BDI Extensions: Obligations, Motivations

Michael Sewell

sewell@uni-koblenz.de

Obligations

Collaboration in multi-agent systems

- to support collaborative behaviour, it is essential to provide individual agents with various forms of *social awareness*
- BDI agents: isolated (non-social)
- wanted: reasoning about the beliefs, desires and intentions of *other* agents

BOID architecture

- *BOID* architecture: framework using Deontic Logic* that extends the BDI model by adding *obligations* (and *norms*)
  - BOID: **B**eliefs-**O**bligations-**I**ntentions-**D**esires

- **Deontic logic*: a form of modal logic
  - operators concern aspects such as *permission* and *obligation*

Mental attitudes in BOID: Beliefs

- **Beliefs**: knowledge (and knowledge defaults)
  - information of an agent about the current state of the world
  - observations are turned into beliefs
  - acts as a filter: only desires and obligations consistent with the beliefs are turned into a goal

Mental attitudes in BOID: Obligations

- **Obligations**: prohibitions and permissions
  - reflect the social nature of individual or groups of agents
  - arise from interactions and relationships with other agents
    - agent A: "If I buy an item from B, I am *obliged* to pay B for the item"
  - obligations can be violated because agents are autonomous
    - e.g. if a desire is stronger than an obligation
      - "I am obliged to pay for this, but I want to keep my money, so I won't pay"
  - obligations of different "sources" can conflict:
    - e.g. obligation to agent B to achieve p and obligation to C to achieve $\neg p$

Mental attitudes in BOID: Intentions

- **Intentions**: commmitents and plans

Mental attitudes in BOID: Desires

- **Desires**: wishes and wants
  - internal motivations and long-term preferences
    - resemble an emotional process
  - generally an agent can select its goals, but not its desires
  - desires can be in conflict with each other
    - as opposed to goals

More on obligations

- obligations are often associated with penalties (sanctions) that apply when they are not fulfilled
  - e.g. if agent *A* does not pay for an item he bought, he will be punished
- there may be some mechanism, organization or other body which is responsible for enforcing the penalty
  - e.g. the police deals with agents who steal
- usually obligations on actions carry a time aspect
  - indicates that the action should be performed before a certain deadline
  - e.g. agent *B* expects *A* to pay for the item now and not later

Authorization

- Authorization is the counterpart of an obligation
- It describes the same dependency between agents as the obligation but from the point of view of the other agent
- Generally: If an agent has the authorization to perform some action it has some basis on which to justify it
- Example:
  - if *A* is authorized to demand payment from *B* then *B* is obligated to pay after the demand to do so
  - this is not the case if *A* is not authorized
- Different ways to implement authorizations (hardcoded, based on current role, etc)

BOID agents

- Different agent types
  - e.g. *realistic*, *social*, *selfish*, *stable*
  - type of agent depends on the **priority** of the mental attitudes (B., O., I. and D.) in the agent‘s goal generation routine
- Priority as a means to conflict resolution
  - e.g. conflict between desire and obligation
    - "I desire to quit my job, but I am obligated to pay my bills"
  - using preference relations of the rules and a feedback loop

Resolving conflicts between attitudes

- a mental attitude conflicts with another mental attitude if both cannot be used to generate a consistent set of goals
- to resolve the conflict, choose one attitude to override the other
- order of conflict resolution determines agent type
  - e.g. in *realistic* agents the belief component overrules any other component (B > OID)

Agent types by conflict resolution approach

- B > OID: *realistic*
- I > DO: *stable* (or *simple-minded*)
- D > O: *selfish*
- O > D: *social*
- agent types can be combined
  - e.g. I > O > D: stable and social
- an agent type is called *primitive* if it contains only one constraint
- an agent type is called *complete* if it induces a total strict ordering on the components

