This article is part of the topic "Everyday Activities" Holger Schultheis and Richard P. Cooper (Topic Editor). For a full listing of topic papers, see https://onlinelibrary.wiley.com/toc/17568765/2021/earlyview.

Action Selection and Execution in Everyday Activities: A Cognitive Robotics and Situation Model Perspective

David Vernon,[a] Josefine Albert,[b,c] Michael Beetz,[a] Shiau-Chuen Chiou,[d] Helge Ritter,[d] Werner X. Schneider[b,c]

[a]*Institute for Artificial Intelligence, University of Bremen*
[b]*Center for Interdisciplinary Research (ZiF), Bielefeld University*
[c]*Neuro-cognitive Psychology, Department of Psychology, Bielefeld University*
[d]*Center for Cognitive Interaction Technology (CITEC), Bielefeld University*

Abstract

We examine the mechanisms required to handle everyday activities from the standpoint of cognitive robotics, distinguishing activities on the basis of complexity and transparency. Task complexity (simple or complex) reflects the intrinsic nature of a task, while task transparency (easy or difficult) reflects an agent's ability to identify a solution strategy in a given task. We show how the CRAM cognitive architecture allows a robot to carry out simple and complex activities such as laying a table for a meal and loading a dishwasher afterward. It achieves this by using generalized action plans that exploit reasoning with modular, composable knowledge chunks representing general knowledge to transform underdetermined everyday action requests into motion plans that successfully accomplish the required task. Noting that CRAM does not yet have the ability to deal with difficult activities, we leverage

---

insights from the situation model perspective on the cognitive mechanisms underlying flexible context-sensitive behavior with a view to extending CRAM to overcome this deficit.

1. Everyday activities: Simple and complex

Everyday activities are carried out routinely by humans. They often seem simple and straightforward when, in fact, they can be complex and demanding, at least at first. Let us consider why this is so. To begin with, some activities are inherently simple. They can be executed without much thought because it is easy to see what needs to be done and how: even if there are several steps in an activity, the relationship between these steps is limited, for example, with one step following another in a well-defined order, and with each step having little or no variety in the way it is carried out. These activities require little or no deliberation. Stacking plates on a shelf is an example because it involves repeatedly picking up plates and putting them down somewhere else. There is little room for variety or creativity. In contrast, filling a dishwasher with cups, saucers, plates, and cutlery is a complex task because doing it well requires objects to be selected and placed in appropriate positions so that they would not shield other objects, preventing water from reaching them. The dishwasher also needs to be filled compactly, and there are many different strategies one can apply to achieve this. However, with most tasks that initially are difficult for us, our experience is that with practice, the task gets easier and easier, and we may arrive at a point where we can perform the initially difficult task more or less habitually and with very little effort.

This leads us to distinguish between the *complexity* of a task versus its *difficulty* for a particular agent.

A task can be taken to have low complexity (i.e., it is simple) when it admits a solution that can be described as a short action pattern ("program"). For instance, stacking many plates can be formulated with a very compact action pattern and thus is simple, even if the sheer number of required action steps may be large. This intuitive characterization of complexity as a task-inherent property has received its rigorous mathematical foundation in Kolmogorov complexity theory (e.g., Li & Vitányi, 2008). It is not without subtleties. For instance, for an interactive robot, the "program" must be able to run within real-time bounds and include all prerequisites (e.g., feedback or even prospection mechanisms) that will be needed to cope with the potential variability of task instances. As a result, complexity depends on what constraints apply on the possible combination of steps involved in the task and the number of degrees of freedom in each step.

But low complexity (in the sense defined above) of a task does not always imply that it is also *easy for a particular agent*. We all remember tasks that we first experienced as very difficult but after some effort we managed to discover a short, perhaps even elegant, way of solving them. Such examples expose *task difficulty* as a property that is—unlike complexity—not a feature of the task itself, but instead measures how far the agent still struggles with unnecessary effort that would not be required, according to the true and intrinsic complexity of the
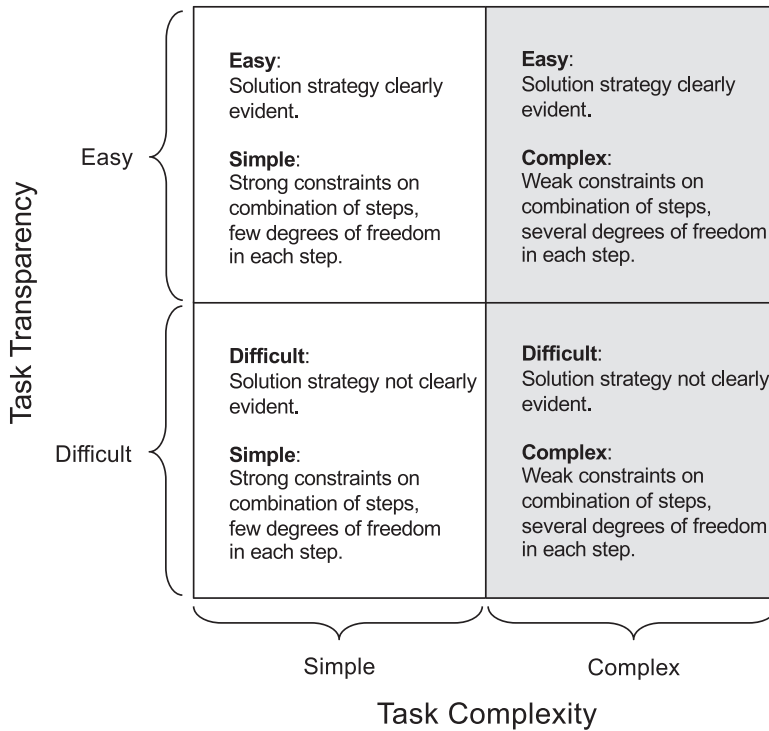
Fig. 1. Tasks can be characterized by their transparency and their complexity. Task transparency reflects the relative difficulty depending on the agent involved, its experience, and therefore its ability to easily identify a solution strategy or not. Task complexity reflects the intrinsic nature of task and depends on what constraints apply on the possible combination of steps involved in the task and the number of degrees of freedom in each step.

task. Thus, task difficulty captures the degree to which the solution strategy is "transparent" (i.e., clearly evident or not clearly evident) to the cognitive agent. We also see that while complexity is a static property of the task, its difficulty may change when the agent is adaptive and has occasion to become, through repeated trials, familiar with a task. To emphasize this distinction between "complexity" and "difficulty," we will in the following use the terms *task complexity* and *task transparency*, respectively.

These two dimensions can vary independently, giving rise to four different "quadrants" that classify situations within which a cognitive agent needs to operate (see Fig. 1).

