ELSEVIER

Cognitive Science 28 (2004) 75–88

COGNITIVE SCIENCE

http://www.elsevier.com/locate/cogsci

A rhythm recognition computer program to advocate interactivist perception

Jean-Christophe Buisson

IRIT (Institut de Recherche en Informatique de Toulouse) and ENSEEIHT, Toulouse, France

Received 21 March 2003; received in revised form 11 July 2003; accepted 6 October 2003

Abstract

This paper advocates the main ideas of the interactive model of representation of Mark Bickhard and the assimilation/accommodation framework of Jean Piaget, through a rhythm recognition demonstration program. Although completely unsupervised, the program progressively learns to recognize more and more complex rhythms struck on the user's keyboard. It does so without any recording of the input flow, and without any pattern matching in the usual sense. On the contrary, internal processes are dynamically constructed to follow and anticipate the user's actions. We show that these processes are representations of the rhythms in the interactivist sense, and that they emerge from non representational grounds, avoiding the symbol-grounding problem. They are not copies or transductions of reality, but ideal internal constructions of the agent, avoiding the circularity pointed out by Piaget. In practice, the active nature of this recognition process allows it to work even with noisy and complex input flows.

Keywords: Perception; Interactivism; Rhythm recognition; Piaget schemas

1. Introduction

According to interactivism (Bickhard, 1996; Bickhard & Terveen, 1995), cognitive representation cannot solely be composed of correspondences (like symbolic encodings), since it would have no genuine content about that reality. Bickhard shows that a more basic form of representation, which is interactive and temporal in nature, is necessary. Piaget (1954) also proved that mental representations cannot be simplified copies of the world, but his theory of perception (Piaget, 1967) fell in the trap of encoding and its contradictions. Nevertheless, he provided an interactive and temporal framework of assimilation schemes which have inspired our research.

E-mail address: buisson@enseeiht.fr (J.-C. Buisson).

0364-0213/$ – see front matter
doi:10.1016/j.cogsci.2003.10.003

Our computer demo program is designed to illustrate these ideas, and not to serve as real music recognition software (Coyle & I., 1998; Shmulevich & Povel, 2000). It provides a detailed and operational model of interactive representation, adding quantification to qualitative theories. As recommended by Rodney Brooks, the working program interacts with human users, in order to ensure that no important feature has been omitted, or modelled improperly. “The physical grounding of a system in the world forces its designer to deal with all the issues. . . . There is no room for cheating” (Brooks, 1991a).

In the first part of this paper, the demo program, its performances and its associated perception model are described in detail. Later, important problems of the psychology of perception are reviewed, and our results are used to defend an interactivist position.

2. The rhythm recognition demo program

2.1. Presentation

The rhythm recognition demo is a computer program which learns to recognize musical rhythms, although completely unsupervised. Fig. 1 shows screenshots of the program, when dealing with simple rhythms.

The user interacts with the program by rhythmically striking a single key on the keyboard. When a rhythm is recognized, a score with a single bar is displayed in the window and a black dot jumps from note to note, following and anticipating the player’s keystrokes. The user provides no indication of any error, so learning occurs without any supervision. The program does not recognize complex rhythms from the start; it must first be exposed to simple rhythms. When simple rhythms have been assimilated, the user may try new and more complex rhythms, which must resemble or be based on previous ones. The richer the musical knowledge of the program, the more able it is to assimilate new complex rhythms.

Some noise may be added in the form of random keystrokes, and learning still works well when the noise level is above 20% of all strokes.

The program is designed to distinguish between half-notes, quarter-notes and eighth-notes. It takes on average 4 or 5 periods for a new rhythm to be assimilated and displayed in the window, provided that the rhythm is indeed assimilable, that is to say it resembles previously assimilated rhythms. The user is not required to have perfect timing, even though people with some background in music are noticeably better trainers.

When a rhythm which has been assimilated in the past is played again on the keyboard, it takes one or two periods for the program to find the correct tempo and display its score on the screen. Therefore it can be assumed that the rhythm was memorized during its first presentation and is subsequently remembered by the program.

The black dot under the notes follows and anticipates the player’s strokes. If the user stops playing abruptly, the dot moves one or two more steps before vanishing with the bar display. A related phenomenon is that, for a well assimilated rhythm, when the user ‘forgets’ to strike a key, the black dot moves under the absent note, as if nothing had been omitted. People sometimes behave similarly when they read without noticing typographical errors.