Twelve primitive agent types

Constraints		Agent type
$B \succ O$	$(O \succ B)$	Realistic relative to obligations (dogmatic)
$B \succ I$	$(I \succ B)$	Realistic relative to intentions (over-committed)
$B \succ D$	$(D \succ B)$	Realistic relative to desires (wishful thinker)
$O \succ I$	$(I \succ O)$	(Un)Stable relative to obligations
$O \succ D$	$(D \succ O)$	Social (selfish)
$I \succ D$	$(D \succ I)$	(Un)Stable relative to desires

Realistic agents

- realistic agents are the most common agent type
- example:
  - a conflict between a **belief** and a **prior intention** means that an intended action should no longer be executed due to the changing environment (and should be removed)
  - a conflict between a **belief** and **obligation or desire** means that a violation has occured
    - beliefs must overrule desires, otherwise there is *wishful thinking*

Six complete realistic agent types

Constraints	Agent type
$B \succ O; O \succ I; I \succ D$	Realistic, unstable-O, stable-D, social
$B \succ O; O \succ D; D \succ I$	Realistic, unstable-O, unstable-D, social
$B \succ I; I \succ O; O \succ D$	Realistic, stable-O, stable-D, social
$B \succ I; I \succ D; D \succ O$	Realistic, stable-O, stable-D, selfish
$B \succ D; D \succ O; O \succ I$	Realistic, unstable-O, unstable-D, selfish
$B \succ D; D \succ I; I \succ O$	Realistic, stable-O, unstable-D, selfish

Conflict example (expanded)

- if the agent is at work, it desires to drink coffee
- if the agent is at the beach, it is obligated to wear a bathing suit

- agents should consider the effects of actions before they commit to them ("think ahead")

Conflict example (cont.)

- goal sets: {{work, drink_coffee}, {swim, wear_bathing_suit}}
- agent must commit to single goal set
- agent desires to be on the beach
  - only way to achieve goal is to quit the job
  - consider necessary actions and their side-effects
    - if the agent quits its job and drives to the beach, then it will be poor
    - if the agent is poor, it does not want to be on the beach, but it wants to work

Working with rules in BOID

- rules are propositional formulas of the type $a \underset{X}{\rightarrow} b$, with b a conjunction of literals and $X \in \{B, O, I, D\}$
  - they define the inference steps to make
- $a \underset{B}{\rightarrow} b$: if a is derived as a goal, then the agent **believes** that as a consequence, b is a goal
  - $\underset{O}{\rightarrow}$: is **obligated**, $\underset{I}{\rightarrow}$: **intends**, $\underset{D}{\rightarrow}$: **desires**
- each component in the BOID goal generation process receives a set of rules (called an *extension*) as the input
- and outputs another extension

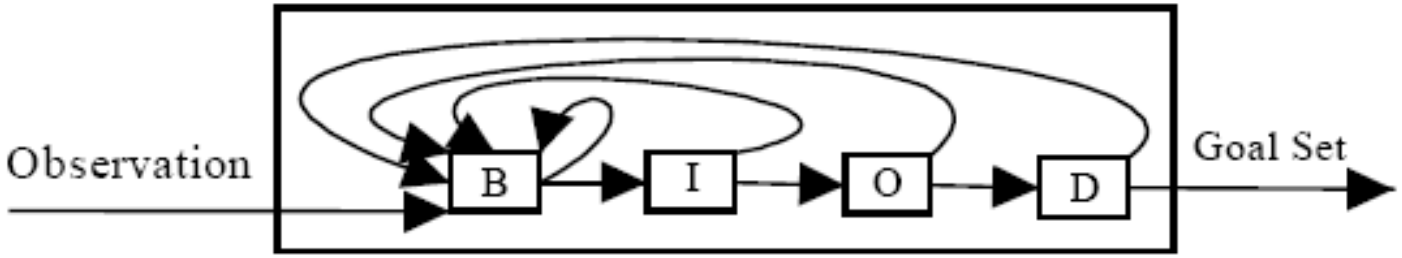
Blocked rules

- There might be other rules blocking the inference, such as $\top \underset{X}{\rightarrow} \neg b$
- *Default logic*
  - rules are only applied if they do not lead to an inconsistency
  - goal sets may conflict

