Towards a Computational Model of Perception and Action in Human Computer Interaction

Pascal Haazebroek and Bernhard Hommel

Cognitive Psychology Unit & Leiden Institute for Brain and Cognition
Wassenaarseweg 52, Leiden, 2333 AK The Netherlands
{PHaazebroek,Hommel}@fsw.leidenuniv.nl

Abstract. The evaluation and design of user interfaces may be facilitated by using performance models based on cognitive architectures. A recent trend in HCI is the increased focus on perceptual and motor-related aspects of the interaction. With respect to this focus, we present the foundations of HiTEC, a new cognitive architecture based on recent findings of interactions between perception and action in the domain of cognitive psychology. This approach is contrasted with existing architectures.

Keywords: Cognitive Architecture, Perception, Action, HCI, action effect learning, PDP, connectionism.

1 Introduction

The evaluation and design of user interfaces often involves testing with human subjects. However, sometimes this is too expensive, impractical or plainly impossible. In these cases, usability experts often resort to analytical evaluation driven by their intuition rather than empirically obtained findings or quantitative theory.

In these situations, computational models of human performance can provide an additional source of information. When applied appropriately, these models can interact with user interfaces and mimic the user in this interaction, yielding statistics that enable the usability engineer to quantitatively compare interaction with alternative interface designs or to locate possible bottlenecks in human computer interaction. As more and more aspects of our lives are becoming increasingly 'computerized', even small improvements that slightly facilitate user interaction can scale up to large financial benefits for organizations. In addition, using computational models of human performance may contribute to deeper insights in the mechanisms underlying human computer interaction.

Usually, models of human performance are task specific instances of a more generic framework: a cognitive architecture [1]. Such an architecture (e.g., ACT-R, [2]; SOAR, [3]; EPIC, [4]) describes the overall structure and basic principles of human cognition, covering a wide range of human cognitive capabilities (e.g., attention, memory, problem solving and learning).

Recently, the focus in Human Computer Interaction is no longer only on purely cognitive aspects, but also on the perceptual and motor aspects of interaction. Computers, mobile phones, interactive toys and other devices are increasingly

V.G. Duffy (Ed.): Digital Human Modeling, HCII 2009, LNCS 5620, pp. 247–256, 2009.

equipped with advanced displays and controls, such as direct manipulation GUI's, touch screens, multi-function keys et cetera, that allow for user interfaces that draw on a rich body of real world perceptual-motor experience in the human user [5].

To account for perceptual and action-related interactions, some cognitive architectures have extended their coverage from primarily cognitive processes to perceptual processing and response execution (e.g. EPIC; ACT-R/PM, [6]). Although these approaches have been shown to be quite successful in modeling human performance in a number of specific tasks, they are still too limited to explain more general phenomena that are relevant in the perception-action domain in cognitive psychology. In this paper, we first examine some existing cognitive architectures and discuss their characteristics with respect to a number of challenging findings from cognitive psychology. Next, we present and describe the characteristics of our HiTEC model for perception and action [7]. Finally, we discuss its promise as a cognitive architecture for digital human modeling in HCI.

2 Cognitive Architectures

A cognitive architecture can be characterized as a broad theory of human cognition based on a wide selection of human experimental data [1]. Whereas traditional research in cognitive psychology tends to focus on specific theories of a very limited range of phenomena, cognitive architectures are attempts to integrate these theories into computer simulation models. Apart from their potential to compare and contrast various theoretical accounts, cognitive architectures can be useful in creating models for an applied domain like HCI that requires users to employ a wide range of cognitive capabilities, even in very simple tasks. Cognitive architectures define an overall structure and general principles. To model a specific task, certain aspects (e.g., prior knowledge, the task goal) need to be filled in by a cognitive engineer. Only then, 'running' the architecture may result in interactions comparable to human behavior.