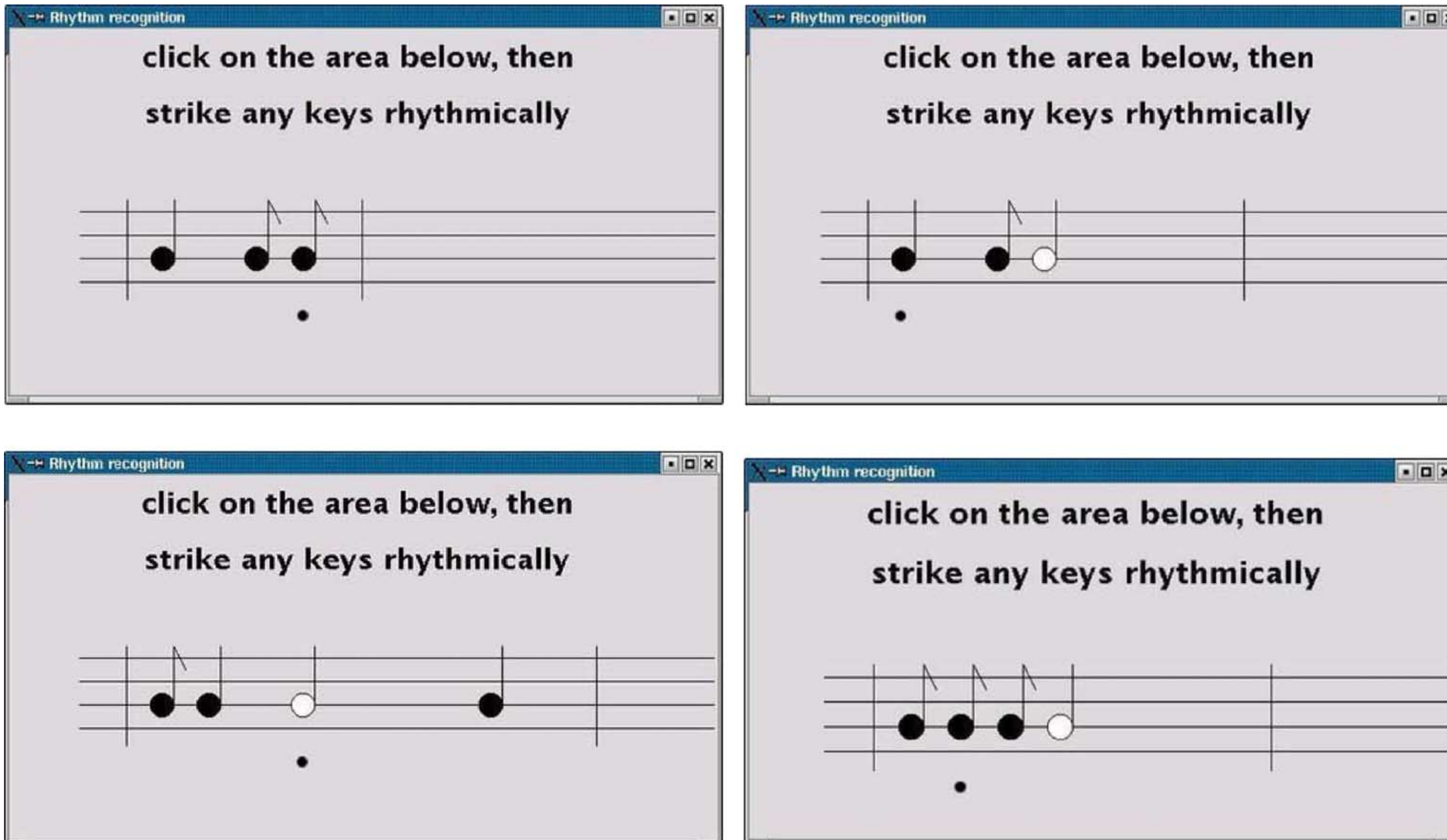

Fig. 1. Screenshots of the rhythm recognition computer program. The black dot under the notes follows and anticipates the player's keystrokes in real time.

When the user's tempo varies slightly, the program accommodates this change, and remains synchronized and in time. However, these tempo variations must not be too abrupt. Of course mutual synchronization often occurs; while the program is trying to accommodate variations in tempo, the user unconsciously tends to synchronize with the program.

2.2. How it works

2.2.1. How it does not work

The program does not match any part of the entered keystrokes against pre-stored patterns. In fact, it does not even record any part of the input strokes. Our solution is dynamic, and does not lie in any static pattern matching method.

2.2.2. Sensory-Motor Schemes

The program manages a set of Sensory-Motor Schemes (SMS), perfectly analoguous to Piaget's assimilating schemes (Piaget, 1954), and which play the same role of reacting to actions from the world (the keystokes) in the functioning of their structure, and accommodate to them in order to maximize this assimilation.

In general, each SMS is associated with a rhythm that the system already knows, and is encoded in the form of a sequence of pauses and keystroke anticipations. For example, the SMS associated with the rhythm [*half-note, quarter-note*] is the cyclic sequence: [*wait for a half-tempo, anticipate a stroke, wait for a quarter-tempo, anticipate a stroke*]. In fact, an SMS is a computer thread, that is to say an independent computer process, which plays its own sequence internally and silently in real time, at a certain tempo. And the main point is that it tries to assimilate the external (strokes) world, by trying to synchronize with it.

