ESRC Centre for Research in
Development, Instruction and Training

DEPARTMENT OF PSYCHOLOGY, UNIVERSITY OF NOTTINGHAM
UNIVERSITY PARK, NOTTINGHAM, NG7 2RD, U.K.

Designing Abstract Visual Perceptual and Motor Action Capabilities for Use by Cognitive Models

Gordon. D. Baxter and Frank E. Ritter

Technical Report No. 36

October 1996

Email: gdb@psychology.nottingham.ac.uk

Phone +44 (0)115 9515151 ext. 8348; Fax +44 (0)115 9515324

Designing Abstract Visual Perceptual and Motor Action Capabilities for Use by Cognitive Models

Gordon D. Baxter and Frank E. Ritter

ESRC Centre for Research in Development, Instruction and Training
Department of Psychology
University of Nottingham
Nottingham, UK NG7 2RD

gdb@psychology.nottingham.ac.uk

Technical Report No. 36

Abstract

Cognitive models typically fail to interact with the external environment when performing a task. Where models have incorporated interaction this has not generally been implemented as a psychologically plausible mechanism. In the real world, where many tasks are now computer based, task context influences performance in all but the most trivial of tasks. Visual perception and motor action are therefore central to task performance, and hence to the generation of erroneous actions. The failure to take account of the constraints imposed by having to interact with the environment helps to explain why cognitive models perform tasks (and learn) too quickly.

We identify here a set of requirements to constrain the design of simulations of visual perception and motor action that form extensions to a cognitive architecture. These simulations operate at a symbolic level of abstraction such that the model can internally represent each object in the environment using a symbol. We also include a set of facilities to support debugging of the interaction mechanism, based on our experiences from earlier attempts at implementing such a capability.

Acknowledgements

Support for this work has been provided by the DRA, contract number 2024/004, and by the ESRC Centre for Research in Development, Instruction, and Training. The views in this paper are those of the authors, and should not necessarily be attributed to the Ministry of Defence.

Table of Contents

1. The Need for Interaction in Cognitive Models

We are fundamentally interested in how people perform complex dynamic tasks. These are tasks where people are required to interact with their immediate surroundings, typically in the shape of a computer display, in order to control some larger system. To facilitate control of the system the user employs an internal model of the state of the task environment. This internal representation is generated using what is perceived from the computer screen, together with any stored information that may be appropriate. Control is maintained by mentally manipulating this representation to generate suggested changes to the system state, which are then physically performed using the computer interface.

The general idea of people constructing an internal model is not new and dates back at least as far as Craik (1943). The key aspect that is often overlooked in cognitive modelling, however, is that the internal model of the environment is developed on the basis of what people perceive around them. Here, perception is defined to be the interpretation of the signals received by the senses. The limitations of the senses thus constrain the content of the representation, rather than all of the data from the environment being readily available without having to search the environment.

The importance of interaction–perception and action–is made apparent when one considers the sequence of steps involved in task performance (Kelley, 1968, p. 10):

1. The course of events is perceived.
2. Two or more possible future states are conceived of, as are the events that would lead to them.
3. One of these future states is chosen as a goal.
4. Bodily movements are employed to initiate a train of events leading to the goal.
5. The train of events initiated by the bodily movements (which in some cases is monitored and modified while in progress) brings about the goal.

Even though Craik's (and Kelley's) ideas pre-date much of cognitive modelling, many cognitive models have, at best, only incorporated limited interaction with their external environment. Models are more often created with superhuman capabilities providing a direct and complete correspondence between the relevant parts of the environment and the model's internal representation of that environment. Thus, when the model "observes" the external world, it sees all of the appropriate features at once. Similarly, when the model wishes to effect a change to the environment, this is only done to the model's representation of the environment which is held in working memory. The change therefore happens instantly and is immediately visible to the model.

Complex dynamic tasks all require the use of interaction and external memory. Models that only use an internal representation of the task will exhibit an artificially high level of performance. Performance will be faster than it should be because the model does not have to search the environment for information–it is all stored in memory. Similarly, performance will also be more accurate than it should be, because the model is not subject to the misinterpretations and ambiguities that can result from visual perception, and the slips of action that people often make.

The influence of interaction on task performance can be illustrated by considering models that solve the Tower of Hanoi puzzle. Typically these models keep a complete representation of the puzzle in working memory. The puzzle is solved by simply manipulating this representation, rather than having to manually move the disks on the real puzzle. Although in this case the interaction is relatively simple, there is a difference between the purely mental task, and its physical equivalent. It is simpler in some ways because the model does not have to physically move the disks. In other ways, however, it is more complicated because the locations of the disks on the pegs have to be stored in memory, rather than simply using the real pegs and disks as a form of external memory.