BOID goal generation

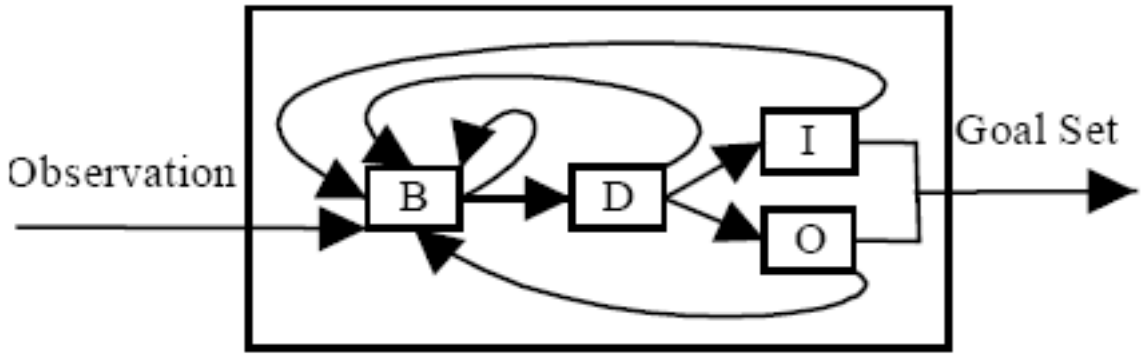
- let B, O, I, D be sets of rules and ρ a *priority function* from $B \cup O \cup I \cup D$ to the integers
- in case of multiple applicable rules, the one with the highest ρ value is applied
- to generate unique goal sets, ρ must be such that $\rho(x) = \rho(y)$ implies $x = y$ *(complete*)
  - this is not always enforced, leading to ambiguities
- If for all $X, Y \in \{B, O, I, D\}$ with $X \neq Y$ we have either $X \succ Y$ or $X \prec Y$, ρ induces a *strict component order*
  - satisfied only by the *complete* agent types

Example: Goal generation component for *stable social* agents



- Apply all **B** rules, generate extensions
- Pass extensions to the **I** component
  - Apply I rules
  - if we can obtain a new extension, feed back extensions to **B**
  - otherwise pass extensions to **O**
- and so on for the other components

Example: Goal generation component for *selfish unstable-D* agents



- After applying all **D** rules, choose either I or O
  - Choice can be implemented in different ways, e.g. deterministically or non-deterministically
- If no rule can be applied by **I**, try **O** (and vice versa)
  - or find some other way of combining extensions of equally-ranked components (probability distribution, ...)

BOID goal generation example

$(\text{go\_to\_conference} \wedge \text{cheap\_room}) \underset{B}{\rightarrow} \neg\text{close\_to\_conf\_site}$ $(\rho = 5)$

$(\text{go\_to\_conference} \wedge \text{close\_to\_conf\_site}) \underset{B}{\rightarrow} \neg\text{cheap\_room}$ $(\rho = 4)$

$\top \underset{I}{\rightarrow} \text{go\_to\_conference}$ $(\rho = 3)$

$\text{go\_to\_conference} \underset{O}{\rightarrow} \text{cheap\_room}$ $(\rho = 2)$

$\text{go\_to\_conference} \underset{D}{\rightarrow} \text{close\_to\_conf\_site}$ $(\rho = 1)$

$$(\text{go\_to\_conference} \wedge \text{cheap\_room}) \underset{B}{\rightarrow} \neg\text{close\_to\_conf\_site} \qquad (\rho = 5)$$

$$(\text{go\_to\_conference} \wedge \text{close\_to\_conf\_site}) \underset{B}{\rightarrow} \neg\text{cheap\_room} \qquad (\rho = 4)$$

$$\top \underset{I}{\rightarrow} \text{go\_to\_conference} \qquad (\rho = 3)$$

$$\text{go\_to\_conference} \underset{O}{\rightarrow} \text{cheap\_room} \qquad (\rho = 2)$$

$$\text{go\_to\_conference} \underset{D}{\rightarrow} \text{close\_to\_conf\_site} \qquad (\rho = 1)$$