Initially, the system is equipped with a single SMS: [*half-note*], which corresponds to a simple 'tap-tap'. It is all it knows, and it tries to assimilate the whole world through this single 'tap-tap' mental structure. Obviously, it will not assimilate much unless the user strikes only this very simple rhythm. Even if the user plays a rhythm which corresponds exactly to an SMS's own rhythm, this SMS will be confronted with the problem of how to synchronize with the sensory (strokes) flow, and how to preserve this synchrony even when the user makes slight mistakes or when the tempo fluctuates.

Tempo synchronization is the first degree of what Piaget would call *accommodation,* but in this case it is only a means of fine tuning and it makes sense only when the SMS is grossly adapted to the situation. A deeper degree of accommodation is needed in the presence of more complex rhythms, when new SMSs need to be created. The second problem is how to leave the SMS population intact when all played rhythms are well assimilated by it, and how to expand this population in the right but minimal way in order to allow for the assimilation of new rhythms.

These two accommodation issues—tempo synchronization and SMS population expansion—will now be addressed.

2.2.3. Sensory-Motor Scheme synchronisation

Let us suppose that an SMS's musical score corresponds to a rhythm present in the real sensory (keystroke) flow, *and* that their respective tempos are identical, *and* that the current

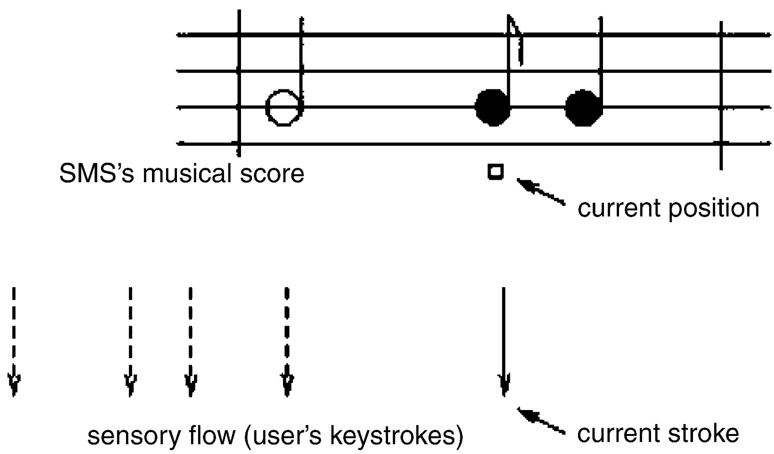

Fig. 2. A Sensory-Motor Scheme synchronized with the user's input keystrokes.

position in the SMS musical score corresponds to the current stroke in the sensory flow. This situation is illustrated in Fig. 2.

In this situation, the SMS is perfectly synchronized with reality, achieving its only goal, but any variation in the keystrokes tempo, for instance, could lead to unsynchronization. Presently the SMS assimilates reality, as Piaget would have put it, but it needs accommodation mechanisms to maintain this assimilation process.

We incorporated such a *temporal accommodation mechanism* in the computer demo to allow an SMS which is almost synchronous with the input strokes to maintain the right pace. Specifically, when an anticipated stroke comes too early, the SMS tempo is slightly quickened, and the SMS position in the score is realigned with the keystroke. A similar but reverse adjustment is made when an anticipated stroke comes a bit late.

Now that there is a mechanism to *maintain* synchronization, another is needed to *start* synchronization. We call it the *resynchronization mechanism*. When an SMS is not at all synchronous with the input strokes, its tempo is increased progressively (up to a certain limit) until it is able to catch up to the real keystrokes, usually within a few cycles.

2.2.4. Activity of a Sensory-Motor Scheme

The degree of activity for each SMS is permanently updated, measuring to what extent it is synchronized with the sensory flow. This is expressed as a number between 0 and 100, with 100 being perfect synchronization and 0 being no synchronization at all. When the synchronization is between the two, the absolute value of the corresponding activity degree (between 0 and 100) is not really significant; what matters is that it must allow the activity of several running SMSs to be reliably compared. This computation of SMS activity was done in an ad-hoc manner and much tuning was necessary to verify this function. It must be understood that there are many situations where it is difficult to determine which of two SMSs is best synchronized to the sensory flow. Finally, the musical score of the most active SMS is displayed on the screen, along with a moving dot which continuously shows the changing current position in the score.

2.2.5. Evolution of the SMSs population: a variation/selection algorithm

Now that SMS behavior has been described, we must explain how they are created and eliminated.