The best known cognitive architectures (e.g., SOAR, EPIC, ACT-R) are theoretically based on the Model Human Processor, the seminal work of [8]. According to this theoretical model, the human 'processor' is composed of three main modules: perception, cognition and action. It describes cognitive processing as a cyclic, sequential process from stimulus perception to cognitive problem solving to response execution. Note that this closely resembles the seven stage model [9] often used to explain human behavior in HCI: (1) users perceive the state of the world, (2) users interpret their perception, (3) users form evaluations based on these interpretations, (4) users match these evaluations against their goals, (5) users form an intention to act, (6) users translate this intention into a sequence of actions and (7) users execute this action sequence. Executing an action sequence subsequently results in a change in the world state which can again be perceived in stage 1.

Traditionally, cognitive architectures are developed to model the middle, cognitive steps of this sequence. It is assumed that the first steps, perceiving and interpreting the world state, are performed relatively easily.. The main focus is on comparing the world state with a goal state and deciding upon which action to take next in order to achieve the goal state. It is further assumed that once an action is chosen, its execution is easy, leading to a predictable new world state.

The core mechanism used by most architectures is a production rule system. A production rule defines the translation of a pre-condition into an action that is known to produce a desired post-condition. This can be interpreted as “IF (x) THEN (y)” rules. By specifying a set of production rules, a cognitive architecture can be given some prior knowledge resulting in tendencies to choose those actions that eventually realize certain goals.

When putting a cognitive architecture, endowed with a set of production rules, in interaction with an environment, conflicts between rules or unexpected conditions may present themselves. Some cognitive architectures have means to cope with these situations. For example, SOAR has a learning mechanism that can learn new production rules [1].

By assuming a set of production rules, a cognitive architecture also assumes a set of action alternatives. However, when a user is interacting with a physical or virtual environment, it is often unclear which actions can be performed. In certain contexts, users may not readily detect all action opportunities and action alternatives may differ in their availability, leading to variance in behavior. [10]. This is hard to capture in a cognitive architecture that assumes a predefined set of (re)actions.

With the increased interest in the perceptual and action-related aspects of human computer interaction, some of the architectures have been extended with perceptual and motor modules that allow for the modeling of aspects related to ‘early’ and ‘late’ stages of the interaction cycle as well. For example, EPIC contains not only production rules which define the behavior of the cognitive processor, but also some perceptual-motor parameters which define the time courses of (simulated) perceptual information processing and (simulated) motor action [4]. Importantly, perceptual processing is modeled as a computation of ‘additional waiting time’ before the production rules can be applied. By defining certain parameters in the model, this waiting time can, for instance, vary for different modalities. Similarly, EPIC does not simulate actual motor movement, but computes the time it would take for a particular motor output to be produced after the cognitive processor has sent the action instruction. This time course depends on specified motor features and the current state of the motor processor (i.e.., the last movement it has prepared). ACT-R/PM is a recent extension of the ACT-R cognitive architecture. It allows a modeler to include time estimates of perceptual and motor processes in a similar fashion as EPIC [1].

In sum, cognitive architectures typically maintain a perception-cognition-action flow of information, where the focus of the modeling effort is primarily on cognitive (i.e., problem solving) aspects. New extensions of some of the leading architectures allow modelers to include perceptual and motor aspects, but this is typically limited to approximations of the time needed to perceive certain features and to produce certain movements.

3 Cognitive Psychology

Existing cognitive architectures are generally based on findings in cognitive psychology. They are mainly inspired by studies on problem-solving and decision-making. However, (recent) findings in the perception-action domain of cognitive psychology may shed some new light on the assumptions of existing cognitive architectures. In the following we discuss a number of these effects. Subsequently, we describe the Theory of Event Coding that aims at integrating these effects into a single meta-theoretical framework. This is the main theoretical basis of the HiTEC architecture that will be described in the next section.

Stimulus Response Compatibility. When studying the design of computer interfaces, Simon [11] accidentally discovered that spatial responses (e.g. pressing a left or right key) to non-spatial stimulus features (e.g., color or shape) are faster if the stimulus location corresponds to the response location. This effect has come to be known as the Simon effect. It suggests that while the only specified task 'rules' are "IF red THEN left" and "IF green THEN right", the non-specified 'rules' "IF left THEN left" and "IF right THEN right" are apparently active as well. It is clear that a cognitive architecture that incorporates perceptual and action related processes needs to explain this type of automatic stimulus-response interaction in a natural way.