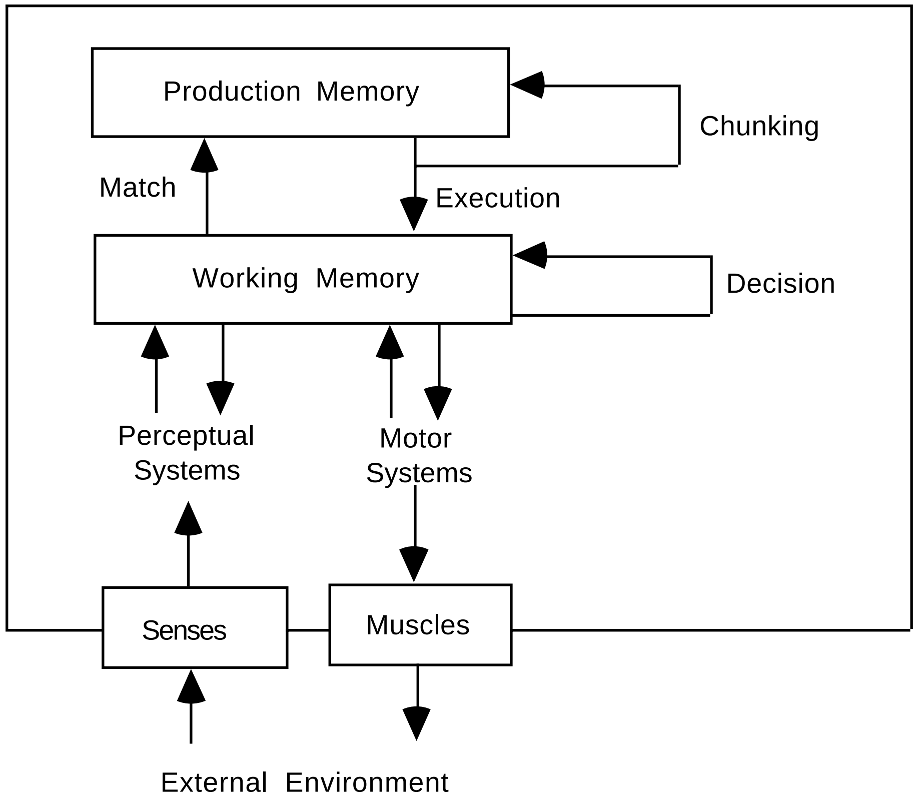

Figure 1. Overview of the Soar Cognitive Architecture (adapted from Newell, Rosenbloom & Laird, 1989).

The role of interaction is tacitly acknowledged in the specifications of the cognitive architectures that are employed in the development of cognitive models. In Soar (Newell, 1990), for example, a full cognitive model includes interaction with the outside environment, as shown in Figure 1. Similar specifications can be found for other cognitive architectures such as ACT-R (Anderson, 1993), which has already started to incorporate the role of visual attention, for example (Anderson, Matessa, & Douglass, 1995). Cognitive models without a strong commitment to a specific architecture are also starting to include models of visual, and auditory senses (e.g., Tabachneck, Leonardo, & Simon, 1995).

This document addresses the need to extend cognitive models in general to incorporate mechanisms to support interaction with complex dynamic tasks. The design of these mechanisms–visual perception and motor action–is based on an earlier attempt which involved getting a Soar model to interact with a simulation of a simple ATC-like task (Bass, Baxter, & Ritter, 1995), developed using Garnet (Myers, Giuse, Dannenberg, Vander Zanden, Kosbie, Pervin, et al., 1990). The psychological plausibility of the mechanisms is maintained by taking account of results from the literature on visual perception and motor action. The design requirements are explicitly listed at the end of each of the appropriate sections of the document, and are captured together in a list in Appendix A.

One of the goals of this work is to demonstrate that the proposed solution to the problem of interaction is a feasible one. In order to do this, the design is to be implemented using the SL-GMS toolkit (Sherrill-Lubinski Corporation, 1994), to provide a way for a Soar model of an electronic warfare task to communciate with the OOPSDG simulation (Ramsay, 1995b). The interface to the simulation is also implemented using SL-GMS.

Although our approach is more detailed than earlier efforts in this area, which either used a completely abstract level of perception or ignored it altogether, it is not as intricate as real perception. In the long run, however, the model may require senses that operate at a less abstract level, and are independent of the implementation of the interface. Models will need to incorporate the ability to misinterpret the incoming signals from perception, for example, if they are to perform the same erroneous actions as humans. In the extreme, this would involve adding a robot arm and a camera-based eye to the model in order to simulate the interaction capabilities at a fine-grained level of detail.

In Section 2 we identify and describe a simple set of interaction mechanisms for cognitive models. Sections 3 and 4 describe the input and output components in more detail before consideration is given to the way that they interact in Section 5. The requirements for the simulated senses are clarified by illustrating how the senses may be used by the model, where appropriate. Section 6 describes the support facilities required for helping to develop and debug the model and its simulated senses.

2. Overview of the Interaction Mechanisms

If cognitive models are to perform computer-based tasks in the same way as real users, they need some way of getting information about the task from the computer. When this information has been suitably processed, they will need some way of performing actions to realise the task. Taken together, these requirements illustrate the need for visual perception and motor action. In order to debug these interaction mechanisms, and to facilitate an understanding of how the model is using the computer interface, support facilities are also required. This section provides a brief overview of each of the desired capabilities, which are described in more detail in later sections.

2.1 Visual perception

The model will need a visual perceptual capability to perceive what is on the display screen. We believe the appropriate level of information to deliver to the model is a symbolic representation of the objects that appear in the interface. This representation approximates what the human perceptual system delivers to cognition once the input stimuli from the retina have been processed by the visual cortex.

A possibly useful way to think about this (or to implement it) is to imagine that the interaction mechanisms effectively form an overlay on the display screen, whilst any changes that are made to the information shown on the user's display, in the form of pop-up menus and so forth, occur below the surface of the perceptual display and can be noticed. (Note, however, that whilst the interaction mechanisms may conceptually reside above the display surface, the implementation of these mechanisms is intrinsically linked to the implementation of the interface.)

2.2 Motor action

Some form of motor action is necessary to interact with a computer system in the same way as a user. Models will need to select objects, pull down menus, type in data values and so on. Again, the capability is relatively abstract and can, in some sense, be regarded as analogous to the visual perceptual mechanism of the model: the motor capability represents the results of programming and performing motor actions rather than the details of the muscle actions themselves.

2.3 Debug support facilities

People modelling task performance will need support facilities to allow investigation and interrogation of the interaction capabilities provided to the model. They will need to know what the model knows, where it is looking, and similar interaction state information. There may also be a need to directly drive the model's interaction mechanisms, especially during testing.

It is important that the support facilities be kept separate from what the model can see or use (unless that is what is the model's environment). The support facilities should not be visible to the model's senses because they do not constitute part of the interface to the task simulation being investigated.

2.4 Overview of implementation

The interaction mechanisms described here are implemented at a level of abstraction that can be supported by the SL-GMS programming toolkit that is also used to implement the interface to the task simulation.

An outline of how the model and task simulation interact is shown in Figure 2. The cognitive part of the model (developed using Soar) decides upon an action to take based upon its current knowledge. If this involves doing something with the simulation (perception or action) it sends out a request via MONGSU for the desired action to be performed. This is translated by the interface between MONGSU and the SL-GMS simulation into a call to an appropriate function. This function call will almost invariably be handled by the interface to the simulation (implemented using SL-GMS), and the results returned to the model. The action will often also be reflected in the interface in a way that is visible to the analyst: either the simulated eye or the simulated mouse pointer will be moved on the display.