The population of SMSs evolves using a variation and selection algorithm, very similar to the evolution of the B-cell population in a mammals' adaptive immune system (Forrest & Hofmeyer, 2000). These algorithms are now actively being studied for their interesting properties, notably associative memory and inductive learning (Forrest & Hofmeyer, 2000; Segel, 1997). Until the middle of the last century, it was believed that pathogens left imprints on the lymphocytes to which they were bound, and that these imprints were duplicated as the lymphocytes replicated. The falseness of this theory is interesting as it is similar to the very common idea, explicit or not, that the mind can form internal copies of the world, using transduction mechanisms from sensory organs.

It is now known that when enough pathogens bind to a B-cell, this cell starts replicating quickly with a high rate of mutation. Some of the mutated B-cells will bind even better with this pathogen. After a few of these replication cycles, a population of highly adapted B-cells will emerge.

In our system, an SMS which is highly synchronized with the sensory flow is analoguous to the B-cell activated by its encounter with a pathogen. When the level of synchronization of an SMS with the actual input keystrokes is above a certain threshold, it starts replicating itself quickly, with a very high rate of mutation. A mutated SMS has a musical score derived from its parent's, with changes compatible with the principles of music. A note may be substituted, deleted or inserted (see Fig. 3).

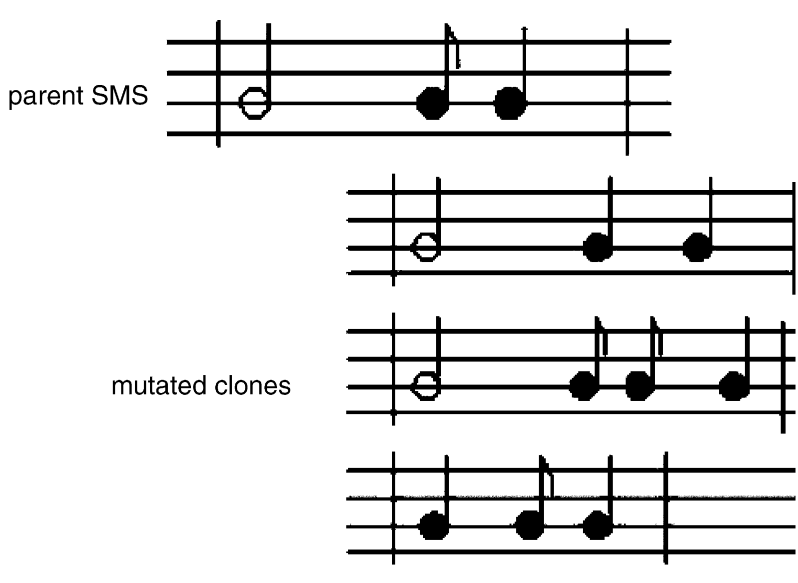

Fig. 3. Possible mutations for a replicated musical Sensory-Motor Scheme.

Among these mutated SMSs, some will synchronize with reality better than their parent, while others will not. The better ones will quickly rise above the threshold and replicate. Then after a few such cycles, a population of highly synchronized SMSs will emerge.

In order to manage an ever expanding population of SMSs, their life expectancy is limited, depending on their origin. The newly born mutated SMSs which perform poorly die after being inactive for too long. A parent SMS (an SMS which has entered the replicating process) has a much longer life expectancy, since it has proved useful at least once. This mechanism is similar to the creation of long lasting memory B-cells in the immune system.

2.2.6. Initial population of SMSs

At the beginning of the process, a single SMS is provided, associated with a one-one rhythm. The user must start with very elementary rhythms, since they must be first assimilated by this very simple SMS. The variation and selection algorithm then starts to produce a population of SMSs which is more and more adapted to the user's keystrokes. It will quickly adapt to changes and at the same time it will remember old rhythms for a long time.

On the other hand, it is also possible to provide an initial population of SMSs which allows the system to recognize more rhythms from the start. Like innate behavior such as breast feeding, this initial knowledge will disappear if it is not used for too long a period.

2.2.7. Noise resilience

The learning and recognition features continue to work when noise is incorporated into the sensory flow in the form of random keystrokes, even up to about 20% of all keystrokes. This resilience is due to the active nature of the perception SMS processes, which wait for the next keystrokes while ignoring all others. However, noise strokes too close to anticipated strokes sometimes lead to the desynchronization of SMSs; this problem would be reduced with a richer sensory flow, where it would be easier to distinguish a noise stroke from a real one.

3. Ideas highlighted by the demo program

3.1. The perception process is fundamentally temporal

Perception is a process, not a static phenomenon. This is very clear for audition and tactile feeling, where perception necessarily takes the same amount of time as the observed situation, and its results depend on the successive steps of the process. It is not merely a matter of accumulating information over time. Consider, for example, a melody slowed down ten times. Why is it so difficult to recognize? As pointed out by Shannon (1993), this should not be the case if we indeed had an internal static representation of it. In fact, it would be even easier!

