

A Review of Case-Based Reasoning in Cognition-action Continuum: a step towards Bridging Symbolic and Non-symbolic Artificial Intelligence

Pinar Öztürk, Axel Tidemann

*Norwegian University of Science and Technology (NTNU),
Department of Computer and Information Science,
Sem Saelandsvei 7-9, NO-7491 Trondheim, Norway
Ph. +47 73551019
E-mail: pinar@idi.ntnu.no, axel.tidemann@gmail.com*

Abstract

In theories and models of computational intelligence, cognition and action have historically been investigated on separate grounds. We conjecture that the main mechanism of case-based reasoning (CBR) applies to cognitive tasks at various levels and of various granularity, and hence can represent a bridge - or a continuum - between the higher and lower levels of cognition. CBR is an artificial intelligence method that draws upon the idea of solving a new problem reusing similar past experiences. In this paper we re-formulate the notion of CBR to highlight the commonalities between higher level cognitive tasks such as diagnosis, and lower level control such as voluntary movements of an arm. In this view, CBR is envisaged as a generic process independent from the content and the detailed format of cases. Diagnostic cases and internal representations underlying motor control constitute two instantiations of the case representation. In order to claim such a generic mechanism, the account of CBR needs to be revised so that its position in non-symbolic AI becomes clearer. The paper reviews the CBR literature that targets lower levels of cognition to show how CBR may be considered as a step toward bridging the gap between symbolic and nonsymbolic AI.

1 Introduction

In theories and models of computational intelligence, cognition and action have historically been investigated on separate grounds. Within the method area of artificial intelligence (AI), symbol-processing approaches have emphasized higher-level cognition, while situated and embodied AI has more focus on the bodily activities of an agent. The issue of representation has become an effective departure point contributing to the escalation of tension between these two lines of AI. In recent years, however, a trend has emerged that is looking for similarities and complementarities between underlying mechanisms of higher-level cognition and lower-level motor behavior. Ongoing research in cognitive science aims at providing clues of a seamless link between cognition and action, resulting from the close cooperation between researchers across disciplines such as cognitive psychology, linguistics, neuroscience and AI, who previously investigated the same problems in isolation from each other. An important hypothesis that has arisen in this context is the equivalence of the neural mechanisms underlying overt motor behavior and motor

The original and a shorter version of this paper was entitled 'Case-Based Reasoning in the Cognition-Action Continuum' and appeared in the Second Norwegian Artificial Intelligence Symposium, November 2010.

imagery (i.e. thinking of a motor action without executing it), which implicates the existence of underlying *representations*. Both cognitive psychology (Robin, Dominique, Toussaint, Blandin, Guillot, & Le Her, 2007), philosophy (Grush, 2004) and neuroscience (Filimon, Nelson, Hagler, & Sereno, 2007; Zimmermann-Schlatter, Schuster, Puhan, Siekierka, & Steurer, 2008) findings provide convincing evidence supporting this hypothesis.

Knowledge models of higher level reasoning tasks include generalized models such as rule-based, frame-based, or network-based models, but also instance-based models such as collections of situation-specific experiences. The latter are typically captured by *case* representations and case-based reasoning (CBR) methods (Kolodner, 1993; de Mántaras, Mcsherry, Bridge, Leake, Smyth, Susan, Faltings, Maher, Cox, Forbus, Keane, Aamodt, & Watson, 2005). CBR draws from research in cognitive science on human memory and was originally targeted at explaining experience-based behavior at higher levels of cognition (Schank, 1982). It rests on the assumption that similar problems are likely to have similar solutions, which can be used to solve new similar problems. Two key indicators of CBR are context-sensitive representations coupling a concrete problem situation with a corresponding solution, and similarity-based search for reusable cases. Besides a plethora of applications in higher level cognition such as medical diagnosis and therapy, design, job scheduling and planning, CBR has also, more recently, been applied in highly dynamic domains that call for non-symbolic representations such as navigation, real time strategy games, robot soccer, and actuator movement. Although CBR has provided outstanding results in higher-levels of the cognition, it has so far scarcely been employed for lower level control and even less so in cooperation with neuroscientists studying movement planning, who have provided interesting findings about internal representation pertinent to actuator-level planning. This paper has a special focus on how CBR is applied at the lower levels of cognition-action continuum, and through a review of the current state of the art, argues that CBR is a generic mechanism that can be used in the entire cognition-action continuum.

The notion of *representation* has triggered a major debate in AI. Brooks (1991), for example, banned representations in robot behavior and maintained that “the world is its own best representation” and “explicit representations and models of the world simply get in the way. It turns out to be better to use the world as its own model”. We conjecture that unification of symbolic and non-symbolic AI is conditioned on a reconciliation of the different views of representation, and where/when they are used in the entire cognition-action continuum. Our standpoint is that representations in the brain exist for motor level control as well, and are used in a CBR process. Planning in both higher and lower levels of cognition-action continuum rely on **representations** pertinent to their own target domains, although the two types of AI have, at the first sight, irreconcilable perspectives on the representations underlying these models. Following Markman & Dietrich (2000), this paper stresses that the non-symbolic “models” should be included in the definition of *representation* because they are stored for later use and can be recalled upon encountering similar situations. Our hypothesis is that the computational framework for case-based reasoning may encompass lower level cognitive tasks such as motor control, as well as higher level reasoning tasks. The reasoning is that actions may be at various abstraction levels (Öztürk, 2009) while the underlying action selection mechanism may be similar in vital ways. Proposing marriage, ordering a blood test, going to the cinema and lifting a tea cup are all actions, albeit at rather different levels of granularity.

The idea we advance is that “action selection” in nondeterministic domains including purposeful actuator movement is *case selection*. But why is applying CBR to such low-level mechanisms as movement learning and control important? From the biological plausibility perspective, the aforementioned research findings in neuroscience indicate that the motor system deals not (only) with purely reactive stimulus-response type of processes but often with stimulus-response-consequence type of processes where the consequences are computed through anticipation, an important part of the whole enterprise (Herwig & Waszak, 2009; Shin et al., 2010). The notion of anticipation in itself is a vital issue with respect to the discussion whether to use representations

or not, providing supporting evidence to the ubiquity of representations in the entire cognition-action continuum. Furthermore, studies in neuroscience on imitation-based motor control also support the existence of multiple context-sensitive representations underlying actuator movement.

The outline of the paper is as follows. Following the introduction, main characteristics of a CBR system and the tasks of the system designer are introduced in section 2. Section 3 conveys the rationale behind our hypothesis that cognition and action constitute a continuum involving action selection at various levels, why CBR underlies the entire cognition-action continuum and why it provides a bridge to unify symbolic and sub-symbolic AI. Section 4 reviews CBR systems that deal with real-time dynamic domains such as navigation, robot soccer and real-time strategic games. Section 5 discusses the role of imitation in various disciplines, with a particular focus on neuroscience, followed by a discussion of the role of representations in human motor control and planning. Section 7 presents an architecture about imitation-based motor control and learning, suggested by neuroscientists. This is a preparation for the other part of the review of CBR systems, which focuses on the lower levels of cognition-action tasks, in sections 8, 9 and 10. Finally, Section 11 discusses why the reviewed CBR systems strengthen the possibility that CBR plays an important role in connecting symbolic and sub-symbolic AI.

2 Main characteristics and design tasks of CBR systems

The main argument of using CBR in general is that it does not need an extensive and deep domain model and relies instead on experience-based, compiled knowledge which humans are known to gather during and after problem solving. The distinctive role of CBR in symbolic AI is its ability of fast reasoning compared to reasoning with deep domain knowledge.

An experience of solving a problem enriches the experiential memory of humans (and animals, in general) and, if properly stored, experiences can be useful in solving new, but similar problems. A CBR process has a number of characteristic subprocesses, as shown in Figure 1. Upon encountering a new problem, the problem description of the new case is matched against those of the past cases in the case base and the most similar ones are used to solve the new problem. This constitutes the *retrieval* subprocess which is often implemented as k-nearest neighbor, where various similarity metrics may be employed. In the *reuse* stage, the solution of the retrieved case is used in solving the new problem. If the problem description of the retrieved case(s) is almost equivalent to the current situation, no adaptation of the solution would be necessary. Otherwise, the solution needs and adaptation before its reuse, and adaptation may be anything from simple modifications of individual actions/steps to a rather complicated operation on an overall plan. *Revise* evaluates how well the past (or adapted past) solution worked in the new situation. Finally, the *retain* stage takes care of the storage of the new case if it is sufficiently different from the cases in the case base.

Typically a case has two main components, a problem description and a problem solution: $C = \{ProbDescription, Solution\}$. In mainstream CBR there may also be a “lessons learned” component, typically a free text note written by the problem solver that may provide hints on the specific conditions that this solution works best, or anything in general that may be useful for the problem solver to decide whether to re-use this solution. The cases are stored in a case base: $CB = \{C_1, C_2, \dots, C_m\}$.

In any CBR system the most important questions that the system designer encounters are

- What is the content of a case, how is it represented?
- How is the case base initially populated?
- How is the similarity between two cases measured?
- How are the retrieved case solutions adapted?

In mainstream CBR, both the problem description and the solution are typically represented explicitly and symbolically. The problem description part embraces a set of attribute-value pairs

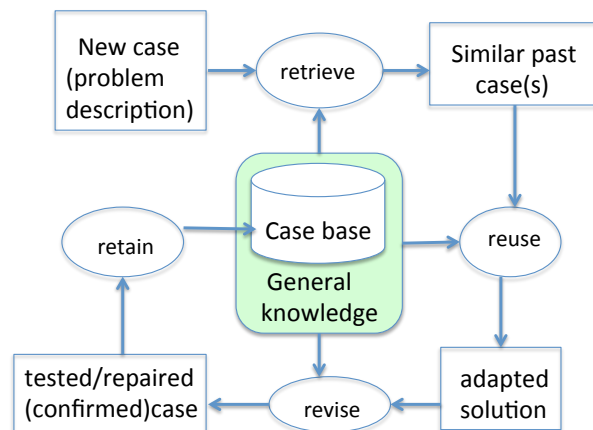


Figure 1 The CBR cycle has four main computational subtasks: *retrieve*, *reuse*, *revise* and *retain*.

(tuples) where the attributes are salient properties defining a situation or an object under consideration: $ProbDescription = \{ \langle a_0 : V_0 \rangle, \langle a_1 : V_1 \rangle, \dots, \langle a_n : V_n \rangle \}$. Attributes are defined concepts in classical CBR, while the values may take numerical or nominal values. The solution can be of various sort such as the diagnosis of a patient, a sequence of actions of a soccer player agent, or a free text interpretation of an anomalous event, depending on the application task.

The content of the case and the used representation language depend on the level of the application in the cognition-action continuum. Representations at the lowest motor level don't involve symbols and concepts as they do in higher level cognitive processes, which is the focus in symbolic CBR. During retrieval, previously solved cases are activated through some indexes typically decided by the designer. The indexes are often a subset of the attributes of the cases.

Initial population of a case base is a daunting task in classical CBR because it is manually crafted by knowledge engineers who makes use domain experts or written material to extract the case content. We can call this "case grounding" problem, which is reminiscent of the much discussed *symbol grounding* problem (Harnad, 1990). We believe case grounding problem is the reason why CBR has not seen wide-spread adoption in the industry - because manual extraction of cases from reports and records is costly and time consuming. Scarcity of CBR in motor level applications, such as voluntary actuator movement, may also be attributed to the problem of initial case acquisition - manual case acquisition is even more impracticable. Recently new ways of case acquisition through training, observation, and imitation have emerged. In addition, a recently popular research subfield of CBR, *textual CBR* (TCBR), has a focus on automated extraction of cases from documents in free text such as anomaly reports (Massie, Wiratunga, Donati, & Vicari, 2007) and patient records that embrace a concrete problem solving experience. TCBR is an example of how CBR brings together different research disciplines such as linguistics, psychology, information retrieval and information extraction. Another line of research related to automated case acquisition is imitation-based learning. Although the CBR examples on real time domains and in particular highly nondeterministic domains like control of actuators are rare, still the current state of the art CBR provides sufficiently strong evidence to suggest that the computational framework for case-based reasoning may encompass lower level cognitive tasks such as motor control, as well as higher level reasoning tasks.