Action influences Perception. In recent experiments, it has been shown [12] that if people prepare a manual grasping or reaching action, they detect and discriminate target stimuli in an unrelated interleaved task faster if these targets are defined on feature dimensions that are relevant for the planned action (e.g., shape and size for grasping, color and contrast for reaching). This finding suggests that action planning can influence object perception. It challenges the traditional view of a strictly sequential flow of information from perceptual stages to stages of action planning and execution.

Learning Action Alternatives. Various studies, including research on infants, show that people are capable of learning the perceptual effects of actions and subsequently use this knowledge to select an action in order to achieve these effects [13]. In this way, initially arbitrary actions may become the very building blocks of goal directed action. This principle could introduce a more grounded notion of goal-directedness in a cognitive architecture than merely responding with a set of reactions.

3.1 Theory of Event Coding

To account for various types of interaction between perception and action, Hommel, Müsseler, Aschersleben, and Prinz [14] formulated the Theory of Event Coding (TEC). In this meta-theoretical framework they proposed a level of common representations, where stimulus features and action features are coded by means of the same representational structures: 'feature codes'. Feature codes refer to distal features of objects and events in the environment, such as distance, size and location, but on a remote, descriptive level, as opposed to the proximal features that are registered by the senses. Second, at this common codes level, stimulus perception and action planning are considered to be similar processes; both involve activating and integrating feature codes into complex structures called 'event files'. Third, action features refer to the perceptual consequences of a motor action; when an action is executed, its perceptual effects are integrated into an event file, an action concept. Following the Ideomotor theory [15], one can plan an action by anticipating the features belonging to this action concept. As a result, actions can be planned voluntarily by intending their perceptual effects. Finally, TEC stresses the role of task context in stimulus and response coding. In particular, feature codes are “intentionally weighted” according to the action goal at hand. In order to computationally specify the mechanisms proposed in TEC and validate its principles and assumptions by means of simulations, we are developing the HiTEC architecture [7]. HiTEC is a generic architecture that can be used to define more specific computational models of human perception and action control and that can serve as a starting point for a cognitive architecture in digital human modeling for HCI. In the following, we describe the HiTEC architecture in terms of its structures and processes and discuss how the architecture incorporates the above mentioned psychological effects.

4 HiTEC

The Theory of Event Coding provides a number of constraints on the structure and processes of the HiTEC cognitive architecture. First, we describe the general structure of HiTEC and its representations. Next, we describe the processes operating on these representations, following the two-stage model for the acquisition of voluntary action control [16].

4.1 HiTEC’s Structure and Representations

HiTEC is architected as a connectionist network model that uses the basic building blocks of parallel distributed processing (PDP, [17]). In a PDP network model processing occurs through the interactions of a large number of interconnected elements called units or nodes. During each update cycle, activation propagates gradually through the nodes. In addition, connections between nodes may be strengthened or weakened reflecting long term associations between nodes.

In HiTEC, the elementary nodes are codes which can become associated. As illustrated in Fig. 1, codes are organized into three main systems: the sensory system, the motor system and the common coding system. Each system will now be discussed in more detail.

Sensory System. The human brain encodes perceived objects in a distributed fashion: different features are processed and represented by different brain areas. In HiTEC, different perceptual modalities (e.g., visual, auditory, tactile, proprioceptive) and different dimensions within each modality (e.g., visual color and shape, auditory location and pitch) are processed and represented in different sensory maps. Each sensory map is a module containing a number of sensory codes that are responsive to specific sensory features (e.g., a specific color or a specific pitch). Note that Fig. 1 shows only a subset of sensory maps. Models based on the HiTEC architecture may include other sensory maps as well.

Motor System. The motor system contains motor codes, referring to proximal aspects of specific movements (e.g., right index finger press, left hand power grasp et cetera). Although motor codes could also be organized in multiple maps, in the present version of HiTEC we consider only one basic motor map with a rudimentary set of motor codes.