Each quadrant thus may require an agent cognitive architecture to recruit different mechanisms, with some tasks being accomplished in a habitual manner and other tasks requiring more deliberation. For instance, in the lower left quadrant, the agent may need to deliberate even for a simple task (for instance, when the task is novel and the agent fails to see a solution in terms of habitual behaviors that are already at its disposal). However, with repeated routine execution, such tasks can be carried out with habitual behavior, moving them into the upper left quadrant. Similarly, under the right conditions, some complex tasks, too, can be carried out with habitual behaviors, moving them from lower right to upper right. We address these

conditions in Section 4. Note also that some everyday activities—simple or complex—that have become amenable to habitual execution can also on occasion require the recruitment of the mechanisms responsible for deliberative behavior, for example, in cases where the action fails or when something unexpected happens.

In the following, we describe the results of a research program to develop a cognitive architecture that allows a robot to successfully carry out easy simple and complex everyday activities and we discuss future plans to address difficult everyday activities. Research in cognitive robotics can contribute to the broader agenda of cognitive science by proposing information processing models, implementing them as robot control programs, and assessing their ability to perform everyday activities. Such research provides insights and let us test whether particular information processing principles and combinations of information processing components are sufficient for accomplishing concrete subclasses of everyday activity. Often, the information processing models used in cognitive robotics are informed through cognitive models of human or animal agency, which further increases the synergies between the fields.

An example of this line of research is presented by Kazhoyan and Beetz (2019a) who have designed and realized a robot control system that uses a single generalized action plan for fetching and placing objects. This robot control system has been demonstrated to be sufficient for accomplishing subsets of human-scale everyday manipulation tasks such as setting the table, cleaning up, and loading the dishwasher. Fig. 2 shows how one of the high-level actions (fetch some object) is translated into different low-level body motions (such as grasping) depending on the object, task, and context so that the action is performed successfully. The approach we describe is of interest for its generality and how the generality is realized: the cognitive architecture enables a robot agent to manipulate different kinds of objects, in different configurations of the environment, replacing the limbs, sensors, and actuators of the robot with different ones, when required.

In the next section, we continue with an overview of cognitive architectures, in general, and the CRAM robot cognitive architecture (Beetz, Mösenlechner, & Tenorth, 2010), in particular, and we show how a robot controlled by CRAM can accomplish easy simple and complex everyday activities such as setting a table for a meal and tidying up afterward. We also explain the limitations of CRAM at present and identify the need for more flexible cognitive behavior to deal with difficult tasks. We then provide a short overview of the situation model framework (Schneider, Albert, & Ritter, 2020) as a basis for achieving the flexible context-sensitive cognitive behavior required for difficult tasks and briefly discuss the implications of using it to extend CRAM in the future.

2. The CRAM cognitive architecture

2.1. Three types of cognitive architecture

The concept of a cognitive architecture is due to Allen Newell who made several landmark contributions to cognitive science and artificial intelligence, including the development
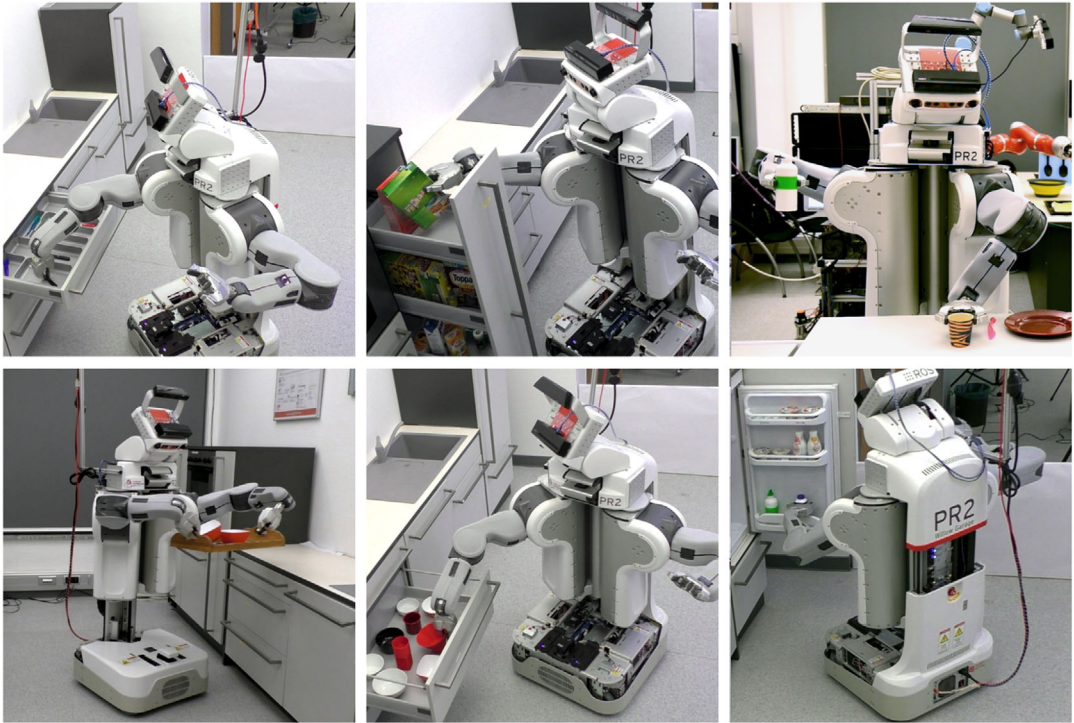
Fig. 2. A high-level action to fetch some object is translated to different low-level body motions, depending on the object, task, and context. The examples in this figure show the different grasps that are selected by the CRAM cognitive architecture when executing a fetch action with various objects. These examples are documented in a short video; see footnote.

of the Soar cognitive architecture for general intelligence (Laird, Newell, & Rosenbloom, 1987) and the concept of a unified theory of cognition (Newell, 1990). A cognitive architecture is a computational framework that integrates all the elements required for a system to exhibit the attributes that are considered to be characteristic of a cognitive agent. A cognitive architecture determines the overall structure and organization of a cognitive system, including the component parts or modules (Sun, 2004), the relations between these modules, and the essential algorithmic and representational details within them (Langley, 2006). The architecture specifies the formalisms for knowledge representations and the types of memories used to store them, the processes that act upon that knowledge, and the learning mechanisms that acquire it.

Existing cognitive architectures may be classified as three different types: the cognitivist, the emergent, and the hybrid. Cognitivist cognitive architectures, often referred to as symbolic cognitive architectures (Kotseruba & Tsotsos, 2020), focus on the aspects of cognition that are relatively constant over time and that are independent of the task (Langley, Laird, & Rogers, 2009; Ritter & Young, 2001), with knowledge providing the task-specific element.

The combination of a cognitive architecture and a particular knowledge set is often referred to as a cognitive model.

Emergent approaches to cognition focus on the development of the agent from a primitive state to a fully cognitive state over its lifetime. As such, an emergent cognitive architecture is both the initial state from which an agent subsequently develops and the encapsulation of the dynamics that drive the development, often exploiting subsymbolic processes and representations.

