BDI & Reasoning

Henry Hexmoor
Department of Computer Sciences
SIUC

Outline

- Mental states and computer programs
- BDI model
- Some BDI implementations
- Example of BDI reasoning
- BDI resources

Mental states and computer programs

- Daniel Dennet:
  - There are three different strategies thet we might use when confronted with objects or systems: the physical stance, the design stance and the intentional stance
  - Each of these stances is *predictive*
  - We use them to predict and thereby to explain the behavior of the entity in question

The Physical Stance

- Stems from the perspective of the physical sciences
- In predicting the behavior of a given entity we use information about:
  - its physical constitution, and
  - the laws of physics
- If I drop a pen, I use physical stance to predict what happens

The Design Stance

- We assume that the entity has been designed in a certain way
- We predict that the entity will thus behave as designed (i.e. Alarm clock, turning on a computer)
- Predictions are riskier than physical stance predictions, but DS adds predictive power compared to the PS

The Intentional Stance

- We treat the entity in question as a rational agent whose behavior is governed by intentional states such as beliefs, desires and intentions
- Riskier than the design stance, but provides useful gains of predicting power
- Great abstraction tool for complex systems and indispensable when when it comes to complicated artifacts and living things

The Intentional Stance

- Consider chess-playing computer, it can be seen in several ways:
  - as a physical system operating according to the laws of physics;
  - as a designed mechanism consisting of parts with specific functions that interact to produce certain characteristic behaviour; or
  - as an intentional system acting rationally relative to a certain set of beliefs and goals
- Given that our goal is to predict and explain a given entity's behavior, we should adopt the stance that will best allow us to do so.
- There are hundreds (or more?) of differently implemented programs that play chess, but we don't have to worry about the implementation.

The Intentional Stance (Cont.)

- The adoption of the IS:
  1. Decide to treat X as a rational agent
  2. Determine what beliefs X ought to have
  3. Determine what desires X ought to have
  4. Predict what X will do to satisfy some its desires in light of its beliefs

Belief-Desire-Intention (BDI) model

- A theory of practical reasoning.
- Originally developed by Michael E. Bratman in his book ”*Intentions, Plans, and Practical Reason”*, (1987).
- Concentrates in the roles of the intentions in practical reasoning.

Practical reasoning

- Practical reasoning is reasoning directed towards actions — the process of figuring out what to do:

  “Practical reasoning is a matter of weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires/values/cares about and what the agent believes.” (Bratman)

  ”We deliberate not about ends, but about means. We assume the end and consider how and by what means it is attained.” (Aristotle)

Practical reasoning

- Human practical reasoning consists of two activities:
  - *Deliberation,* deciding *what* state of affairs we want to achieve
    - the outputs of deliberation are *intentions*
  - *Means-ends reasoning,* deciding *how* to achieve these states of affairs
    - the outputs of means-ends reasoning are *plans*.

Theoretical reasoning

- Distinguish practical reasoning from *theoretical reasoning*. Theoretical reasoning is directed towards beliefs.
- Example (syllogism):
  - ”Socrates is a man; all men are mortal; therefore Socrates is mortal”

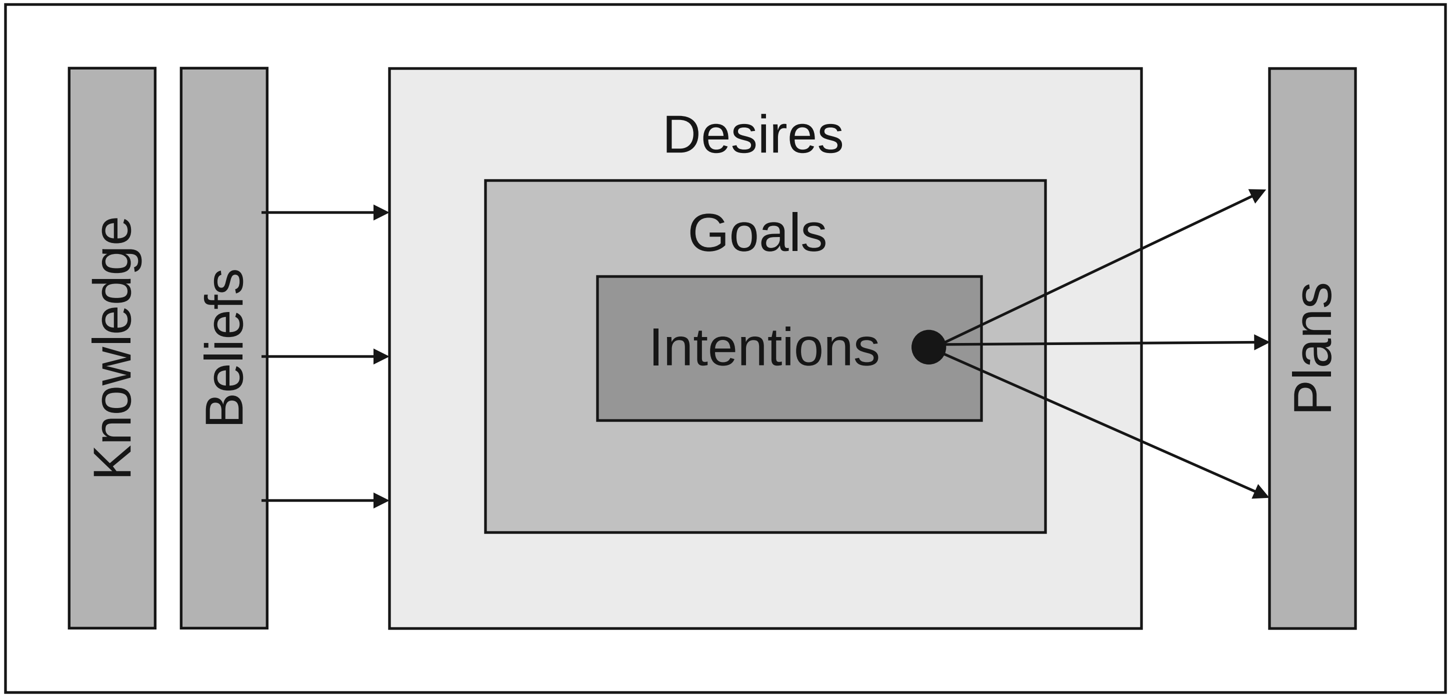
Belief-Desire-Intention (BDI) model

- *Beliefs* correspond to information the agent has about the world
- *Desires* represent states of affairs that the agent would (in an ideal world) wish to be brought about
- *Intentions* represent desires that it has *committed* to achieving

Belief-Desire-Intention (BDI) model

- A philosophical component
  - Founded upon a well-known and highly respected theory of rational action in humans
- A software architecture component
  - Has been implemented and succesfully used in a number of complex fielded applications
- A logical component
  - The theory has been rigorously formalized in a family of BDI logics

Belief-Desire-Intention



[Rao and Georgeff, 1995]

Belief-Desire-Intention

- Beliefs:
  - Agent’s view of the world, predictions about future.