Classical CBR has used various similarity measures. Cunningham (2009) presents a taxonomy of similarity measures used in CBR research, and examples in each category. Euclidean distance is one of the most common metrics for measuring the distance/similarity between two cases, each of which is represented as a feature vector. Case adaptation has much to say about the success of case reuse and various methods varying between simple and superficial changes to complex and

radical changes have been studied (Leake, Kinley, & Wilson, 1995, Lee-Urban & Muñoz-Avila, 2009). Throughout the paper we will review various similarity and adaptation methods.

3 Rationale behind the “CBR in cognition-action continuum”

A large number of tasks at various levels of cognition involve selection of the best next action. However, the notion of **action selection** (AS) is not well defined, since it has been used both for selection of an action defined at a level as high as “getting married” or “declaring war”, as well as at a much lower level, such as “grasping a tea mug”. Öztürk (2009) attempts to define action selection at different levels of cognition as a unified mechanism. Briefly, she makes a distinction between two types of action which we dubbed *Type1* and *Type2* actions. The border between the Type1 and Type2 AS is where the planning starts to involve the limbs. Type1 actions are higher level intentions, not involving the motor system while Type2 actions are movement related; we may call them *intention in action* (Searle, 1983). CBR has so far been mostly applied to Type1 AS, while the current paper argues that CBR applies also to Type2 action selection and describes how motor planning and motor-command selection rests on CBR. Öztürk (2009) speculated that the Type1 action selection may take place in the basal ganglia (BG), while the cerebellum may be the loci for Type2 action selection. This argument relies on neuroscience findings that the BG may be facilitating a winner-takes-all mechanism while the cerebellum, which is known to take a major role in motor control, involves a mixture-of-experts process. Despite these differences, both action selection mechanisms rely on the same notions of matching and similarity. This speaks for why a CBR mechanism should be common to action selection in both Type1 and Type2 action levels.

In domains where a high level planning is needed CBR has long proven to perform well where plans involve “conceptual actions” close to goal or task descriptions such as “get money from machine”, “fly to San Francisco” and “take taxi to the hotel” (Muñoz-Avila & Cox, 2008). Such a plan is done when preparing for a trip. The **purpose** of making the plan is important. In higher levels of cognition, the plans are for setting up the goals (i.e. *what* to do), while in the lower levels planning involves increasingly more *how* to do things. In motor control, action selection is very much about how to compute the force to be exerted on the actuators to move them to the desired state/goal.

At the lower levels of the cognition-action continuum, the body (the actuators in particular) becomes a focus in intelligent artificial agents. Systems at this level involve non-symbolic, continuous representations contrary to their counterparts in the higher level cognitive tasks. Therefore, in non-symbolic AI, neural networks have often been used for learning and control of motor level tasks. On the other side, related research in neuroscience suggests that the motor level representation is not necessarily a single generic large network. We propose accordingly that an artificial motor system should embrace a case base instead of, or in addition to, a generic large network. There are two reasons why the system should have a case base and why the underlying mechanism is case-based reasoning. The first is the findings in neuroscience illustrating that the motor subsystem (i.e. cerebellum and cerebral cortex) learns **multiple representations**. The other reason is of a more technical nature and is due to a problem in artificial neural networks, called catastrophic forgetting (French, 1994). Briefly, catastrophic forgetting arises when a neural network is forced to learn new concepts reaching beyond its capacity, which leads to the overwriting of old information. Having several networks instead of one huge enables learning of a large repertoire of different movements. In this paper we don’t deal with the technical aspects of the discussion of a single network versus a case base. We focus rather on the coherence between the findings and models in neuroscience, a CBR explanation of artificial motor control, and subsequently a unified account of symbolic and subsymbolic AI suggested by the role and use of CBR in the entire cognition-action continuum. Our account is grounded in two arguments:

- Human motor planning involves a planning similar to planning in higher levels of cognition because both involve representations. An important point of departure, however, is that

models the corresponding models are about the external world (i.e. the environment) and the actuators (i.e. the body), respectively.

- Planning in motor control also involves a CBR process because it relies on selection between previously learned context-sensitive representations - among a set of alternative ones - that provide an action to be taken at the next moment.

In the rest of the paper we review the CBR accounts at the lower levels of cognition-action characterized as real-time, dynamic, nondeterministic domains. This will illustrate why we conceive CBR as a compelling candidate to bridge the symbolic and non-symbolic AI, or more correctly, the suitability of CBR as a generic control mechanism.

4 CBR in dynamic and real-time domains

Despite the broadness of the areas in which CBR has been used, it has predominantly been employed for problems that involve conceptual and higher levels of cognition, such as recipe creation (Hammond, 1989), diagnosis/therapy (Lopez & Plaza, 1993; Leake & Kinley, 1998), newspaper story understanding (Ram, 1993), legal argumentation (Ashley & Hammond, 1988), dispute resolution (Kolodner et al., 1985), software reuse (Grabert & Bridge, 2003; Koehler, 1996), navigation planning (Ram & Francis, 1996), and process planning (Velooso et al., 1996; Muñoz Avila & Hüllen, 1996).

The review of the CBR systems in this section is organized according to the application tasks such as navigation, robot soccer, and realtime strategy games. The CBR applications of purposeful actuator control is reviewed in depth in Section 9 and 10.

Ram & Santamara (1993) make a distinction between two types of CBR: "... high-level reasoning in problem domains that can adequately be described using discrete, symbolic representations. However, many real- world problem domains, such as autonomous robotic navigation, are better characterized using continuous representations". Our paper makes a similar distinction between the use of CBR in higher levels of cognition and its use in lower levels of the cognition-action continuum. We may pronounce this distinction along a number of dimensions: 1) the characteristics of the application domain, i.e. whether it is a dynamic and real-time application or a more static one, 2) the type of underlying representations, i.e symbolic or non-symbolic, 3) the type of planning that are required, i.e. episodic, one shot, sequential or incremental planning (Russel & Norvig, 2010), and 4) according to what is modelled, e.g. the external world or the effectors of the body.

In this section we review CBR applications in dynamic and real-time domains that require continuous representations. We should note however in advance that although the domains discussed in this section are continuous by nature, the case representations themselves are not necessarily always so. We will elaborate on this aspect when discussing each system.

4.1 Navigation

Ram & Santamara (1993) and Ram et al. (1997) are one of the pioneers using CBR for adaptive reactive navigation, an example of online real-time behavior in dynamic environments. They call this type of CBR as *continuous CBR* because they maintain, it is suitable in problem domains that are continuous in three senses: "continuous representations", "continuous performance" (i.e. continuous sensorimotor interaction with the environment) and "continuous adaptation and learning". Navigation behavior of the robot is modelled as a relatively small set of behavior schemas. Some examples are *avoid-obstacle*, *move-ahead*, *move-to-goal*, and *noise* (move randomly in a wandering fashion) schemas, which are similar to layers in Brooks' subsumption architecture (Brooks, 1991). However the big difference between them is the control mechanism, that determines how much each behavior/layer has to say about the control of the robot at each moment. In the subsumption architecture, coordination among the layers is enabled through inhibition and suppression of inputs and outputs of finite state automatons (FSAs) composing the

layers. Which layer will be dominant at each moment is regulated through hard-coded decisions, for example, “explore” layer takes over the control every 50 seconds for just 10 seconds. In (Ram & Santamara, 1993) and (Ram et al., 1997) coordination is managed by two integrated methods. First a number of generic behavior schemas (Arkin, 1989) are activated based on the environment and the robot’s current state. Before the schemas are used in determining the robot’s next action they undergo an adaptation process performed using CBR. That is, CBR is employed only after the selection of the behavior schema assemblage done by the higher level schema-based planner. The final decision of the robot is a combination of suggestions from all the behavior schemas in the selected behavior assemblage, a summed vector representation.

Each reactive behavior schema implements a function with a set of parameters and each function computes as output a vector suggesting the robot’s speed and direction in the next time step. However, the parameters in the schema function are generic (empirically decided) and may need to be altered, when the environment to be navigated changes, to correspond to the current sensory information. Some behavioral parameters, for example within *avoid-obstacles* schema, are *goal gain*, *noise gain*, *noise persistence*, *object gain*, and *sensible distance*. Gain parameters are the multiplicative weights of the corresponding schemas, *noise persistence* controls the frequency with which the noise vector changes its direction randomly, and *sensible distance* controls the distance within which the robot reacts to obstacles with the *avoid-obstacles* schema.

Parameter adaptation is done by the CBR module. The solution part of each case comprises a set of adaptable behavioral parameters specifying how they can/should be adapted to the current environmental situation and robot’s current status. This knowledge is represented by a numeric range for each parameter that limits the changes that the system is allowed to make on the concerned parameter in the current problem situation. The problem description part of the cases represents the problem state through two kinds of situational description; the environmental state and the robot’s own state. Examples on the attributes of the environmental states are *clutter*, a metric of how many obstacles are around the robot, *goal-nearby* which is a binary value, *goal-direction*, a value in radian angle measure while robot’s state is represented by attributes such as *no-progress-with-obstacles* which indicates that the robot is moving but not towards the goal and the robot is not within the sphere of any obstacle. Naturally, one of the most important attribute of robot’s state is its current location, on a two-dimensional grid. Most environmental and robot states are abstracted representations of perceptual information such as “circles”, “no-movement-flag”, “movement-to-goal-flag”, and “clutter”, many with binary values, and often abstracted in an oversimplified way from an actual experience which consists of the time series of real values. The authors mention that “one of the open issues in their research is the automatic extraction of such symbolic representations from the actual continuous experiences of the system”. This issue is still one of the most challenging problems in AI today. We conceive the currently predominant conception of representation as the source of the problem. Contrary to what was advocated in the context of reactive systems (Brooks, 1991) *there is a need for representations all the way in the cognition-action continuum*, albeit these are not necessarily only symbolic representations. The notion of “representation” should have a broader sense, not limited to symbolic representations but embracing them. The hypothesis of this paper is that CBR can be used in “reactive” systems but abstracted, symbolic frame-based representations are not sufficient at this level of the cognition-action continuum. It is simply not possible to represent in a case base all the possible combinations of real-valued attributes, within the frame-based representations. So, something other than symbolic representations should be used that allow the use of actual sensory data. Throughout the paper we will discuss various attempts made to this end.

Regarding the similarity judgment, Ram et al. (1997) use a similarity metric that maps the environmental and robot state of the current case with their counterpart in the past cases and computes partial matches when the discrepancy between the values of a certain attribute in the

new and the past case is within a predefined range. The cases are hard coded and the initial case base is manually populated, leaving the case acquisition problem unresolved.

Likhachev et al. (2002) also use behaviour schemas where a higher-level finite state automate-based deliberative planner generates a high level plan consisting of navigational states of type *goto*. Each such state is associated with a behavior assemblage that consists of a set of behavior schemas such as *MovetoGoal*, and *AvoidObstacles*. Similar to Ram & Santamara (1993), in each particular situation the parameters used in the schema functions needs to be adapted to the situation. The paper however suggests a method to learn optimal behavioral parameterization that avoids the need for manual configuration of behavioral parameters in the cases. In this way, initial configuration of the case library can be configured automatically. The suggested learning process can either be a separate training process or be a part of the mission execution. The role of CBR is, similar to (Ram et al., 1997), switching between different sets of behavior schema parameters, i.e. adaption of the parameters. Here also cases involve interpreted sensory inputs (e.g. goal-nearby: 0 / 1) but this work includes a *feature identification* module that automatically converts the sensory environmental data into an abstracted representation that is suitable for comparison of cases. Details of this computation is presented in (Likhachev & Arkin, 2001). Briefly, a spatial vector is computed by the feature identification submodule that takes as input the sensor data and the goal information (i.e. position) and computes the relevant spatial features as a *traversability* vector F of size k . The space around the robot is divided into a predefined k number of angular regions and each element of F represent the degree to which the region can be traversed. Traversability depends on sensor readings indicating crowd of obstacles in the region and is represented as a scalar in the range between 0 and 1, for each of k regions. Each case is represented by a traversability vector and a temporal vector. The temporal vector is also computed by the feature identification module and has two scalar components: short-term and long-term velocities of the robot relative to the maximum possible velocity of the robot. Similarity measurement between the new case and a past case happens in two stages. First, similarity along spatial dimension is measured using a weighted Euclidean distance between spatial feature vectors. Subsequently, the best cases enter into a second similarity process, this time for temporal match. The resultant cases are then used to adapt the behavior schema parameters.