Hybrid architectures endeavor to combine the strengths of the cognitivist and emergent approaches. Most hybrid architectures focus on integrating symbolic and subsymbolic processing. Hybrid cognitive architectures are the most prevalent type: 48 of the 84 cognitive architectures surveyed by Kotseruba and Tsotsos (2020) are hybrid.

The majority of cognitive architectures focus on modeling human cognition. Some cognitive architectures are also a candidate unified theory of cognition. The cognitive architectures Soar (Laird, 2012; Laird et al., 1987), ACT-R (Anderson, 1996; Anderson et al., 2004), CLARION (Sun, 2007, 2016, 2017), and LIDA (Franklin, Madl, D'Mello, & Snaider, 2014; Ramamurthy, Baars, D'Mello, & Franklin, 2006) are archetypal candidate unified theories of cognition, all of which are classified as hybrid in the survey by Kotseruba and Tsotsos (2020).

In the following, we consider an alternative hybrid cognitive architecture, CRAM (Beetz et al., 2010; Beetz, Tenorth, & Winkler, 2015; Kunze & Beetz, 2017), to illustrate how cognitive robots can accomplish everyday activities, in general, and how they can tackle the problem of action selection and action execution, in particular. Rather than choosing a cognitive architecture that is a candidate-unified theory of cognition, for example, Soar, ACT-R, CLARION, or LIDA, we selected CRAM because it is a robot cognitive architecture that was specifically designed to enable robot agents to carry out everyday manipulation tasks. We acknowledge that Soar and LIDA, and a version of ACT-R—ACT-R/E (Trafton et al., 2013)—have been applied in robotics and that Soar has addressed the two-system approach (Laird & Mohan, 2018) entailed by the situation model framework described in Section 3.

2.2. Challenges of everyday activities

The CRAM cognitive architecture addresses five challenges in carrying out everyday activities.

First, requests for accomplishing everyday tasks are typically underdetermined. Requests such as "set the table," "load the dishwasher," and "prepare breakfast" typically do not fully specify the intended goal state, yet the requesting agent has specific expectations about the results of the activity. Consequently, the robot agent carrying out the everyday activity needs to acquire the missing knowledge and disambiguate information to accomplish the task in the expected manner. To set the table, a robot agent also needs to generate subtasks, such as putting each object on the table where it is expected to be.

Second, accomplishing actions requires context-specific behavior. The underdetermined request "fetch the next object and put it where it belongs" has to generate different behaviors depending on the object, its state, the current location of the object, the scene context, the destination of the object, and the task context. The behavior has to be carefully chosen

to match the current contextual conditions. Variations in these conditions require adaptive behavior.

Third, competence in accomplishing everyday manipulation tasks requires decisions based on knowledge, experience, and prediction. The knowledge required includes common sense, such as knowing that the tableware to be placed on the table should be clean and that clean tableware is typically stored in cupboards. It also requires intuitive physics knowledge, for example, that objects should be placed with their center of gravity close to the support surface to avoid them toppling over. Domain knowledge might include the fact that plates are made of porcelain, which is a breakable material. Experience allows the robot agent to improve the robustness and efficiency of its actions by tailoring behavior to specific contexts. Prediction enables the robot to take likely consequences of actions into account, such as predicting that a specific grasp would require the object to be subsequently regrasped in order to place it at the intended location.

Fourth, accomplishing everyday manipulation tasks requires the robot agent to reason at the motion level, predicting the way the parameterization of motions alters the physical effects of these motions, and thereby identifying the best way to achieve the intended outcomes and avoid unwanted side effects. For example, in order to pour something from a pot onto a plate, a robot agent has to infer that it has initially to hold the pot horizontally and then tilt it. To do this, it has to grasp the pot with two hands such that the center of mass is in between the hands because the tilting motion is the easiest when grasping the handles and rotating the pot around the axis between the handles.

CRAM also addresses a fifth challenge: a robot agent should be able to answer questions about what it is doing, why it is doing it, how it is doing it, what it expects to happen when it does it, how it could do it differently, what are the advantages and disadvantages of doing it one way or another, and so on. This applies both to making decisions and reasoning about motions. This ability is important because it allows the robot agent to assess the cost of not doing something effectively or failing to do it. This understanding also allows a robot agent to discover possibilities for improving the way it currently does things.

2.3. Overview of the CRAM approach to carrying out everyday activities

The components of the CRAM cognitive architecture are shown in Fig. 3. These are: (a) the CRAM Plan Language (CPL) executive; (b) a suite of knowledge bases and associated reasoning mechanisms, collectively referred to as KnowRob 2.0 (Beetz et al., 2018); (c) a perception executive referred to as RoboSherlock; (d) an action executive referred to as Giskard; and (v) a metacognition system referred to as COGITO. CRAM operates via a dialog between the plan executive and KnowRob 2.0, in which the plan executive presents a series of queries for motion plan parameter values[1] and KnowRob 2.0, as an implementation of the generative model,[1] provides the corresponding responses containing the required values.

The current abilities of CRAM to accomplish everyday activities have been demonstrated in an exercise requiring a robot agent to set and clean a table, given vague task requests. The approach we have adopted, explained more fully below, only requires a carefully designed *generalized action plan* for fetching and placing objects. This plan can be executed with
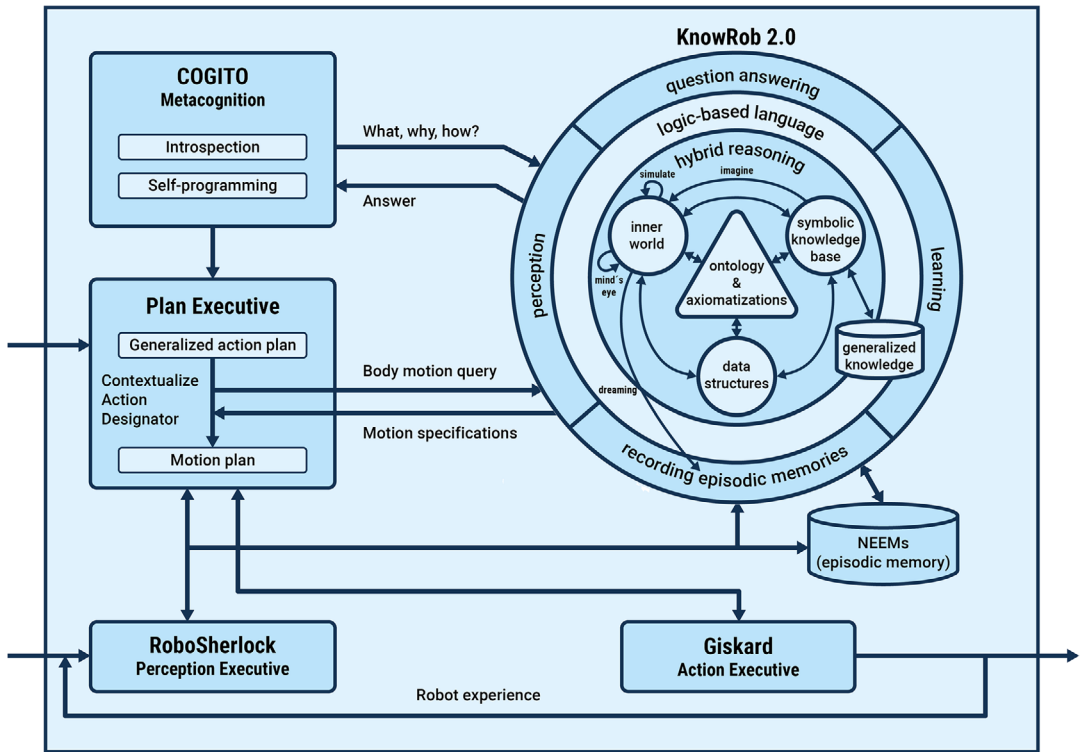
Fig. 3. The CRAM cognitive architecture. The inputs are underdetermined instructions, that is, underdetermined action or task descriptions (top left) and sensor data (bottom left). The output is the robot body motion which maximizes the likelihood of success of the action.