- Desires:
  - Follow from the beliefs. Desires can be unrealistic and unconsistent.
- Goals:
  - A subset of the desires. Realistic and consistent.
- Intentions:
  - A subset of the goals. A goal becomes an intention when an agent decides to commit to it.

- Plans:
  - Intentions constructed as list of actions.

What is intention? (Bratman)

- We use the concept of intention to characterize both our actions and our minds.
- I intend to X vs. I did X intentionally
- Intentions can be present or future directed.
- *Future directed* intentions influence later action, *present directed* intentions have more to do with reactions.

Intention vs. desire (Bratman)

- Notice that intentions are much stronger than mere desires:

  "My desire to play basketball this afternoon is merely a potential influencer of my conduct this afternoon. It must vie with my other relevant desires [. . .] before it is settled what I will do. In contrast, once I intend to play basketball this afternoon, the matter is settled: I normally need not continue to weigh the pros and cons. When the afternoon arrives, I will normally just proceed to execute my intentions." (Bratman, 1990)

Intention is choice with commitment (Cohen & Levesque)

- There should be ”rational balance” among the beliefs, goals, plans, intentions, commitments and actions of autonomous agents.
- Intentions play big role in maintaining ”rational balance”
- An autonomous agent should act on its intentions, not in spite of them
  - adopt intentions that are feasible, drop the ones that are not feasible
  - keep (or commit to) intentions, but not forever
  - discharge those intentions believed to have been satisfied
  - alter intentions when relevant beliefs change

(Cohen & Levesque, 1990)

Intentions in practical reasoning

1. Intentions normally pose problems for the agent.
   - The agent needs to determine a way to achieve them.

2. Intentions provide a ”screen of admissibility” for adoptin other intentions.
   - Agents do not normally adopt intentions that they believe conflict with their current intentions.

(Cohen & Levesque, 1990)

Intentions in practical reasoning

3. Agents ”track” the success of their attempts to achieve their intentions.
   - Not only do agents care whether their attemts succeed, but they are disposed to replan to achieve the intended effects if earlier attemts fail.

4. Agents believe their intentions are possible.
   - *They believe there is at least some way that the intentions could be brought about.*

Intentions in practical reasoning

5. Agents do not believe they will not bring about their intentions.
   - It would not be rational to adopt an intention if one doesn't believe it is possible to achieve.
6. Under certain conditions, agents believes they will bring about their intentions.
   - It would not normally be rational of me to believe that I would bring my intentions about; intentions can fail. Moreover, it does not make sense that if I believe ϕ is inevitable that I would adopt it as an intention.

Intentions in practical reasoning

7. Agents need not intend all the expected side-effects of their intentions.
   - If I believe $\phi \rightarrow \psi$ and I intend that ϕ, I do not necessarily intend ψ also. (Intentions are not closed under implication.)
   - This last problem is known as the *side effect* or *package deal* problem. I may believe that going to the dentist involves pain, and I may also intend to go to the dentist — but this does not imply that I intend to suffer pain!
   - Agents do not track the state of the side effects.

Planning Agents

- Since the early 1970s, the AI planning community has been closely concerned with the design of artificial agents
- Planning is essentially automatic programming: the design of a course of action that will achieve some desired goal

Planning agents

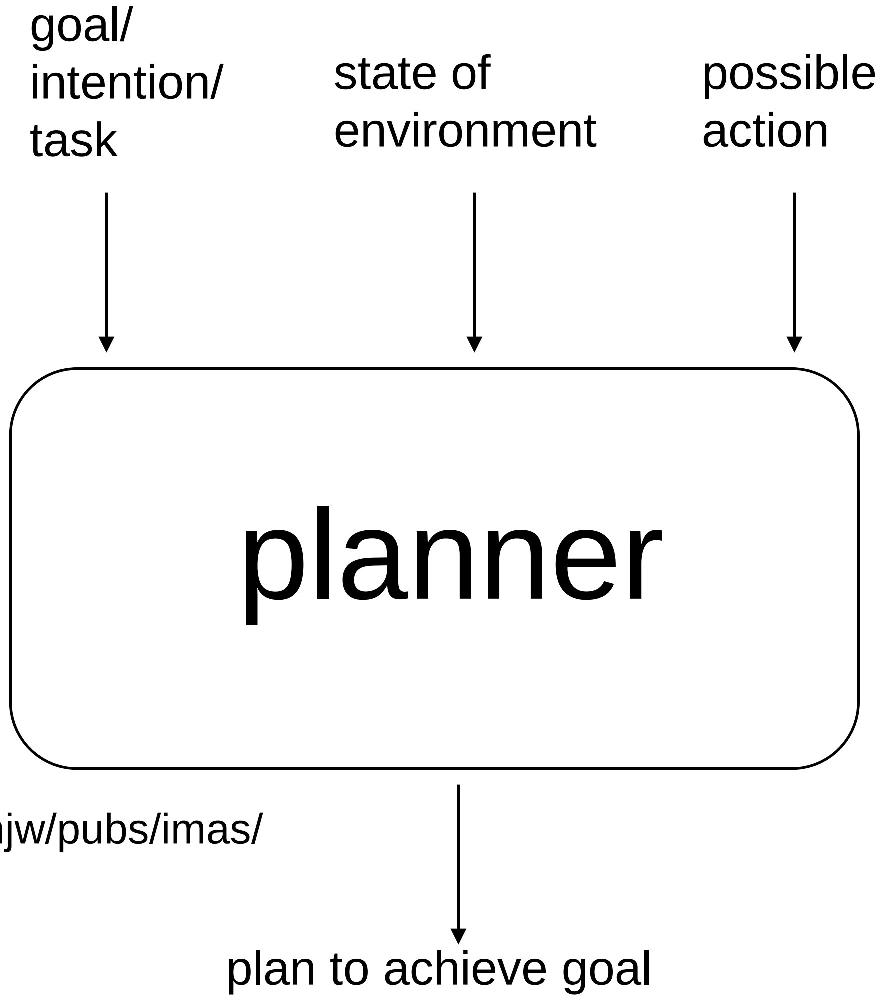
- Within the symbolic AI community, it has long been assumed that some form of AI planning system will be a central component of any artificial agent.
- Building largely on the early work of Fikes & Nilsson, many planning algorithms have been proposed, and the theory of planning has been well-developed.