Before we go further, a point worth to stress - which applies to most existing CBR systems in dynamic domains - is that at this “lower level” the actions are executed in the environment, not *on* actuators, but “implicitly” and *via* actuators. The result of an action consists of a certain speed and a direction, but it remains implicit in the paper how/whether this is issued to the actuators. Possibly, it is not. In a sense the actuators and the environment seem to be conflated and reduced to “body”. What kind of actions are at the actuator level has not much been the subject matter research focus. Still, this type of CBR is at a closer level to the real reactive and overt behavior.

To sum up, the “behavior level” CBR does not correspond to the truly lowest level in the cognition-action continuum. This is partly because it is mainly concerned with the outcomes of actions on the environment and on the “body”, and not on the actuators per se, and partly because the involved representations are symbolic, fairly arbitrary abstractions of sensor data.

4.2 Robot soccer

Karol et al. (2003) select game plays in robot soccer using CBR. The new case is matched with the past cases based on the similarity between the positions of the players and their degree of possession of the ball, which are denoted as *Field* and *Degree of Possessions* in the problem description part of cases. The main idea underlying the similarity of cases is that the positions of the players should be as similar as possible meaning that more similar cases would need less time to move all the players from the field state of the first case to the state of the second one. To do this, the game space is divided into 30 strategical regions, each of which is given a number and a weight indicating its strategic importance. In the Field state, each player is represented by

the region it currently is in. Then the Field states of two cases are compared using a weighted Euclidean metric. Similarity involving the possession of the ball is based on *Degree of Possession* of players, measured as a scalar value between 0 and -3. For example, “no possession” has 0 value, “no possession but in a scrum” has value 1, and “possession by the opponent team in the scrum” has value -2. Similarity is based on the distance between the possession values in the two cases under consideration. The cases involve interpreted (i.e. abstracted) data; sensory data is converted into the described higher representations on which Euclidean distance metrics are used for similarity measurement. It is not clear though how the abstraction is done.

Gabel & Veloso (2001) use CBR to enhance the performance of the online coach in the simulated robot soccer matches. This is a multiagent system with two teams where a coach agent for each team selects the team player agents. The system enables the coach to decide the team line-up using past experiences. There is a number of player types which differ from each other in specific characters represented in terms of mostly real-valued 11 attributes such as maximum player speed, player’s size, and inertia moment of the player. For example, the *kickable margin* attribute of player types defines a circular area around the player in terms of its radius in which the player is able to control the ball. The value of this attribute is between 0.7 and 0.9. A player type is represented as a vector of real numbers each corresponding to one of these attributes. The problem part of a case represents each player type as such a vector and specifies how many players in the concerned soccer match belongs to each player type. Each player suits certain positions in a game best and the task of a coach is to decide the main strategy of the game, such as how many players will play in each position (e.g. midfielder, defender) and to deploy each player in its most suitable position. The coach may also make changes in the position of players during the match. The solutions of the past cases are examples of the assignment of the players to the regions/positions each player stayed most of the time during the match. The first part of a solution is the main strategy which is the specification of the number of players deployed in each position (e.g. defender or forward) in the game, for example 4-3-3. The second part of the solution is about performance statistics of the suggested team such as ball possession time and pass success rate as well as the score of the match. Similarity between two cases are done for player types, more specifically the attributes that characterize them. Similarity measure is based on a simple distance metric that normalizes the difference between the value of each attribute in the compared cases:

$$Sim_A(v_q - V_c) = \left(1 - \frac{v_q - V_c}{max_A - min_A}\right)$$

where A is an attribute of the player type, *max* and *min* are its maximum and minimum values and v_q and v_c are the values of this attribute in the new case and the past case. respectively. The total similarity of player types is computed through a weighted measure of the similarity between all the attributes. The weights used in the measure are not fixed but may be changed by a “similarity teacher” that learns how to do that through a genetic algorithm.

Solutions may be adapted through substitution of player types. The underlying mechanism relies on the statistics in the case solutions in the case base and ensures that the player types that are most frequently used in past cases are given priority to substitute the players to be changed. The cases are generated from the past games’ logfiles eliminating the need for manual crafting of the case base.

In (Ros et al., 2006) a special robot, called “captain”, decides the next actions of all the agents and instructs them. The captain chooses next actions by comparing snapshots of the current game with the initial state of the game play in past cases. The problem description of a case captures the environmental states (e.g. positions of the ball, teammate robots, opponent robots) and the game/strategy-related aspects (e.g. time and score) while the solution represents a sequence of actions at the, for example, “turn” and “kick to the right” level. Case retrieval is based on two mechanisms: similarity measurement and adaptability cost. A distinction is made between controllable and uncontrollable indexing features. Positions of teammate robots

are controllable features, while the opponents' and the ball's positions are uncontrollable by the robot that makes the decision. Non-controllable features are very important in measuring similarity between the current case and a past case while non-controllable ones are used for computing the adaptation cost. Past cases are filtered out if the similarity along a non-controllable feature is below a certain threshold, where different features are assigned different thresholds. The overall degree of similarity between the new case and a past case is computed as an aggregated function (e.g. harmonic mean) of the separate similarities along the environmental states and the game/strategy-related aspects. Similarity between environmental states in two cases, in turn, is computed using different similarity functions to compare different features pertinent to the environmental state. For example, a 2D Gaussian function computes the degree of similarity between two points, i.e. positions of teammates.

A novel characteristic of this system is the concern of *adaptability* during case retrieval. The degree of adaptability reflects the correspondence between the robots described in the case and the ones described in the problem. Adaptation-guided retrieval has been studied in mainstream CBR, for example, in poetry generation (Díaz-Agudo et al., 2003). Adaptability is measured in (Ros et al., 2006) in terms of the cost of moving the teammate robots in the current situation to the initial positions of the teammates in a past case. The idea is that teammate robots are controllable and can be moved to adapt the current problem description to the description of a promising past case being considered for retrieval. This is to make the current situation more similar to a case in the case base that is similar to the new case in other respects. It is assumed that a past case with the minimal cost of such pre-adaptation would be more effectively re-used. Previously (Smyth & Keane, 1998) computed the cost of modifying the solution of a case in a similar way. The difference is here it is the problem to be adapted to solve to the case while in (Smyth & Keane, 1998) the solution is adapted.

Ros et al. (2009) explores some of the ideas envisaged in (Ros et al., 2006) as future work. The retrieval process has three elements: The similarity regarding the teammates' and the ball's positions in the two cases, the cost of pre-adaptation, and the applicability of the candidate solution. The first two are similar to (Ros et al., 2006). The third one is new. The solution of a case embraces a sequences of actions such as "kick". The role of the third element is to ensure that the ball trajectory, which the actions in the solution leads to, is free from the opponents and that the opponents' situation in the current case is similar to that of the considered past case. This component involves a function called *free path* function that measures how free of opponents are the trajectories the ball follows during the execution of the past case solution in the current situation. If the the current problem includes opponents that are either on the trajectory or in a position to reach the ball before the teammates, then the case is not a good candidate for retrieval. Hence, the free path function simulates the solution on-line during retrieval to see how effective it would be under current circumstances. Since the ball's movement, for example after a "kick" action, is not deterministic the paper uses a specific representation for the trajectory. It represents a trajectory by means of a fuzzy set whose membership function indicates the degree of membership of an opponent or the ball in the trajectory such that the closer their location to the center of the trajectory, the higher their membership value. A distinguishing property of the case description is the *case scope*. Precise values of the positions of the opponents and the ball on the field may be misleading because the incoming information about the world state is uncertain. To resolve this problem, positions of the opponents and the ball are represented as elliptic regions with small predetermined radius centered in the position of the object. The *case scope* maintains the radius values of the scopes of the opponents and the ball. The reasoning component of the system does position-related computations accordingly. Another rationale behind representing the positions of the opponents and the balls as regions with small radius is that it is easier for the system to reason with qualitative locations. Qualitative information is advantageous when generalization of similar situations are needed. As we have seen earlier in this section, it is a common system development strategy to fall back on the abstracted/interpreted version of the

sensory data to ensure effective similarity-related reasoning For example, in (Kerkez & Cox, 2003) the state representation on (object1 table) and on (object2 table) is abstracted to ontable (object 2) meaning there are 2 objects on the table, because this makes case comparison much easier and flexible. Such a generalization is needed for the type of similarity-reasoning method the system proposed in (Ros et al., 2009) uses. As the authors maintain, “it is more general to reason about a defender being in a region in front of the ball, rather than the defender being in position (x, y)”. In a nut shell, Ros et al. (2009) has two important contributions to CBR applications in nondeterministic and dynamic domains: anticipation of the applicability of a candidate solution through simulation and the pre-adaptation, both during retrieval.

4.3 Real-time strategic games

Case-based plan recognition is another task studied in connection with interactive virtual environments such as characters in training simulators and computer games, highly similar to robot soccer and american football types of tasks but with a special focus on how a counter-player’s future actions, intentions or goals can be anticipated by observing their actions like in “mind reading”. This involves predicting the intentions and actions of the observed agent through retrieving plan-cases from the case base. For example, Fagan & Cunningham (2003) use CBR in “space invaders”, a strategic game, to predict other players’ actions. The states in the plans are abstractions of the real-time data to states at the conceptual level of e.g., “safe” and “very unsafe” while the actions are of type *fire*, *hide*, *emerge*.

The system stores intermediate states in the plans in form of action-state pairs instead of actions only. That is, the case solutions maintain the states that result from the observed actions. The approach does not require a priori case bases because cases are extracted from the problem solving episodes. Representation of the consequences of actions in the solution facilitates prediction of the consequences, which is useful when an action is considered as a candidate next action. This capability of the system is crucial particularly in nondeterministic contexts. The system can find past cases that “match a current observation at arbitrary points in a stream of observed actions” (Kerkez & Cox, 2003). It can predict only one step ahead, contrary to Dinerstein et al. (2005).

Dinerstein et al. (2005) describe a CBR approach for incremental prediction of another agent’s actions in interactive virtual environments involving mixtures of synthetic agents (non-playable characters (NPCs)) and humans. The idea is that by learning on-line through interaction with the human the NPC becomes a better opponent when playing against synthetic agents in the system. The NPCs **predict** the human players’ actions using case-based plan recognition. The agent observes the state-action pairs employed by the human, and records each nondeterministic mapping of a state-action pair as a case.

While many systems with interactive virtual characters involve on-line learning of high-level behavioral guidelines from user feedback in social contexts, this work is concerned with relatively lower level behavior and is more physically oriented. The state space itself is represented as a real-valued, n -dimensional feature vector where n is small and the features are decided by the designer ensuring a compact state representation. This makes the generalization easy and the retrieval easy. Some salient features are the relative distance between an observed agent and its nearest opponent, the velocities of the observed agent and the nearest opponent, and an angle representing the average direction towards obstacles.

Actions are also represented as real-valued vectors. The agent predicts the other agents’ action, and then uses this information to perform an informed tree search to plan its own actions. The agent predicts another agent’s actions by retrieving a case given the current state, and can do that iteratively several steps into future, so that the agent selects its actions based on non-local information. Each time the predicted next action is used to predict the next state and this goes on a certain number of times. In retrieval, the system uses a continuous k -nearest neighbor

algorithm to find the k cases closest to the query state and to compute the next action through a distance-weighted normalized sum of the actions associated to the retrieved k cases.

