Belief - Desire - Intention (BDI) Model

BDI Introduction, Applications and Analyses

Massimo Innocentini

University of Hamburg
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics
Technical Aspects of Multimodal Systems

04. December 2017

Outline

1. Introduction
   BDI Scope
2. Implementations
   Why multiple implementations?
3. Applications
4. Case Scenario
   Possible approaches
   BDI Approach
5. Results
6. Comparison
7. Observations
8. Conclusion

BDI is a software programming paradigm used for implementing intelligent agents.

BDI stands for:

- Belief
- Desire
- Intention

The original principles were set by Michael Bratman during the 80s.

Definitions

BDI as whole can be represented by the following components:

- Belief = The knowledge of the world, `state of the world`.
- Desire = The objective to accomplish, `desired end state`.
- Intention = The course of actions currently under execution to achieve the desire of the agent.[1]
- Set of `plans` supplied at design time.

Reduce action decision time by eliminating inconsistent choices relative to the intention.[2]

[1] V. Mascardi, D. Demergasso, D. Ancona, (2005). Languages for Programming BDI-style Agents: an Overview.. 9-15.

[2] Georgeff M., Pell B., Pollack M., Tambe M., Wooldridge M. (1999) The Belief-Desire-Intention Model of Agency.

Execution Cycle

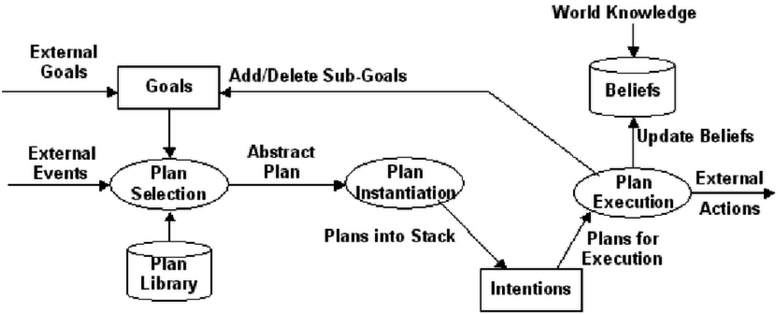

Fig. 1 BDI Execution Cycle [3]

[3] G. Jakobson, A. Corp, N. Parameswaran, J. Buford, L. Lewis, R. Pradeep (2006) Situation-Aware Multi-Agent System for Disaster Relief Operations Management.

BDI Scope

"Software agents (in particular, BDI agents) provide the essential components necessary to cope with the real world."

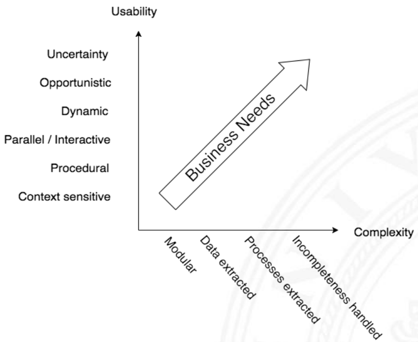

Fig. 2 Graph shows scope of BDI [4]

[4] Georgeff M., Pell B., Pollack M., Tambe M., Wooldridge M. (1999) The Belief-Desire-Intention Model of Agency.

Limitations

- Lack of learning competences.
- Lack of explicits architecture for multi-agents behaviour.
- Overthinking in certain scenarios.

Implementations

Different agent architectures:

- `Procedural Reasoning System (PRS)`
  Developed for embedded applications.
  - `distributed Multi-Agent Reasoning System (dMARS)`
    Evolution of PRS including multi-agent behaviour.
  - `JACK`
    Build for defence simulation.
- `AgentSpeak(L)`
  Agent-oriented programming language.
  - `JASON`
    Development platform for AgentSpeak.

Architectures

BDI model itself does not specify how to handle each component behaviour.

- PRS uses database for beliefs.
- AgentSpeak agent is a reactive planning system. [5]
- dMARS plans represents procedural knowledge.
- ...

Overcoming original limitations:

- Agent systems needs to be distributed.
- Adapting to changes from experience.

[5] Dr. Smith Rao M.S, Jyothsna.A.N (2013) BDI: Applications and Architectures, IJERT Vol. 2 Issue 2

Type of problems

BDI agents can be used to solve problems with partial information in a complex and dynamic environment.