What is means-end reasoning?

- Basic idea is to give an agent:
  - representation of goal/intention to achieve
  - representation actions it can perform
  - representation of the environment

and have it generate a *plan* to achieve the goal

STRIPS planner

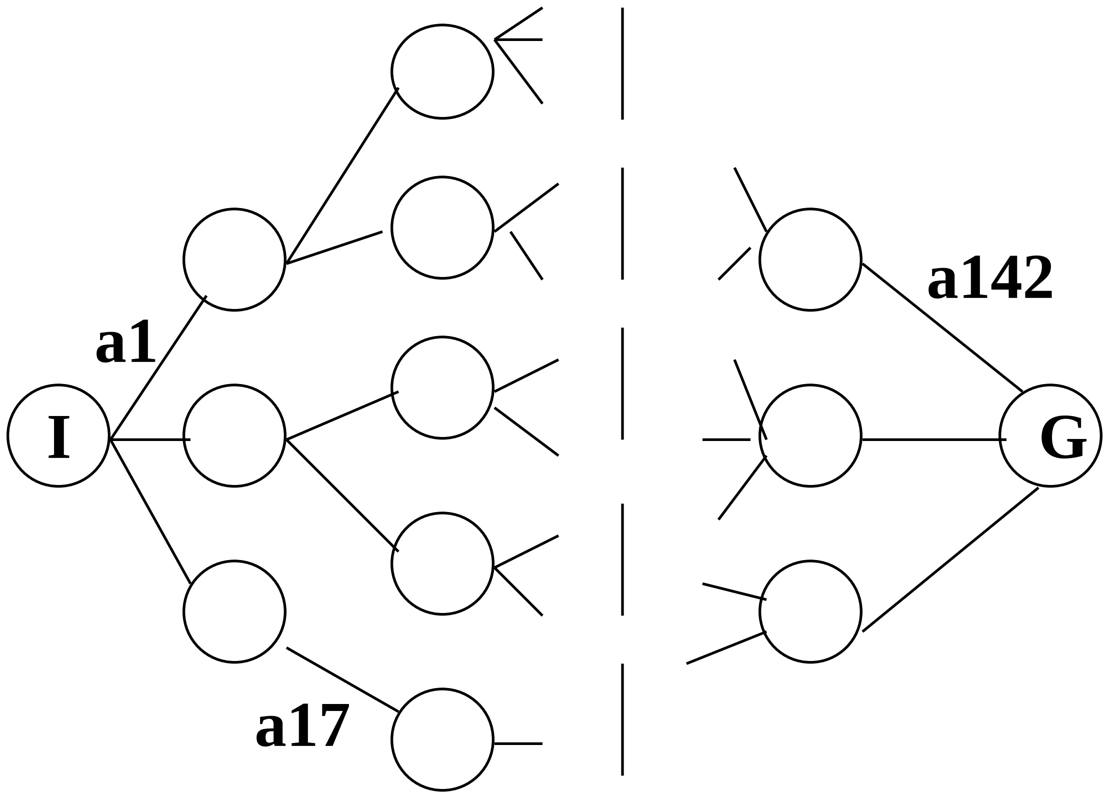


http://www.csc.liv.ac.uk/˜mjw/pubs/imas/

Actions

- Each action has:
  - a *name*
    which may have arguments
  - a *pre-condition list*
    list of facts which must be true for action to be executed
  - a *delete list*
    list of facts that are no longer true after action is performed
  - an *add list*
    list of facts made true by executing the action

A Plan



- A plan is a sequence (list) of actions, with variables replaced by constants.

http://www.csc.liv.ac.uk/˜mjw/pubs/imas/

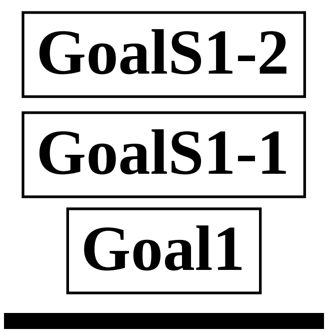
The STRIPS approach

- The original STRIPS system used a goal stack to control its search
- The system has a database and a goal stack, and it focuses attention on solving the top goal (which may involve solving sub goals, which are then pushed onto the stack, etc.)

The Basic STRIPS Idea

- Place goal on goal stack:

- Considering top Goal1, place onto it its subgoals:



- Then try to solve subgoal GoalS1-2, and continue...

STRIPS approach to plans

- Most BDI agents use plans to bring about their intentions.
- These plans are usually pre-written by the software developer. This means that the agent does not construct them from its actions.
- So, the plans are like recipies that the agent follows to reach its goals.

BDI plans

- In BDI implementations plans usually have:
  - a *name*
  - a *goal*
    invocation condition that is the triggering event for the plan
  - a *pre-condition list*
    list of facts which must be true for plan to be executed
  - a *delete list*
    list of facts that are no longer true after plan is performed
  - an *add list*
    list of facts made true by executing the actions of the plan
  - a *body*
    list of actions

The challenge of dynamic environments

1. At any instant of time, there are potentially many different ways in which the environment can evolve.
2. At any instant of time, there are potentially many different actions or procedures the system can execute.
3. At any instant of time, there are potentially many different objectives that the system is asked to accomplish.
4. The actions or procedures that (best) achieve the various objectives are dependent on the state of the environment (context) and are independent of the internal state of the system.
5. The system can only be sensed locally.
6. The rate at which computations and actions can be carried out is within reasonable bounds to the rate at which the environment evolves.