For a fast retrieval in this real-time complex domain the continuous state space is partitioned by the programmer, and a set of cases are associated with each region of the space. This has been criticized in (Hartley et al., 2005) because of low accuracy attributed to compact representation of the state space. To resolve the speed problem, Hartley & Mehdi (2009) takes another stance and introduces two types of features: primary and secondary features. Primary features are of more abstract nature and they are used in a first round of retrieval while in the second round a more elaborative similarity evaluation is performed using the secondary features with continuous or less abstract representation of the features.

The work presented in (Aha, Molineaux, & Ponsen, 2005) investigates methods for plan selection in WARGUS game. Example actions are “selecting a building to construct”, “researching a specific new technology”, “setting a destination for a group”, and “assigning a task to a group”. The main task is action selection at any moment, hence, incremental planning. The actions comprise *tactics* in a strategy plan. The state space is a 128 x 128 map that involves various units and buildings. This space is abstracted to a lattice to reduce the complexity, while a set of actions is an abstraction of the decision space. Cases are for mapping the game situations to the tactics. The system retrieves cases when a new situation in the state lattice is entered. Each case represents in the problem description part 8 features that describe the situation such as the number of opponent buildings ever created, number of own worker units currently existing, and number of buildings currently existing. The solution of the case provides an action and a value in $[0, 1]$ indicating the utility of selecting this action in the current situation.

The Darmok system (Sugandh, Ontañón, & Ram, 2008) learns plans from observing a human and reuses plans when playing new games. It puts a special emphasis on on-line adaptation of plans. The system learns the plan cases from observing and recording the traces of the behavior of the human player. The traces are annotated by human experts where the actions are connected to the goals they are pursuing. The case learning module then extracts the *behavior* from the annotated traces and saves them as cases. The behaviors have two components, a *declarative* part and a *procedural* part. The declarative parts capture the goal, preconditions that must be satisfied before the behavior can be executed and the success condition that needs to be satisfied for the behaviour to be successful. A significant difference from the “classical” planning is that here there is no postconditions because a behaviour is not guaranteed to success. Goals are symbolic non-operational descriptions, may have parameters, and define the success conditions which are predicates that can be checked to evaluate whether the goal is satisfied in any game state. The initial goal is “win the game”. A behaviour is retrieved for this goal, which in turn, sets up a number of subgoals. The procedural part consists of an executable code for accomplishing the goal. A *plan execution* module executes the plan and marks the goals that are achieved. A *plan expansion* module does then re-planning for the goals that are not yet achieved and relies on *behavior retrieval* module to retrieve cases from the case base. Adaptation itself is of two types: parameter adaptation and structural adaptation. For adaptation the system relies on a dependency graph which is inferred from retrieved plans. Dependencies between actions are deduced from the preconditions and the success conditions. Adaptation may be needed, for example, when preconditions of an action are not satisfied, which triggers a new subgoal that target the satisfaction of the preconditions. This is analogous to general problem solving in mainstream AI such as in SOAR (Rosenbloom & Newell, 1987).

The authors argue that CBR is a better alternative for planning in real-time strategy games, a domain characterised by nondeterminism and is not fully-observable, since the decision space is huge, which renders a search-based planning infeasible. Although the system aims to reduce the search, the plans, goals, actions and preconditions are represented in a way rather similar to planning in search-based AI and involves relatively high level planning. Furthermore, the

adaptation itself relies on a rule-based like mechanism to remove the unnecessary operations and adding new ones into the plan.

The CBR systems reviewed in this section are applications in domains that require continuous representations and continuous performance, such as autonomous robotic navigations. In such applications continuous sensory-motor interaction with the environment is required because the environment changes dynamically. Therefore, this type of applications extend the use of CBR beyond relatively deterministic domains signalling its role in the entire cognition-action continuum. However, the focus of the accounts reviewed in this section is not the control of the end-effectors. The reason possibly is that the actuator are assumed to behave deterministically; when an action is applied to an end-effector, its outcome is assumed to be certain. For example when a certain force is applied to an arm in a certain state, its end state is presumed to be known. This is simply a too unrealistic assumption. A large community of neuroscience researchers investigate the motor behavior to understand the mechanism of the non-deterministic motor behavior and the feedback mechanism underlying humans' smooth motion. In recent years, in addition to the applications such as robot navigation, soccer, and realtime game playing, research efforts have appeared that employ CBR also for the lower level motor control focusing on the end-effector control. Although the reviewed CBR applications and the effector-control level applications have some commonalities such as the need for continuous performance and sensorimotor feedback, they have important differences. Most notably, neuroscience-inspired voluntary end-effector control rely on anticipatory representations of the controlled actuator. This is, rather similar to mainstream AI planning where the outcomes of actions - in the environment - are simulated during the action selection process. Contrary to reactive planning accounts disfavoring representations, neuroscientists inform that the very lowest level of motor behavior uses models of the limbs.

The CBR systems reviewed in this section provide evidence that CBR is a suitable mechanism for action selection in dynamic and nondeterministic domains in general. Various approaches uses different types of non-symbolic representations including continues values and vectors of them. Most typically robot soccer, strategical games and navigation all have their main focus on making a decision about other agents in the environment, reducing the actuators to more or less placeholders. Therefore, these system do not prove (or defeat) CBR's suitability at the actuator level of actions. Shortly, we review other CBR accounts that put a special emphasis on actuator-level CBR. However, we first elaborate some imitation-related research in various disciplines. Imitation is important first of all from the case grounding perspective. It is a promising candidate for automated knowledge (i.e. "case" in CBR) acquisition which is a challenge in CBR and in AI in general. Manual population of the initial case base is not always possible and it is not even desirable. For example, in inverse kinematic tasks, the cases are not discrete representations which makes manual construction of such cases impossible. We have already seen in this section some CBR applications that eliminate manual case generation, through extraction from activity. Imitation is particularly effective at actuator-level control.

Some of the systems reviewed in this section were examples of case extraction from activity, for example, of opponent, human agents' or self playing experiences. Imitation is a justified target for AI research because it plays an important role in natural computation and intelligence.

5 Imitation in psychology, neuroscience and AI

Learning by imitation is a multi-disciplinary field of research. In developmental psychology, Piaget (1962) describes imitation learning as an ongoing process where the infant adjusts internal sensory-motor schemas to what it perceives in the outside world. Meltzoff & Moore (1977) found that newborn infants are able to imitate facial gestures shortly after birth, suggesting that imitation is an innate mechanism that combines vision and proprioception in a way that can be used to guide the motor system of the infant. Meltzoff & Moore (1997) describe this mechanism in a framework called *active intermodal mapping* (AIM) that unifies perception and action and

is able to notice differences between observed and produced motor actions. The AIM process can subsequently correct the erroneous motor action, guiding the infant towards the correct imitative behaviour.

Imitation was put in a neuroscientific context when Rizzolatti et al. (1996a) discovered neurons that were active both when observing and performing the same action, which were named “mirror neurons”. Even though the initial discovery was in monkeys, Rizzolatti et al. (1996b) found similar activations in human brains as well. These neurons were thus hypothesized to be a “neural implementation” of the imitative capability (Schaal, 1999). Since the location of the mirror neurons is close to the language areas of the brain (Broca’s area, specifically), Arbib (2002) consider the mirror neurons crucial for our ability to develop a language. Kohler et al. (2002) found that mirror neurons were active also when hearing the sounds of an action, not only seeing the action being performed, i.e. the sound of cracking open a peanut elicited the same neural activity as when seeing or opening the peanut by the monkey itself. However, only seeing the peanut being opened without hearing the sound itself did not trigger the same activity, showing the importance of hearing the associated sound when understanding the action performed - if there was no sound of the opening of the peanut, the action was not perceived as successful. This is further evidence of the importance of mirror neurons as a basis for language, i.e. a combination of motor activity and sound perception. Gallese & Goldman (1998) link the mirror neurons to our ability to empathize with others; mirror neurons provide the basis for identifying oneself with another person. It has been hypothesized that the lack of this ability can identify (amongst other symptoms) autism spectral disorder (ASD), and Williams et al. (2001) suggest that people with ASD might suffer from a dysfunctional mirror neuron system. This idea finds support in EEG (Oberman, Hubbard, McCleery, Altschuler, Ramachandran, & Pineda, 2005) and fMRI (Dapretto, Davies, Pfeifer, Scott, Sigman, Bookheimer, & Iacoboni, 2005) studies. However, other fMRI studies demonstrate a lack in direct mapping of brain activity when humans are observing and performing the same action (Dinstein et al., 2008, 2007; Lingnau et al., 2009). A mirror neuron as described above could still exist, but in humans these are not the only neurons active during observation and execution of movement. The concept of a mirror neuron system in humans analogous to the brain activity in monkeys remains controversial.

In artificial intelligence research, the imitative process has gained interest as a biologically plausible method for transferring motor skills that can be implemented on artificial entities. Robotics has had a special interest in this issue in recent years. Schaal (1999) considers model-based approaches as most suitable to implement imitative behaviour in robots, consisting of pairing an inverse model (i.e. behaviour or controller) with a forward model (i.e. predictor). This is an established approach in the control literature (Jordan & Rumelhart, 1992), which has also been implemented in AI architectures for imitation learning (Demiris & Khadhour, 2006; Wolpert, Doya, & Kawato, 2003; Tiedemann & Öztürk, 2008). Wolpert et al. (1998) argue that inverse/forward models are present in the cerebellum, leading to an architecture based on those principles (Wolpert & Kawato, 1998). fMRI studies suggest that such an ordering is present in the brain (Imamizu, Kuroda, Yoshioka, & Kawato, 2004). Gaussier et al. (1998) and Matarić (2002) propose architectures for motor imitation that have predefined modules for the various stages of sensorimotor processing, i.e. perception, recognition and action selection. There are other approaches to imitation learning that focus on neural network architectures instead of a modular solution. Cangelosi & Riga (2006) use a neural network to ground symbols; the student learns the association between actions and words by imitating the actions of the teacher. Billard & Hayes (1999) employ a recurrent associative network to connect symbols to sensory perception. The student learns the association between labels and motor actions by imitating the teacher, using Hebbian learning to store the knowledge.

6 The role of representations in motor control: a neuroscience view

Humans are able to flexibly interact with a rich variety of objects in very different environments. This can be explained by their huge repertoire of motor behaviors (Wolpert & Kawato, 1998). Proper motor control demands the selection of the correct motor action (command) that would cause the goal output, i.e. the desired sensory feedback. For example, when a person wants to reach and lift a cup, the motor system decides the motor action that will achieve a smooth reaching movement to the desired location. In control theory and neuroscience terminology, the decision is taken by the *controller*. The movement should have a certain speed and timing in order not to overshoot and to be smooth. Some parameters of the motor control are internal to the body, such as the joint angles of the arm, masses, moment of inertia etc. Other necessary parameters relate to the external world. For example, if the goal is to lift a cup, properties of the cup (i.e. its geometry, shape, affordances and the content) should also be known. Wolpert & Kawato (1998) calls these properties “context”, since these variables are outside of the motor system. The dynamics of an arm involves both internal and contextual parameters. They further suggest that, given the abundance of the context that the humans should cope with, the motor system should have multiple controllers, each specialized for different types and contexts of movements. Otherwise, a single controller would try to adapt each time it encounters a new context before it could produce appropriate motor actions and this would produce large performance errors. Hence, Wolpert & Kawato (1998) suggest a modular approach with several controllers. The system should then select the controller(s) that suit best the current situation.

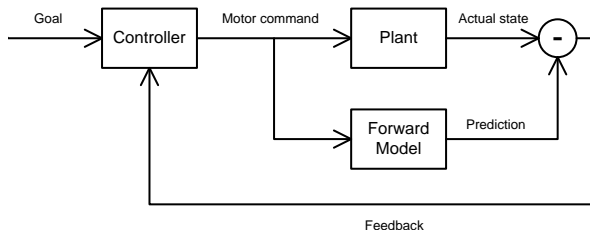


Figure 2 The representations underlying motor control. The *controller* is an inverse model.