In the case of tactile perception, Merlau-Ponty (1962) showed how it is necessary to manipulate a fabric with the right timing in order to correctly determine its nature.

The temporal nature of perception is perhaps less evident for modalities such as vision, where we have been trained to use the common analogy of the eye as a camera taking snapshots, but James Gibson among others has demonstrated that this is not the case (Gibson, 1966). Probably his major insight was to show that we detect invariances over time (as well as over space) in the sensory flows, and this detection cannot be carried out on one or even a few still images.

Intrinsic temporality is visible in our demo: each SMS possesses its own ‘standard’ tempo. It can adapt to a certain range of faster or slower tempos, but it is not infinitely extensible.

3.2. Sensation and action are inseparable in the perception process

To smell, you need to breathe in. To feel an object with your fingers, you need to manipulate it, or at least to move your fingers over its surface. In both these cases, there would be no perception without action.

Even with vision, activity is all important. Many experiments have been conducted with computers displaying images to users while connected to eye movement measuring devices. One of these consists in moving the computer’s image with the eye’s movements, so that the retinal image remains fixed. Under these circumstances, vision becomes completely ineffective! More precisely, a variety of perceptual phenomena occur, ranging from loss of contrast, to fragmentation, to the visual field becoming gray or black (Ditchburn, 1973).

In another experiment, observers were asked to discriminate between three previously unknown Chinese symbols, with conditions arranged so that they could only see each symbol when they fixed their eyes on its center; the symbol disappeared as soon as their eyes moved. The subjects found the task extremely unpleasant and difficult and needed hundreds of trials to complete the discrimination task (Nazir & O’Regan, 1990). As O’Regan and Noë put it: “It is like trying to recognize an object lain on your hand without manipulating it” (O’Regan & Noë, 2001).

In our demo, the active part of the agent is limited to timing aspects, since it does not move or act on anything; it just waits the right amount of time between keystrokes. However as previously mentioned, timing is an integral part of action.

3.3. Perceptive processes are composed of action/anticipation pairs

To smell, we first breathe in. Our theory is that we breathe in, *in order to test* whether the particular smell anticipated is there or not. We wouldn’t breathe in this way or at this time if we were not engaged in this particular perceptual activity. We move our fingers *in order to check* an anticipated sensation, etc. Even with vision, we make eye movements to verify anticipated predictions. As O’Regan and Noë put it (O’Regan & Noë, 2001):

> ‘Red’ is the structure of the changes that ‘red’ causes . . . the sensation that a patch of color is red, say, does not primarily derive from the fact that it is stimulating long-wavelength retinal cones. Rather, it generally derives from the *structure of the changes* in sensation that occur when you move your eyes around relative to the patch.

This is exactly what we have tried to model in our demo. The most elementary SMSs have the structure depicted in Fig. 4.

In this figure, the arrow with the ‘Action’ label indicates that an action is first performed, at its own tempo. ‘? Sensation’ indicates that, when ‘Action’ has been performed, the anticipated sensory feature ‘Sensation’ is tested.

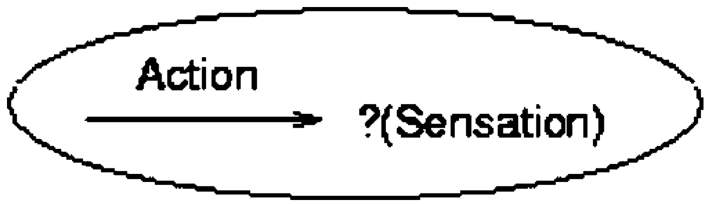

Fig. 4. Structure of a basic Sensory-Motor Scheme.

For example, 'Action' = 'breathe in' and 'Sensation' = 'coffee smell feature'. Or: 'Action' = 'eye movement' and 'Sensation' = 'a change that red causes'. In our recognition demo, it is clear that the musical SMSs are chained elements of this sort, one for each note in the SMS score. Clearly, 'Sensation' = 'anticipated stroke', but it seems that there is no real action. In fact, 'Action' = 'wait for the amount of time associated with the note'; it does not consist of any moving action, but timing is all that matters here (Fig. 5).

Let us call these Elementary SMSs E-SMS. They have several important properties:

- An E-SMS is *sensory-motor*, that is to say motor *and* sensory, the two aspects being inseparable.
- An E-SMS takes time to be performed; it takes a necessary tempo to perform 'Action', then to check 'Sensation'
- An E-SMS is normative: it can be true or false, depending on the occurrence of 'Sensation' after 'Action'. It can also be measured on a continuous scale between 0 and 1.

In our demo program, an SMS is a sequence of elementary SMSs, but other kinds of combinations are possible and still remain to be explored. Consider for instance the modular combination mode depicted in Fig. 6.