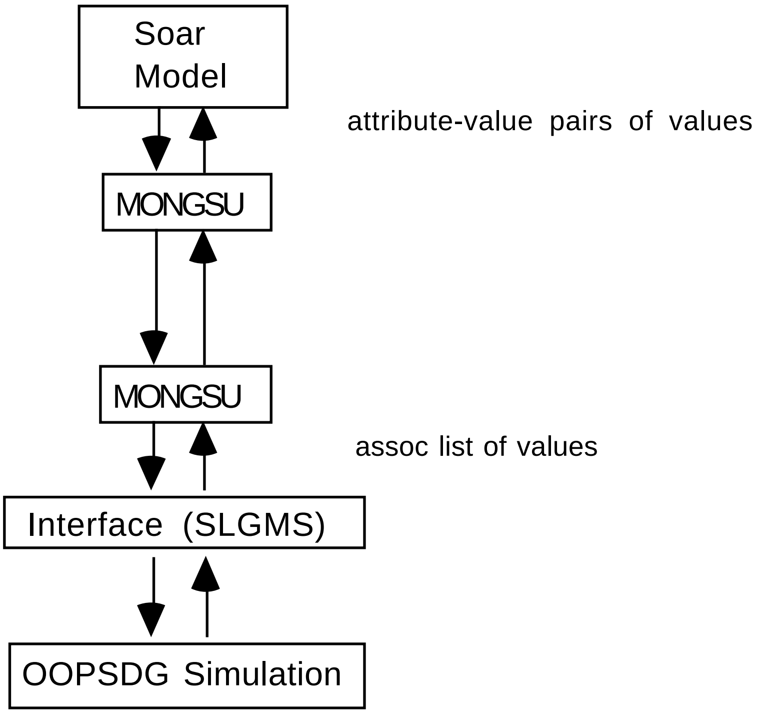

Figure 2. Outline of communication between model and task interface.

In order to simplify the implementation while maintaining the maximum ability to amend or extend the model, all measurements are computed in pixels in this design. Many behaviours are modified by variables, such as maximum eye speed, and these should be read out of an initialisation file. We believe the use of variables in terms of pixels provides the most general way for modifications and additions to be computed and passed in to the interaction model without modifying the interaction model itself. It provides the analyst with the ability to move the model user's head closer to the screen by modifying the eye's size (bigger) and relative speed across the screen (slower).

3. Visual Perceptual Capabilities

The fundamental requirement of the visual perceptual capability is that it should be approximately functionally equivalent[1] to that possessed by humans. In humans the visual perceptual system (including the visual cortex) delivers to central cognition a single image of what is seen by the eyes. The simulated perceptual capability needs to work in a similar way, although there are some fundamental differences. One such difference is the reduced importance of the need for depth information (for three dimensional objects): the model described here is working with an essentially flat plane–the display screen—hence perception in two dimensions is sufficient. Another difference is because the objects to be recognised, that is, the interface objects, are known, low level aspects of vision, such as feature detection, do not have to be included.

There are basically two dimensions that need to be considered when implementing the perceptual capability of the eye. Firstly, the physiological aspects of the eye that influence what is seen need to be mimicked, and secondly, the behavioural aspects of the eye that influence how objects are seen need to be modelled. The level of abstraction of the visual perceptual capability is allowed to be relatively high, ignoring a lot of lower level empirical vision data (see Boff & Lincoln, 1988, for a range of examples). Even though this makes the model relatively crude, it still provides a sufficient level of detail to investigate human-like task performance, particularly interaction with the task simulation.

3.1. Physiology of the eye

For our purposes we are essentially dealing with a symbolic representation of perception that is at a level of abstraction significantly higher than that of the physiology of the retinal cells. There are, however, three regions on the retina which need to be considered, since they differ in the degree of acuity that they provide. These parts are: the fovea; the parafoveal region; and the periphery. These are not arbitrary parts, but are based on the distribution of cells on the retina of a real eye and how and what information is provided by the eye.

3.1.1. Fovea

The fovea is the part of the retina where the greatest visual acuity occurs. Roughly 2° of visual arc are projected onto the fovea; complete visual arc is approximately 208° horizontally and 120° vertically. The central part of the horizontal visual arc is viewed using binocular vision; the more extreme parts are viewed using monocular vision, since the view from the other eye is blocked by the nose. As a heuristic, the area projected onto the fovea is approximately the size of one's thumb nail when viewed at arm's length. It is also not quite circular, being slightly wider than it is tall. Since the fovea is the area of greatest visual acuity, all the basic details of the objects in the fovea need to be represented, that is, object type and attributes (colour, location and so on). In reality, the categorisation of the object as a particular type is part of cognition (object recognition), rather than part of perception.

The objects that can be identified by the fovea should correspond to the interface widget level. These objects are the sort of things that you would expect to see referenced in verbal protocols of users performing the task. So, for example, radio buttons would be identified as an object, rather than as a number of individual sub-components. In general, object recognition should be relatively straightforward, since most widgets are of a sufficiently small size that the whole of their image can be projected onto the fovea.

There are, however, a number of cases where this is not the case. In an ATC task interface, for example, range rings form part of the display screen, and each of these occupies an area of the

[1] By functionally equivalent we mean that on a black-box level the two should be indistinguishable. So, for example, presenting a radio button widget to the model's perceptual capability should cause the generation of a corresponding representation (symbol) that is made available to cognition, just as it would in real perception once the image has been processed by the visual cortex.

screen greater than that which gets projected onto the fovea. The problem of recognition of such large structures is addressed in the Behavioural Aspects section of this paper.

- A fovea displayed and internally represented with its size configurable at start-up or dynamically via a command from the analyst or from the model.

3.1.2. Parafovea

The parafovea is the part of the retina immediately surrounding the fovea, and extends approximately 5° beyond the fovea. In this region, visual acuity starts to diminish, so the level of available information about objects in this area should also be reduced. So, for example, shape and location information needs to be available, but possibly not much more.