As Wolpert and Kawato maintain, “we can view the motor system as forming a loop in which motor commands cause muscle contractions, with consequent sensory feedback which, in turn, influence future motor commands” (Wolpert, Doya, & Kawato, 2003). Figure 2 illustrates this. The neuroscience community has proposed that the central nervous system mimics the aspects of the sensorimotor loop in planning, control and learning (Miall & Wolpert, 1996; Wolpert, Doya, & Kawato, 2003). Borrowed from terminology in control theory, the neural circuits in the central nervous system responsible for mimicking the behavior of the sensorimotor loop are called **internal models**. There are two types of internal models: *inverse models* and *forward models*. Inverse models provide the motor action/command that leads to the desired outcome (i.e. the state of the arms). As such, they act as controllers. Forward models simulate the causal (forward) process and predict the consequences of a motor action. They model the causal relationship between actions and their consequences in the motor system. Forward models predict how the motor system’s state changes when a motor action is sent to the muscles, given its current state. Prediction is essential in the motor domain for various reasons. One reason is the delays in the sensorimotor loop which render perceptual feedback alone insufficient in the control of fast movements. Forward models can estimate the outcome of an action and give *internal feedback* before the perceptual feedback arrives (Ito, 1984; Miall, Weir, Wolpert, & Stein, 1993). Another reason is that by predicting the sensory outcome of a candidate action, before actually performing it, a forward model can be used to judge the suitability of the action (Wolpert & Kawato, 1998) and plays a crucial role in action selection. These explain partly, for example, why a person can move her arm in the dark: she can estimate the position of her arm with some degree of accuracy.

Neuroscience studies provide evidence that there exist multiple inverse models, learned and switched when the context changes. In laboratory experiments (Shadmehr & Mussa-Ivaldi, 1994), subjects used longer times to learn a motor task when they were visually perturbed, but could swiftly re-adapt when they were back to the conditions prior to perturbation, illustrating that the previous models were kept during perturbation and switched back to after the perturbation.

Context has been stressed as having a vital effect in motor control. Data/information from various modalities have been used as context. For example, Wolpert et al. (2003) highlight how the emptiness/fullness of a cup to be lifted influences the selection of motor commands. Sound data have also been used in neuroscience to study context (Kravitz & Yaffe, 1972). These studies suggest that subjects can switch immediately between two learned behaviors based on the context. This means that the brain hosts different plans when the same task (e.g. “send the ball”) is performed in different contexts (e.g. material and size of the ball) or using different actuators or tools (use a stick or a foot) (Wolpert & Flanagan, 2001; Blakemore, Goodbody, & Wolpert, 1998). Motor plans that are suggested to be stored, for example in the cerebellum (Wolpert, C., & Kawato, 1998; Ito, 2008), have the same function as the planning cases.

Neurophysiological studies also provide evidence about the existence of alternative action options in the brain. It is maintained that during this time more information is gathered that will ease the selection process (Cisek, 2005).

The work referred to in this section argues for the existence of multiple paired inverse/forward models in the brain, and that each of these contains motor knowledge acquired through a learning process. When a similar situation is encountered, the most appropriate controller is selected. This is similar to how cases (i.e. acquired knowledge) are stored in a CBR system, and how cases are reused by retrieval of similar cases.

The architecture by (Wolpert & Kawato, 1998) described in the following section is worth discussing for two reasons. First, this is suggested by neuroscientists to explain human imitation and its role in motor control. Second, the architecture has been implemented in AI as a CBR system (Tidemann & Öztürk, 2008).

7 A neuroscience architecture of learning motor control by imitation

In neuroscience terminology, voluntary movements involve “motor planning”. This planning however is different from the mainstream AI planning in that there exist no plans comprising several steps of action ahead, and the “plans” here are not represented symbolically. In the motor domain, planning (read “action selection”) and plan execution are intertwined, and planning occurs in real time and continuously, until the intended overall goal is attained. That is, motor planning is not a one-shot process.

Briefly, as we mentioned in section 6, motor control involves selection of the motor command (i.e. controller) at each time step that best suits the current state, the goal and the desired state of the body, as well as the current context. The whole movement therefore involves a series of controller selections which are called (following motor control terminology) *switching of controller*.

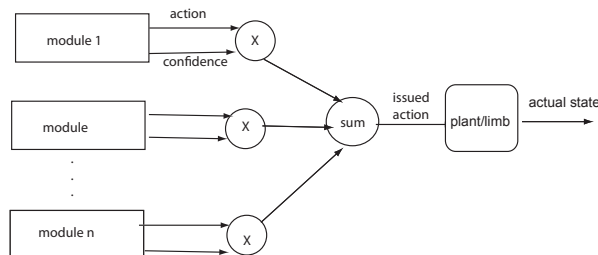


Figure 3 Modules and their confidence determines the final action to be issued to the motor.

Figure 3) illustrates an architecture suggested by neuroscientists (Wolpert & Kawato, 1998) to explain imitation by humans. The architecture comprises a set of *modules* each of which captures specific parts and characteristics of the movement. At each time step, the system selects a module (or a small set of modules) from memory to compute the next motor command to be issued to the involved limbs. A module is a controller, and at each time step one (or several) controller(s) decides the next action u_t . We will shortly describe how it is decided which module(s) will control the limbs, and how the next action will be decided. Each module has three main components an *inverse model*, a *forward model*, and a *responsibility predictor* (see Figure 4). The inverse model generates an action (i.e. motor command) u_t^i , given the current state x_t and the desired state x_{t+1}^i , while the forward model predicts (i.e. simulates) the next state \hat{x}_{t+1}^i of the limbs given the motor command u_t^i and the current state x_t .

At each time step, each module suggests an action and its judgment of its own confidence (see Figure 3) to take the control. The final action is computed as a mixture of suggestions from the most confident modules.

The system is context-sensitive where context models, *responsibility predictor* (RP) in Figure 4, represents the relationship between context signals (y_i) and the modules. The scalar value of the RP of a module in a given context is written as p_t^i .

In addition to the prediction of the relevance on the basis of the context signal, another predictive mechanism further elaborates the module's relevance. This second prediction relies on the forward model which models a limb and can compute what the next state will be, given a motor command. The *likelihood* model shown in Figure 4 computes the suitability of the module based on how well its inverse and forward models would do in the current situation, based on the difference between the predicted and the actual states of the limb at time step $t+1$, i.e. $|\hat{x}_{t+1}^i - x_{t+1}|$.

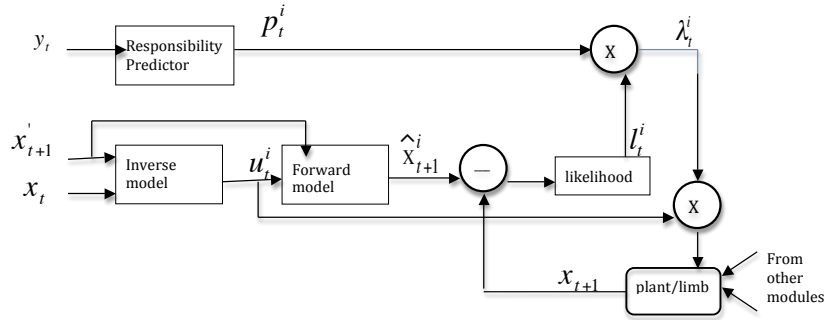


Figure 4 A module with its three components, an inverse a forward model and a responsibility predictor. They collectively decide the next action and how confident the module is at taking over the control in the next time step.

The model outputs a scalar l_t^i , by assuming the presence of Gaussian noise. The scalar quantifies how well the forward model predicted the next state, see equation (1). If $|\hat{x}_{t+1}^i - x_{t+1}|$ is small, the likelihood value will be high.

$$l_t^i = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-|\hat{x}_{t+1}^i - x_{t+1}|^2 / 2\sigma^2} \quad (1)$$

The final confidence, λ_t^i , of the module is a combination of these two relevance estimations. The simulated performance of the suggested action and the output of the responsibility predictor (which we prefer to name the *relevance predictor* p_t^i) determine the λ_t^i signal. The λ_t^i vector describes how much a controller influences the limb. λ_t^i is calculated according to Equation (2).

$$\lambda_t^i = p_t^i l_t^i \quad (2)$$

The ultimate motor command to be issued to the motor is computed according to Equation 3 (see also Figure 3)

$$u_t = \sum_i \lambda_t^i u_t^i \quad (3)$$

In the next section we review some CBR systems applied at the level of end effectors, hence the same level as this architecture operates on.

8 CBR at actuator level

As Hamilton & Grafton (2007) puts it, the same action may be viewed from different perspectives and levels:

An ordinary action such as a pouring a glass of wine can be understood on many levels. The barman will reach for and grasp the bottle, then transport it and tilt it over the glass, carefully controlling the angle to avoid spilling. These kinematic components each require a precisely orchestrated pattern of muscle activity guided by proprioceptive and visual feedback. But a drinker waiting for the barman to finish pouring would likely consider only the goal of the action to provide some refreshment

This part of the paper is closer to or at the kinematics level. Jurisica & Glasgow (1995) is one of the early examples of rather few CBR applications at the truly low level of cognition-action continuum where the controlled object is the actuators and cases involve non-symbolic representations. Their TA3 system employs CBR for an inverse kinematic task where the system computes the joint angles of the robot’s arm that would correspond to a desired state (i.e. coordinates) of the arm. This application is different from the other CBR applications in dynamic, real-time applications because it relates to the brain-actuator interaction. The controlled entity is not the merged body and environment but a particular actuator which behave nondeterministically in the sense that the results of the actions are not necessarily like anticipated. Therefore, the control of actuators rely on feedback, which comes only with delay. The system presented in Jurisica & Glasgow (1995) and Jurisica & Glasgow (1997) doesn’t deal with the delayed feedback problem but it makes significant contributions regarding the use of CBR for such a nonlinear phenomenon. The second system we will review, SHEILA, deals also with the delayed feedback mechanism.

9 TA3

TA3 (Jurisica & Glasgow, 1997) investigate a dynamic context mechanism that allows partial matching between the new case and a past case. It introduces a special parameter of relevance relation the authors called “context”.

TA3 deals with a three-link robot and predicts the joint angles of the robot that will enable the robot to reach the desired positions (i.e. the coordinates of the end effectors). An example is welding where a robot (or human as well) needs to follow a smooth trajectory and should be fast. Jurisica & Glasgow (1997) characterizes this as a classification task with infinite number of classes because the results are continuous values. This is an inverse kinematics task having either more than one solution or it may not have a known analytical solution. TA3 uses CBR for this purpose and employs a special notion of relevance assessment. Relevance is assessed with respect to an explicitly represented context. The underlying similarity process is different than most systems reviewed above computing the distance between two vectors representing the current problem and a past case, respectively. Regarding the population of the initial case base in TA3, the first method used was to create cases distributed uniformly over the working space of the robot. A set of special equations are used for computing the end-effector coordinates, which is a forward kinematics task. Another set of equations are used for inverse kinematics computations, which uses results from the first equation set. These are used to compute the solutions. The

second method for initial population of the case base was through online learning when a robot performs a normal task, i.e. learned by observing the process. Details of the online learning are not elaborated, though. The authors suggest a combination of online and off-line learning of cases. They indicate a need for further empirical study but it is not clear whether they followed up this line of research - we were not able to find more recent literature related to this work.

Before we dwell on TA3's details, it is important to note that this CBR account has been employed for different tasks. In addition to robot kinematics, it is also applied, among others, in the servo domain and for letter recognition. This illustrates that Jurišica and Glasgow share our hypothesis that CBR underlies different tasks on the cognition-action continuum, although they do not articulate this with the same emphasis as is done in this paper.