- let the input of the agent be empty
- we derive {go_to_conference, cheap_room, ¬close_to_conf_site}

$$(\text{go\_to\_conference} \wedge \text{cheap\_room}) \underset{B}{\rightarrow} \neg\text{close\_to\_conf\_site} \qquad (\rho = 5)$$

$$(\text{go\_to\_conference} \wedge \text{close\_to\_conf\_site}) \underset{B}{\rightarrow} \neg\text{cheap\_room} \qquad (\rho = 4)$$

$$\top \underset{I}{\rightarrow} \text{go\_to\_conference} \qquad (\rho = 3)$$

$$\text{go\_to\_conference} \underset{O}{\rightarrow} \text{cheap\_room} \qquad (\boldsymbol{\rho = 1})$$

$$\text{go\_to\_conference} \underset{D}{\rightarrow} \text{close\_to\_conf\_site} \qquad (\boldsymbol{\rho = 2})$$

- we derive {go_to_conference, close_to_conf_site, ¬cheap_room} (selfish behavior)

$(\text{go\_to\_conference} \wedge \text{cheap\_room}) \underset{B}{\rightarrow} \neg\text{close\_to\_conf\_site}$ $(\rho = 5)$

$(\text{go\_to\_conference} \wedge \text{close\_to\_conf\_site}) \underset{B}{\rightarrow} \neg\text{cheap\_room}$ $(\rho = 4)$

$\top \underset{I}{\rightarrow} \text{go\_to\_conference}$ $(\rho = 3)$

$\text{go\_to\_conference} \underset{O}{\rightarrow} \text{cheap\_room}$ $(\boldsymbol{\rho = 2})$

$\text{go\_to\_conference} \underset{D}{\rightarrow} \text{close\_to\_conf\_site}$ $(\boldsymbol{\rho = 2})$

- we derive both $\{\text{go\_to\_conference}, \text{cheap\_room}, \neg\text{close\_to\_conf\_site}\}$ and $\{\text{go\_to\_conference}, \text{cheap\_room}, \neg\text{cheap\_room}\}$
- conflict! agent has to select one of the candidate goal sets by some other means

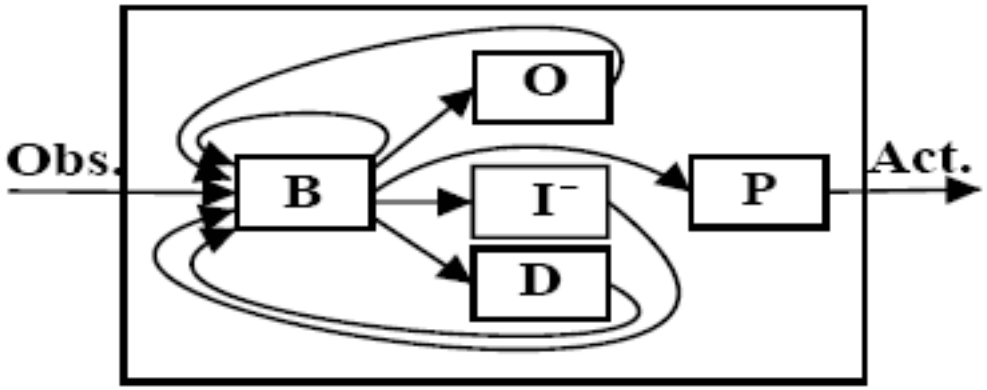
Working with rules in BOID (cont.)

