed in ELP:

$$attribute(lightbulb01, on\_off\_state, on).$$

- Example of causal hypothesis expressed in ELP:

$$attribute(lightbulb01, on\_off\_state, on) \leftarrow action(X, press, button01).$$

IK-STRIPS

- Modification of classical STRIPS formalism for representing actions, making it usable with **incomplete knowledge and ELP representation**.
- Just like STRIPS, IK-STRIPS describes preconditions and effects of each action. However, there are **two sets of preconditions** - positive and negative. Action/effect is applicable if **each positive and no negative precondition is satisfied** in a current world model.

IK-STRIPS

- In addition to that, IK-STRIPS provides more intuitive action representation by allowing **preconditions not only for actions, but also for effects.** It also can process **variables in effect descriptions**, which are substituted by constants from current model. This way, one action can have different effects on different world models.
- Experimental planning engine using IK-STRIPS as an action-representation formlism is also presented at this conference.

Task Universality

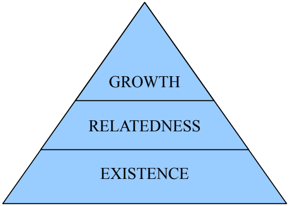
- Ability to perform any **practical task** which can be represented by ELP.

Practical Tasks and Actions

- **Goals/tasks** of UKA are represented by **incomplete descriptions of world** (non-empty set of ELP facts). Any such goal can be submitted to UKA agent, which then tries to **make a plan** leading it to a world state corresponding to that goal.
- If an agent has **actions** which potentially lead to that goal, it can achieve it. Otherwise it obviously can't.
- We call tasks like that (achievable by certain agent's actions) **"practical tasks"**.

Solution by ERG Motivational Model

- Psychological model of motivation, based on classical Maslow's hierarchy.
- 3 levels of needs, sorted by priority (Existence needs have highest priority).

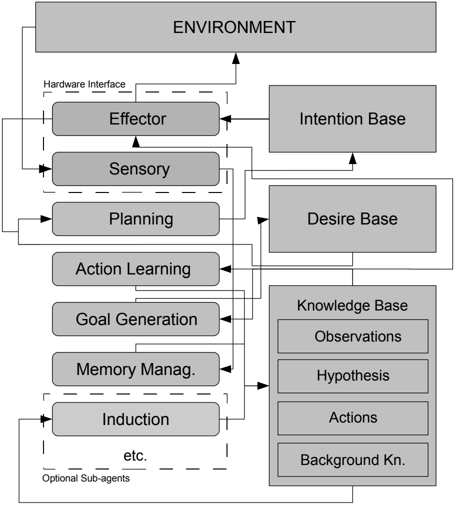


Solution by ERG Motivational Model

- ERG model adapted for artificial agents.
- **Existence needs** - ensuring agent's continuous existence (avoiding damage, recharging power supply...). Autonomy-related.
- **Relatedness needs** - carrying out tasks/goals submitted from outside (by its owner).
- **Growth needs** - learning and exploratory behaviour, validating its hypotheses...
- Goals representing existence and growth needs are **generated by agent** itself, while goals representing its relatedness needs are **submitted** from outside (tasks).

Minsky's Philosophy

- Marvin Minskys modular scheme - Society of Mind.
- Minsky tried to explain how mind works by splitting it into many simpler processes, which he called mental agents.
- Inspiration for designing UKA architecture as a **multi-agent system** composed of several **sub-agents**.
- Each sub-agent performs **single cognitive function**, which shouls according to Minsky lead to emergence of intelligent behaviour.
- **Complexity** of individual sub-agents is however kept **low**. Such modularity enables a programmer to modify an agent easily.


ENVIRONMENT
Hardware Interface
Effector
Sensory
Intention Base
Planning
Desire Base
Action Learning
Knowledge Base
Goal Generation
Observations
Memory Manag.
Hypothesis
Induction
Actions
etc.
Background Kn.
Optional Sub-agents