Cambridge University Press & Assessment
978-1-107-03727-4 — Automated Planning and Acting
Malik Ghallab , Dana Nau , Paolo Traverso
Excerpt
More Information

# CHAPTER 1 Introduction

This chapter introduces informally the concepts and technical material developed in the rest of the book. It discusses in particular the notion of *deliberation*, which is at the core of the interaction between planning and acting. Section 1.1 motivates our study of deliberation from a computational viewpoint and delineates the scope of the book. We then introduce a conceptual view of an artificial entity, called an *actor*, capable of acting deliberately on its environment, and discuss our main assumptions. Deliberation models and functions are presented next. Section 1.4 describes two application domains that will be simplified into illustrative examples of the techniques covered in rest of the book.

## 1.1 PURPOSE AND MOTIVATIONS

### 1.1.1 First Intuition

*What is deliberative acting*? That is the question we are studying in this book. We address it by investigating the computational reasoning principles and mechanisms supporting how to choose and perform actions.

We use the word *action* to refer to something that an agent does, such as exerting a force, a motion, a perception or a communication, in order to make a change in its environment and own state. An agent is any entity capable of interacting with its environment. An agent acting deliberately is motivated by some intended objective. It performs one or several actions that are justifiable by sound reasoning with respect to this objective.

*Deliberation* for acting consists of deciding which actions to undertake and how to perform them to achieve an objective. It refers to a *reasoning process*, both before and during acting, that addresses questions such as the following:

- If an agent performs an action, what will the result be?
- Which actions should an agent undertake, and how should the agent perform the chosen actions to produce a desired effect?

Such reasoning allows the agent to predict, to decide what to do and how do it, and to combine several actions that contribute jointly to the objective. The reasoning consists in using predictive models of the agent's environment and capabilities to simulate what will happen if the agent performs an action. Let us illustrate these abstract notions intuitively.

**Example 1.1.** Consider a bird in the following three scenes:

- To visually track a target, the bird moves its eyes, head, and body.
- To get some food that is out of reach, the bird takes a wire rod, finds a wedge to bend the wire into a hook, uses the hook to get the food.
- To reach a worm floating in a pitcher, the bird picks up a stone and drops it into the pitcher, repeats with other stones until the water has risen to a reachable level, and then picks up the worm. □

Example 1.1 mentions actions such as moving, sensing, picking, bending and throwing. The first scene illustrates a precise coordination of motion and sensing that is called visual servoing. This set of coordinated actions is certainly purposeful: it aims at keeping the target in the field of view. But it is more *reactive* than deliberative. The other two scenes are significantly more elaborate: they demand reasoning about causal relations among interdependent actions that transform objects, and the use of these actions to achieve an objective. They illustrate our intuitive notion of acting deliberately.

The mechanisms for acting deliberately have always been of interest to philosophy.[1] They are a subject of intense research in several scientific disciplines, including biology, neuroscience, psychology, and cognitive sciences. The deliberative bird behaviors of Example 1.1 have been observed and studied from the viewpoint of how deliberative capabilities are developed, in species of corvids such as crows [597] or rooks [71, 70]. Numerous other animal species have the ability to simulate their actions and deliberate on the basis of such simulations.[2] The sophisticated human deliberation faculties are the topic of numerous research, in particular regarding their development in infants and babies, starting from the work of Piaget (as in [478, 479]) to the recent diversity of more formal psychology models (e.g., [563, 19, 461]).

We are interested here in the study of computational deliberation capabilities that allow an *artificial* agent to reason about its actions, choose them, organize them purposefully, and act deliberately to achieve an objective. We call this artificial agent an *actor*. This is to underline the acting functions on which we are focusing and to differentiate them from the broader meaning of the word "agent." We consider *physical actors* such as robots, as well as *abstract actors* that act in simulated or virtual environments, for example, through graphic animation or electronic Web transactions. For both kinds of actors, sensory-motor functions designate in a broad sense the low-level functions that implement the execution of actions.

[1] In particular, the branch of philosophy called *action theory*, which explores questions such as, "What is left over if I subtract the fact that my arm goes up from the fact that I raise my arm?" [610].