9.1 Context

Similar to other CBR systems, a case consists of a set of attribute-value pairs:

$$C = \{ \langle a_0 : V_0 \rangle, \langle a_1 : V_1 \rangle, \dots, \langle A_n : V_n \rangle \}$$

where a_i is an attribute name and V_i is the value of this attribute in case C . The paper mentions that the important attributes and their characteristic values may be selected using machine learning or knowledge mining. However, it seems that in the described work human experts specify the attribute importance off-line.

The context defines the important attributes and how close they should be in the new and a past case to consider them a match. Context is defined as a set of attributes, a subset of the case attributes, with associated constraints on the attribute values:

$$Cxt = \{ \langle a_0 : CV_0 \rangle, \langle a_1 : CV_1 \rangle, \dots, \langle A_n : CV_m \rangle \}$$

where a_i is an attribute name and CV_i specifies the set of "allowable" values for attribute a_i .

Context varies across different tasks and for each domain and task, it specifies which attributes may be ignored or be considered important. It seems that the context in TA3 is specified by the user but other possible methods, both automated and semi-automated (i.e. with partly user involvement), are also mentioned.

9.2 Similarity assessment

In the robot domain of TA3, a problem has six attributes containing real values: three lengths describing robotic arms and three attributes describing a desired end-effector position while the solution is a class with three continuous values: the three joint angles. Individual attributes of the problem are grouped into two categories: robot parameters, and end-effector positions.

Two cases are judged to be similar when they *satisfy* the same context, that is when each attribute defined in the context behaves according to the corresponding constraints in the context. The paper defines *satisfaction* as the following:

$$sat(C, Cxt) \text{ iff } \forall \langle a_i : CV_i \rangle \in Cxt, \exists \langle a_i : CV_i \rangle \in C \mid V_i \in CV_i$$

9.3 Retrieval and adaptation

Input to the retrieval process is the case base, the context, and the new problem. If only few relevant cases are retrieved or too many are retrieved using the initial context, the system modifies the context and starts the retrieval anew.

Context can be modified through *relaxation* or *restriction* either of values or cardinalities. In both cases the constraints of the original contexts are changed. Context modification is used during retrieval to control which cases are considered relevant. When too few cases are retrieved in the first round, the similarity context is relaxed. Cxt1 is called relaxation of Cxt2, denoted by $Cxt1 \succ Cxt2$, if the following holds:

$$\forall \langle a_i : CV_i \rangle \in Cxt1, \exists \langle a_i : CV_j \rangle \in Cxt2 \mid CV_i \supseteq CV_j \wedge Cxt1 \neq Cxt2$$

If Cxt1 is a relaxation of Cxt2 then all cases that satisfy Cxt2 also satisfy Cxt1. Context relaxation can be implemented through either *reduction* or *generalization* where the latter transforms the context by enlarging the set of allowable values for an attribute. Reduction means that only a subset of the attributes of the original context need to be matched, hence reduction is managed by reducing the number of attributes required to be matched. It is not clear in the papers how the required context hierarchies are constructed. They may be manually generated. Context *restriction* works in the opposite way of context *relaxation* implemented through *expansion* and *specialization*. Expansion strengthens constraints by enlarging the number of attributes required to match while specialization strengthens constraints by removing values from a constraint set for an attribute, both amounts to a decreased number of cases that satisfy the resulting context. It was maintained that restriction of the context lead to the increase in similarity between cases.

The retrieval function is founded on an iterative retrieval driven by such context manipulations. Hence, depending on the result of retrieval, “the system allows for manual or automatic query modifications through context restriction or relaxation” (Jurisica & Glasgow, 1997). Precision and recall, which are typically used in information retrieval, are used for evaluation of retrieval.

A weighted average of attribute values is computed to find the final solution, for example in the servo domain and for inverse kinematics. This may not be possible in other domains when the values are nonnumeric, the authors note, and sometimes specific adaptation rules may be needed that use more domain models.

TA3 does not employ a solution adaptation in the classical sense. It merges solutions from several cases. There is a somewhat similarity between TA3 and the CBR approach employed in (Ros et al., 2009), which we discussed in section 4.2, in the sense that they both “adapt” the *query*, the new problem through reformulating or manipulating the problem description.

10 SHEILA

SHEILA (Software for Hierarchical Extraction and Imitation of drum patterns in a Learning Agent) is a system where the motor system is trained to produce various rhythmic patterns by imitating a human drum player’s movements (Tidemann & Öztürk, 2008), i.e. the system acts as a “groovy” drummer. SHEILA was tested on a simulated robot. It combines sound and movement processing through imitation of movements that render the intended rhythmical patterns. It has a motor planning component as well as a movement execution component. The movements that produce a particular pattern are segmented into a set of planning cases on a self-organized basis, and these cases are stored in the motor case base. The stored cases (i.e. patterns) can then be reused to produce novel sequences of drum patterns. An inherent quality of the system is its neural network representation; this makes it possible to generate novel sequences that will be *similar but not identical* to the original training sequence, more on this in the following section. Motor planning happens in a nondeterministic domain which gives a dynamic characteristic to it; planning depends on the visual feedback. This leaves out the possibility of one-shot planning of movements.

The underlying architecture is based on the model in neuroscience (Wolpert & Kawato, 1998) that is described in Section 7. However, in Wolpert & Kawato (1998), the context signal is a visual cue, while in SHEILA the context signal is some intrinsic properties pertinent to a drum track, coding high-level structure of the track, like the ones dancers use as cues to a choreographed dance. Hence, the context signal represents rhythmic patterns drummers play. During imitation, the modules self-organize their learning, which leads to representation of an arm movement that is distributed across a subset of modules. The training data consist of the current state of the limbs, the desired next state, and the context signal. The motor system learns to generate the next action to be issued to achieve the desired next state. The context signal is “extra help” for the motor system, so the most appropriate modules can be chosen *before* the actual movement is initiated.

Motor control involves selection of the motor command at each time step that best suits the current and the desired states of the body, and the context signal biases this selection. Motor command is selected through selection of the controller(s). The whole movement, therefore, involves a series of controller selections. Each “controller selection” involves a *retrieval* in CBR terminology.

10.1 Case representation

A motor planning case captures a situation-action-consequence representation, contrary to the situation-action structure predominant in classical CBR. Figure 5 depicts the case structure. The problem description part consists of three vectors that specify the desired next state of the limb, its current state, and the current context signal, respectively. States can be coordinates or joint angles of limbs.

$$\left\{ \left[\begin{array}{c} currentState \\ desiredNextState \\ contextSignal \end{array} \right], \left[\begin{array}{c} ActGenNet \\ ActSimulNet \\ CtxModel \end{array} \right] \right\}$$

Figure 5 Case Representation in the motor domain. A case has two main components: a problem description and a solution description. A problem description, in turn comprises three subcomponents: current state, desired state and context signal. The solution part also embraces three elements: ActGenNet, ActSimulNet, and CxtModel.

The case solution embraces three models, functionally linked together (see Figure 6). In SHEILA, all three are implemented as Echo State Networks (ESNs), a novel neural network architecture characterized by its large hidden layer and fast training algorithm (Jaeger & Haas, 2004). One of the components is the action generator (i.e. ActGenNet in Figure 6) that, given the current and desired states, generates the next action to the motor. ActGenNet corresponds to an Inverse model in neuroscience terminology. The second component is a context model (CxtModel) that makes a preliminary prediction of the degree of relevance of the case in the current situation, based on the context signal. This is the *relevance predictor*, similar to the responsibility predictor in (Wolpert & Kawato, 1998).

The third component, ActSimulNet, which corresponds to a forward model in (Wolpert & Kawato, 1998), constitutes a departure point between the conventional CBR and CBR at the motor level. Retrieval has a peculiarity pertinent to motor control; it has an internal (and somewhat premature) “revise-like” additional stage. As a consequence, the solution part incorporates the “action simulation” (i.e. ActSimulNet) component. This model, through predicting the consequences of the action (i.e. the next state of the limbs when this action is issued to the motor system) suggested by the ActGenNet in the same case solution, enables the system to anticipate how close the actual next state will be to the desired next state, without executing the action (i.e. issuing the motor command). This resembles, in mainstream AI, determination of the consequences of an action in the external world using a symbolic domain model. However, the two simulations are radically different representation-wise. ActSimulNet is not a symbolic representation and it models the body/actuators, rather than the environment.

A case does not represent a whole movement and, in that sense, is not similar to a case representing a whole “summer vacation in New Delhi in August 98” episode. This is because the unit of representation that can be useful and can be re-used is not at the “summer vacation” level, but is at the much more fine-grained and “movement characteristics” level. In this respect SHEILA and TA3 (Jurisica & Glasgow (1997)) are similar; their case representations heavily involve continuous values. They are similar also because they rely on the notion of context-sensitivity in the similarity measurement and retrieval. However, they radically differ in the way they define the context, the change/switch of context and its particular role in the

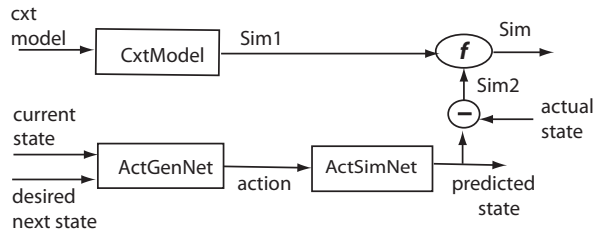


Figure 6 Similarity judgment underlying retrieval in motor control.

similarity/retrieval process, albeit context changes much more dynamically in SHEILA and it is not manually defined but extracted from the sound data, i.e. the context signal is *data driven*. The nature of the self-organizational approach that is employed in SHEILA yields a system that dynamically defines context signals and the creation of cases. This is an obvious feature for a system that implements an imitative capability; the ability to learn unknown movements by imitating them.

According to Wolpert & Kawato (1998), re-usable memories in the motor subsystem are not comprised of a single generic function that can produce the required action in all situations, but represented through multiple networks.

10.2 Retrieval of movement cases

In classical CBR, relevance is measured by similarity, typically between the problem descriptions of the new and a past case, using various similarity metrics. In SHEILA's CBR account, relevance is measured through two factors: (i) context similarity, and (ii) predicted (i.e. simulated) performance of the case. Each case evaluates its own similarity/confidence at the current time step on the basis of how good its context model matches the current context and the prediction made by the ActSimulNet of how well the action generated by the ActGenNet performs in controlling the limbs. The cases with the best combination of contextual similarity and predicted performance are assigned high relevance.

Figure 6 illustrates how the three models represented in a case solution are involved in the similarity judgment process during retrieval. The underlying similarity measurement algorithmically shown in Figure 7. The first component, *Sim1*, refers to the similarity between the current context and the context of $case_i$ in the case base, while the second component, *Sim2* represents the predicted performance of the couple of ActGenNet and ActSimulNet.

1. $Sim_1 = sim(context_{new}, context_{case_i})$
2. $Sim_2 = sim(predicted_next_state, actual_state)$
3. $Sim(case_{new}, case_i) = f(Sim_1, Sim_2)$

Figure 7 The algorithm of similarity judgement.

The following two sections describe the two components of this similarity judgment process.

10.2.1 Contextual Similarity

Context signal is part of the mechanism behind the relevant judgment of cases. The context model, CxtModel, reveals how much the current context signal coheres with the context representation of the past case under consideration. The motor commands issued to an arm in order to lift a heavy hard-cover book and a paper-back pocket book are different. Humans can, based on their understanding of the object they intend to interact with, flexibly prepare their motor commands.

A CxtModel is a predictor. It uses context information to predict how well the case may control the limbs - that is, how well the action provided by the case solution can do in the current context. This is depicted as *Sim1* in figure 6 and 7. In situations where the current state is ambiguous, the context information helps the architecture choose the best case without having to try out several case solutions, and converge towards the correct one. This leads to increased performance, since non-optimal cases need not be considered for controlling the limb. The output of the CxtModel is a scalar measuring contextual similarity, hence *Sim1* (see Figure 6). *Sim1* may be calculated as explained in Section 10, similar to the way the relevance predictor (or responsibility predictor in Wolpert’s terms) computes p_t^i .