There is some evidence suggesting that the parafovea may be dynamic in size. When the complexity of the stimulus currently projected onto the fovea is increased, the size of the parafovea is correspondingly reduced, because there are less attentional resources available for dealing with what is currently located in the parafoveal region. For the purposes of the initial model, this feature will not be required, although it may be desirable in the future. The idea of a dynamic parafovea, and the role of visual attention are discussed under the section on saccades.

- A parafovea that provides more limited information than the fovea.

3.1.3. Periphery

The periphery constitutes the remainder of the visual field. Information about objects located in the periphery is very limited; this is a corollary of the relative scarcity of cells on the retina as the distance away from the fovea increases. Typically, location information will be available together with an indication as to whether the target is moving or not. More detailed information can only be found by making saccadic movements to the object to bring its image onto the fovea, so that it can be processed in detail.

One other piece of information is also held for objects in peripheral vision, to record when an object has been looked at. In order to achieve this effect a unique id can be used to represent each object. Thus, when each object moves it is known that the object has been seen before, and is not a new object.

- Objects not in the periphery get reported with location, an indication as to whether they are moving or not, and a unique identifier.

3.2. Behavioural aspects

There are a number of fundamental behavioural aspects of perception that the model's visual perceptual mechanism is required to possess. These include:

- saccadic eye movements
- pursuit movements or tracking
- fixations (and gaze)

Although there are some other behavioural aspects that can be considered, such as microsaccades (which serve to keep the image from fading on the retina) these are currently beyond the scope of this work. The remainder of this section assumes a basic grasp of the role of visual perception, and the way that this role can be fulfilled (a short review is provided in Appendix B).

The behaviour of the visual perceptual capability will in general be controlled by the cognitive model. The implementation of visual perception needs to provide the appropriate facilities to support this behaviour. So, for example, if the model decides that the eye should move to a new location, it sends a “move-eye” message to the visual perceptual mechanism, together with the parameters that define the target for that movement. The visual perceptual mechanism will translate this message into a call to SL-GMS which will then cause the desired movement to be performed.

3.2.1. Saccadic eye movements

Saccades are approximately ballistic movements of the eyes that take place when the eye moves from looking at one place to looking at another place. The peak velocity of a saccade is about 800° per second, and typically—around 85% of cases—moves the eye through less than 15° of visual angle. The duration of a saccade is generally in the range 20-100 ms. This duration is dependent on the size of the saccadic movement: for saccades larger than 5° in amplitude, the duration is approximately 20-30 ms plus about 2 ms for every degree of amplitude. For small saccades, the duration is approximately constant, since it is dominated by the mechanical properties of the eye. For saccades larger than about 20° a corrective second saccade occurs at about 200 ms after the start of the initial saccade; although this can be ignored for the current implementation, it may be required in later versions, since saccades in humans typically fall short of their target by an amount approximately linear with respect to the size of the original saccade.

When the target of the saccade lies outside of the fovea, there is a delay in the range of 150-250 ms between the appearance of that target and the start of the saccadic movement. The main reason for the delay is the complexity of the calculation needed to translate the retinal distances involved in the saccade into eye movements. As the amplitude of the saccade starts to get very large, the latency of the saccade increases. So, for example, although the latency for movements of 10° and 20° may be approximately the same, the increase in latency at 40° is about 40 ms. This time does not increase as the number of targets increases beyond two, and may actually be reduced if the target’s location and the time at which it will appear are known in advance.

For our purposes we are assuming that the amplitude of eye movements needed to perceive all of the display screen is relatively small. Consequently, all of the display can be viewed without including the complication of head movements, which are involved when the angle between the location currently being fixated by the eye and the target location is sufficiently large. It also means that we avoid the problems of non-linearity that seem to arise when the eye movements extend beyond about 20°.

During the saccade, the image that is projected onto the retina is, more or less, smeared out. We believe that the implementation of this effect can occur in the cognitive model.

The visual system has to determine a point to saccade to, and this becomes more complicated as the size of the target object increases. There are two things that need to be taken into account: the first is that the centre of gravity of an object is attractive, as are its edges and corners; the second is that the interpretation of the object can be influential, particularly in the case of more complex objects.

Determining the target of a saccade is a non-trivial problem. We have adopted the attentional resources metaphor as a means of describing how this process can be done. Objects that have their images projected closer to the fovea receive more attention: more resources are dedicated to the processing of that object. Similarly, if the objects happen to be complex, then more attentional resources will be required to process them. A consequence of this increase in resource requirements is that there will be fewer resources available for processing objects whose images project onto the retina at some distance from the fovea. It is therefore unlikely that the more peripheral objects will form the target of a saccade. In the literature there do not appear to be any quantitative measures associated with this phenomenon, however.

There has been much made of the idea of a searchlight of attention, which implies that attentional resources are something like a variable sized searchlight, and anything that the beam of the searchlight falls on will be allocated some attention resources to enable it to be processed in some way. In more recent work, however, this idea has been called into question (see Allport, 1989 for a general review) . There is now growing evidence to suggest that rather than a spreading beam, attention is allocated in discrete packets. The vast majority of the attentional resources are still allocated to processing those objects that are projected onto the fovea. Outside this region, the distribution is much less uniform, and will tend to be based on expectations arising from experience of doing a particular task.

The saccadic movement of the eye can be performed using a number of different co-ordinate systems. The movement must be specified in terms of co-ordinates rather than named objects when the object lies more than about 4° away from the fovea; for our purposes the movement can always be described in terms of co-ordinates. The main reason behind this is the physiological construction of the retina of the eye. The cells in the eye are most densely located around the fovea, and as the distance from the fovea is increased, this density tails off quite sharply. Any object that appears in the periphery can only be identified with any degree of certainty by making saccadic movements to project the image of that object onto the fovea.

For the purposes of the implementation of the visual perceptual mechanism, the co-ordinate system can be based on the co-ordinate system used by the display. Nominally, the top left corner of the screen will be designated as being at co-ordinates (0,0). In addition, the co-ordinate range can be limited to the range of the display screen. In humans, however, a different range is used. So, for example, the eyes can move from the display screen, to the mouse, or the keyboard, and back again—it would not be possible to model this directly in the simulation as it currently stands. Also, the version of the OOPSDG simulation that we are using is limited to a single-head display, that is, only one screen is used. In this way we circumvent the complication of how the visual perceptual mechanism tracks the movement of the mouse between different screens.

