'Look No Goals!': A Sufficient Model of Simple Algebra Problem Solving Without Explicit Goal Representation

Gareth E. Miles (GMiles@Glam.ac.uk)
School of Psychology, University of Glamorgan
Trefforest, CF37 1DL, UK

Abstract

Theories of cognitive control suggest that goal representation is both amodal and specialized. The GLAM-PS (Glamorgan Problem Solver) Cognitive Architecture has no specialised goal representation or goal handling. In addition, GLAM-PS, a distributed production system theory, does not use amodal representation (it is an embodied/grounded architecture). The current paper demonstrates that it is, nevertheless, possible to model 'off-line' and abstract problem solving within GLAM-PS. A model of linear equation solving is presented. The processes used by the model to solve equations are described in detail, with a particular focus on the way control of thought and action is achieved. Instead of goals, the model's problem solving is guided by the use of naturally occurring control states derived from existing internal and external representations. The conclusion that *specialized goal representation isn't needed* may also apply to architectures using amodal as well as modal representation (e.g. J. R. Anderson's ACT-r, 2007).

Keywords: Embodied Cognition; Problem Solving; ACT-r; Cognitive Architecture

Introduction

Newell and Simon's (1972) work on problem solving highlights the critical role of goals in complex behaviour. Contemporary accounts of the role of goal representations in behaviour view the goal as a special type of amodal representation, in much the same way as Newell and Simon did. For instance ACT-r 6.0 (Anderson, 2007) represents goals as part of declarative memory, but stipulates that they influence behaviour through the action of a goal buffer (which in turn is part of an amodal cognitive module).

Memory for Goals (MFG, Altmann & Trafton, 2002) adopts a similar approach, though MFG highlights the importance of a goal's activation level in explaining how goals influence behaviour. A third notable account of goal processing has recently emerged from the ACT-r community, Salvucci and Taatgen's (2008) Threaded Cognition. This theory extends the ACT-r account of goal processing by allowing two or more goals to control cognition simultaneously, each goal pursued by simultaneously active 'threads' of behaviour. If the resources required by the threads are separate then each thread can continue as if being pursued by itself.

Whilst most specialised theories of goal processing have emerged from work with production system architectures, there is also a class of theories that deal with executive function more generally. These theories include at least some detail on how goals are processed and how they influence behaviour, an example is Baddeley's (1996) Central Executive component of his Working Memory model. Broadly speaking these accounts are consistent with the aforementioned theories of goal processing. This consensus is reflected in the use of goals as a theoretical construct in other areas of Psychology.

Key commonalities in these accounts are that goal representation is a) special, and b) amodal. It is special in the sense that specialized mechanisms are required to handle the role of goals in behaviour. These mechanisms store and manipulate amodal goal representations and are assumed to be primarily located in particular areas of the brain (the Prefrontal Cortex and/or the Anterior Cingulate Cortex; e.g. Anderson, 2007, maps ACT-r's goal module to the latter).

In the current paper an account of simple algebra problem solving is outlined in the form of a production system model using the GLAM-PS (Glamorgan Problem Solver) Cognitive Architecture. The account is novel because it a) does not assume any special goal processing and b) only makes use of modal representation.

Glamorgan Problem Solver (GLAM-PS)

Barsalou (2009) highlights the need for computationally implemented, theoretical accounts of Grounded Cognition. Glamorgan Problem Solver (GLAM-PS; Miles, 2011) is an attempt to address this need within a production system formalism. Architecturally, GLAM-PS is a collection of interacting modal subsystems. Each of these subsystems has its own working memory and long term/production memory. All information is processed modally. GLAM-PS assumes that central areas of the system (i.e. the brain) are primarily concerned with re-representing information between subsystems and modulating other modal processes (e.g. learning) rather than performing any form of transformational processing of amodal representations. The basic GLAM-PS architecture is illustrated in Figure 1.

Each module is an independently functioning production system with its own central bottleneck. Hence only one production can be executed per system cycle in each module. Whilst each module is representationally independent its processing is influenced by information about currently active representations in other modules. This is achieved through the production matching process. All productions are able to match up to two active representational elements from each modules working memory. So matching in any given module is influenced not only by local working memory but also by information about Modal Memory Elements (MMEs) active in the working memory of other non-local modules.

An important feature of GLAM-PS is the ability of the architecture to represent an action or sequence of action without necessarily executing it (Miles, 2009). This allows for action representations to influence the processing in all modules. For instance the consequences of a represented Manual action can be simulated in the Visual Input and Aural modules without that action having been executed. Actions are only executed once an Action Execution Threshold (of MME activation) is exceeded.