- for the full BOID goal generation process, the input and output are sets of extensions
- need another component that selects **one** extension from the set of extensions
  - this extension (the new intentions) is the input for the planning component

Selecting the goal set

- different strategies for choosing the "best" extension set
  - select the smallest extension set
  - use domain knowledge (such as information about costs)
  - based on similarity with previously selected extensions (*persistent* agent type)

From goals to actions



- *Planning*: decide which **actions** should be performed to achieve the **goals** given by the calculated **extension**
- additional planning component P determines which actions to perform
- input of P: extension
- output of P: set of actions to be performed
- if the planning component decides to $do(b)$, then $\top \underset{I}{\rightarrow} b$ is added to the intentions in the next goal generation

Agent deliberation process

select ρ ;
repeat
    Obs := read_environment();
    S := generate_candidate_goal_sets(Obs, B, O, I, D, ρ);
    P := select_goal_set_and_generate_plans(S);
    update(B, O, I, D, ρ, P)
until forever

Desires as states instead of actions

- An agent may have two individual desires D(spend time with family) and D(give conference talk)
- But probably not the combined desire D(spend time with family $\wedge$ give conference talk)
  - in the sense of doing both concurrently
  - even if desires are not inconsistent, doing them at the same time might be undesirable
- Problem can be avoided by using states as desires: D(have

Norms

Norms

- agents joining a group (society) of agents must undergo a process of socialisation
- they are required to accept the normative conventions (*norms*), the rules of behaviour of the group
  - norms can be seen as the desires of society
- while obligations are agreements and commitments between individual agents, norms are "obligations" to the society as a whole

Why bother with norms at all?

- norms reduce the amount of computation required to make a decision
  - behaviours of others can be anticipated (with some degree of reliability)
  - knowledge of norms allows for easier coordination

How does an agent become aware of norms?

- norms can be hard-coded or given to an agent by an authoritative leader in the system
  - this is called the *top-down approach*
- better: agent is able to infer norms of a newly joined society
  - infer/identify norms by observing patterns of interactions and their consequences
  - can be done by e.g. association rule mining
  - *bottom-up approach*
- norms may change over time and an agent should be able to adapt to a changing environment

Where do norms come from?

- *legalistic* (or *prescriptive*) *view:* norms used to regulate emerging behavior of open systems
  - commonly, agents are "motivated" by sanctions to stick to norms
- *interactionist view* (or *emergent*): autonomous norms as regularities of behavior which emerge without any enforcement systems
  - agents conform to them e.g. because their goals happen to coincide
  - (authoritative) sanctions are not always necessary
    - instead, social blame or exclusion from the group is often enough

Different types of norms

- Prohibition norms
  - occurence of an event causes a sanction to occur
  - e.g. littering causes a sanction in a park
- Obligation norms
  - *absence* of an event causes a sanction to occur
  - e.g. waiter in a restaurant may sanction a customer for not tipping
  - harder to detect than prohibitions

What about hardwiring the norms?

- Besides autonomy, an important characteristic of agents is that they can react to a changing environment
- Hardwiring in the sense of a simple filter on the possible goals of an agent
  - i.e. always obey the norms
  - obeying a norm should be a motivated, 'conscious' decision
  - have to be able to reason about applying the norms
    - norms may conflict with desires or other norms
- We want to allow explicit reasoning about norms and obligations, because circumstances might change, which may make norms obsolete or suggest new or modified norms
- although in many implementations, this is fixed

Hardwiring obligations

- Same reasoning applies: if an agent notices that another agent is cheating it should be able to switch to another protocol to protect itself
- Generally, there might be circumstances in which the agent violates a convention in order to adhere to a private goal that it considers to be more important (more profitable)

Formalization of norms and obligations

- As a modal logic
- $N^z(p|q)$ - it is a norm of the society or organization z that p should be true when q is true
- $O^z_{ab}(p|q)$ - when q is true, individual a is obliged to b that p should be true. z is the organisation/society that is responsible for enforcing the penalty
- plus a preference ordering on norms and obligations