Rao and Georgeff (1995)

The challenge of dynamic environments (2)

- Agent can’t trust that the world remains constant during the whole planning process
  - While you are trying to figure out which grocery store has the best price for flour for the cake, your children may drink up the milk
  - And if you spend a long time recomputing the best plan for buying flour, you just may lose your appetite or the store closes before you’re done.

Pollack (1992)

The challenge of dynamic environments (3)

- Real environments may also change while an agent is executing a plan in ways that make the plan invalid
  - While you are on your way to the store, the grocers may call a strike
- Real environments may change in ways that offer new possibilities for action
  - If your phone rings, you might not want to wait until the cake is in the oven before considering whether to answer it

Pollack (1992)

The challenge of dynamic environments (4)

- Intelligent behaviour depends not just on being able to decide *how to achieve one's goals*
- It also depends on *being able to decide which goals to pursue in the first place*, and *when to abandon or suspend the pursuit of an existing goal*

Pollack (1992)

Resource bounds and satisficing

- A rational agent is *not* one who always chooses the action that does the most to satisfy its goals, given its beliefs
- A rational agent simply does not have the resources always to determine what that optimal action is
- Instead, rational agents must attempt only to ”satisfice”, or to make good enough, perhaps non-optimal decisions about their actions

Pollack (1992)

Using plans to constrain reasoning

- What is the point of forming plans?
  - Agents reside in dynamic environments, any plan they make may be rendered invalid by some unexpected change

  - The more distant the intended execution time of some plan, the less that can be assumed about the conditions of its execution

Pollack (1992)

Using plans to constrain reasoning

- Agents form/use plans in large part *becouse* of their resoure bounds
- An agent's plans serve to frame its subsequent reasoning problems so as to constrain the amount of resources needed to solve them
  - Agents *commit* to their plans
  - Their plans tell them *what to* reason about, and *what to not* reason about

Pollack (1992)

Commitment

- When an agent commits itself to a plan, it commits both to:
  - *ends (i.e.* the state of affairs it wishes to bring about, the goal), and
  - *means* (i.e., the mechanism via which the agent wishes to achieve the state of affairs, the body).

Commitment

- Commitment implies *temporal persistence.*
- An intention, once adopted, should not immediately evaporate.
- A critical issue is just *how* committed an agent should be to its intentions.
- A mechanism an agent uses to determine when and how to drop intentions is known as *commitment strategy*.

http://www.csc.liv.ac.uk/˜mjw/pubs/imas/

Commitment strategies

- Blind commitment (fanatical commitment):
  - An agent will continue to maintain an intention until it believes the intention has been achieved.
- Sinlge-minded commitment:
  - An agent will continue to maintain an intention until it believes that either the intention has been achieved or it cannot be achieved.
- Open-minded commitment:
  - An agent will continue to maintain an intention as long as it is still believed to be possible.

(Wooldridge, 2000)

Intention reconsideration

- Intentions (plans) enable the agent to be goal-driven rather than event-driven.
- By committing to intentions the agent can pursue long-term goals.
- However, it is necessary for a BDI agent to reconsider its intentions from time to time:
  - The agent should drop intentions that are no longer achievable.
  - The agent should adopt new intentions that are enabled by opportunities.

http://www.csc.liv.ac.uk/˜mjw/pubs/imas/

Intention reconsideration

- Kinny and Georgeff experimentally investigated effectiveness of intention reconsideration strategies.
- Two different types of reconsideration strategy were used:
  - *bold* agents
  - *cautious* agents

http://www.csc.liv.ac.uk/˜mjw/pubs/imas/

Intention reconsideration

- Bold agent never pauses to reconsider its intentions.
- Cautious agent stops to reconsider its intentions after every action.
- *Dynamism* in the environment is represented by the *rate of world change*, φ.

http://www.csc.liv.ac.uk/˜mjw/pubs/imas/

Intention reconsideration

- Results:
  - If φ is low (i.e., the environment does not change quickly), then bold agents do well compared to cautious ones. This is because cautious ones waste time reconsidering their commitments while bold agents are busy working towards — and achieving — their intentions.
  - If φ is high (i.e., the environment changes frequently), then cautious agents tend to outperform bold agents. This is because they are able to recognize when intentions are doomed, and also to take advantage of serendipitous situations and new opportunities when they arise.

http://www.csc.liv.ac.uk/˜mjw/pubs/imas/

Some implemented BDI - architectures

- IRMA - Intelligent, Resource-Bounded Machine Architecture. Bratman, Israel, Pollack.
- PRS - Pocedural Reasoning System. Georgeff, Lansky.
  - PRS-C, PRS-CL, dMARS, JAM…