different types of robot and it can be used with other objects and in different environments. The plan is automatically contextualized for each individual object transportation task. Thus, the robot infers the body motions required to achieve the respective object transportation task depending on the type and state of the object to be transported (be it a spoon, bowl, cereal box, milk box, or mug), the original location (be it the drawer, the high drawer, or the table), and the task context (be it setting or cleaning the table, loading the dishwasher, or throwing away items) (Kazhoyan & Beetz, 2019b). In doing this, it avoids unwanted side effects (e.g., knocking over a glass when placing a spoon on the table). The body motions generated to perform the actions are varied and complex and, when required, include subactions such as opening and closing containers, as well as coordinated, bimanual manipulation tasks. The results of the exercise are documented in a short video[2] and Fig. 2 shows some examples of the robot grasping objects in different ways during this exercise. All of these grasp strategies are inferred from the robot's knowledge base: the spoon in the drawer is grasped from the top because it is a very flat object; the tray is grasped with two hands because the center of mass would be too far outside the hand for a single-hand grasp; the mug is grasped from the side and not at the rim because the purpose of grasping it is pouring liquid into the mug.
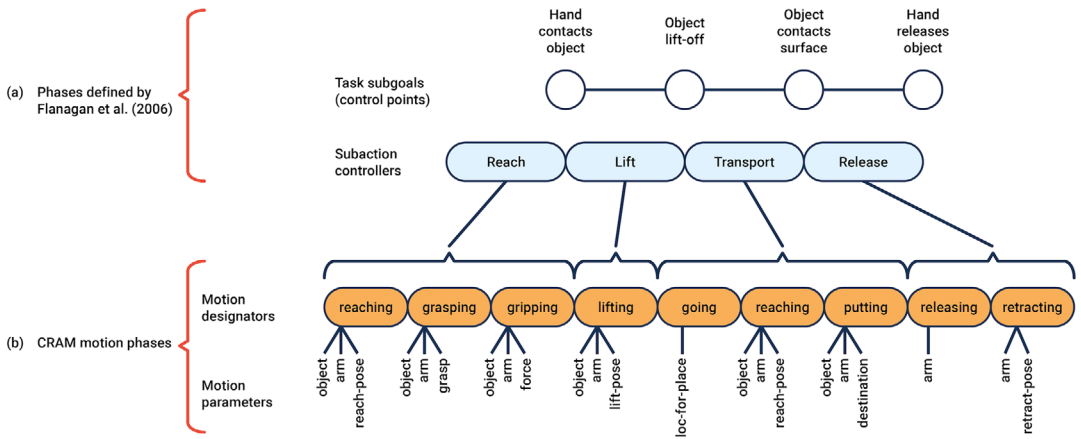
Fig. 4. Motion plan for the fetch and place generalized action plan. (a) The motion phases as described in (Flanagan, Bowman, & Johansson, 2006): motion phases, each governed by a subaction controller, are depicted in blue boxes, while the control points depicted by circles represent the task subgoals. (b) The realization of this approach in CRAM, showing the motion designators that comprise each motion phase depicted in orange boxes, together with their associated motion parameters. Note that Flanagan et al. (2006) refer to *action phase* rather than *motion phase*. We use *motion phase* here to connote a low level of abstraction and avoid confusion with the high level of abstraction generalized action plan.

There are five key elements in the CRAM approach to carrying out everyday activities.

First, there is the *underdetermined action description*. This is a high-level abstract specification of the robot actions required to carry out an everyday activity, for example, "set the table." As we noted already, these action specifications are framed in incomplete terms, that is, they do not provide all knowledge required to complete the task.

Second, there is the related concept of a *generalized action plan*. This is a computational encapsulation of an underdetermined action description that can be deployed in many everyday contexts. There is a generalized action plan for every category of underdetermined action description, each of which typically corresponds to action verbs such as *fetch*, *place*, *pour*, and *cut*. A generalized action plan specifies an action schema, that is, a template of how actions of this category can be executed.

Third, there is the *motion plan*. This is a sequence of parameterized motion phases. Each motion phase has several parameters that can be set to adapt the motions to the current context. The generalized action plan is expanded into a motion plan, and the motion plan is executed. The motion phase parameter values determine the exact nature of the motion. In structuring the motion plan, an approach similar to that described by Flanagan et al. (2006) is adopted, but with a slightly finer-grained representation of motion phases; see Fig. 4. Each motion phase has a goal. When the goal is achieved, the start of the subsequent motion phase is triggered. Goals can be force-dynamic events, for example, the robot finger coming into contact with the object to be grasped, or other perceptually distinctive events, for example, a milk carton becoming visible when a fridge door is opened.

The fourth element is the process of transforming the generalized action plan into the motion plan and the subsequent identification of the motion parameter values that maximize the likelihood that the associated body motions successfully accomplish the desired action. This process is called *contextualization*. It operates on an element of the generalized action plan known as an *action designator*, effectively a placeholder for information that is not yet known and that must be provided to carry out the motion plan.

The fifth element is the *generative model*. This is the means by which the motion parameter values are identified. It is a joint distribution of motion parameter values and the associated effects of performing these motions. It provides the basis for a mapping from desired outcomes of an action to the motion parameter values that are most likely to succeed in accomplishing the desired action. In CRAM, the generative model is realized through knowledge representation and reasoning, based on a robot's tightly coupled symbolic and subsymbolic knowledge, the tasks it is performing, the objects it is acting on, and the environment in which it is operating.