Here what is anticipated in not an elementary sensation, it is a complete assimilation activity. SMS1 is performed (if possible) *in order* to test whether SMS2 occurs. Such a higher form of SMS retains all the properties of the elementary ones: it is sensory-motor, since each of its parts is and so also is its manner of construction. Its temporal aspects are a combination of those of SMS1and SMS2. Finally, it is normative, being true if, when SMS1 is true, *then afterwards* SMS2 is true.

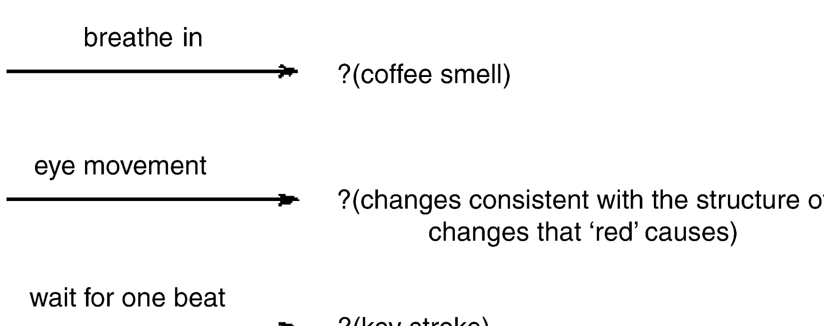

Fig. 5. Examples of elementary Sensory-Motor Schemes.

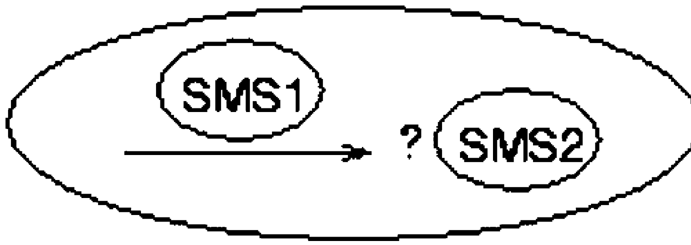

Fig. 6. Modular combination of Sensory-Motor Schemes.

3.4. Cognitivism and interactivism

In the western traditional way of thinking, intelligence is characterized in terms of relationships between ideas, and between ideas and the environment. In this view, which is often called *cognitivism*, representation plays the central role as the fundamental unit in the cognitive machinery. It is associated with a *perception* → *semantic inference* → *action* model of the mind, where perception plays the role of information gathering. The physical aspects of the system have no part except in input or output, which are never closely intertwined since they are separated by the *semantic inference* layer.

Until now this tradition has dominated artificial intelligence and cognitive science research, but its foundations have been strongly criticized recently by advocates of more dynamic and situated approaches to cognition.[1]

Notably, Rodney Brooks (1991a,b) has argued that the *sense* → *think* → *act* architecture is incapable of producing effective real world interaction for mobile robots. In his view, it is too difficult to build a detailed enough representation of the world to allow a robot to operate in an ordinary environment such as an office. Even if it were possible, the *sense* → *think* → *act* cycle would lead to unacceptably slow robotic behavior. In Brooks' view, the embodiment of intelligent systems is crucial. To the question of whether or not the mind can be disembodied he replies, "what is human about us is very directly related to our physical experiences" (Brooks, 1991a). This is very similar to Piaget's views where all competencies, even at the seemingly most abstract level, are rooted in the sensori-motor intelligence developed during the pre-language period. Set inclusion relations, for instance, are rooted in similar practical operations on objects (Piaget, 1967).

Other researchers think that intelligent action requires factors so distributed and temporarily structured that dynamic systems theory is the only framework needed by cognitive science (Thelen & Smith, 1994; Van-Gelder & Port, 1995).

Mark Bickhard has described in great detail the contradictions of the cognitivist position (Bickhard & Terveen, 1995). In particular, he has shown that no standard approach to representation is able to provide error detection *for the agent*. To do so, the system would have to be able to compare what a representation is supposed to be representing with what is actually being represented. Yet determining what is actually being represented is precisely the problem to solve.

The key to Bickhard's solution—interactivism—lies in considering the system while it interacts with its environment. At any one particular time, an agent must somehow indicate to itself what the next possible interactions in the environment are, and each of these interactions must be associated with indications of anticipated outcomes. For instance, one SMS might advise waiting for a quarter tempo and another waiting for a half tempo; both interactions have as an anticipated outcome a keystroke. Since the success or failure of the anticipated outcomes can be detected, these two interactive representations allow the system to differentiate two classes of (musical) environments. The agent must then decide which course of action to take among all the possible interactions, with a rational algorithm of its own. For example, in our demo, we update the activity level for each SMS, and choose the SMS which has the highest level. After the interaction is performed, the agent can test whether the anticipated outcomes are present or not. If they are not, the system can *internally* detect that it is in error and that another course of action must be taken. An agent with limited or no learning capabilities will have fixed indicators of interaction and anticipated outcomes and a fixed decision system, while a learning agent will be able to enlarge and refine both aspects. Our demo, for example, expands the population of SMSs, the success or failure of which adjusts the activity values, which in turn affects the decision algorithm.