[2] In the interesting classification of Dennett [150], these species are called *Popperian*, in reference to the epistemologist Karl Popper.

### 1.1.2 Motivations

We address the issue of how an actor acts deliberately by following the approaches and methods of *artificial intelligence* (AI). Our purpose proceeds from the usual motivations of AI research, namely:

- To understand, through effective formal models, the cognitive capabilities that correspond to acting deliberately.
- To build actors that exhibit these capabilities.
- To develop technologies that address socially useful needs.

Understanding deliberation is an objective for most cognitive sciences. The specifics of AI are to model deliberation through computational approaches that allow us to explain as well as to generate the modeled capabilities. Furthermore, the investigated capabilities are better understood by mapping concepts and theories into designed systems and experiments to test empirically, measure, and qualify the proposed models. The technological motivation for endowing an artificial actor with deliberation capabilities stems from two factors:

- *autonomy* – that is, the actor performs its intended functions without being directly operated by a person and
- *diversity* in the tasks it can perform and the environments in which it can operate.

Without autonomy, a directly operated or teleoperated device does not usually need to deliberate. It simply extends the acting and sensing capabilities of a human operator who is in charge of understanding and decision making, possibly with the support of advice and planning tools, for example, as in surgical robotics and other applications of teleoperation.

An autonomous system may not need deliberation if it operates only in the fully specified environment for which it has been designed. Manufacturing robots autonomously perform tasks such as painting, welding, assembling, or servicing a warehouse without much deliberation. Similarly, a vending machine or a driverless train operates autonomously without a need for deliberation. For these and similar examples of automation, deliberation is performed by the designer. The system and its environment are engineered so that the only variations that can occur are those accounted for at the design stage in the system's predefined *functioning envelope*. Diversity in the environment is not expected. A state outside of the functioning envelope puts the system into a failure mode in which a person takes deliberate actions.

Similarly, a device designed for a unique specialized task may perform it autonomously without much deliberation, as long the variations in its environment are within its designed range. For example, a vacuum-cleaning or lawn-mowing robot does not deliberate yet can cope autonomously with its specialized tasks in a reasonable range of lawns or floors. But it may cease to function properly when it encounters a slippery floor, a steep slope, or any condition outside of the range for which it was designed.

When a designer can account, within some functioning envelope, for all the environments and tasks a system will face and when a person can be in charge of deliberating outside of this envelope, by means of teleoperation or reprogramming, then deliberation generally is not needed in the system itself. Such a system will be endowed with a library of *reactive* behaviors (e.g., as the bird's visual target tracking in Example 1.1) that cover efficiently its functioning envelope. However, when an autonomous actor has to face a diversity of tasks, environments and interactions, then achieving its purpose will require some degree of deliberation. This is the case in many robotics applications, such as service and personal robots, rescue and exploration robots, autonomous space stations and satellites, or even driverless cars. This holds also for complex simulation systems used in entertainment (e.g., video games) or educational applications (serious games). It is equally applicable to many control systems that manage complex infrastructures such as industrial or energy plants, transportation networks, and urban facilities (smart cities).

Autonomy, diversity in tasks and environments, and the need for deliberation are not binary properties that are either true or false. Rather, the higher the need for autonomy and diversity, the higher the need for deliberation. This relationship is not restricted to artificial systems. Numerous natural species (plants and some invertebrates such as sponges or worms) have been able to evolve to fit into stable ecological niches, apparently without much deliberation. Species that had to face rapid changes in their environment and to adapt to a wide range of living conditions had to develop more deliberation capabilities.

### 1.1.3 Focus and Scope

We address deliberation from an AI viewpoint. Our focus is on the *reasoning functions* required for acting deliberately. This focus involves two restrictions:

- We are not interested in actions that consists solely of internal computations, such as adding "2 + 3" or deducing that "Socrates is mortal." These computations are not actions that change the state of the world.[3] They can be used as part of the actor's deliberation, but we take them as granted and outside of our scope.
- We are not concerned with techniques for designing the sensing, actuation, and sensory-motor control needed for the low-level execution of actions. Sensory-motor control (e.g., the visual servoing of Example 1.1) can be essential for acting, but its study is not within our scope. We assume that actions are performed with a set of primitives, which we will call *commands*, that implement sensory-motor control. The actor performs its actions by executing commands. To deliberate, it relies on models of how these commands work.