When a saccade is made to a moving object, there is an element of prediction involved in the calculation of the size of the saccade. In between the target appearing and the saccade being performed, an adjustment based on the velocity of the moving object has to be incorporated. This adjustment is required to make sure that the saccade lands on the object at its new location, rather than on the location where it was detected as a target for the saccade. The adjustment mechanism is also involved in shifting the gaze from one moving target to another: the size of the required saccade is always of the appropriate magnitude.

There are a few salient features that can be used to help determine the target of a saccade. The first is movement: a fixation reflex occurs when an isolated target is presented to the peripheral part of the retina. The eyes are automatically moved to the target's location. The second feature that plays a role is colour.

In the cases where there does not appear to be any need for a saccade, the visual perceptual mechanism should move into a visual scanning or sampling mode. This type of behaviour has been noted elsewhere among experienced operators (e.g., Moray, 1976). The precise movements will be based on expectations, and will consist of a number of saccadic movements around the displayed data. All that is required of the implementation of the visual perceptual mechanism to do this is the ability to make the saccadic movements; the cognitive part of the model will control such movements. Initially, a simple guiding heuristic will be used to perform scanning/sampling. The search procedure will move the fovea around the outer radar rings to look for targets. In the OOPSDG simulation it is likely that a real operator would scan the outer range rings first, and possibly start at a particular point based on the expected direction of attack. It should be noted that the scanning behaviour of people is not as straightforward as would naively be expected. Experts tend to only scan about half of a scene, even when there are no pressing constraints to suggest that they need to analyse the scene within a time limit.

- The fovea needs to move by a relative amount when told.
- The movements, if they are within the simulation's cycle time, should take realistic amounts of time.
- Vision should return a complete list of items when asked.
- The attributes of what is returned should be comprehensive, and locations may be in screen co-ordinates.

3.2.2. Pursuit Movements or Tracking

Pursuit movements are used whenever the eyes try to track a moving object in order to maintain a stable image of the target on the retina. The velocity of the eye movement depends on the target object being tracked, with an upper limit of around 30° per second. At velocities greater than this, the eyes lag behind. The update rate for displays for information systems of the sort that we are interested in is of the order of once per second, although the system may receive data from the environment at a much higher rate. There are two direct consequences of this, the first is that it is extremely unlikely that the physical upper limit for tracking speed will ever be exceeded (the display screen occupies less than 30° of the visual field); the second is that the objects will actually be tracked in a staccato fashion, since they do not move continuously across the display screen, but rather tend to move in discrete steps. This tracking behaviour may be implemented by small saccades so as to ensure that the tracked object remains within the projection of the fovea.

- The eye must accept a command to lock-on to an object.
- After each saccade, the results of looking are/are not sent to cognition.

3.2.3. Fixations and Gaze

People generally maintain their gaze in one direction for a second or two at most. The length of the fixation dwell, nominally around 400 ms, is influenced by the complexity of the stimulus being fixated. For a complicated object, the fixation dwell will generally be longer than that required for a simple object, like a text label, for example. Overall about 90% of perceptual experience occurs as fixations, where the eyes are approximately stable, and an image of the world is continuously present on the retina (Irwin, 1992) .

Fixations are only made on the image that is projected onto the fovea, that is to say, in the direction that the model is currently looking. Detailed processing of an object therefore requires that the image of that object be projected onto the fovea. In some cases a fixation may be of a relatively short duration, usually because the objects that are present are obviously irrelevant, or contain little or no useful information. Fixation errors are not considered here, so corrective saccades need not be implemented.

4. Motor Action Capabilities

A representation of the user’s hands is required for use by the model. Conceptually there needs to be a mouse and a keyboard for the model to use to interact with the computer system’s interface. The model has to utilise these devices in the same way that a real user would, so it needs to be able to perform actions corresponding to moving the mouse, pressing the mouse button(s), releasing the mouse button(s), clicking, double-clicking, and typing on the keyboard. The requirements are laid out in detail below. The physiological and behavioural aspects of human motor capability are listed, and the capabilities and constraints that are required for the model’s motor action mechanism are identified.

4.1. Physiological aspects

There are a few global constraints that arise from the physiological aspects of hand motions (Park, 1987) that must be kept in mind when designing simulated motor motions The main constraints that need to be considered are:

- there is a physical limit to the speed at which the hand can move,
- the hand has to accelerate and decelerate over the course of a movement,
- smooth continuous curved movements are faster than staccato straight-line movements involving changes of direction,
- movements along the line of direction running between 7:30 and 1:30 on a clock face are faster than movements in other directions.

The most important aspect that needs to be included during the initial implementation is the first constraint. The others will need to be accounted for in future implementations.

- There is a maximum speed for hand movements (30cm per second can be used as an initial guide).

4.2. Behavioural aspects

There are essentially six different kinds of movement available (Park, 1987) :

- Positioning: Consists of a primary gross movement and a secondary corrective movement, usually involving visual feedback. The reaction time is roughly constant, and the movement time is related to (but not proportional to) the distance. The time taken is affected by the direction, as noted above.
- Continuous: Tracking movements. Again the direction influences the time taken.
- Sequential: for example, typing on a QWERTY keyboard.
- Repetitive: the same movement is performed repeatedly. These can either be paced or unpaced.
- Manipulative: intricate manipulating movements, e.g., in watch mending.
- Static reaction: holding a load, which requires muscle co-ordination to prevent the load from dropping.

Of these, the first three are the most important for the simulated motor capability. The need to make movements in a particular direction faster than movements in other directions can initially be ignored, although it may be required in future implementations. The need to provide repetitive movements as a separate type of behaviour will also be significant in some situations. The last two effects on the list can be ignored for the purposes of the current specification.

As with the simulated perceptual sense, the behaviour of the motor action capability will generally be controlled by the cognitive part of the model. The SL-GMS implementation of motor action has to provide the appropriate facilities to support this behaviour. So, for example, if the model decides that the mouse should move to a new location, it sends a “move-mouse” message to the simulation, together with the parameters that define the target for that movement. At the simulation, this may be translated into a call to SL-GMS that causes the simulated mouse pointer to be moved on the display screen.