Let us see how the five elements and the five components of the CRAM cognitive architecture fit together and how they address the five challenges.[3] The CPL executive allows underdetermined action descriptions (element 1) expressed as generalized action plans (element 2) to be transformed into motion plans (element 3) through the contextualization process (element 4), using KnowRob 2.0 to fulfil the role of the generative model (element 5). RoboSherlock provides the perceptual input for the contextualization process, while Giskard is responsible for executing the motion plans. Cogito provides a capacity for improving performance through plan transformation, tailoring behaviors to specific tasks and context by enhancing the generative model. Thus, the CPL executive addresses challenge 1 (underdetermined action requests) through the use of generalized action plans and it addresses challenge 2 (context-specific behavior) by exploiting run time contextualization. KnowRob 2.0 uses reasoning and its several knowledge bases (see Fig. 3), as well as internal simulation with its inner world model, to address challenge 3 (decisions based on knowledge, experience, and prediction). In particular, it uses reasoning with the episodic memory knowledge base comprising the NEEMS[4] knowledge base (including motor control procedural data) and a generalized knowledge base to address challenge 4 (reasoning at the motion level). While performing an activity, the robot logs its perception and execution data, including sensory data and control signals. These records of external perceptions and the internal semantically annotated control structures allow KnowRob 2.0 to address challenge 5 (the ability to answer questions about what the robot is doing, how it is doing it, and what it expects to be the outcome, for example).

2.4. Accomplishing the various types of tasks in CRAM

Complex tasks are accomplished in CRAM using knowledge, reasoning, and internal simulation in KnowRob 2.0 to identify the parameter values to successfully carry out the action. Sensory information is available from the perception executive to resolve object designators and provide information about the pose of the objects to be manipulated. The action executive

then controls the robot by mapping the parameterized motion plans to adaptive trajectories in real time.

The plan executive can also execute plans without engaging in a dialog with KnowRob 2.0. In this case, all the knowledge required to execute the plan, that is, to resolve the action designators, can be compiled into the plan.[5] This is how simple tasks are accomplished in CRAM.

In its current state, CRAM is only able to accomplish easy everyday activities, but not difficult ones, in the sense described in Section 1. This is because CRAM depends on the existence of an appropriately designed generalized action plan corresponding to each class of underdetermined action description. In other words, the solution strategy has already been identified. Simple tasks can be accomplished without the aid of KnowRob 2.0 because of the strong constraints on the combination of steps, and degrees of freedom in each step, and these constraints are encapsulated in rules in the plan executive. Complex tasks require KnowRob 2.0 to compensate for weak constraints by bringing to bear knowledge and reasoning, using its several knowledge bases and ontologies to infer possible motion parameter values and then to simulate the use of these in the inner world knowledge base before identifying those that are most likely to succeed, and returning them to the plan executive for subsequent action execution.

An alternative way to capture the current capabilities of CRAM is to distinguish between old actions and new actions, and known outcomes and new outcomes; see Fig. 5. Difficult tasks (the bottom two quadrants in Fig. 1) correspond to tasks requiring either new actions or new outcomes; easy tasks (the top two quadrants in Fig. 1) correspond to the situation where the actions are known and the outcomes are known. At present, CRAM can accomplish tasks in the top-left quadrant: using known actions to accomplish known outcomes. If we wish to accomplish difficult tasks using new actions or with new outcomes, we need flexible cognitive behavior. In the next section, we describe the situation model framework as a possible way of realizing such flexible behavior in cognitive agents, in general, and in the next generation of the CRAM cognitive architecture, in particular.

3. The situation model framework

The Situation Model Framework was introduced by Schneider et al. (2020) as the basis for understanding how flexible context-sensitive cognitive behavior is realized in humans, animals, and machines. It has five elements: (a) behavioral episodes, (b) the two-system approach to thought and action, (c) capacity-limited working memory, (d) attentional control, and (e) cognitive maps (see Fig. 6).

A behavioral episode links the functional elements of perception, long-term memory, and motor control for action. Thus, a behavioral episode is a joint representation of perception and action and comprises objects, scenes (i.e., arrangements or layouts of objects), actions, and action outcomes, capturing the causal link between observation of a current state, an action, and the perception of the outcome of that action.
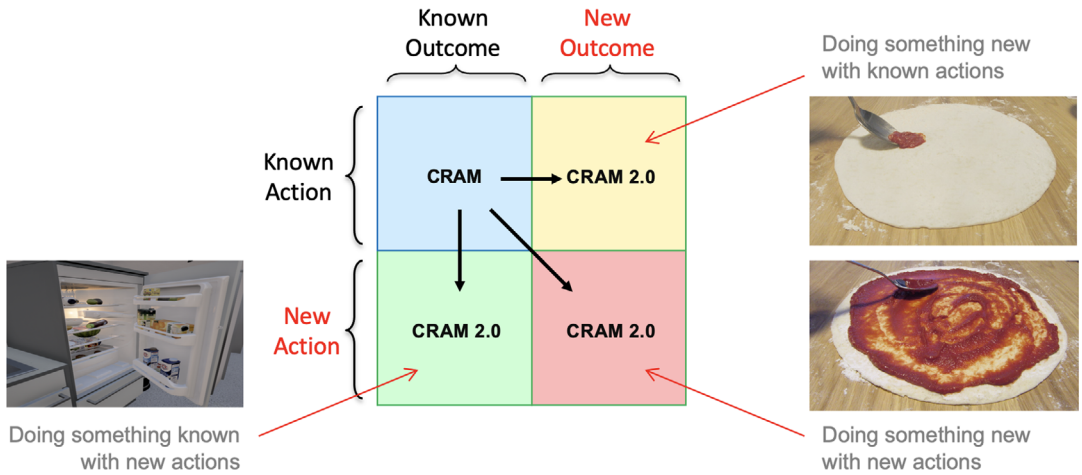
Fig. 5. The manner in which a robot approaches everyday activities can be characterized by whether it uses a known action or a new action with a known outcome or a new outcome. Bottom left: first moving aside a bowl in the fridge to gain access to the carton of milk would require the introduction of a new action (shift the bowl) to achieve a known outcome (picking up the carton of milk). Top right: a pizza can be covered with tomato sauce by using a known action (pick and place) to spoon the sauce from the jar to many different locations on the pizza (a new outcome). Bottom right: a pizza can be covered with tomato sauce by using a new action (spreading) to distribute the sauce over the pizza (a new outcome). Top left: at present, CRAM can accomplish simple and complex tasks using known actions to accomplish known outcomes. The other three quadrants represent what the next generation of CRAM should be able to accomplish by adopting the situation model framwork outlined in Section 3.
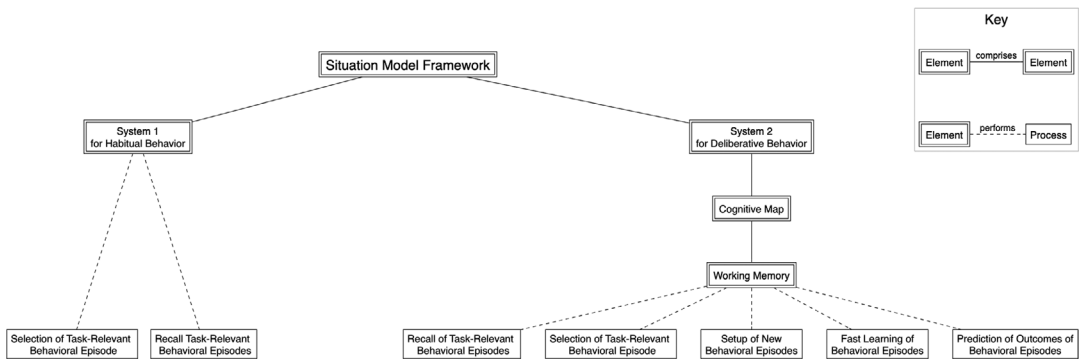

Fig. 6. The key elements of the situation model framework.