Motivation

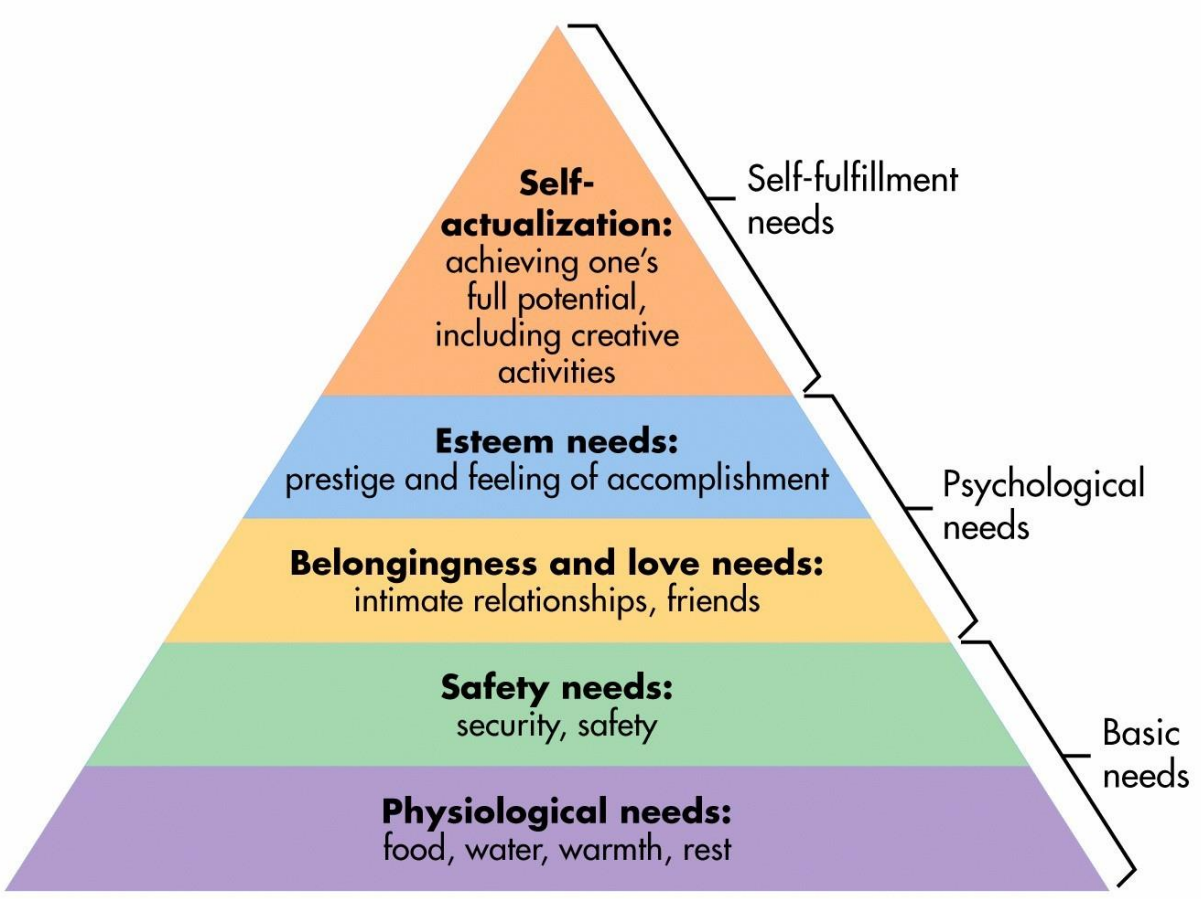
Motivation

- *motivations* (or *motives*) are preference relations over the desires
- in *motivated agents*, desires are added by a motivational component
- example motivations: greed, altruism, hunger, love
  - commonly more abstract (high-level) than desires
  - cannot be considered desires, but rather the reasons for desires

More on motivation

- agents may have multiple motivations
  - in this case, a motive hierarchy has to be established
    - e.g. hierarchy of needs
- motivations further characterize agents

Maslow's Hierarchy of Needs



How do motivations affect desires?

- link motives to desires by a *coupling strength* $\in [-1,1]$ and a condition
  - e.g.:
    - (thirst, drink_water, 0.3, at_girlfriends_place)
    - (thirst, drink_beer, 0.9, at_home)
- motives have an associated weight that can change over time
  - e.g. thirst increases over time
- generate the *motivation value* of a desire by combining all applicable couplings and the respective motive weights
- (if the motivation value exceeds a certain treshold, add it to the list of desires)

Why bother with motivation?