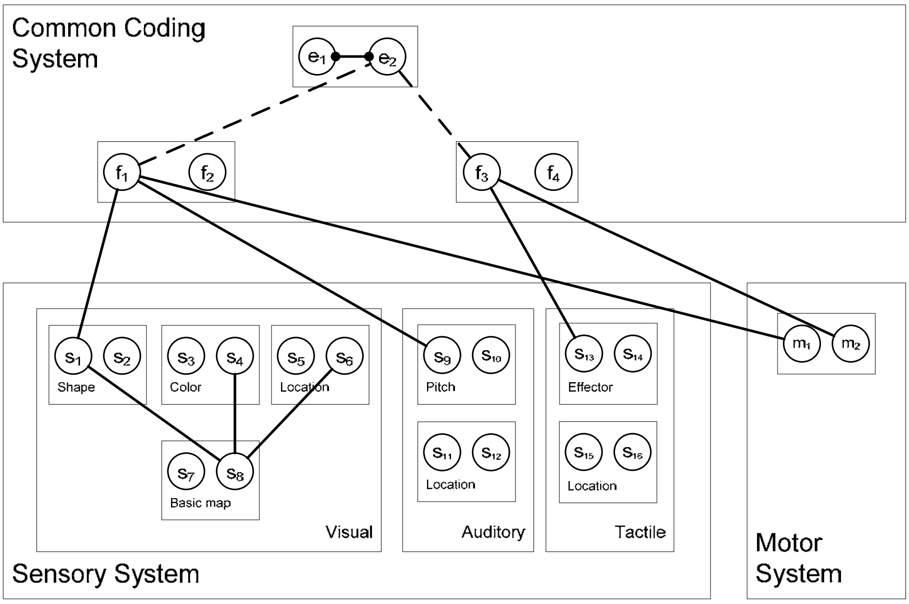

Fig. 1. HiTEC Architecture

Common Coding System. According to TEC, both perceived events and events that are generated by action are coded in one common representational domain [14]. In HiTEC, this is implemented as a common coding system that contains common *feature codes*. Feature codes refer to distal features of objects (e.g., global location in scene, overall object color, size, et cetera) as opposed to the proximal features coded by the sensory codes and motor codes. Feature codes may be associated to both sensory codes and motor. They can combine information from different modalities and are in principle unlimited in number. TEC assumes that feature codes are not fixed, but that they emerge by extracting regularities from sensorimotor experiences. For example, as a result of frequently using the left hand to grasp an object perceived in the left visual field, the distal feature code 'left' may emerge which both codes for left-hand actions and for objects perceived in the left visual field. As a result, feature codes gradually evolve and change over time.

Associations. In HiTEC, codes can become associated, both for short term and for long term. Short term associations between feature codes reflect that these codes 'belong together in the current task or context' and that their binding is actively maintained in working memory. In Fig. 1, these temporary bindings are depicted as dashed lines. Long term associations can be interpreted as learned connections reflecting prior experience. These associations are depicted as solid lines in Fig. 1.

Event Files. Another central concept of TEC is the *event file* [18]. In HiTEC, the event file is modeled as a structure that temporarily associates to feature codes that 'belong together in the current context' in working memory. An event file serves both the perception of a stimulus as well as the planning of an action. When multiple events are present in working memory, choosing between these events (e.g., deciding between different action alternatives) is reflected by competition between the associated event files. This competition is computationally modeled by means of negative associations between event files, depicted as solid lines with filled disk ends in Fig. 1.

4.2 HiTEC's Processes

Following Elsner and Hommel [16] two-stage model of acquisition of voluntary action we now describe the HiTEC processes that enable the learning of action alternatives. Next, we discuss how HiTEC allows for action and task mediated perception as well as stimulus response compatibility.

Stage 1: Acquiring Action – Effect Associations. Feature codes are perceptually grounded representations since they are derived by abstracting regularities in activations of sensory codes. Associations between feature codes and motor codes reflect acquired knowledge of action-effect contingencies: motor codes m_i are activated, either because of some already existing action-effect associations or simply randomly (e.g., an infant trying out some buttons on an interactive toy). This leads to a change in the environment (e.g., pressing a button produces a sound) which is registered by sensory codes s_i. Activation propagates from sensory codes towards feature codes f_i. Eventually, these feature codes are integrated into an event file e_i which acts as an action concept.