The situation model framework makes a distinction between habitual and cognitive (i.e., deliberative) behaviors that, reflecting work by Norman and Shallice (1986), posit that action is controlled in two distinct ways. One is based on well-learned habits that demand little attentional control. The other, which can override habitual control, involves a second process, a supervisory attentional system that deals with situations that cannot be handled by habit-based processes. This "two systems" approach to action control is related to dual process/two

systems approaches to human thought, for example, Kahneman (2003) and Stanovich and West (2000). These approaches have a long history, dating back to James (1890); see Sloman (1996) and Stanovich and West (2000) for details. In habitual actions, System 1 retrieves a number of behavioral episodes, subjects them to a winner-take-all competition, and selects one winning behavioral episode, a process referred to as "contention scheduling." This winning behavioral episode then controls the action by filling in the required sensor or motor information in real time. In System 2, the behavioral episodes can be recalled and optionally modified, or newly constructed, and then simulated to assess the outcome of the action. Thus, System 1 involves reactive or "automatized" (Schneider et al., 2020) control of action based on previous experience encapsulated in behavioral episodes retrieved from long-term memory (see also Neumann, 1984), while System 2 involves cognitive (i.e., deliberative) control of action.

A key function of working memory refers to the ability to manipulate information (Baddeley, 2012; Baddeley & Hitch, 1974), a function that Oberauer (2009) attributes to the limited "region of direct access" that is able to actively process a small number of chunks of information by a "focus of attention." The situation model framework suggests that this manipulation function of working memory is a capacity-limited System 2 property that is missing within System 1. Flexible cognitive behavior is achieved in System 2 by exploiting three working memory mechanisms linked to behavioral episodes: (i) a mechanism to set up novel episodes that are not yet a part of long-term memory, (ii) a mechanism to predict covertly, that is, by simulation, the outcome of a behavioral episode, and (iii) a mechanism to improve or refine a behavioral episode through fast learning. Thus, the behavioral episodes that enable cognitive behaviors have three features. They can be constructed on the fly in working memory. They can be tested through internal simulation (they can also be tested in reality by enacting them). Finally, they can be quickly learned and assimilated into long-term memory so that they can be deployed again in the future.

The information that is retrieved from the large amount of knowledge available in long-term memory must be constrained, just as information flowing from perception must be constrained. In effect, knowledge is bound to the current context and combined in a flexible manner that is constrained by the current task. This is achieved by deploying a process of internal selective attention when retrieving information from long-term memory and external selective attention when extracting information from the environment through perception (Chun, Golomb, & Turk-Browne, 2011). These processes of selective attention—external and internal—serve to constrain the complexity of the activity by restricting the number of behavioral episodes that need to be selected or constructed and evaluated to find the one that suits the current task. In addition, attentional processes select spatial movement parameters of the desired action target object (Schneider, 1995).

Within the situation model framework, the term *situation model* describes the computational space enabling flexible and context-sensitive behavior. Such a space has also been examined under the concept of a *cognitive map*, a term coined by Tolman (1948) to refer to a rich internal model of the world that encapsulates the relationship between events and that can be used to predict the consequences of events (Behrens et al., 2018). Thus, while the concept of a cognitive map is traditionally linked with the neuroscience underpinning spatial

behaviors (O'Keefe & Nadel, 1978), for Tolman, a cognitive map provides a way of organizing knowledge in all domains of behavior in a systematic manner (Behrens et al., 2018), that is, "an abstract map of causal relationships in the world" (Wikenheiser & Schoenbaum, 2016). At present, the situation model framework is not specific about how cognitive maps are to be constructed.

4. Carrying out everyday activities from the perspective of the situation model framework

Easy simple activities (refer to Fig. 1) are candidates for habitual behavior effected with System 1, where behavioral episodes that match the current context are recruited and deployed directly without the need to adapt them. Specifically, the selected episode controls behavior by filling the open parameters of the current episode (object, scene, action, and outcome) with appropriate sensory information (environment, body) and knowledge retrieved from memory. If this parameter filling generates behavior sufficient to achieve the intended action goal without the use of additional decision making or attentional processes, then the habitual mode of action control by System 1 is suggested to be at work. If not, or if the action fails, the agent will revert to deliberative behavior by engaging System 2.

Complex activities typically require deliberative System 2 behavior by virtue of their complexity. However, here too an agent may benefit from accumulating experience and succeed in reducing the initial difficulty of the task. Importantly, such reduction may bring the task into the range of feasibility for the agent. But it is inevitable that here a significant level of difficulty will remain: the task will only move closer toward the upper right quadrant in Fig. 1.

Returning to the dishwasher loading problem, we know that the involved geometry planning problems can become extremely hard on "adversarial" problem instances. This shows us that we never can reduce the difficulty of this complex problem to a level that is completely within the range of purely habitual behavior. But assimilation of sufficiently many successful deliberation outcomes into long-term memory and association with their specific contexts may gradually refine our repository of habitual behaviors such that they begin to cover more and more realistically occurring situations. This will enable System 1 to "almost always" select a habitual behavior that is a strong candidate for a solution, ultimately making the need for System 2 deliberation and creation of new behaviors and their validation using internal simulation a rare event.

Difficult simple activities, too, will require deliberation and the deployment of System 2, with recall, construction, recombination, and internal simulation of behavioral episodes. With experience, the solution strategy may become more evident to the agent on subsequent encounters with the activity, in which case the activity may become easy and simple, requiring only habitual System 1 mechanisms for successful action execution.

Finally, as in the dishwasher example, some activities—simple or complex—that have become amenable to habitual execution can also on occasion require the recruitment of the mechanism responsible for deliberative behavior, for example, in cases where the action fails

or when something unexpected happens. That is, even easy simple tasks may, on occasion, require the deployment of System 2.

Let us now address the alternative characterization of everyday activities in Section 2.4, that is, whether they involve known or new actions and known or new outcomes. Activities involving a known action and a known outcome can be either simple or complex and, as such, will require either habitual System 1 or deliberative System 2 approaches, as discussed above. The CRAM cognitive architecture is currently capable of accomplishing everyday activities in this class.