4.2.1. In-transit Hand Movements

There should be a finite time lag for the hand moving between the mouse and the keyboard. This should nominally be of the order of one second. The length of time should be variable, however, based on the expertise level of the model. The values used for expert users should approximate those reported for the keystroke level model (Card, Moran, & Newell, 1983) .

4.2.2. Mouse Movements

Mouse movements typically occur in two phases, so the model's motor action mechanism must be able to support both of these. In the first phase an approximately ballistic movement takes place, in which the mouse moves in the general direction of the desired target object. The second phase is much more controlled and relies on visual feedback to correctly position the mouse pointer over the desired target object. The two phases occur sequentially, and to all intents and purposes appear to be a single relatively smooth movement. To correctly match the behaviour of a real user, the simulation should incorporate a means for generating erroneous movements, otherwise the need for visually controlled feedback will never arise, since the model will simply follow a straight line trajectory directly to the target.

In the first phase of the movement the mouse typically moves something like 75% of the required distance (Carlton, 1981) . Note that this is almost invariably not the same as moving 75% of the x-distance, and 75% of the y-distance. There is some contention over whether behaviour can be described in such a way, but for our purposes, 75% provides a useful initial guiding heuristic.

For the second phase of movement, the response time will need to be relatively quick, since there has to be some interplay between the simulations of motor action and perception in a relatively tight feedback loop. The simulated perceptual sense needs to check the direction of movement of the mouse, and be able to provide sufficient information to cognition to allow it to (dynamically) modify the movement as appropriate.

Mouse movements to a desired object should be terminated when the pointer is located over some part of that object. This restriction means, for example, that when the mouse pointer starts down and to the right of the desired target object, the movement should stop soon after the mouse pointer is above and to the left of the bottom right hand corner of that object. The cognitive model will handle when to terminate mouse movements.

4.2.3. Mouse Button Actions

There are a number of fundamental actions that need to be supported:

- mouse button press and hold, so that the model can select items from pull-down menus (and drag items around the display)
- mouse button release, so that the model can terminate the selection of an item (or to do the drop in a drag-and-drop operation)
- mouse button click
- mouse button double click

Each of these actions should take a finite (albeit small) amount of time.

4.2.4. Typing

There needs to some sort of buffering mechanism available for typing so that the model can demonstrate the necessary degree of expertise in typing. Typically expert transcription typists can buffer up to three or four words ahead—this is nominally somewhere around 15-20 characters. Although this feature goes beyond what is required for the current model (which is to perform the EW task) it would be useful to support this capability.

The time taken to simulate the pressing of a key should be some small, finite value. There are tables of times taken to press the different keys on the keyboard, and it would be sensible, and improve the plausibility of the model, if these tables were to be used.

A list of regularities for transcription typing has appeared elsewhere (Salthouse, 1986) . Some of these will be applicable to the task we are concerned with, but most will not. Initially the simulation of motor action should meet those regularities that are identified as being influential to the way that the EW task is performed.

Table 1. A more complete sequence of events necessary for changing the heading of an aircraft.

- move eyes to the “command” item on the menu-bar
- move the mouse pointer to the “command” item on the menu-bar
- press the left mouse-button
- move the eyes to the “heading” item on the “command” sub-menu that is now visible
- move the mouse-pointer to the “heading” item on the “command” sub-menu
- release the left mouse-button
- move the eyes to the desired radio-button on the dialogue box
- move the mouse-pointer to the desired button on the dialogue box
- click the left mouse button
- move the eyes to the “OK” command button on the dialogue box
- move the mouse-pointer to the “OK” command button on the dialogue box
- click the left mouse-button

- The model needs to be able to move and click the mouse, type, and move the hand between keyboard and mouse, paying heed to the various factors included as variables that can influence these actions.

5. Eye-hand Co-ordination

Eye-hand co-ordination can appear to be a relatively straightforward task. When put simply, the sequence of operations needed to change the heading of an aircraft in a simple ATC-like task (Bass, et al., 1995) is just:

- select the heading option from the command menu
- pick the desired heading from the dialogue box

On closer inspection, however, the sequence of events is far longer and involves several co-ordinated hand and eye movements. Table 1 is a more complete version of events, but is still a simplified sequence of events in that it ignores the role of visual search, and the fact that the user may look for visual feedback to confirm that the operations have been successfully completed.

All of the behaviour in this example has to be supported by the model’s interaction mechanisms. There are several implications worth noting. The main one is that the response of the simulated senses must be short enough to enable feedback to take place *and* the model to respond within a reasonable time frame. This implication is probably most crucial during the second phase of mouse movements that is controlled using visual feedback (as described earlier). The processing of feedback is done by the cognitive model, since it has to process information from the simulated perceptual capability, compare it with expectations, and then send out movements via the simulated motor capability. The entire process needs to happen within a reasonable time frame (somewhat less than a second), since mouse movements from a starting point to a target object typically take of the order of 1 second.

In order to support future enhancement of the model, due consideration must be given to the truly dynamic nature of mouse movements. Mouse movements can be reprogrammed dynamically on-the-fly, in such a way that the movement gets corrected based on visual feedback of distance and direction errors (Wickens, 1992; Thierry Baccino, personal communication).

In the initial implementation, it is unlikely that Fitts' (1954) law will be observed. The reason for this is that we do not take into account the speed-accuracy trade-off that naturally arises in mouse movements—this is what Fitts' law captures in a formal way—since we ignore the size of the target of the mouse movement. In future implementations, as we introduce more fine-grained detail into the perceptual and motor capabilities, Fitts' law-like behaviour should naturally emerge.

6. Support Facilities

Support facilities are required to help debug the cognitive model and its interaction with the simulation by explaining the inputs to the model and the behaviour of the interaction elements. These facilities must be invisible to the model, providing it access to only the same displays that a user would see. The displays for the support facilities should be implemented in such a way as to allow the task to maximise the use of available screen space. Thus the support facilities should not initially appear on the display, but should have to be explicitly invoked by a unique key sequence (e.g., control-A), or as a pop-up menu invoked via an unused button sequence on the mouse.