Implemented BDI Agents: IRMA

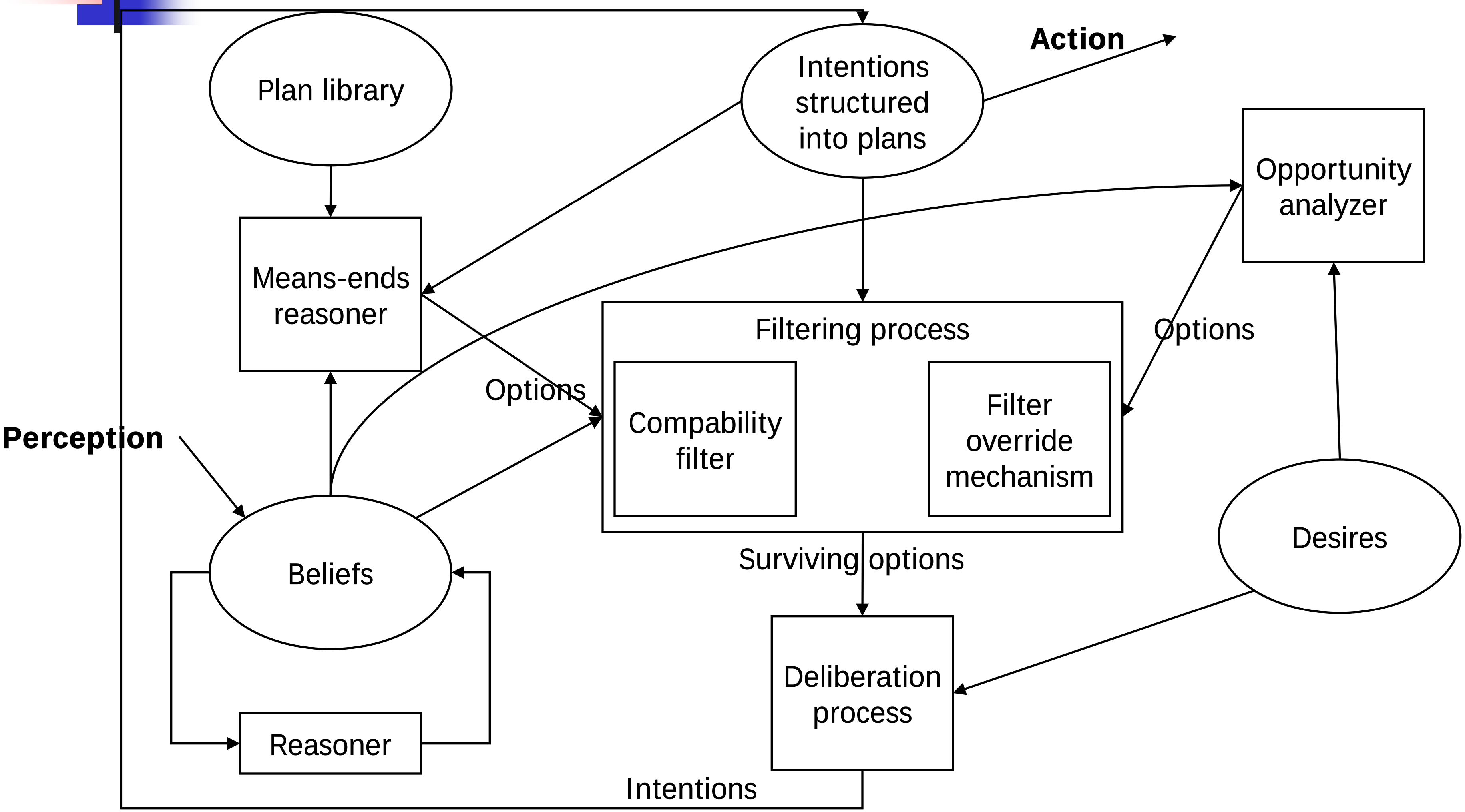
- IRMA has four key symbolic data structures:
  - a *plan library*
  - explicit representations of
    - *beliefs*: information available to the agent — may be represented symbolically, but may be simple variables
    - *desires*: those things the agent would *like* to make true — think of desires as *tasks* that the agent has been allocated; in humans, not necessarily logically consistent, but our agents will be! (goals)
    - *intentions*: desires that the agent has *chosen* and *committed to*

http://www.csc.liv.ac.uk/˜mjw/pubs/imas/

IRMA

- Additionally, the architecture has:
  - a *reasoner* for reasoning about the world; an inference engine
  - a *means-ends analyzer* determines which plans might be used to achieve intentions
  - an *opportunity analyzer* monitors the environment, and as a result of changes, generates new options
  - a *filtering process* determines which options are compatible with current intentions
  - a *deliberation process* responsible for deciding upon the ‘best’ intentions to adopt