- BDI models assume the desires of an agent are given
  - for simple agents with limited capabilites and tasks this is often sufficient
- for more complex agents/environments, motivations are mechanisms that allow an agent to generate and set its own desires
- in this case, motivation drives the behaviour of the agent

Motivated cooperation

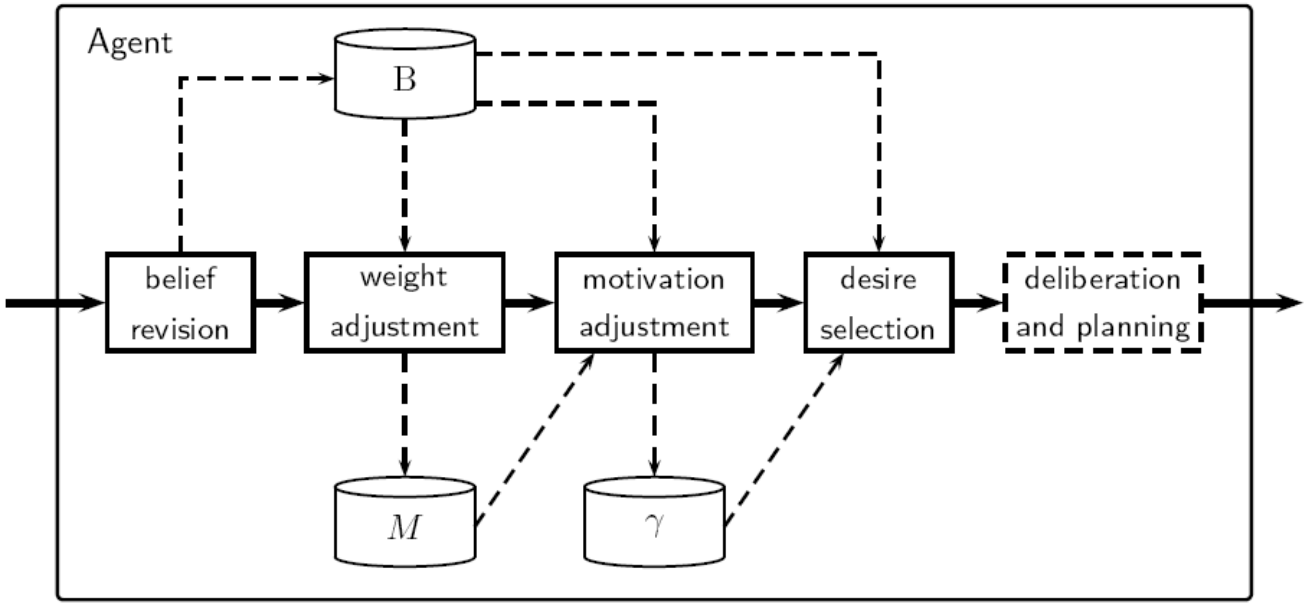
- cooperation arises when an agent's goal:
  - cannot be achieved alone
  - or is better achieved through cooperation
- in such situations an agent uses its knowledge and trust of others to determine who to ask for assistance
  - on receiving a request for assistance, agents inspect their own motivations and commitments (or intentions) to decide how to respond
- motivation determines whether agents *want* to cooperate

Issues with motivated cooperation

- motivations change over time
- cooperations that would be valuablee in the long-term may be rejected
- short-term view of cooperation leads to missed opportunities for cooperation
  - agents' goals may be "out of step"

Model of a motivated agent

- solid lines: action flow
- dashed lines: information flow



Motivated agents

- Motivated agents are agents whose autonomy results from motivations
- Motivations can be though of as an agent's high level desires, guiding all aspects of its behaviour
- The intensities of an agent's motivations change in accordance with its beliefs (which are determined by perceptions)
- When the intensity of a motivation exceeds its threshold, a set of goals is generated