The scope of this book is not limited to the most studied deliberation function, which is *planning* what actions to perform. Planning consists in choosing and organizing the actions that can achieve a given objective. In many situations, there is not much need for planning: the actions to perform are known. But there is a need for significant deliberation in deciding *how* to perform each action, given the context and changes in the environment. We develop the view that planning can be needed for deliberation but is seldom sufficient. We argue that acting goes beyond the execution of low-level commands.

[3] The borderline between computational operations and actions that change the external world is not as sharp for an abstract actor as for a physical one.

**Example 1.2.** Dana finishes breakfast in a hotel restaurant, and starts going back to his room. On the way, he notices that the elevator is not on his floor and decides to walk up the stairs. After a few steps he becomes aware that he doesn't have his room key, which he left on the breakfast table. He goes back to pick it up. □

In this example, the actor does not need to plan the simple task of going to his room. He *continually* deliberates while acting: he makes opportunistic choices, simulates in advance and monitors his actions, stops when needed and decides on alternate actions.

Deliberation consists of reasoning with predictive models as well as *acquiring* these models. An actor may have to *learn* how to adapt to new situations and tasks, as much as to use the models it knows about for its decision making. Further, even if a problem can be addressed with the actor's generic models, it can be more efficient to transform the explicit computations with these models into low-level sensory-motor functions. Hence, it is natural to consider learning to act as a deliberation function. Section 7.3 offers a brief survey on learning and model acquisition for planning and acting. However, our focus is on deliberation techniques using predefined models.

## 1.2 CONCEPTUAL VIEW OF AN ACTOR

### 1.2.1 A Simple Architecture

An actor interacts with the external environment and with other actors. In a simplified architecture, depicted in Figure 1.1(a), the actor has two main components: a set of *deliberation functions* and an *execution platform*.

The actor's sensory-motor functions are part of its execution platform. They transform the actor's commands into actuations that execute its actions (e.g., the movement of a limb or a virtual character). The execution platform also transforms sensed signals into features of the world (e.g., to recognize a physical or virtual object, or to query information from the Web). The capabilities of the platform are explicitly described as models of the available commands.

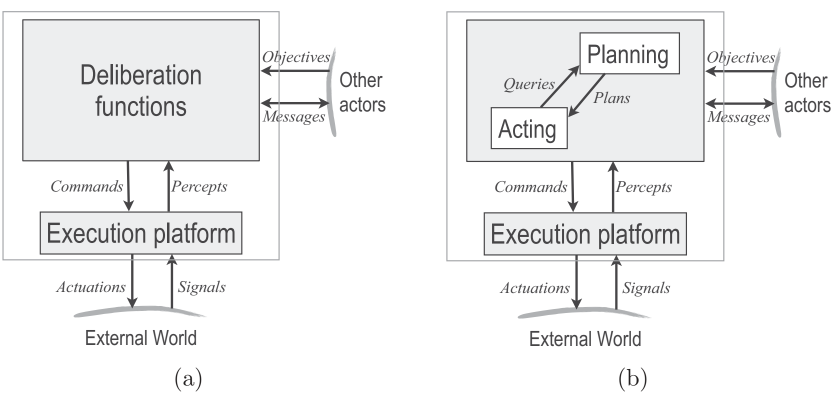


**Figure 1.1.** Conceptual view of an actor (a); its restriction to planning and acting (b).

Deliberation functions implement the reasoning needed to choose, organize, and perform actions that achieve the actor's objectives, to react adequately to changes in the environment, and to interact with other actors, including human operators. To choose and execute commands that ultimately achieve its objectives, the actor needs to perform a number of deliberation functions. For example, the actor must commit to intermediate goals, plan for those goals, refine each planned action into commands, react to events, monitor its activities to compare the predicted and observed changes, and decide whether recovery actions are needed. These deliberation functions are depicted in Figure 1.1(b) as two main components: planning and acting. The acting component is in charge of refining actions into commands, reacting to events, and monitoring.