IRMA

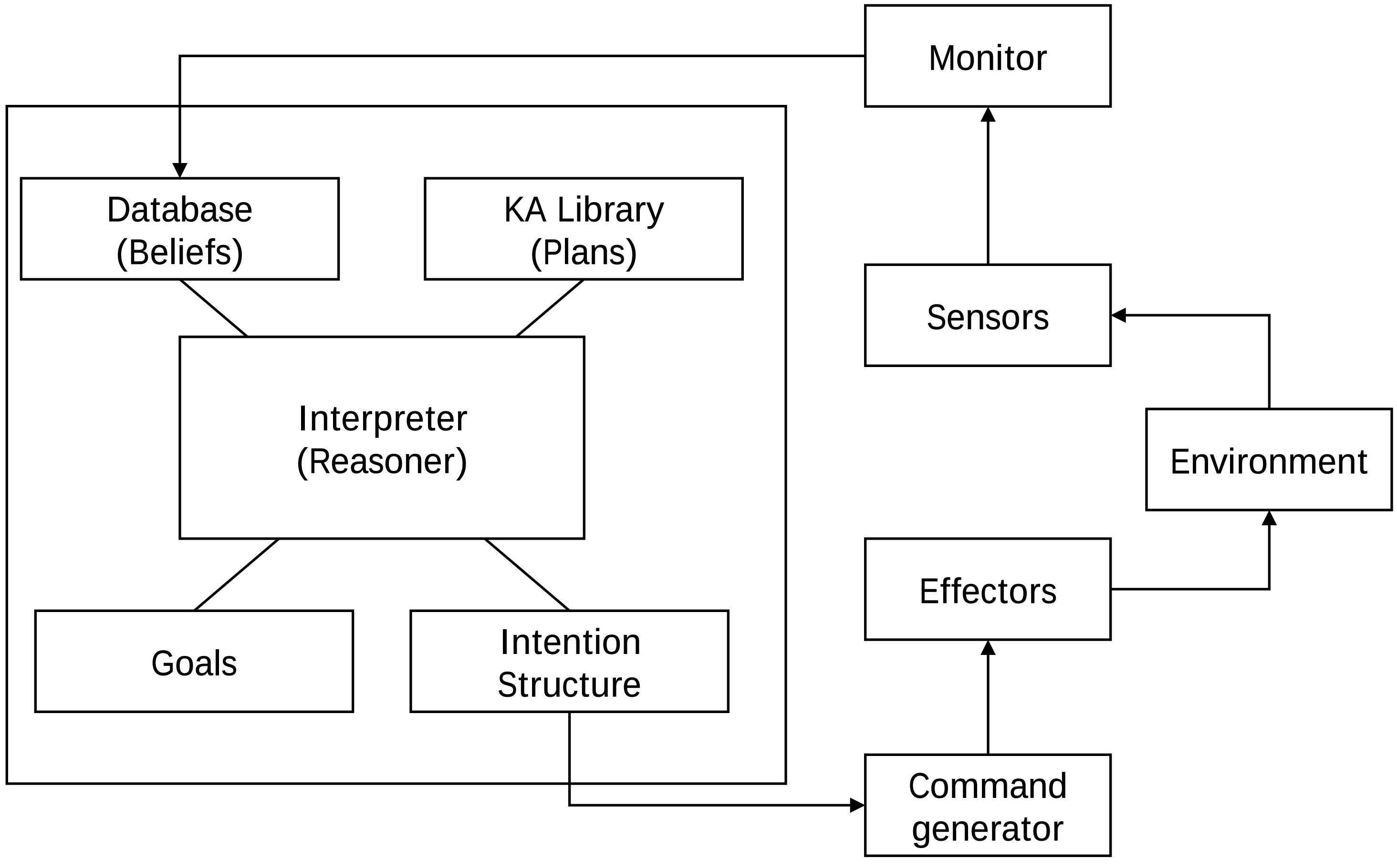


Plan library
Intentions structured into plans
Action
Opportunity analyzer
Means-ends reasoner
Filtering process
Options
Options
Compability filter
Filter override mechanism
Perception
Desires
Beliefs
Surviving options
Deliberation process
Reasoner
Intentions

Implemented BDI Agents: PRS

- In the PRS, each agent is equipped with a *plan library*, representing that agent's *procedural knowledge*: knowledge about the mechanisms that can be used by the agent in order to realize its intentions.
- The options available to an agent are directly determined by the plans an agent has: an agent with no plans has no options.
- In addition, PRS agents have explicit representations of beliefs, desires, and intentions, as above.

PRS


Monitor
Database
(Beliefs)
KA Library
(Plans)
Sensors
Interpreter
(Reasoner)
Environment
Effectors
Goals
Intention
Structure
Command
generator

JAM Plan

```
Plan: {
  NAME: "Clear a block"
  GOAL:
              ACHIEVE CLEAR $OBJ;
  CONTEXT:
              FACT ON $OBJ2 $OBJ;
  BODY:
              EXECUTE print "Clearing " $OBJ2 " from on top of " $OBJ "\n";
              EXECUTE print "Moving " $OBJ2 " to table.\n";
              ACHIEVE ON $OBJ2 "Table";
  EFFECTS:
              EXECUTE print "CLEAR: Retracting ON " $OBJ2 " " $OBJ "\n";
              RETRACT ON $OBJ1 $OBJ;
  FAILURE:
              EXECUTE print "\n\nClearing block " $OBJ " failed!\n\n";
}
```

Plan actions (JAM)

ACTION	DESCRIPTION	ACTION	DESCRIPTION
ACHIEVE	Subgoal	AND	Do all branches; try in order
ASSERT	Add to world model	ASSIGN	Set variable value
ATOMIC	Perform without interruption	DO-WHILE	Iterate
DO_ALL	Do all branches in random order	DO_ANY	Do one random branch
EXECUTE	Perform primitive action	FACT	Check world model values
FAIL	Unconditionally fail	LOAD	Parse JAM input file
MAINTAIN	Subgoal	NEXTFACT	Get the next matching world model relation retrieved with RETRIEVALL

Plan actions (JAM)

ACTION	DESCRIPTION	ACTION	DESCRIPTION
OR	Do any branch; try in order		
PARALLEL	Execute all branches simultaneously	PERFORM	Subgoal
POST	Add top-level goal	QUERY	Subgoal
RETRACT	Remove from world model	RETRIEVE	Get values from world model
RETRIEVALL	Get all matching world model relations	SUCCEED	Unconditionally succeed
TEST	Check condition	UNPOST	Remove goal
UPDATE	Change world model	WAIT	Wait for condition/goal
WHEN	Conditional execution	WHILE	Iterate

Goals and Intentions in JAM

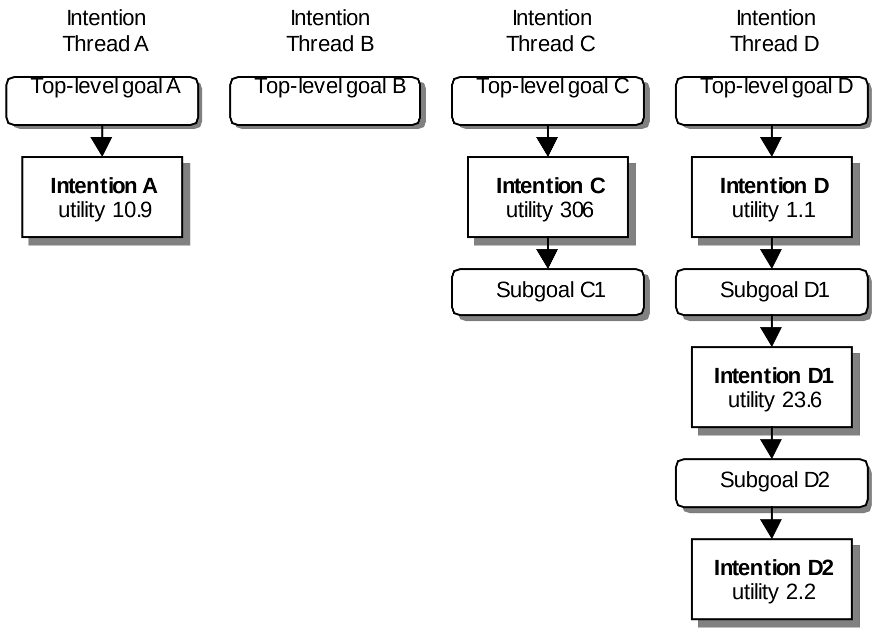


Figure 3.1 A depiction of a Jam agent's intention structure in the middle of execution. Top-level goals A, C, and D have had plan intended to them. The plans for Intention C and D have subgoaled. Subgoal C1 has not yet had a plan intended to it yet, however. As only plans are executable, Subgoal C1 is not considered for execution selection. Intention Thread D has subgoaled an additional level, for which Intention D2 has been selected. In the situation depicted, only Intention Thread A and D are executable. Because Intention D2 has a higher utility than Intention A, the plan for Intention D2 will be executed in the next cycle. Note that the utility values for Intention D and Intention D1 are ignored.