Subsequently, the cognitive system learns associations between the feature codes f_i belonging to this action concept and the motor code m_i that just led to the executed motor action. The weights of these associations depend on activation of the motor code and the feature code. Crucially, this allows the task context to influence the learning of action effects, by moderating the activation of certain feature codes. Due to this top-down moderation, task-relevant features (e..g, button look and feel) are weighted more strongly than task-irrelevant features (e.g., lighting conditions in the room). Nonetheless, this does not exclude task-irrelevant but very salient action effects to become involved in strong associations as well.

Stage 2: Using Action – Effect Associations. Once associations between motor codes and feature codes exist, they can be used to select and plan voluntary actions. By anticipating desired action effects, feature codes become active. Now, by integrating the feature codes into an action concept, the system can treat the features as constituting a desired state and propagate their activation towards associated motor codes. Initially, multiple motor codes m_i may become active as feature codes typically connect to multiple motor codes.

However, some motor codes will have more associated features that are also part of the active action concept and some of the $m_i – f_i$ associations may be stronger than others. Therefore, in time, the network will converge towards a state where only one code m_i is strongly activated, which will lead to the selection of that motor action. In addition to the mere selection of a motor action, feature codes also form the actual action plan that specifies (in distal terms) how the action should be executed [e.g., global button location]. This action plan is kept active in working memory, allowing the system to monitor, evaluate and adjust the actual motor action. Crucially, action alternatives can be learned and selected in terms of their perceptual effects.

Task Preparation. In human computer interaction, users may have tendencies to respond differently to different stimulus elements. To model this, different event files are created and maintained for the various options [e.g., choosing among buttons that produce different sounds]. Due to the negative links between these event files, they will compete with each other during the task.

Perception and Action. When the environment is perceived, sensory features will activate a set of feature codes. Activation propagates towards one or more event files (that were formed during task preparation). Competition takes place between these event files. Simultaneously, activation propagates from event files to action effect features and motor codes, resulting in the execution and control of motor action. Note that task preparation already sensitizes feature codes both for the to-be-perceived stimuli and for the to-be-planned responses. Therefore, the cognitive system is biased in perceiving elements in the environments and anticipating responses in terms of these feature codes. As the common feature codes are used for both perception and action, perceptual coding can influence action coding and vice versa.

Stimulus Response Compatibility. When feature codes for perceived elements and anticipated responses overlap, stimulus-response compatibility effects can arise: if a stimulus element activates a feature code (e.g., picture of an animal) that is also part of the event file representing/of the correct response (e.g., the sound of that animal), this response is activated more quickly, yielding faster reactions. If, on the other hand, the feature code activated by the stimulus element is part of the incorrect response, this increases the competition between the event files representing the correct and incorrect response, resulting in slower reactions.

5 Discussion

We discussed existing cognitive architectures and highlighted their limitations with respect to a number of psychological findings highly relevant to HCI. We subsequently described our HiTEC architecture and discussed how these findings could be explained from HiTEC's basic structures and processes.

Like other cognitive models, HiTEC also consists of perception, motor and cognitive modules. However, in contrast to the sequential architecture of existing models, the modules in HiTEC are highly interactive and operate in parallel. Perception of a stimulus does not need to be completed before an action plan is formed (as suggested by linear stage models). Furthermore, the cognitive module contains common codes that are used for encoding perceived stimulus as well as for anticipated actions. Actions are represented as motor programs in the motor module, but they are connected to their (learned) perceptual action effects (e.g., a resulting visual effect or a haptic sensation of a key press) as proposed by Ideomotor theory [15]. The way in which tasks are encoded in HiTEC (by using competing event files) shows similarities to a system where multiple production rules compete for 'firing'.

However, it is important to note that action alternatives are selected on the basis of their distal feature effects, rather than on the basis of their proximal, motor characteristics.

We acknowledge that HiTEC, in its current incarnation, is not yet capable of the rich body of simulations that other cognitive architectures have demonstrated. Indeed, we do not exclude a production rule component in future versions, but we emphasize that the core of HiTEC consists of perception-action bindings.