### 1.2.2 Hierarchical and Continual Online Deliberation

The view presented in Section 1.2.1 can be a convenient first approach for describing an actor, but one must keep in mind that it is an oversimplification.

**Example 1.3.** To respond to a user's request, a robot has to bring an object o7 to a location room2 (see Figure 1.2). To do that, it plans a sequence of abstract actions such as "navigate to," "fetch," and "deliver." One of these refines into "move to door," "open door," "get out," and "close door." Once the robot is at the door, it refines the "open door" action appropriately for how it perceives that particular door.

The robot's deliberation can be accomplished by a collection of hierarchically organized components. In such a hierarchy, a component receives tasks from the component

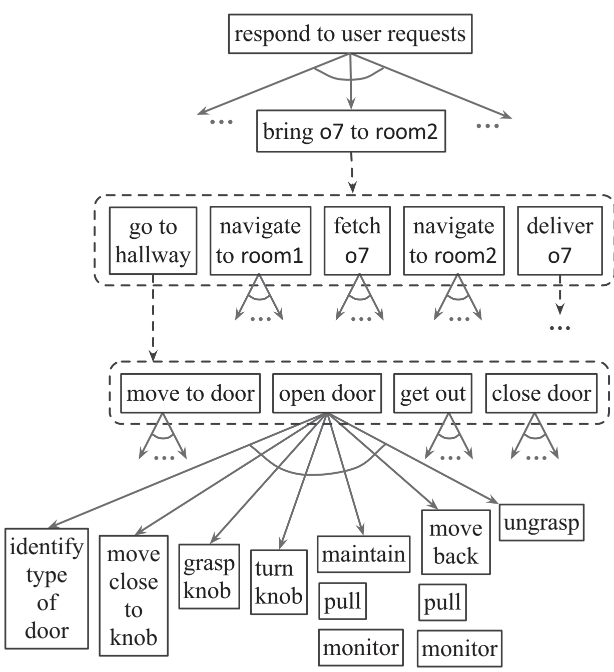


Figure 1.2. Multiple levels of abstraction in deliberative acting. Each solid arrow indicates a refinement of an abstract action into more concrete ones. Each dashed arrow maps a task into a plan of actions.

above it, and decides what activities need to be performed to carry out those tasks. Performing a task may involve refining it into lower-level steps, issuing subtasks to other components below it in the hierarchy, issuing commands to be executed by the platform, and reporting to the component that issued the task. In general, tasks in different parts of the hierarchy may involve concurrent use of different types of models and specialized reasoning functions. □

This example illustrates two important principles of deliberation: hierarchical organization and continual online processing.

- *Hierarchically organized deliberation.* Some of the actions the actor wishes to perform do not map directly into a command executable by its platform. An action may need further refinement and planning. This is done online and may require different representations, tools, and techniques from the ones that generated the task. A hierarchized deliberation process is not intended solely to reduce the search complexity of offline plan synthesis. It is needed mainly to address the heterogeneous nature of the actions about which the actor is deliberating, and the corresponding heterogeneous representations and models that such deliberations require.
- *Continual online deliberation.* Only in exceptional circumstances will the actor do all of its deliberation offline before executing any of its planned actions. Instead, the actor generally deliberates at runtime about how to carry out the tasks it is currently performing. The deliberation remains partial until the actor reaches its objective, including through flexible modification of its plans and retrials. The actor's predictive models are often limited. Its capability to acquire and maintain a broad knowledge about the current state of its environment is very restricted. The cost of minor mistakes and retrials are often lower than the cost of extensive modeling, information gathering, and thorough deliberation. Throughout the acting process, the actor refines and monitors its actions; reacts to events; and extends, updates, and repairs its plan on the basis of its perception focused on the relevant part of the environment.