10.2.2 Predicted performance relevance

The ActSimGen predicts the next state of the system when/if the $case_i$ takes the control, based on the current state and the output action from its paired ActGenNet. Based on a prediction of next state and the actual state of the limbs, the system calculates a predicted performance of the couple of ActGenNet and ActSimulGen. Higher performance estimation (i.e. *Sim2* in figures 7 and 6) indicates higher similarity between this case and the current case. This similarity judgment, hence, involves a simulation of the case solution, without the action suggested by the case necessarily being executed.

10.2.3 Overall similarity

Each case computes its own overall relevance/similarity, *Sim*. It is the function (i.e. f) in figures 7 and 6, and corresponds to λ_t^i in Figure 4. It may be computed using Equation (2) in SHEILA. *Sim* represents how much control the case should exert over the robot. It is a combination of a *prior* prediction of how well the module performs (i.e. output of the CxtModel, *Sim1*) and “posterior” knowledge of how well the ActSimulGen and ActGenNet perform together (represented by *Sim2*).

In this account of CBR, the solution of a case is simulated to find out how good the suggested action is. In SHEILA experiments, only a few modules contributed significantly at each time step. The final action (i.e. the motor command represented as a vector) is computed based on the actions suggested by the relevant cases and in proportion with their relevance. This may be done as in SHEILA (see Equation (3)). The final action computed in this way is sent to the motor system for execution.

10.3 Reuse

In “reuse”, solutions of these cases (if more than one with λ value higher than a certain threshold) are fused on the basis of their λ values that serve as weights indicating the confidence (or relevance) of each case.

A weighted fusion of actions suggested by retrieved cases is performed where the involved cases influence the computation of the resultant motor command based on the degree to which they are confident in their match with the current situation (Equation 3). This would provide a resultant vector representing the λ -weighted motor commands (which were suggested by the cases). In SHEILA experiments (Tidemann & Öztürk, 2008), only a few modules seem to contribute significantly at each time step.

These weighted motor commands are summed and applied to the robot (Equation 3). Hence, modules that make good predictions will have more influence over the robot than modules with bad predictions. This is how the architecture implements switching of control between modules. In this way the architecture allows for multiple modules to collaborate to control the robot, through a fusion of actions suggested by a set of relevant/confident planning cases.

The action generated in this way, is adjusted according to the perceived performance error feedback (i.e. the discrepancy between the perceived and intended states in the previous time step). The resultant action (i.e. motor command) is subsequently adjusted on the basis of available sensory feedback. Thus, reuse is characterized by (a) fusion of relevant planning cases, if there

are more than one case, (b) adapting/correcting the computed action, if necessary. These will now be explained.

Plan retrieval and plan reuse are not two sequential processes; planning is incremental and consequently, retrieval and adaptation are intertwined. At each time step a planning case (or multiple cases, if more than one collaborate) is retrieved. The planning case provides the next motor command to be issued. If necessary, the motor command is adapted before being actuated. Consequently, plan adaptation/correction also is an incremental process, whereby not a complete plan is adapted ahead of execution, but an action in the plan is adapted at each time step, immediately before it is issued to the plan executor.

The retrieved case(s) may not perfectly match the current situation. In addition, in an embodied setting, perturbations and noise are likely to occur. Therefore, an on-line correction ability is crucial for good performance of the robot. This is similar to the functionality of the human cerebellum (Kandel et al., 2000); the cerebellum receives signals about the current state, the desired next state and the motor commands about to be issued to the muscles. The cerebellum then detects discrepancies between the desired next state and the consequences of applying the motor commands given the current state, and makes a last-minute adjustment before the motor commands are propagated down the spinal cord. If the action computed by the CBR mechanism does not fulfill the intended goal, the action is corrected before being sent to the robot. Hence, here are three sources of the need for adaptation: (i) the plan does not suit the current situation well, (ii) the environment can be modified between two time steps, or (iii) the body part (i.e. effector) can be perturbed. The latter two will lead to higher performance error - notice the difference between *prediction error* (which was used in confidence calculation) and the *performance error*. Performance error may be compensated by online adaptation in the reuse stage, also called online correction in the motor domain, an optional process that ensures stability.

10.4 Retain

Retain takes place during learning. More than one module can learn and how much each inverse and forward module would learn is determined on the basis of how well, relative to other cases, it already does in the current situation. Those that are doing best are encouraged to refine their representation even more. In the SHEILA implementation, the inverse and forward models, implemented as ESNs, learn through updating the weights in the output layer. The learning rule of the forward model involves the prediction error, the discrepancy between the actual next state and the next state that the model itself had predicted. The inverse model, on the other hand, learns using the discrepancy between the actual motor action and the motor command that the model itself had suggested. In general, only a small subset of cases truly learn at each step. In this way, the system self-organizes representation and the learning of cases.

Now we will draw the commonalities between the CBR applications in the real-time, dynamic nondeterministic domains and point to the characteristics of the tasks peculiar to the lower levels of cognition-action continuum and discuss how CBR is revised to tackle the action selection and planning at the motor level.

11 Discussion

A common property of classical CBR applications are the symbolic, conceptual representations. We suggest that the constraints on the format and the content of cases need to be revised. This is also in line with recent attempts with regard to the revision of the notion of “representation”. For example, to broaden the scope of “representation”, Markman & Dietrich (2000) introduce “transitional representations”, which are examples of non-conceptual, non-symbolic representations. An episodic memory triggered by a rare scent may be different from a physician’s memory by a patient’s complaint about red spots on their skin but they may be similar in principal, regarding the activation process. An assumption of classical CBR seems to be that translation of a perceptual input to a conceptual form is required before CBR can be applied. This may

not necessarily be the case. It may be possible that a scent, without a language-like attribution may activate a past memory in which the same scent was perceived. Obviously, indexing and representation of such an experience will be anything but in terms of concepts. We suggest, therefore, to relax the constraint that cases can involve (only) symbolic representations.

Non-symbolic cases in most of the systems reviewed in this paper qualifies as representations, since they are enduring, although they may be outside the scope of the traditional, symbolic definition of representation. Our notion of representation is in line with the effort in (Markman & Dietrich, 2000) which lay down some criteria for representation.

The genuine mechanism of CBR is shared by various planning task both where cases involve symbolic and nonsymbolic representations, aligning with the hypothesis that both are managed by the same functional neural circuit despite three facts: 1) plans are represented differently in the classical and the new CBR, 2) the plans do not involve only the characteristics and the states of the external world in the revised CBR, but also that of actuators, 3) the target of planning changes along the cognition-action continuum and as the target gets closer to the actuators, the involved representations moves from symbolic to non-symbolic representations. The reviewed systems reveal that learning cases on-line is a natural part of CBR systems at lower levels (Sugandh et al., 2008; Gabel & Veloso, 2001; Karol et al. (2003)); imitation, learning by observation, recording and then annotation of cases by human experts all constitute a step toward solution of case grounding problem.

When nondeterminism is concerned, prediction becomes an integral part of planning along the entire cognition-action scope. In actuator level, prediction is facilitated by the “forward models” of the actuators (Tidemann & Öztürk, 2008) which were part of the case representation. Prediction enabled better estimation of the relevance of various cases and decreased the need and reduce the scope of adaption. In slightly higher levels but still lower side of the cognition-action continuum, case solutions either directly represent action-consequence couples (Kerkez & Cox, 2003), or outcome of replaying a retrieved solution is predicted by executing an anticipatory function (Ros et al., 2009). All these minimize the need for adaptation of the retrieved solution. Yet another way is to do pre-adaptation as a way to evaluate the fitness of a candidate case, as has been done in (Jurisica & Glasgow, 1997; Ros et al., 2009). The second way is to use predictive models (e.g. forward models in Tidemann & Öztürk, 2008) to better estimate the relevance of various cases and to decrease the need and reduce the scope of adaption.

Although far from proving our hypotheses that CBR may unify symbolic and subsymbolic AI through its applicability on the broad scope of cognition and action, the systems we reviewed in this paper and the common characteristics we have inferred would provide enough motivation for further pursuit of this line of research.

References

- Aha, D. W., Molineaux, M., & Ponsen, M. (2005). Learning to win: Case-based plan selection in a real-time strategy game. In *in Proceedings of the Sixth International Conference on Case-Based Reasoning*, (pp. 5–20). Springer.
- Arbib, M. (2002). The mirror system, imitation, and the evolution of language. In K. Dautenhahn, & C. L. Nehaniv (Eds.) *Imitation in animals and artifacts*, (pp. 229–280). Cambridge: MIT Press.
- Arkin, R. C. (1989). Motor schema-based mobile robot navigation. *International Journal of Robotic Research*, 8(4), 92–112.
- Ashley, K., & Hammond, K. (1988). Compare and contrast, a test of expertise. In *in Proceedings of the Sixth National Conference on Artificial Intelligence*, (pp. 273–284).
- Billard, A., & Hayes, G. (1999). DRAMA, a connectionist architecture for control and learning in autonomous robots. *Adaptive Behavior*, 7(1), 35–63.

- Blakemore, S. J., Goodbody, S. J., & Wolpert, D. M. (1998). Predicting the consequences of our own actions: The role of sensorimotor context estimation. *Journal of Neuroscience*, *18*, 7511–7518.
- Brooks, R. A. (1991). Intelligence without representations. *Artificial Intelligence Journal*, *47*, 139–159.
- Cangelosi, A., & Riga, T. (2006). An embodied model for sensorimotor grounding and grounding transfer: Experiments with epigenetic robots. *Cognitive Science*, *30*(4), 673–689.
- Cisek, P. (2005). Neural correlates of reaching decisions in dorsal premotor cortex: Specification of multiple direction choices and final selection of action. *Neuron*, *45*.
- Cunningham, P. (2009). A taxonomy of similarity mechanisms for case-based reasoning. *IEEE Trans. Knowl. Data Eng.*, *21*(11), 1532–1543.
- Dapretto, M., Davies, M. S., Pfeifer, J. H., Scott, A. A., Sigman, M., Bookheimer, S. Y., & Iacoboni, M. (2005). Understanding emotions in others: mirror neuron dysfunction in children with autism spectrum disorders. *Nature Neuroscience*, *9*, 28–30.
- de Mántaras, R. L., Mcsherry, D., Bridge, D., Leake, D., Smyth, B., Susan, C. R. A. W., Faltings, B. O. I., Maher, M. L. O. U., Cox, M. T., Forbus, K., Keane, M. A. R. K., Aamodt, A., & Watson, I. A. N. (2005). Retrieval, reuse, revision and retention in case-based reasoning. *The Knowledge Engineering Review*, *20*(03), 215–240.
- Demiris, Y., & Khadhour, B. (2006). Hierarchical attentive multiple models for execution and recognition of actions. *Robotics and Autonomous Systems*, *54*, 361–369.
- Díaz-Agudo, B., Gervás, P., & González-Calero, P. A. (2003). Adaptation guided retrieval based on formal concept analysis. In *ICCB*, (pp. 131–145).
- Dinerstein, J., Ventura, D., & Egbert, P. K. (2005). Fast and robust incremental action prediction for interactive agents. *Computational Intelligence*, *21*(1), 90–110.
- Dinstein, I., Gardner, J., Jazayeri, M., & Heeger, D. (2008). Executed and observed movements have different distributed representations in human aIPS. *Journal of Neuroscience*, *28*(44), 11231–11239.
- Dinstein, I., Hasson, U., Rubin, N., & Heeger, D. (2007). Brain areas selective for both observed and executed movements. *Journal of neurophysiology*, *98*(3), 1415–1427.
- Fagan, M., & Cunningham, P. (2003). Case-based plan recognition in computer games. In *Proceedings of the Fifth ICCBR*, (pp. 161–170). Springer.
- Filimon, F., Nelson, J. D., Hagler, D. J., & Sereno, M. I. (2007). Human cortical representations for reaching: Mirror neurons for execution, observation, and imagery. *NeuroImage*, *37*(4), 1315–1328.
- French, R. M. (1994). Catastrophic forgetting in connectionist networks: Causes, consequences and solutions. In *Trends in Cognitive Sciences*, (pp. 128–135).
- Gabel, T., & Veloso, M. M. (2001). Selecting heterogeneous team players by case-based reasoning: A case study in robotic soccer simulation. Tech. Rep. CMU-CS-01-165.
- Gallese, V., & Goldman, A. (1998). Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, *2*(12).