Interactivism postulates that pairs (indications of interactive potential, indications of anticipated outcomes) are the fundamental forms of representation and knowledge from which all others derive.[2] Indeed, they constitute knowledge of potentialities and are a way for an agent to differentiate its environment and to gain access to its properties. At the same time, these pairs can be falsified. Anticipated outcomes may or may not hold; the representation is true or false, depending on the situation. The crucial point is that this error criterion is purely internal to the agent and does not depend on an external observer or interpreter to assess its normative value. This can be seen in the demo, where learning is totally unsupervised.

Such interactive representations are intrinsically general, since any control system that can control any particular interaction can in principle control a different interaction that manifests the same general interactive properties. For instance, in the previous example SMSs can in principle differentiate two classes of rhythmic environments, that of a hard rock concert as well as that of chamber music.

An indication of an interaction and its associated outcomes makes presuppositions about the environment, and these presupposed interactive properties are the representations of that environment, the predictions of that environment. The representational *content* in this model is *implicit* rather than explicit, and this is one of its major strengths, in that it avoids many foundational issues concerning representation, including the frame problem (Bickhard & Terveen, 1995).

3.5. Perception processes are all permanently active

Reoccuring images come to me after playing basketball for a long time. In my mind, I keep playing basketball before sleeping. My children report the same phenomena after playing video games. And after spending all day programming, it takes more than an hour to return to a ‘normal’ state. And probably the most obsessing of all are musical melodies, which can haunt us for hours or even days.

All these remarks are very ordinary yet significant. They inspired one of the main points in our model—the fact that all the SMS processes, once created, remain permanently active, helping to guide the agent's actions. I see basketball images at night, but the model predicts that I should also see them while I am playing if the real sensations were not so vivid as to overrun them. The evoked images do not have the power to impede the agent's activity. On the contrary, they GUIDE this activity. When actually shooting, all my previous shots try to synchronize with my present activity, and at the same time influence it. When I shoot, I am guided in real time by all my previous analogous shots.

To further illustrate this point of view, let us take a different example. When we listen to some background music distractedly, for instance when cooking, we can be caught humming synchronously at certain times, or merely listening at others. If the sound level drops too low or becomes even inaudible for short periods of time, we will be able to catch up. We argue that in all these cases, an internal dynamic sensori-motor scheme attempts and achieves synchronicity with the music and that its motor outputs are more or less inhibited. In this case, listening and singing are two aspects of the same process.

In our recognition program, all created SMSs remain active permanently, and they guide the agent's action; this feature is one of the most salient of our model. To prevent it from being too memory intensive in a computer simulation, remember that precautions were taken to limit the evolution of the SMS population.

This tendency of an SMS to apply whenever possible is also closely related to the problem of the agent's *needs* or *drives*.

Needs are endogenous activities that tend toward, and reach steady-state only when a discrepancy has been eliminated or minimized. In our recognition program, the relevant discrepancy is that of mismatches between endogenous and exogenous rhythms. Indeed, as soon as a new activity is created (with a newly born SMS), a drive to perform it is created. This activity, composed of SMSs, orients its movements and actions to function naturally, and it fulfils its associated needs by definition. Thus our agent's needs can be said to be created by its activity, whereas the classical model states the opposite, namely that the needs of an agent are the starting point of its mental activity. As Piaget has shown (Piaget, 1954), this position does not explain how such needs are able to orient movements and actions necessary to their satisfaction, nor how they are created.

3.6. Perceptive processes are the ideal creations of the agent, and not imperfect copies of reality

Piaget argued against copy theories of representation and knowledge throughout his career (see for example, Piaget, 1970). His argument is that, to construct a copy of something, what is to be copied must already be known, a fatal circularity.

To illustrate this contradiction, consider for example two music tunes played simultaneously at similar sound levels but that we are listening to only one of them, unknown to us. Our mind will try to make a mental copy of this first tune in order to learn it. To this end, it will need to eliminate from the global audio signal the second tune by using, in real time, extracting operations such as frequency filtering. However to apply these extracting operations at the right time and the right parameters, our mind will need to know this first tune in detail already, or else it risks damaging it in the process. Thus the contradiction: to make a copy of the tune in order to learn it, we need to know it beforehand.

Now, if we adopt a process framework, and only attempt to *discriminate* the first tune, the previous vicious circle becomes a virtuous spiral. It is not necessary to completely know one of the tunes to be able to somehow separate it from the other. If we are able to distinguish its basic tempo and rhythm (and our general musical culture will help us with that), these will help us gain access to finer clues such as the detailed rhythms of each bar which in turn will allow us to discriminate the note values, etc.