Different parts of the actor's hierarchy often use different representations of the state of the actor and its environment. These representations may correspond to different amounts of detail in the description of the state and different mathematical constructs. In Figure 1.2, a graph of discrete locations may be used at the upper levels, while the lower levels may use vectors of continuous configuration variables for the robot limbs.

Finally, because complex deliberations can be compiled down by learning into low-level commands, the frontier between deliberation functions and the execution platform is not rigid; it evolves with the actor's experience.

### 1.2.3 Assumptions

We are not seeking knowledge representation and reasoning approaches that are effective across every kind of deliberation problem and at every level of a hierarchically organized actor. Neither are we interested in highly specialized actors tailored for a single niche, because deliberation is about facing diversity. Instead, we are proposing a few generic approaches that can be adapted to different classes of environments and, for a given actor, to different levels of its deliberation. These approaches rely on restrictive assumptions that are needed from a computational viewpoint, and that are acceptable for the class of environments and tasks in which we are interested.

Deliberation assumptions are usually about how variable, dynamic, observable, and predictable the environment is, and what the actor knows and perceives about it while acting. We can classify them into assumptions related to the dynamics of the environment, its observability, the uncertainty managed in models, and how time and concurrency are handled.

- *Dynamics* of the environment. An actor may assume to be in a static world except for its own actions, or it may take into account exogenous events and changes that are expected and/or observed. In both cases the dynamics of the world may be described using discrete, continuous or hybrid models. Of these, hybrid models are the most general. Acting necessarily involves discontinuities in the interaction with the environment,[4] and these are best modeled discretely. But a purely discrete model abstracts away continuous processes that may also need to be modeled.
- *Observability* of the environment. It is seldom the case that all the information needed for deliberation is permanently known to the actor. Some facts or parameters may be always known, others may be observable if specific sensing actions are performed, and others will remain hidden. The actor may have to act on the basis of reasonable assumptions or beliefs regarding the latter.
- *Uncertainty* in knowledge and predictions. No actor is omniscient. It may or may not be able to extend its knowledge with specific actions. It may or may not be able to reason about the uncertainty regarding the current state of the world and the predicted future (e.g., with nondeterministic or probabilistic models). Abstracting away uncertainty during a high-level deliberation can be legitimate if the actor can handle it at a lower level and correct its course of action when needed.
- *Time and concurrency*. Every action consumes time. But deliberation may or may not need to model it explicitly and reason about its flow for the purpose of meeting deadlines, synchronizing, or handling concurrent activities.

Different chapters of the book make different assumptions about time, concurrency, and uncertainty. Except for Section 7.4 on hybrid models, we'll restrict ourself to discrete approaches. This is consistent with the focus and scope discussed in Section 1.1.3, because it is primarily in sensory-motor functions and commands that continuous models are systematically needed.

## 1.3 DELIBERATION MODELS AND FUNCTIONS

### 1.3.1 Descriptive and Operational Models of Actions

An actor needs predictive models of its actions to decide *what* actions to do and *how* to do them. These two types of knowledge are expressed with, respectively, descriptive and operational models.

[4] Think of the phases in a walking or grasping action.

- *Descriptive models* of actions specify the actor's "know what." They describe which state or set of possible states may result from performing an action or command. They are used by the actor to reason about what actions may achieve its objectives.
- *Operational models* of actions specify the actor's "know how." They describe how to perform an action, that is, what commands to execute in the current context, and how organize them to achieve the action's intended effects. The actor relies on operational models to perform the actions that it has decided to perform.

In general, descriptive models are more abstract than operational models. Descriptive models abstract away the details, and focus on the main effects of an action; they are useful at higher levels of a deliberation hierarchy. This abstraction is needed because often it is too difficult to develop very detailed predictive models, and because detailed models require information that is unknown at planning time. Furthermore, reasoning with detailed models is computationally very complex. For example, if you plan to take a book from a bookshelf, at planning time you will not be concerned with the available space on the side or on the top of the book to insert your fingers and extract the book from the shelf. The descriptive model of the action will abstract away these details. It will focus on where the book is, whether it is within your reach, and whether you have a free hand to pick it up.