- Gaussier, P., Moga, S., Banquet, J. P., & Quoy, M. (1998). From perception-action loops to imitation processes: A bottom-up approach of learning by imitation. *Applied Artificial Intelligence*, 1(7), 701–727.
- Grabert, M., & Bridge, D. (2003). Case-based reuse of software examplets. *Journal of Universal Computer Science*, 9, 627640.
- Grush, R. (2004). The emulation theory of representation: Motor control, imagery, and perception. *Behavioral and Brain Sciences*, 03, 377–396.
- Hamilton, A. F., & Grafton, S. T. (2007). The motor hierarchy : from kinematics to goals and intentions 22.
- Hammond, K. (1989). Case-based planning: Viewing planning as a memory task. In *in Perspectives in Artificial Intelligence*. Boston, MA: Academic Press.
- Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42, 335–346.
- Hartley, T., & Mehdi, Q. (2009). Online action adaptation in interactive computer games. *Comput. Entertain.*, 7(2), 1–31.
- Hartley, T., Mehdi, Q., & N., G. (2005). Online learning from observation for interactive computer games, hartley, t., mehdi, q., gough n (2005). July, Louisville, Kentucky, USA.
- Herwig, A., & Waszak, F. (2009). Intention and attention in ideomotor learning. *The Quarterly Journal of Experimental Psychology*, 62(2), 219–227.
- Imamizu, H., Kuroda, T., Yoshioka, T., & Kawato, M. (2004). Functional magnetic resonance imaging examination of two modular architectures for switching multiple internal models. *Journal of Neuroscience*, 24(5), 1173–1181.
- Ito, M. (1984). *The cerebellum and neural control*. New York: Raven Press.
- Ito, M. (2008). Control of mental activities by internal models in the cerebellum. *Nature Reviews Neuroscience*, 9(4), 304–313.
- Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication. *Science*, 304(5667), 78–80.
- Jordan, M. I., & Rumelhart, D. E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16, 307–354.
- Jurisica, I., & Glasgow, J. (1995). Applying case-based reasoning to control in robotics. In *In 3rd Robotics and Knowledge-Based Systems Workshop*, (pp. 189–196).
- Jurisica, I., & Glasgow, J. (1997). Improving performance of case-based classification using context-based relevance. *International Journal of Artificial Intelligence Tools*, 6, 3–4.
- Kandel, E. R., Schwartz, J. H., & Jessell, T. M. (2000). *Principles of neural science*. New York: McGraw-Hill.
- Karol, A., Nebel, B., Stanton, C., & Williams, M.-A. (2003). Case based game play in the robocup four-legged league part 1 the theoretical model. In *RoboCup*, (pp. 739–747). Springer.
- Kerkez, B., & Cox, M. (2003). Incremental case-based plan recognition with local predictions. *International Journal on Artificial Intelligence Tools: Architectures, Languages, Algorithms*, 12, 413–463.

- Koehler, J. (1996). Planning from second principles. *Artificial Intelligence*, *87*, 148–187.
- Kohler, E., Keysers, C., Umiltà, M., Fogassi, L., Gallese, V., & Rizzolatti, G. (2002). Hearing sounds, understanding actions: action representation in mirror neurons. *Science*, *297*(5582), 846–848.
- Kolodner, J. (1993). *Case-Based Reasoning (Morgan Kaufmann Series in Representation and Reasoning)*. Morgan Kaufmann.
- Kolodner, J., Simpson, R., & Sycara, K. (1985). A process model of case-based reasoning in problem solving. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Los Angeles, CA*, (pp. 284–290).
- Kravitz, J. H., & Yaffe, F. (1972). Conditioned adaptation to prismatic displacement with a tone as the conditional stimulus. *Percept Psychophysiology*, *12*, 305–308.
- Leake, D., & Kinley, A. (1998). Integrating cbr components within a cbr components within a case-based planner. In *Proceedings of the AAAI-98 Workshop on Case-Based Reasoning Integration*, (pp. 96–105).
- Leake, D. B., Kinley, A., & Wilson, D. (1995). Learning to improve case adaptation by introspective reasoning and cbr. In *In Proceedings of the First International Conference on Case-Based Reasoning*, (pp. 229–240). Springer-Verlag.
- Lee-Urban, S., & Muñoz-Avila, H. (2009). Adaptation versus retrieval trade-off revisited: An analysis of boundary conditions. In *ICCBR*, (pp. 180–194).
- Likhachev, M., & Arkin, R. C. (2001). Spatio-temporal case-based reasoning for behavioral selection. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, (pp. 1627–1634).
- Likhachev, M., Kaess, M., & Arkin, R. C. (2002). Learning behavioral parameterization using spatio-temporal case-based reasoning. In *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 1282–1289).
- Lingnau, A., Gesierich, B., & Caramazza, A. (2009). Asymmetric fMRI adaptation reveals no evidence for mirror neurons in humans. *Proceedings of the National Academy of Sciences*, *106*(24), 9925–9930.
- Lopez, B., & Plaza, E. (1993). Case-based planning for medical diagnosis. In *Proceedings of the 7th international symposium on methodologies for intelligent systems (ISMIS'93)*, (pp. 96–105).
- Markman, A. B., & Dietrich, E. (2000). In defense of representation. *Cognitive Psychology*, *40*, 138–171.
- Massie, S., Wiratunga, N., Donati, A., & Vicari, E. (2007). From anomaly reports to cases. In *7th International CBR Conference (ICCBR'07)*, Springer, (pp. 359–373).
- Matarić, M. J. (2002). Sensory-motor primitives as a basis for learning by imitation: Linking perception to action and biology to robotics. In K. Dautenhahn, & C. L. Nehaniv (Eds.) *Imitation in animals and artifacts*, (pp. 392–422). Cambridge: MIT Press.
- Meltzoff, A. N., & Moore, M. K. (1977). Imitation of facial and manual gestures by human neonates. *Science*, *198*, 75–78.
- Meltzoff, A. N., & Moore, M. K. (1997). Explaining facial imitation: A theoretical model. *Early Development and Parenting*, *6*, 179–192.

- Miall, R. C., Weir, D. J., Wolpert, D. M. W., & Stein, J. F. S. (1993). Is the cerebellum a smith predictor? *Journal of Motor Behavior*, *25*(3), 203–216.
- Miall, R. C., & Wolpert, D. M. (1996). Forward models for physiological motor control. *Neural Networks*, *9*(8), 1265–1279.
- Muñoz Avila, H., & Hüllen, J. (1996). Feature weighting by explaining case-based planning episodes. In *EWCBR '96: Proceedings of the Third European Workshop on Advances in Case-Based Reasoning*, (pp. 280–294). London, UK: Springer-Verlag.
- Muñoz-Avila, H., & Cox, M. T. (2008). Case-based plan adaptation: An analysis and review. *IEEE Intelligent Systems*, *23*(4), 75–81.
- Oberman, L., Hubbard, E., McCleery, J., Altschuler, E., Ramachandran, V., & Pineda, J. (2005). Eeg evidence for mirror neuron dysfunction in autism spectrum disorders. *Cognitive Brain Research*, *24*(2), 190–198.
- Öztürk, P. (2009). Levels and types of action selection: The action selection soup. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, *17*(6), 537–554.
- Piaget, J. (1962). *Play, dreams and imitation in childhood*. New York: W. W. Norton.
- Ram, A. (1993). Indexing, elaboration and refinement: Incremental learning of explanatory cases. *Machine Learning*, *10*, 201–248.
- Ram, A., Arkin, R. C., Moorman, K., & Clark, R. J. (1997). Case-based reactive navigation: a method for on-line selection and adaptation of reactive robotic control parameters. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, *27*(3), 376–394.
- Ram, A., & Francis, A. J. (1996). Multi-plan retrieval and adaptation in an experience-based agent. In D. Leake (Ed.) *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, (pp. 167–183). AAAI Press.
- Ram, A., & Santamara, J. C. (1993). Continuous case-based reasoning. *Artificial Intelligence*, *90*, 86–93.
- Rizzolatti, G., Fadiga, L., Gallese, V., & Fogassi, L. (1996a). Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, *3*, 131–141.
- Rizzolatti, G., Fadiga, L., Matelli, M., Bettinardi, V., Paulesu, E., Perani, D., & Fazio, F. (1996b). Localization of grasp representations in humans by PET: 1. observation versus execution. *Experimental Brain Research*, *111*(2), 246–252.
- Robin, N., Dominique, L., Toussaint, L., Blandin, Y., Guillot, A., & Le Her, M. (2007). Effects of motor imagery training on returning serve accuracy in tennis: the role of imagery ability. *International Journal of Sport and Exercise Psychology*, *2*, 177–188.
- Ros, R., Arcos, J. L., de Mántaras, R. L., & Veloso, M. (2009). A case-based approach for coordinated action selection in robot soccer. *Artif. Intell.*, *173*(9-10), 1014–1039.
- Ros, R., Veloso, M., de Mántaras, R. L., Sierra, C., & Arcos, J. L. (2006). Retrieving and reusing game plays for robot soccer. In *Advances in Case-Based Reasoning*, (pp. 47–61). Springer.
- Rosenbloom, L., & Newell, A. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, *33*, 1–64.
- Russel, S., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. 3rd Edition: Prentice Hall.

- Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6), 233–242.
- Schank, R. C. (1982). *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. New York, NY: Cambridge University Press.
- Searle, J. (1983). *Intentionality: An Essay in the Philosophy of Mind*. New York: Cambridge University Press.
- Shadmehr, R., & Mussa-Ivaldi, F. (1994). Adaptive representation of dynamics during learning of a motor task. *Journal of Neuroscience*, 14(5), 3208–3224.
- Shin, Y., Proctor, R., & Capaldi, E. (2010). A review of contemporary ideomotor theory. *Psychological Bulletin*, 136(6), 943–974.
- Smyth, B., & Keane, M. T. (1998). Adaptation-guided retrieval: Questioning the similarity assumption in reasoning. *Artificial Intelligence*, 102, 249–293.
- Sugandh, N., Ontañón, S., & Ram, A. (2008). On-line case-based plan adaptation for real-time strategy games. In D. Fox, & C. P. Gomes (Eds.) *AAAI*, (pp. 702–707). AAAI Press.
- Tidemann, A., & Öztürk, P. (2008). Learning dance movements by imitation: A multiple model approach. In *31st Annual German Conference on AI*, vol. 5243 of *Lecture Notes in Computer Science*, (pp. 380–388). Springer.
- Veloso, M. M., Muñoz Avila, H., & Bergmann, R. (1996). Case-based planning: Selected methods and systems. *AI Communications*, 9(3), 128–137.
- Williams, J., Whiten, A., Suddendorf, T., & Perrett, D. (2001). Imitation, mirror neurons and autism. *Neuroscience and Biobehavioral Reviews*, 25(4), 287–295.
- Wolpert, D. M., C., M. R., & Kawato, M. (1998). Internal models in the cerebellum. *Trends in Cognitive Science*, 2.
- Wolpert, D. M., Doya, K., & Kawato, M. (2003). A unifying computational framework for motor control and social interaction. *Philosophical Transactions: Biological Sciences*, 358(1431), 593–602.
- Wolpert, D. M., & Flanagan, J. R. (2001). Motor prediction. *Current biology*, 11(18), 729–732.
- Wolpert, D. M., & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, 11(7-8), 1317–1329.
- Zimmermann-Schlatter, A., Schuster, C., Puhan, M. A., Siekierka, E., & Steurer, J. (2008). Efficacy of motor imagery in post-stroke rehabilitation: a systematic review. *Journal of NeuroEngineering and Rehabilitation*, 5(8), 8–18.