Under normal circumstances, when the model is performing the task, there may be little need to access the support facilities. Their main use will be when the simulation and the model have been (temporarily) halted, so the state of the world can be interrogated and changed.

6.1. Visual representation of the simulated senses

A visual representation of what the model is currently interacting with during task performance is required on the display screen, as follows:

- The model needs a distinct mouse pointer object to indicate the current location of its mouse. The model's mouse pointer must be visually distinct from, and appear separately from the real mouse pointer, which will must also appear on the screen. Although the model's mouse pointer should be visible to the model, the real mouse pointer should not. If the model performs an action that changes the system mouse pointer, the model's mouse pointer should change in a similar manner.

- The area of the display currently considered to be projected onto the fovea should be shown on the screen in some visually distinct way. The simplest, and least intrusive is to have it surrounded by a containing line. Ideally the shape should be ovoid, but we believe that in an initial implementation a rectangle would suffice. This object should not be visible to the model, and merely serves to give the modeller some indication of where the model is currently looking.

- Simulated mouse pointer, indicating where the model's mouse is.

- Busy icon for mouse pointer (if necessary).

- Visible foveal projection on the screen.

- Model should only perceive elements of the display that form part of the task interface, that is, should not perceive the support facilities or the real mouse pointer.

6.2. Manual control and display of the simulated senses

Manual control of the interaction mechanisms senses is a useful aid to debugging the model. The analyst should be able to move the mouse pointer and fovea box by clicking and dragging them to a new position.

Table 2. Interaction system variables and their initial values.

- Homing time: the time to move the hand to the mouse or to the keyboard [400 ms]
- Typing or button press speed [280 ms for average non-secretary typist]
- Eye movement parameters [see Section 3.2.1]
- Hand movement speed [400 pixels/sec]
- Eye location [at the centre point of the display]
- Mouse location [at the centre of the display]
- Fovea size [approximately 60 pixels wide, and 40 pixels deep]

Simple textual displays of what the fovea will send on a look command and what the model sent as its motor outputs are desirable as well. This facility allows the modeller to interrogate the display, without having to physically inspect what appears in the model (if indeed anything), and allows the modeller to ascertain whether the model is currently seeing the expected thing. Executing this function may be an expensive operation (redisplaying derived information), so a reasonable use of default behavior and turning window updates off is acceptable.

- Manual interrogation of simulated senses.
- Manual interrogation of displayed objects.

6.3. Pause/resume capability

In order to make use of the manual control facility described above, it will be necessary to be able to temporarily halt the execution of the task simulation and the Soar model. The simulation needs to support this in such a way that when the modeller pauses the simulation, a command is sent to the model to cause it to halt, and the simulation also needs to halt. When both have been halted, the modeller will be able to interrogate the current state of affairs in the task environment. It should also be possible to restart the execution of the task simulation, using a similar mechanism.

The typical scenario would require that the modeller be able to pause the simulation, via the interface, which would then cause the model to temporarily halt too. In order to resume, however, the model needs to be resumed first, since it is not possible to get the model to resume execution without manual intervention, and have it send a resume command to the simulation.

- Simulation pause facility.
- Simulation resume facility.

6.4. Parameter and status values

There are a number of parameters that will facilitate the debugging of the model. These parameters should be loaded at initialisation time to perform actions such as determining the initial location of the eye on the display. Whilst the model is executing, it should be possible for the modeller to read the parameters, and it may be possible to modify some of them (dependent on the synchronisation between the model and the interaction mechanisms). Table 2 shows the minimum set of values that should be adjustable by the modeller (default values are shown in brackets).

In addition to the adjustable parameters, it may prove useful to have some representation of the current internal status of the model. As a bare minimum the status information should include the current location of the simulated hands: at the keyboard or at the mouse.

- Control parameters in a controlling display: homing time; typing speed; eye movement speed; hand movement speed.

6.5. Model-simulated sense communication

The model needs to pass its commands to the simulation and have the data structures passed back in a usable form. In order for the model to communicate with the interaction mechanisms the data structures used by the model and the simulated senses need to be compatible. The compatibility is achieved by using MONGSU, which translates the data structures into the appropriate format. MONGSU is also used to physically realise the transfer of the data by means of a Unix-style socket mechanism.

The data from the SL-GMS interface, which sits on top of the OOPSDG simulation, will be transferred as ASCII characters. MONGSU (Ong & Ritter, 1994) converts the data it receives from the simulation as a list of attributes and values into attribute-value pairs that can be added to the Soar model's state representation. For transfer in the other direction, the process is reversed.

At present the sockets/MONGSU communication link transmits ASCII characters. If there are problems in attaining an appropriate rate of throughput, the communication mechanism may be modified to utilise a faster mechanism, such as CRTCP (Ramsay, 1995a) .

- Display data structures need to be compatible with MONGSU.

- MONGSU must be used on the Soar side.

7. References

Allport, A. (1989). Visual attention. In M. I. Posner (Ed.), *Foundations of cognitive science.* 631-682. Cambridge, MA: MIT Press.

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Anderson, J. R., Matessa, & Douglass (1995). The ACT-R theory and visual attention. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*. 61-65. Hillsdale, NJ: Lawrence Erlbaum Associates.

Bass, E. J., Baxter, G. D., & Ritter, F. E. (1995). Using cognitive models to control simulations of complex systems. *AISB Quarterly, 93*, 18-25.

Boff, K. R., & Lincoln, J. E. (1988). *Engineering data compendium: Human perception and performance*. Wright-Patterson Air Force Base, OH: Harry G. Armstrong Aerospace Medical Research Labratory.

Card, S., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Carlton, L. G. (1981). Processing visual feedback information for movement control. *Journal of Experimental Psychology: Human Perception and Performance, 7*, 1019-1030.

Craik, K. (1943). *The Nature of Explanation*. Cambridge, UK: Cambridge University Press.

Irwin, D. E. (1992). Visual memory within and across fixations. In K. Rayner (Ed.), *Eye movements and visual cognition.* 146-165. Berlin, Germany: Springer-Verlag.