For instance: [6]

- Air-traffic control
- Autonomous space-craft control
- Health care services
- Industrial control systems
- Robot soccer

[6] Dr. Smith Rao M.S, Jyothsna.A.N (2013) BDI: Applications and Architectures, IJERT Vol. 2 Issue 2

Example

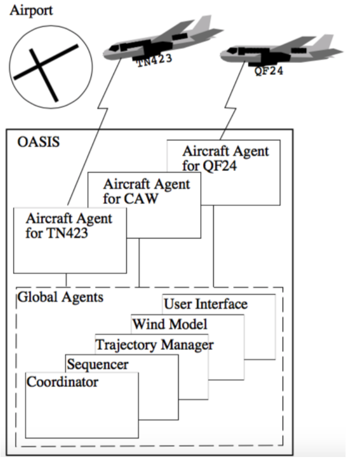

Fig. 3 OASIS System Architecture

OASIS (Optimal Aircraft Sequencing using Intelligent Scheduling).

Tested successfully at Sydney Airport in 1995. Implemented using PRS (Procedural Reasoning Systems).

Multiple Agents, each tackling sub-problems. Agents communicate using asynchronous messages.[7]

[7] M. Ljungberg, A.Lucas (1992) The OASIS air-traffic management system. PRICAI, Seoul, Korea

Example

List of agents:

- SEQUENCER Agent
- AIRCRAFT Agent
- WIND MODEL Agent
- ...

Possible BDI instance in this scenario:

- Belief = Planes position.
- Desire = Decrease speed of aircraft.
- Intention = Adopted plan.

Changes in the environment leads to reassessing intentions.[8]

[8] M. Ljungberg, A.Lucas (1992) The OASIS air-traffic management system. PRICAI, Seoul, Korea

Case Scenario

Applying reinforcement learning and BDI model to create a better strategy for Robot Soccer. [9]

Multi-Agent cooperation overtakes individual optimisation. All the agents pursue a common optimum solution.

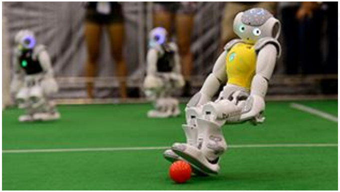

Fig. 4 NAO Robot model [10]

[9] Guo Qi, Wu Bo-ying (2009) Study and Application of Reinforcement Learning in Cooperative Strategy of the Robot Soccer Based on BDI Model, IJRS Vol. 6 No. 2 pp. 91-96 PRICAI, Seoul, Korea

[10] http://static.nautil.us (2016)

Possible approaches

- `Pure reactive`
  If something happened, I am going act on it.

- `Behaviour tress + Fuzzy Logic`
  Leaf nodes used as action to change state of the robot.
  Non-leaf node are used to move within the tree.[11]

- `BDI`
  Define Belief, Desire and Intention. Provide a plan library.

[11] R. Abiyev, I. Gunsel, N. Akkaya, E. Aytac, A. Cagman, S. Abizada, (2016) Robot soccer control using behaviour trees and fuzzy logic, ICAFS

Architecture of the Agent is Implemented in dMars as a several set of plans:[12]

- Plan for managing Agent's role.
- Plan for managing Agent's responsibility.
- Plan for managing Agent's strategies.

There are also two intention threads:[12]

- Intention thread for Agent's role.
- Intention thread for Agent's responsibility.

[12] S. Ch'ng, L. Padgham (1998) From roles to teamwork: A framework and architecture, Applied Artificial Intelligence

BDI Approach

Procedure to choose the role of an Agent:[13]

1. Update beliefs.
2. Select a role.
3. Become the role.

4. New intention thread.
5. Might discard old responsibilities.

Failing a responsibility cause the role to terminate.

---

[13] S. Ch'ng, L. Padgham (1998) From roles to teamwork: A framework and architecture, Applied Artificial Intelligence

Example of Corner kick used in the paper.

