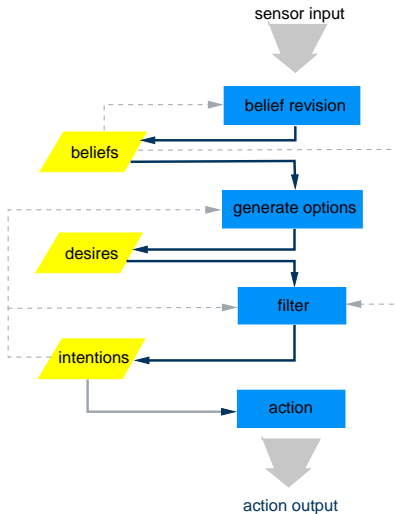


The BDI Architecture

Control process in the BDI model:



An Example: Passing CSE304

- A student agent perceives the following beliefs:

$$\text{Beliefs}_1 = \text{brj} \left\{ \emptyset, \left\{ \begin{array}{l} \text{workhard} \rightarrow \text{passCourse} \\ \text{attendLectures} \cap \text{completeCoursework} \cap \text{review} \\ \rightarrow \text{workhard} \end{array} \right\} \right\}$$

- The agent has an initial intention to pass the course:

$$\text{Intentions}_0 = \{ \text{passCourse} \}$$

- The agent's desires are freshly generated each for cycle (they do not persist). The option generation function leads to desires to pass the course and its consequence:

$$\text{Desires}_1 = \text{options}(\text{Belief}_1, \text{Intentions}_0) =$$

$$\{ \text{workhard}, \text{attendLecture}, \text{completeCoursework}, \text{review} \}$$

Generating intentions

- The filter function leads to some new intentions being added:

$$\begin{aligned} \text{Intentions}_1 &= \text{filter}(\text{Belief}_1, \text{Desires}_1, \text{Intentions}_0) \\ &= \left\{ \begin{array}{l} \text{passCourse}, \text{workHard}, \text{attendLectures}, \\ \text{completeCoursework}, \text{review} \end{array} \right\} \end{aligned}$$

- One or more of which will then be executed before the agent's deliberation cycle recommences.

Obtaining new beliefs

- Suppose the agent perceives new information which leads to his beliefs being revised:

$$\begin{aligned} Beliefs_2 &= brf \left(Beliefs_1, \left\{ \begin{array}{l} cheat \rightarrow passCourse, \\ cheat \prec workhard \end{array} \right\} \right) \\ &= \left\{ \begin{array}{l} workHard \rightarrow passCourse, \\ attendLecture \cap completeCoursework \cap review \\ \rightarrow workHard, \\ cheat \rightarrow passCourse, \\ cheat \prec workHard \end{array} \right\} \end{aligned}$$

Revising desires and intentions

- The agent recomputes his current desires

$$Desires_2 = options(Beliefs_1, Intentions_1) = \{cheat\}$$

- And intentions

$$\begin{aligned} Intentions_2 &= filter(Beliefs_2, Desires_2, Intentions_1) \\ &= \{passCourse, cheat\} \end{aligned}$$

- The agent drops his original intention to work hard (and its consequences) and adopts a new one to cheat

Adding even more beliefs...

- Subsequently, the agent perceives that if caught cheating, he will no longer pass the course. What's more, he is certain to be caught

$$Beliefs_3 = brf \left(Beliefs_2, \left\{ \begin{array}{l} cheat \cap caught \rightarrow \neg passCourse, \\ caught \end{array} \right\} \right)$$

$$= Beliefs_2 / \{cheat \rightarrow passCourse\} \cup$$



$$\{cheat \cap caught \rightarrow \neg passCourse, caught\}$$

- Because the new beliefs lead to an inconsistency, the agent has had to drop his belief in

$$cheat \rightarrow passCourse$$

Revising desires and intentions: again

- The agent recomputes his desires and intentions

$$\begin{aligned} Desires_3 &= options(Beliefs_2, Intentions_2) \\ &= \{workHard, attendLectures, completeCoursework, review\} \end{aligned}$$

$$\begin{aligned} Intentions_3 &= filter(Beliefs_3, Desires_3, Intentions_2) \\ &= \left\{ \begin{array}{l} passCourse, workHard, \\ attendLectures, completeCoursework, review \end{array} \right\} \end{aligned}$$

- Because it's not longer consistent to cheat (even though it may be preferable to working hard), the agent drops that intention and re-adopts workHard (and consequences)

Summary of BDI architecture

- It is a practical reasoning architecture.
- Basic component are data structures representing beliefs, desires and intentions of agent, and functions representing its deliberation and means-ends reasoning.
- Intentions play a central role: proving stability for decision-making and focus for practical reasoning (or planning).

Summary of BDI architecture

- Must decide on type of environment (rate of world change) to enable a good balance between being committed to intentions and reconsidering them.
- BDI model is appealing because:
 - It is intuitive – we can recognize decision process
 - It gives a clear functional decomposition, indicating what sorts of subsystems are required to build an agent
- Main drawback – how to implement efficiently.