PRS example: An Abstract BDI Interpreter

- Based on a classic sense-plan-act procedure:
  1. Observe the world.
  2. Plan actions.
  3. Execute actions.

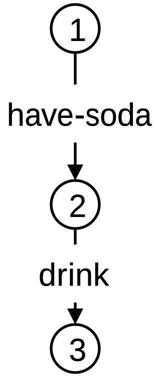
An Abstract BDI Interpreter

- The system state comprises three dynamic data structures representing the agent's beliefs, desires and intentions. The data structures support update operations
- Assume agent's desires mutually consistent, but not necessarily all achievable. Such mutually consistent desires are called goals.
- The inputs to the system are atomic events, received via an event queue. The system can recognize both external and internal events.
- The outputs of the system are atomic actions.

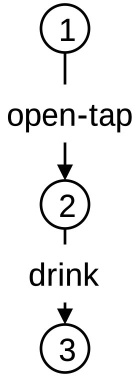
(Singh et al, 1999)

Plans (for quenching thirst)

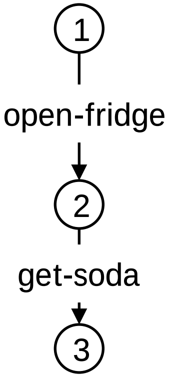
Type: drink-soda
Invocation:
g-add(quenched-thirst)
Precondition: have-glass
Add List:{quenched-thirst}
Body:



Type: drink-water
Invocation:
g-add(quenched-thirst)
Precondition: have-glass
Add List:{quenched-thirst}
Body:



Type: drink-water
Invocation:
g-add(have-soda)
Precondition: have-glass
Add List:{have-soda}
Body:



(Singh et al, 1999)

Plans

- Having a plan means that its body is believed to be an option whenever its invocation condition and precondition are satisfied.
- A plan represents the belief that, whenever its invocation condition and precondition are satisfied and its body is successfully executed, the propositions in the add list will become true.
- The agent can execute plans to compute new consequences. These consequences can trigger further plans to infer further consequences.

(Singh et al, 1999)

Intentions

- Once the plans are adopted, they are added to the intention structure (stack). Thus, intentions are presented as hierarchically related plans.
- To achieve intended end, the agent forms an intention towards a means for this end: namely, the plan body.
- An intention towards a means results in the agent adopting another end (subgoal) and the means for achieving this end.
- This process continues until the subgoal can be directly executed as an atomic action. The next subgoal is then attempted.

(Singh et al, 1999)

An Abstract BDI Interpreter

- Simplified PRS interpreter:

```
BDI-Interpreter
initialize-state();
do
   options = option-generator(event-queue,B,G,I)
   selected-options = deliberate(options,B,G,I)
   update-intentions(selected-options,I)
   execute(I)
   get-new-external-events()
   drop-succesful-attitudes(B,G,I)
   drop-impossible-attitudes(B,G,I)
until quit.
```

(Singh et al, 1999)

Option-generator

```
option-generator(trigger-events)
  options := {}
  for trigger-event ∈ trigger-events do
    for plan ∈ plan-library do
      if matches(invocation(plan), trigger-event) then
        if provable(precondition(plan), B) then
          options := options ∪ {plan};
  return options.
```

(Singh et al, 1999)

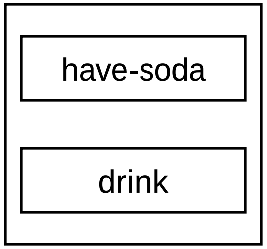
Deliberate

```
deliberate(options)
 if length(options) ≤ 1 then return options;
 else metalevel-options := option-generator(b-add(option-set(options)));
  selected-options := deliberate(metalevel-options);
  if null(selected-options) then
   return (random-choice(options));
  else return selected-options.
```

(Singh et al, 1999)

Example

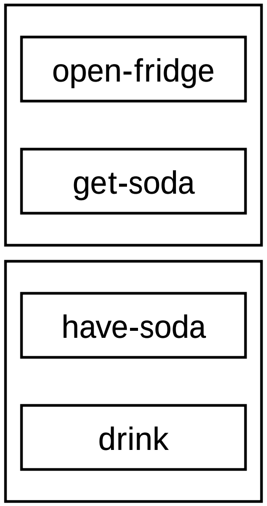
1. Suppose the agent is thirsty and a goal ”quenched-thirst” has been added to its event queue.
2. The agent has two plans to quenche its thirst: ”drink-soda” and ”drink-water”
3. Assume the agent selects the plan ”drink-soda” first (possibly by random choice) and commits to it. The intention structure looks like:



4. The action of the ”drink-soda” plan is adding a sub goal ”have-soda” to the event queue.

Example (cont.)

5. Now the deliberate function finds a plan ”get-soda” whic satisfies the goal ”have-soda” and it is added to the intention structure. Situation is now:



Example (cont.)

6. Next action in the intention structure is ”open-fridge”. So, the agent opens the fridge but discoveres that no soda is present.

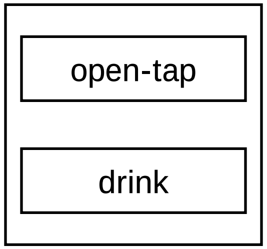
7. The agent is now forced to drop its intention to get soda from the fridge.

8. As there is no other plan which satisfies the goal ”have-soda”, it is forced to drop the inention to ”drink-soda”.

9. The original goal ”quenched-thirst” is added again to the event queue.

Example (cont.)

10. The agent chooses the plan ”drink-water” and adds it to the intention structure:



11. The agent executes ”open-tap”.

12. The agent executes ”drink”.

13. The belief ”quenched-thirst” is added to beliefs.

BDI applications (ok, some are pretty academic...)