The strengths of HiTEC lay primarily in its ability to learn perceptual action effects in a principled way and using these effects for action selection and control. This naturally results in a mechanism that enables stimulus response translation on an abstract level, thereby allowing the system to generalize over different but similar perceptual features, both in object perception and in action planning. This leniency may avoid creating a 'production rule' for each and every minute variant that the system may encounter. This may increase the system's robustness against variability in perception and action.

Of course, expending our simulations to environments and tasks that currently can be simulated by other architectures, requires more intricate techniques to actually learn the sensorimotor contingencies (i.e., feature codes) that we now assume. Also, we now discussed a situation where a simple set of task rules was predefined. Further research is necessary to assess the role of long term memory and motivational influences in this respect.

With the rise of new HCI environments, such as various mobile devices, virtual reality and augmented reality, HCI within these virtual environments increasingly resembles interaction in the physical world. This trend stresses the importance of studying the implications of findings in the perception-action domain for the field of HCI.

Acknowledgments. Support for this research by the European Commission (PACO-PLUS, IST-FP6-IP-027657) is gratefully acknowledged.

References

1. Byrne, M.D.: Cognitive architecture. In: Jacko, J.A., Sears, A. (eds.) Human-Computer Interaction Handbook, pp. 97–117. Erlbaum, Mahwah (2008)
2. Anderson, J.R.: Rules of the mind. Erlbaum, Hillsdale (1993)
3. Newell, A.: Unified theories of cognition. Harvard University Press, Cambridge (1990)
4. Kieras, D.E., Meyer, D.E.: An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. Human-Computer Interaction 12, 391–438 (1997)
5. Welsh, T.N., Chua, R., Weeks, D.J., Goodman, D.: Perceptual-Motor Interaction: Some Implications for HCI. In: Jacko, J.A., Sears, A. (eds.) The Human-Computer interaction Handbook: Fundamentals, Evolving Techniques, and Emerging Applications, pp. 27–41. Erlbaum, Mahwah (2008)
6. Byrne, M.D., Anderson, J.R.: Perception and action. In: Anderson, J.R., Lebiere, C. (eds.) The atomic components of thought, pp. 167–200. Erlbaum, Hillsdale (1998)
7. Haazebroek, P., Hommel, B.: HiTEC: A computational model of the interaction between perception and action (submitted)

8. Card, S.K., Moran, T.P., Newell, A.: The psychology of human-computer interaction. Erlbaum, Hillsdale (1983)
9. Norman, D.A.: The Design of Everyday Things. Basic Book, New York (1988)
10. Kirlik, A.: Conceptual and Technical Issues in Extending Computational Cognitive Modeling to Aviation. In: Proceedings of Human-Computer Interaction International, pp. 872–881 (2007)
11. Simon, J., Rudell, A.: Auditory s-r compatibility: The effect of an irrelevant cue on information processing. Journal of Applied Psychology 51, 300–304 (1967)
12. Fagioli, S., Hommel, B., Schubotz, R.I.: Intentional control of attention: Action planning primes action-related stimulus dimensions. Psychological Research 71, 22–29 (2007)
13. Hommel, B., Elsner, B.: Acquisition, representation, and control of action. In: Morsella, E., Bargh, J.A., Gollwitzer, P.M. (eds.) Oxford handbook of human action, pp. 371–398. Oxford University Press, New York (2009)
14. Hommel, B., Muesseler, J., Aschersleben, G., Prinz, W.: The theory of event coding (TEC): A framework for perception and action planning. Behavioral and Brain Sciences 24, 849–937 (2001)
15. James, W.: The principles of psychology. Dover Publications, New York (1890)
16. Elsner, B., Hommel, B.: Effect anticipation and action control. Journal of Experimental Psychology: Human Perception and Performance 27, 229–240 (2001)
17. Rumelhart, D.E., Hinton, G.E., McClelland, J.L.: A general framework for parallel distributed processing. In: Rumelhart, D.E., McClelland, J.L., The PDP Research Group (eds.) Parallel distributed processing: Explorations in the microstructure of cognition. MIT Press, Cambridge (1986)
18. Hommel, B.: Event files: Feature binding in and across perception and action. Trends in Cognitive Sciences 8, 494–500 (2004)