Kelley, C. (1968). *Manual and automatic control*. London, UK: Wiley.

Myers, B. A., Giuse, D. A., Dannenberg, R. B., Vander Zanden, V., Kosbie, D. S., Pervin, E., Mickish, A., & Marchal, P. (1990). Garnet: Comprehensive support for graphical, highly-interactive user interfaces. *IEEE Computer*, *23*(11), 71-85.

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

Ong, R. (1994). *Mechanisms for routinely tying cognitive models to interactive simulations*. MSc thesis (2 vols). Available as ESRC Centre for Research in Development, Instruction and Training Technical report #21 and as ftp://granby.ccc.nottingham.ac.uk/pub/lpzfr/mongsu-2.1.tar.Z, U. of Nottingham.

Ong, R., & Ritter, F. E. (1994). Mechanisms for routinely tying cognitive models to interactive simulations. In K. Hurts & K. van Putten (Eds.), *Overheads included in the Proceedings of the EuroSoar 8 Workshop*. 71-87. U. of Leiden, NL: Graduate School of Experimental Psychology.

Park, K. S. (1987). *Human reliability: Analysis, prediction, and prevention of human errors*. Amsterdam, NL: North-Holland.

Ramsay, A. (1995a). Compressed real time communications protocol (CRTCP) specification No. TDS/10/AFR/001). DRA/Portsdown.

Ramsay, A. F. (1995b). OOPSDG Modelling Environment for the Centre for Human Sciences No. DRA/CIS(SS5)/1026/9/2). DRA Portsdown.

Salthouse, T. A. (1986). Perceptual, cognitive, and motoric aspects of transcription typing. *Psychological Bulletin*, *3*(3), 303-319.

Sherrill-Lubinski Corporation (1994). SL-GMS Technical Overview. Corte Madera, CA: Sherrill-Lubinski Corporation.

Sloman, A. (1987). On designing a visual system. *Journal of Experimental and Theoretical AI*.

Tabachneck, H., Leonardo, A. M., & Simon, H. A. (1995). *How Does an Expert Use a Graph? CaMeRa: A Computational model of Multiple Representations*. Unpublished.

Appendix A: Summary Requirements Checklist for Initial Implementation

A.1. Simulated perception

- Thumb-nail sized rectangular fovea (covering 2° of visual angle).

- Rectangular parafovea (covering 5° to the left, and 5° to the right of the fovea).

- Periphery - anything beyond the parafovea.

- Movement of fovea, based on relative screen co-ordinates.

- Dynamic saccadic movement of fovea (and corresponding movement of parafovea and periphery).

- Tracking of moving objects.

- Fixations at current location for a duration based on complexity of object.

A.2. Simulated motor action

- Upper limit to mouse movement speed.

- Two phase positioning movements: first phase is 75% of total movement. Second phase is controlled by visual feedback.

- Continuous tracking movements.

- Sequential movements.

- Repetitive movements.

- Time lag for moving between mouse and keyboard.

- Mouse button event actions: press and hold; release; click, double-click.

- Internal buffer of 15-20 characters as a type-ahead store.

- Finite typing speed.

- Finite mouse button operation speed.

A.3. Eye-hand co-ordination

- Dynamic mouse movements (?).

- Fast response time to enable control of mouse movements via visual feedback.

A.4. Support facilities

- Simulated mouse pointer, for manipulation by the model.

- Busy icon for mouse pointer.

- Visible foveal projection on the screen.

- Model should only perceive elements of the display that form part of the task interface, i.e., should not perceive the support facilities or the real mouse pointer.

- Manual interrogation of simulated senses.

- Manual interrogation of displayed objects.

- Simulation pause facility.

- Simulation resume facility.

- Control parameters: homing time; typing speed; eye movement speed; hand movement speed.

- Display data structures need to be compatible with MONGSU.

Appendix B: A Brief Review of Human Perception[2]

Visual perception involves three aspects: where the objects are, when they occur, and whether they are moving. These characteristics are defined with respect to a frame of reference, such that that statements about the motion of an object, for example, are typically made with respect to the surface of the earth. This type of perception is termed geocentric. The first level involved in recovering geocentric information in the eye is the retinocentric level, where information is coded in terms of the retinal co-ordinate system. Note that the term retinocentric relates to the nature of the information, rather than its anatomical location, so a response that is retinocentric in character can occur in the visual cortex.

At the next level, there is an egocentric frame of reference, which uses an integrated binocular signal from the two eyes together with information about their movement. The origin of this frame of reference lies somewhere between the eyes. The directions of the objects is perceived with reference to this egocentre. (The required simulated perceptual capability is like a cyclopean eye located in the middle of the forehead., i.e., based around the egocentre.)

Motion perception involves the consideration of information from a number of sensory systems. For example, information from the vestibular system is involved in determining the acceleration of the head in three dimensions, which is essential in unambiguously determining the pattern of retinal change caused by head movement.

The basic process of recovering geocentric movement occurs as follows. Retinocentric motion is detected in each eye, which gives rise to a single binocular (cyclopean) retinocentric signal. The signal expresses the motion with respect to a point located between the eyes. An egocentric representation can then be formed by integrating a single signal for eye movement (since both eyes move by the same amount) with the binocular retinocentric signal. This representation captures changes in object direction providing the observer is stationary. If the observer's head moves, then extra information is needed to scale the egocentric information by the perceived distance to get a three dimensional location of objects with respect to the observer. A geocentric representation of the movement can then be obtained by correcting the three-dimensional location for the effects of self-movement (which displace the image on the retina).

At the geocentric level of analysis, the nature of the information required is defined, rather than the sources. The geocentric representation conveys information which is independent of observer movements. It contains the properties needed for perceptual constancies, such as size, shape, and location. As such, it forms the basis for object recognition, since objects are represented in a manner which remains consistent even when the viewing conditions are changed.

Geocentric representations can be used to perform tasks expressed in terms of other co-ordinate systems, for example, to judge the location of one object with respect to another. The frame of reference is no longer explicit, but can be recovered, if required, using information about the articulation of the eyes, head and body is kept. It follows that all perception is necessarily geocentric in situations where eye movements are involved.

[2] This information is based on Wade & Swanston (1991).