What is constructed during this spiral movement is not a copy, but *a discrimination process based on the activity and anticipation* of notes and rhythms, and will usually occur after several listenings and refinements. This is exactly what is done in the demonstration program. The partially adapted SMSs are the discrimination processes at intermediate stages, which finally evolve and stabilize to become the best discrimination processes, fundamentally temporal and sensori-motor.

Another important point about these interactive perception processes is that they are the ideal constructions of the agent. If they are created out of a noisy sensory flow, noise is not incorporated into them. They may at first oversimplify the situation, but they have a tendency to incorporate more and more complexity. In our demo, a rich rhythm may be incorrectly identified for a few seconds, for example two quarter notes being mistakenly taken for one half note, but this kind of error is quickly corrected.

Indeed the ability to extract meaningful information out of noisy sensory flows may have important practical applications which need to be researched.

4. Conclusion

Our rhythm recognition software is a fairly simple, 400 line Java program. Nevertheless, it has allowed the implementation of the most important ideas of interactivist representation as well as assimilation schemes *à la* Piaget, including the inherent temporality of interaction and anticipation; how representation emerges out of interactive anticipatory organization; the emergence of truth value in anticipations, using an internally detectable representational error; the inherent variation and selection constructivist nature of learning, which is controlled by this internal error detection.

An original algorithm analoguous to the mammal immune system has been used to model the variation/selection process needed to accommodate existing SMSs and finely adapt them to the situation. The system accomodates each new rhythm by spawning new assimilating SMSs. This contrasts with most learning algorithms based on neural networks, where past experience is melted into a small set of synaptic weights.

Future experiments on other sensory modalities and beyond perception will show if our model exhibits the same properties and remains computationally tractable.

Notes

1. These alternate views on cognition did previously exist, most notably in Europe with Bergson (1929), Piaget (1954), and Merlau-Ponty (1962).

2. A discrete version of interactivism is presented here. There are continuous versions of the model as well (Bickhard & Terveen, 1995).

References

Bergson, H. (1929). *Matter and memory*. New York: Macmillan.

Bickhard, M., & Terveen, L. (1995). *Foundational issues in artificial intelligence and cognitive science*. New York: Elsevier, North-Holland.

Bickhard, M. H. (1996). The emergence of representation in autonomous embodied agents. In *Proceedings of the Fall 1996 AAAI symposium on embodied cognition and action*.

Brooks, R. A. (1991a). Intelligence without reason. In *International joint conference on artificial intelligence*, pp. 569–595.

Brooks, R. A. (1991b). Intelligence without representation. *Artificial Intelligence, 47*, 139–160.

Coyle, E., & I., S. (1998). A system for machine recognition of music pattern. In *Proceedings of the IEEE international conference on acoustics, speech and signal processing*.

Ditchburn, R. W. (1973). *Eye-movements and visual perception*. Oxford: Clarendon Press.

Forrest, S., & Hofmeyer, S. (2000). Immunology as information processing. In I. L. S. Cohen (Ed.), *Design principles for the immune system and other distributed autonomous systems*. Oxford University Press.

Gibson, J. (1966). The problem of temporal order in stimulation and perception. *Journal of Psychology, 62*, 141–149.

Merlau-Ponty, M. (1962). *The Phenomenology of perception*. London: Routledge and Kegan Paul.

Nazir, T. A., & O'Regan, J. K. (1990). Some results on translation invariance in the human visual system. *Spatial Vision, 5*(2), 81–100.

O'Regan, J. K., & Noë, A. (2001). A sensori-motor account of vision and visual consciousness. *Behavioral and Brain Sciences, 24*, 5.

Piaget, J. (1954). *The construction of reality in the child*. New York: Basic.

Piaget, J. (1967). *Biology and knowledge*. Chicago: University of Chicago Press.

Piaget, J. (1970). *Genetic epistemology*. New York: Columbia.

Segel, L. A. (1997). The immune system as a prototype of autonomous decentralized systems. In *Proceedings of the IEEE conference on systems, man and cybernetics*.

Shannon, B. (1993). *The representational and the presentational*. Harvester Wheatsheaf.

Shmulevich, I., & Povel, D. (2000). Measures of temporal pattern complexity. *Journal of New Music Research, 29*(1).

Thelen, E., & Smith, L. B. (1994). *A dynamic systems approach to the development of cognition and action.* Cambridge, MA: Bradfort Books.

Van-Gelder, T., & Port, R. (1995). It's about time: an overview of the dynamical approach to cognition. In R. Port & T. V. Gelder (Eds.), *Mind as motion: Explorations in the dynamics of cognition.* MIT Press.