Activities involving a known action and a new outcome, such as the example in Fig. 5 of covering a pizza base with tomato sauce by repeatedly picking and placing the tomato sauce with the aid of a spoon, require, initially at least, deliberative System 2 behavior to generate a new behavioral episode, involving a new object (tomato sauce), a known action (pick and place), and a new outcome (tomato sauce at multiple locations on the pizza base). It also requires deliberative System 2 behavior to identify the solution strategy of chaining together multiple instances of these behavioral episodes, each instance at a different location on the pizza.

In the case of activities involving a new action and a known outcome, deliberative System 2 behavior is also required, this time to generate a new behavioral episode to shift the bowl out of the way of the milk carton (a new action, comprising a known movement—push—to achieve a new goal). It also requires, as in the previous case, deliberative System 2 behavior to generate the sequential strategy of first pushing the bowl to the side and then picking up the milk carton (a known outcome).

Finally, in the case of activities involving a new action and a new outcome, again, we require the construction of a new behavioral episode comprising a new action (spread, based on push but with the goal of relocating only part of a deformable object) and a new object (tomato sauce) with a new outcome (evenly spread tomato sauce covering the pizza base).

The concept of a new action requires some clarification. An action is not just a movement: an action has an implied goal. Thus, new actions can be achieved in three ways: by using an existing movement in a new context (i.e., new goal and outcome), using a completely new movement in a known context, and using a completely new movement in a new context. Thus, a push might involve a sideways motion intended to relocate a rigid object (a known action) or it might form the basis of a new action as a sideways motion with the goal of relocating some of a deformable object, for example, the new action of spreading tomato paste on a pizza base. We view the action in a behavioral episode as a motion phase in a CRAM motion plan, not as a high-level action as in a CRAM-generalized action plan. Thus, a high-level action can comprise a sequence of behavioral episodes, as in the case of pushing aside the bowl in the fridge before picking up the milk carton or multiple instances of picking and placing the tomato sauce on the pizza base. In other words, a new action, in the sense of a high-level action, can also be formed by a new sequence of behavioral episodes.

As a shared feature, we see that it is always the occurrence of some element of novelty that poses the challenge to both robots and natural agents. Those with higher brains then resort

to System 2 to trigger a process of deliberation. Thus, the structures that are in the focus of the situation model framework (behavioral episodes, interplay of System 1 and System 2, capacity-limited working memory, and attention) can be seen as *aimed at a transformation of novelty into familiarity*. Together, they realize an intricate machinery that incrementally creates powerful computational guidance structures to make acting as swift and successful as possible within a complex and uncertain world: new episodes compile and accumulate the outcomes of deliberation into a rapidly reusable form, attention helps a capacity-limited working memory to identify, set up, and predict the prospects of required new combinations of action elements, and cognitive maps support this generative activity by providing optimized feature spaces that afford flexible internal navigation and simulation. Identifying this interplay of processes in greater depth and bringing it to robots is the promise that the situation model framework has for cognitive robotics.

5. Conclusion

We close by highlighting what the CRAM model reveals about the information processing capabilities that are sufficient for accomplishing some everyday activities. Our hope is that these information processing models might be used to construct hypotheses that can also be studied in human cognition. First, generalized action plans together with modular, composable knowledge chunks representing general knowledge, including common sense and intuitive physics, are sufficient to explain a large part of what is needed to accomplish underdetermined, human-scale everyday task requests. Second, generalized action plans, together with knowledge and reasoning, provide a sufficient basis for internal simulation and, if one assumes that other agents use similar plans, this makes the interpretation of the everyday actions of other agents much easier. Third, robustness in achieving tasks can be realized through failure handling methods that exploit the fixed structure of the generalized action plans and the knowledge about the goal implied by the action description.

The situation model framework with its behavioral episode representation and the causal relations between objects, scene, action, and outcome, coded within it, together with the mechanisms for recall, construction, recombination, and internal simulation of behavioral episodes, offers a way to realize the flexibility required for difficult everyday activities. Any such realization in the next generation of CRAM will require many computational modeling and design decisions. In particular, two issues must be addressed: (i) the representation of behavioral episodes and cognitive maps, and the instantiation of the many processes that operate on these representations; and (ii) the integration of these representations and attendant processes in the CRAM plan language.

Regarding the former, inspired by the object-action-complex (OAC) (Krüger et al., 2011), we are currently investigating the implementation of behavioral episodes as a form of joint episodic-procedural associative memory (Vernon, Beetz, & Sandini, 2015) using convolutional and recurrent neural networks, such as the sequence autoencoder-based approach proposed by Noda, Arie, Suga, and Ogata (2014). We plan on implementing cognitive maps

either as a network of joint episodic-procedural memories, each one encoding a given action, or as a single recurrent joint episodic-procedural memory that encodes all possible actions.

Regarding the latter, we will exploit a form of self-programming in the CRAM plan language executive so that, in the event that a generalized action plan fails (either prospectively in simulation or actually during action execution), it will invoke a plan generation process that will use the cognitive map as a generative model.

The representation of (hierarchical) networks of behavioral episodes poses a significant challenge because of the danger of an exponentially large space of permutations. Possible lines of enquiry to deal with this include the use of graphical models of causal relationships (Didelez, 2018). Understanding the relationship between networks of behavioral episodes and cognitive maps is perhaps the ultimate goal, the pursuit of which will surely shed further light on the nature of the mechanisms underlying cognitive behavior when engaged in everyday activities.

Acknowledgments

Notes

1 The motion plan and the generative model are two elements of the CRAM approach to carrying out everyday activities; we explain them in the text that follows.
2 https://www.ease-crc.org/link/video-ease-robot-day
3 It is a coincidence that there are five challenges, elements, and components in CRAM, and five elements in the situation model framework. No direct correspondence between any of them should be inferred.
4 Narrative enabled episodic memories (NEEMs) are CRAM's generalization of episodic memory, encapsulating subsymbolic experiential episodic data, motor control procedural data, descriptive semantic annotation, and the accompanying mechanisms for acquiring them and learning from them in KnowRob 2.0.
5 It can also be embedded in KnowRob 2.0.

References

Anderson, J. R. (1996). ACT: A simple theory of complex cognition. *American Psychologist*, *51*, 355–365.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036–1060.

Baddeley, A. (2012). Working memory: Theories, models, and controversies. *Annual Review of Psychology*, *63*, 1–29.

Baddeley, A., & Hitch, G. (1974). Working memory. In G. A. Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory* (pp. 47–89). New York: Academic Press.

Beetz, M., Beßler, D., Haidu, A., Pomarlan, M., Bozcuoglu, A. K., & Bartels, G. (2018). KnowRob 2.0 – A 2nd generation knowledge processing framework for cognition-enabled robotic agents. In *IEEE International Conference on Robotics and Automation, ICRA 2018* (pp. 512–519).

Beetz, M., Mösenlechner, L., & Tenorth, M. (2010). CRAM – A cognitive robot abstract machine for everyday manipulation in human environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1012–1017), Taipei, Taiwan.