- Applying Conflict Management Strategies in BDI Agents for Resource Management in Computational Grids http://crpit.com/confpapers/CRPITV4Rana.pdf
- AT Humbold in RoboCup Simulation League http://www.robocup.de/AT-Humboldt/team_robocup.shtml http://sserver.sourceforge.net/
- Capturing the Quake Player: Using a BDI Agent to Model Human Behaviour http://cfpm.org/~emma/pubs/Norling-AAMAS03.pdf
- A BDI Agent Architecture for Dialogue Modelling and Coordination in a Smart Personal Assistant http://www.cse.unsw.edu.au/~wobcke/papers/coordination.pdf
- Space shuttle RCS malfunction handling http://www.ai.sri.com/~prs/rcs.html

BDI resources

- Jadex BDI Agent System http://vsis-www.informatik.uni-hamburg.de/projects/jadex/features.php
- Agent Oriented Software Group: JACK http://www.agent-software.com/shared/home/index.html
- JAM Agent & UMPRS Agent http://www.marcush.net/IRS/irs_downloads.html
- PRS-LC Lisp version of PRS http://www.ai.sri.com/~prs/rcs.html
- Nice list of agent constructing tools (not all BDI, some links not working) http://www.paichai.ac.kr/~habin/research/agent-dev-tool.htm
- Subject: 1.2.1 Practical reasoning/planning and acting http://eprints.agentlink.org/view/subjects/1_2_1.html
- Subject: 1.1.1 Deliberative/cognitive agent control architectures and planning http://eprints.agentlink.org/view/subjects/1_1_1.html

References

- Bratman, Michael E., (1990), *What is Intention?* Interaction in communication, MIT Press, 1990.
- Cohen, P. R. and Levesque, H. J. (1991), *Teamwork*, SRI Technical Note 504.
- Cohen, P. R. and Levesque, H. J. (1990), *Intention Is Choice With Commitment*, Artificial Intelligence **42,** (1990), 213-261.
- Dictionary of Philosophy of Mind. http://www.artsci.wustl.edu/~philos/MindDict/intentionalstance.html
- Munidar P. Singh, Anand S. Rao and Michael P. Georgeff (1999), *Formal Methods in DAI: Logig-based Representation and Reasoning*. In: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press, 1999.
- Pollack (1992), *The Uses of Plans*, Artificial Intelligence **57** (1), (1992), 43-68.
- Rao, Anand S., Georgeff, Michael P. (1995), BDI Agents: From theory to practice, In: *Proceedings of ICMAS-95)*, San Francisco, USA, June, 1995.
- Wooldridge, M. (2000). *Reasoning About Rational Agents*, MIT Press, 2000.
- Wooldridge, M. (2002). *Introduction to MultiAgent Systems*, John Wiley & Sons, 2002. http://www.csc.liv.ac.uk/~mjw/pubs/imas/.

Thank you!

Possible World Semantics

- We can think of a possible world as a consistent collection of propositions. The collection must be consistent since under the ordinary rules of logic an inconsistent collection would entail every proposition.
- The notions of “necessity” and “possibility” can be examined by considering the accessibility relations between one designated world and the other worlds.

Accessibility

- If the accessibility relation is reflexive, then the designated world has access to its own propositions.
- If the accessibility relation is symmetric, then if possible world A has access to possible world B, then B has access to A.
- If the accessibility relation is transitive then if world a has access to world B and world B has access to world C, then world A has access to world C.

Definition

- Allowing for different combinations of these accessibility relations provides a foundation for different modal logics.
- With these ideas in mind, we can say the proposition is possible relative to world `H`, if it is true in some world, `Wn`, that is accessible form `H`.
- Likewise a proposition is necessary if it is true in every world, `W0-n`, that is accessible from `H`.

Not Truth-Functional

- In propositional logic, validity can be defined using truth tables. A valid argument is simply one where every truth table row that makes its premises true also makes its conclusion true. However truth tables cannot be used to provide an account of validity in modal logics because there are no truth tables for expressions such as ‘it is necessary that’, ‘it is obligatory that’, and the like.

Valuation

- In propositional logic, a valuation of the atomic sentences (or row of a truth table) assigns a truth-value (T or F) to each propositional variable p. Then the truth-values of the complex sentences are calculated with truth tables.
- In modal semantics, a set W of possible worlds is introduced. A valuation then gives a truth-value to each propositional variable *for each of the possible worlds* in W. This means that value assigned to p for world w may differ from the value assigned to p for another world w′.

Basic Interpretations

- The truth-value of the atomic sentence p at world w given by the valuation v may be written `v(p, w)`. Given this notation, the truth values (T for true, F for false) of complex sentences of modal logic for a given valuation v (and member `w` of the set of worlds `W`) may be defined by the following truth clauses. (‘iff’ abbreviates ‘if and only if’.)

(¬) `v(¬A, w)=T iff v(A, w)=F`.

(→) `v(A→B, w)=T iff  v(A, w)=F or v(B, w)=T`.

(5) `v(`***N***`A, w)=T iff  for every world w′ in W, v(A, w′)=T`.

Relation to Quantification

- Clauses (¬) and (→) simply describe the standard truth table behavior for negation and material implication respectively.
- According to (5), ***N***A is true (at a world w) exactly when A is true in *all* possible worlds. Given the definition of ***P***, (namely, ***P***A = ¬***N***¬A) the truth condition (5) insures that ***P***A is true just in case A is true in *some* possible world. Since the truth clauses for ***N*** and ***P*** involve the quantifiers 'all' and 'some' (respectively), the parallels in logical behavior between ***N*** and ∀x, and between ***P*** and ∃x is as expected.

Validity

- Clauses ($\neg$), ($\rightarrow$), and (5) allow us to calculate the truth-value of any sentence at any world on a given valuation. An argument is *5-valid for a given set* `W` (of possible worlds) if and only if every valuation of the atomic sentences that assigns the premises T at a world in `W` also assigns the conclusion T at the same world. An argument is said to be *5-valid* iff it is valid for every non empty set of `W` of possible worlds.
- It has been shown that S5 is sound and complete for 5-validity.