The simplifications allowed in a descriptive model are not possible in an operational model. To actually pick up the book, you will have to determine precisely where the book is located in the shelf, which positions of your hand and fingers are feasible, and which sequences of precise motions and manipulations will allow you to perform the action.

Furthermore, operational models may need to include ways to respond to *exogenous* events, that is, events that occur because of external factors beyond the actor's control. For example, someone might be standing in front of the bookshelf, the stool that you intended to use to reach the book on a high shelf might be missing, or any of a potentially huge number of other possibilities might interfere with your plan.

In principle, descriptive models can take into account the uncertainty caused by exogenous events, for example, through nondeterministic or probabilistic models (see Chapters 5 and 6), but the need to handle exogenous events is much more compelling for operational models. Indeed, exogenous events are often ignored in descriptive models because it is impractical to try to model all of the possible joint effects of actions and exogenous events, or to plan in advance for all of the contingencies. But operational models must have ways to respond to such events if they happen because they can interfere with the execution of an action. In the library example, you might need to ask someone to move out of the way, or you might have to stand on a chair instead of the missing stool.

Finally, an actor needs descriptive models of the available commands in order to use them effectively, but in general it does not need their operational models. Indeed, commands are the lower-level sensory-motor primitives embedded in the execution platform; their operational models correspond to what is implemented in these primitives. Taking this remark to the extreme, if one assumes that every known action corresponds to an executable command, then all operational models are embedded in the execution platform and can be ignored at the deliberation level. This assumption seldom holds.

### 1.3.2 Description of States for Deliberation

To specify both descriptive and operational models of actions, we will use representational primitives that define the state of an actor and its environment; these are called *state variables*. A state variable associates a value, which changes over time, to a relevant attribute of the world. The definition of a state with state variables needs to include enough details for the actor's deliberations, but it does not need to be, nor can it be, exhaustive.

In a hierarchically organized actor, different deliberative activities may need different amounts of detail in the state description. For example, in actions such as "grasp knob" and "turn knob" at the bottom of Figure 1.2, to choose the commands for grasping the handle and operating it, the actor needs to reason about detailed parameters such as the robot's configuration coordinates and the position and shape of the door handle. Higher up, where the actor refines "bring o7 to room2" into actions such as "go to hallway" and "navigate to room1," such details are not needed. It is more convenient there to reason about the values of more abstract variables, such as $\mathsf{location(robot)} = \mathsf{room1}$ or $\mathsf{position(door)} = \mathsf{closed}$. To establish correspondences between these abstract variables and the detailed ones, the actor could have definitions saying, for example, that $\mathsf{location(robot)} = \mathsf{room1}$ corresponds to a particular area in an Euclidean reference frame.

The precise organization of a hierarchy of data structures and state representations is a well-known area in computer science (e.g., [522]). It may take different forms in application domains such as robotics, virtual reality, or geographic information systems. Here, we'll keep this point as simple as possible and assume that at each part of an actor's deliberation hierarchy, the state representation includes not only the variables used in that part of the hierarchy (e.g., the robot's configuration coordinates at the bottom of Figure 1.2), but also the variables used higher up in the hierarchy (e.g., location(robot)).

An important issue is the distinction and correspondence between *predicted* states and *observed* states. When an actor reasons about what might happen and simulates changes of state to assess how desirable a course of action is, it uses predicted states. When it reasons about how to perform actions in some context, it relies on observed states; it may contrast its observations with its expectations. Predicted states are in general less detailed than the observed one; they are obtained as a result of one or several predictions starting from an abstraction of the current observed state. To keep the distinction clear, we'll use different notations:

- $s \in S$ is a predicted state;
- $\xi \in \Xi$ is an observed state.

Because of partial and inaccurate observations, there can be uncertainty about the *present observed* state as well as about the *future predicted* states. Furthermore, information in a dynamic environment is ephemeral. Some of the values in $\xi$ may be out-of-date: they may refer to things that the actor previously observed but that it cannot currently