Beetz, M., Tenorth, M., & Winkler, J. (2015). Open-EASE – A knowledge processing service for robots and robotics/AI researchers. In *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, Washington, USA.

Behrens, T. E. J., Muller, T. H., Whittington, J. C., Mark, S., Baram, A. B., Stachenfeld, K. L., & Kurth-Nelson, Z. (2018). What is a cognitive map? Organizing knowledge for flexible behavior. *Neuron*, *100*, 490–509.

Chun, M. M., Golomb, J. D., & Turk-Browne, N. B. (2011). A taxonomy of external and internal attention. *Annual Review of Psychology*, *62*, 73–101.

Didelez, V. (2018). Causal concepts and graphical models. In M. Maathuis, M. Drton, S. L. Lauritzen, & M. W. Boston (Eds.), *Handbook of graphical models* (pp. 353–380). Boston: CRC Press.

Flanagan, J. R., Bowman, M. C., & Johansson, R. S. (2006). Control strategies in object manipulation tasks. *Current Opinion in Neurobiology*, *16*(6), 650–659.

Franklin, S., Madl, T., D'Mello, S., & Snaider, J. (2014). LIDA: A systems-level architecture for cognition, emotion, and learning. *IEEE Transactions on Autonomous Mental Development*, *6*(1), 19–41.

James, W. (1890). *The principles of psychology* (Vol. *1*). Cambridge, MA: Harvard University Press.

Kahneman, D. (2003). A perspective on judgement and choice — Mapping bounded rationality. *American Psychologist*, *58*(9), 697–720.

Kazhoyan, G., & Beetz, M. (2019a). Accomplishing variations of mobile manipulation actions through generalized robot plans. In *Proceedings of the Seventh Annual Conference on Advances in Cognitive Systems*.

Kazhoyan, G., & Beetz, M. (2019b). Accomplishing variations of mobile manipulation actions through generalized robot plans. In *Proceedings of the Seventh Annual Conference on Advances in Cognitive Systems*.

Kotseruba, I., & Tsotsos, J. (2020). 40 years of cognitive architectures: Core cognitive abilities and practical applications. *Artificial Intelligence Review*, *53*(1), 17–94.

Krüger, N., Geib, C., Piater, J., Petrickb, R., Steedman, M., Wörgötter, F., … Dillmann, R. (2011). Object-action complexes: Grounded abstractions of sensory-motor processes. *Robotics and Autonomous Systems*, *59*, 740–757.

Kunze, L., & Beetz, M. (2017). Envisioning the qualitative effects of robot manipulation actions using simulation-based projections. *Artificial Intelligence*, *247*, 352–380.

Laird, J. E. (2012). *The soar cognitive architecture*. Cambridge, MA: MIT Press.

Laird, J. E., & Mohan, S. (2018). Learning fast and slow: Levels of learning in general autonomous intelligent agents. In *Proceedings of AAAI Conference on Artificial Intelligence*.

Laird, J. E., Newell, A., & Rosenbloom, P. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, *33*, 1–64.

Langley, P. (2006). Cognitive architectures and general intelligent systems. *AI Magazine*, *27*(2), 33–44.

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, *10*(2), 141–160.

Li, M., & Vitányi, P. (2008). *An introduction to Kolmogorov complexity and its applications* (3rd ed.). Cham: Springer.

Neumann, O. (1984). Automatic processing: A review of recent findings and a plea for an old theory. In W. Prinz & D. A. Allport (Eds.), *Cognition and motor processes* (pp. 255–293). Heidelberg: Springer.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Noda, K., Arie, H., Suga, Y., & Ogata, T. (2014). Multimodal integration learning of robot behavior using deep neural networks. *Robotics and Autonomous Systems*, *62*(6), 721–736.

Norman, D. A., & Shallice, T. (1986). Attention to action: Willed and automatic control of behaviour. In R. J. Davidson, G. E. Schwartz, & D. Shapiro (Eds.), *Consciousness and self-regulation. Advances in research and theory* (pp. 1–18). New York: Plenum Press.

Oberauer, K. (2009). Design for a working memory. In B. H. Ross (Ed.), *The psychology of learning and motivation: Advances in research and theory* (Vol. *51*, Chapter 2, pp. 45–100). London, UK: Elsevier.

O'Keefe, J., & Nadel, L. (1978). *The hippocampus as a cognitive map*. Oxford: Clarendon.

Ramamurthy, U., Baars, B., D'Mello, S. K., & Franklin, S. (2006). LIDA: A working model of cognition. In D. Fum, F. D. Missier, & A. Stocco (Eds.), *Proceedings of the 7th International Conference on Cognitive Modeling* (pp. 244–249).

Ritter, F. E., & Young, R. M. (2001). Introduction to this special issue on using cognitive models to improve interface design. *International Journal of Human-Computer Studies*, *55*, 1–14.

Schneider, W. X. (1995). VAM: A neuro-cognitive model for visual attention control of segmentation, object recognition, and space-based motor action. *Visual Cognition*, *2*(2–3), 331–376.

Schneider, W. X., Albert, J., & Ritter, H. (2020). Enabling cognitive behavior of humans, animals, and machines: A situation model framework. *ZiF-Mitteilungen* (1), 21–34.

Sloman, S. A. (1996). The empirical case for two systems of reasoning. *Psychological Bulletin*, *119*, 3–22.

Stanovich, K. E., & West, R. F. (2000). Individual differences in reasoning: Implications for the rationality debate. *Behavioral and Brain Sciences*, *23*, 645–665.

Sun, R. (2004). Desiderata for cognitive architectures. *Philosophical Psychology*, *17*(3), 341–373.

Sun, R. (2007). The importance of cognitive architectures: an analysis based on CLARION. *Journal of Experimental & Theoretical Artificial Intelligence*, *19*(2), 159–193.

Sun, R. (2016). *Anatomy of the mind: Exploring psychological mechanisms and processes with the Clarion cognitive architecture*. Oxford: Oxford University Press.

Sun, R. (2017). The CLARION cognitive architecture: Toward a comprehensive theory of the mind. In S. E. F. Chipman (Ed.), *The Oxford handbook of cognitive science*. Oxford: Oxford University Press.

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, *55*, 189–208.

Trafton, J. G., Hiatt, L. M., Harrison, A. M., Tamborello, F. P., Khemlani, S. S., & Schultz, A. C. (2013). ACT-R/E: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction*, *2*(1), 30–55.

Vernon, D., Beetz, M., & Sandini, G. (2015). Prospection in cognitive robotics: The case for joint episodic-procedural memory. *Frontiers in Robotics and AI*, *2*(Article 19), 1–14.

Wikenheiser, A. M., & Schoenbaum, G. (2016). Over the river, through the woods: Cognitive maps in the hippocampus and orbitofrontal cortex. *Nature*, *17*, 513–523.