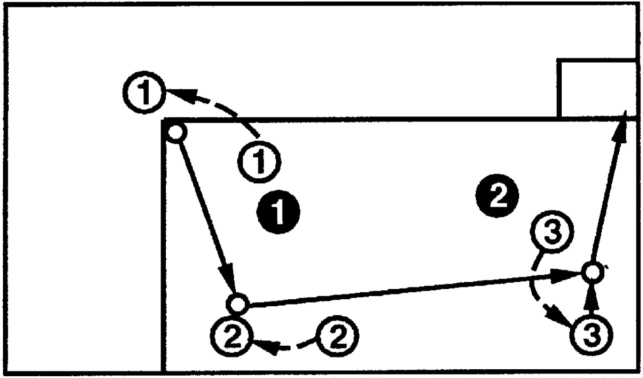

Fig. 5 Show a pass team strategy, where one by one each player select a role and take position.[14]

[14] S. Ch'ng, L. Padgham (1998) From roles to teamwork: A framework and architecture, Applied Artificial Intelligence

Case Scenario Results

The article showed that roles can be assigned quickly and dynamically.

It is prone to errors if something happens in between decisions.[15]

Specifically needs to research more when to drop a plan and move on.

[15] S. Ch'ng, L. Padgham (1998) From roles to teamwork: A framework and architecture, Applied Artificial Intelligence

Comparison

Experiments to compare coordinated action selection (BDI), against Reactive actions.

Both executed using simulation and real robots in a Two vs Two scenario:

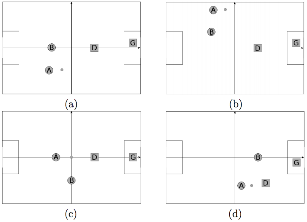

Fig. 6 Scenario used in the experiments [16]

[16] R. Ros, J. L. Arcos, R. L. de Mantaras, M. Veloso (2009) A case-based approach for coordinated action selection in robot soccer IIIA, CSIC

Comparison

The experiment simulates two robot while they attack.

Each simulated experiment uses a different configuration for defence:

- Defender and Goalie.
- Midfield defender and defender.

However for real robot experimentation, only Defender/Goalie configuration (Time Constraint).[17]

[17] R. Ros, J. L. Arcos, R. L. de Mantaras, M. Veloso (2009) A case-based approach for coordinated action selection in robot soccer IIIA, CSIC

Comparison

- The simulation showed that the robots implementing BDI performed better overall.
- The reaction method only outperformed the BDI in Scenario 4.
- They noticed that the Reactive approach is faster at attacking the ball. [18]

[18] R. Ros, J. L. Arcos, R. L. de Mantaras, M. Veloso (2009) A case-based approach for coordinated action selection in robot soccer IIIA, CSIC

Comparison

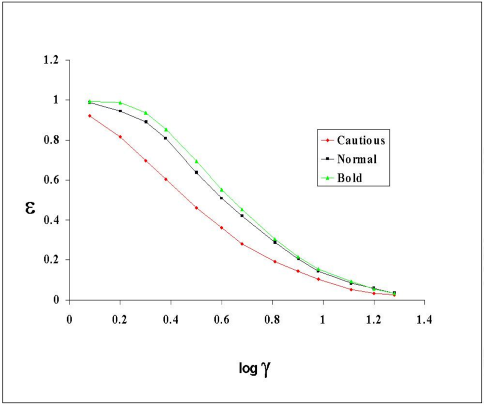

Fig. 7 The graph shows how commitment to a decision affect the final outcome.[19]

---

[19] Georgeff M., Pell B., Pollack M., Tambe M., Wooldridge M. (1999) The Belief-Desire-Intention Model of Agency.

Advantages / Disadvantages

Advantages:

- Saving computation power, no need to build a new plan every time.
- Stay flexible by changing subgoals based on the changes in the environment.

Disadvantages:

- Needs to supply plan library at design time.
- Some implementations however jump from one plan to another when their Belief changes too often.
- A true BDI system that behave like humans is hard to implement.

Why BDI?

The challenges encountered during development fall under the BDI scope area:

- The environment is non deterministic.
- Players have to change roles based on environment (Beliefs).
- The changes have to be low in computational power.
- Multi-Agent system.
- Actions can be gathered in plans.

Conclusion

Solid model to implement human-like practical reasoning Agents.

Multi-Agent coordination needs to be part of the system even if it is not specified in the original BDI model.

Still needs more research, probably a perfect system would incorporate a DBI deliberation system and Reactive system in synergy.