A further novel feature of GLAM-PS is the way the timing of Inter-Module Communication (IMC) is handled. In GLAM-PS once an MME is created or activated then it can immediately be used to match productions in the module it belongs to. However, this MME will not be available to other modules until it has remained active for a number of system cycles equal to a global parameter (set to 5 in the model presented in this paper). In essence, there is a delay associated with providing information about what is happening in one module to another module. The implications of this simple IMC restriction are manifold but are not the focus of the current paper. However, the principle impact of this feature is that processes that don't require new IMC (automatic behaviour/attention) occur more rapidly than those that do (controlled behaviour).

Goal Representation in GLAM-PS

In GLAM-PS Modal Memory Elements (MMEs) acting as goals are not explicitly tagged as goals. Indeed the system of goal representation in GLAM-PS is implicit rather than explicit. It is implicit because an MME acting as a goal is not identifiable in itself as a goal in any way. Its role as a goal emerges from the way it is used to guide behaviour. This is done in a similar manner to how explicit goals guide behaviour in ACT-r, i.e. by being a necessary element in the matching of productions.

There is potential for any MME to act as a 'goal'. This is true whether the MME is a visual object, an auditory object, a manual intention/action, a sub-vocal/vocal articulation, or belongs to another module. Depending on the situation, different MMEs might act to guide behaviour. An example of this is seen in Miles (2009) where a particular manual intention/action (moving a disk) structures behaviour in the Tower of London problem. The structure that is imposed is functionally equivalent to hierarchical subgoaling.

Often, in GLAM-PS, control over behaviour will rely on coalitions of two or more MMEs, perhaps from different modules. Hence multiple MMEs can potentially constitute a collective control state in the same way that an individual MME might act as a control state / guide behaviour. A final point about the technical implementation of goal representation in GLAM-PS is that multiple threads of behaviour can coexist in a way similar to the Threaded Cognition model of Salvucci and Taatgen (2008), with the possibility of minimal or even no interaction between the threads.

Finally, it is necessary to note that the language abilities of GLAM-PS do provide a way of defining explicit goals modally. The simple maintenance in the Speech production module of a word (e.g. 'phone') might be sufficient to guide behaviour in future (e.g. prompting the actor to first find a phone and then make a telephone call). An even more explicit and structured verbal representation might be maintained using words that imply the necessity of action – 'I need to..', 'I must aim to..', or even 'my goal is to..'

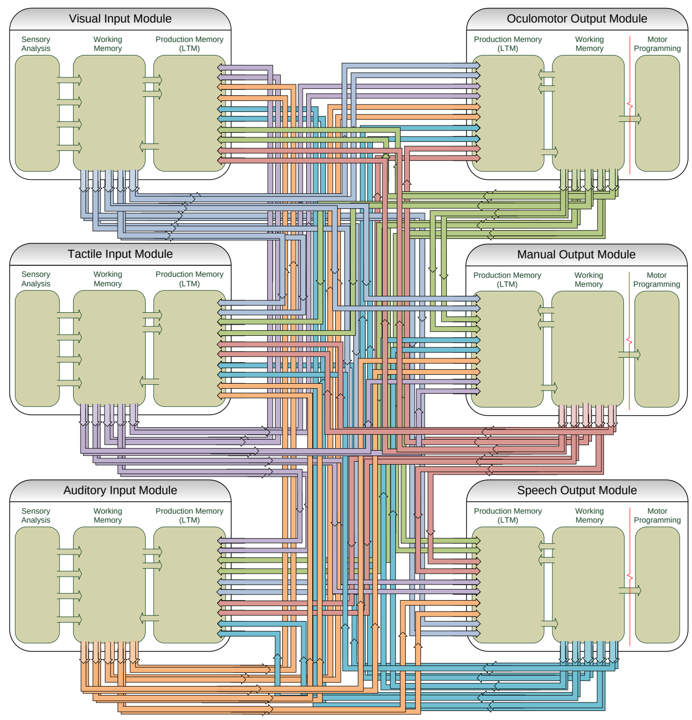

Figure 1: The structure of GLAM-PS. Perceptual (left) and Gestural (right) modules are shown. The flow of information between Module Working Memory and Module Production Memory is indicated with colour-coded arrows.

A New Domain for GLAM-PS: Linear Equations

Previously, the representation of goals has been explored in GLAM-PS using a model of Tower of London problem solving (Miles, 2009). This model used representations of intended actions to guide behaviour. For instance, a representation of a disk move in the Manual module of GLAM-PS was used to 'drive' behaviour. Initially this representation is underspecified and cannot be acted on, but it is subsequently used as a control state in productions that add details about the intended move (where the disk is to be moved to, etc.). In this way the representation of the motor action under consideration can be used for control.

In the current paper a different type of problem solving domain is modeled, simple algebra. This was chosen because i) it has become an important paradigm for accounts of goal usage (e.g. Anderson, 2005) and ii) it is an example of 'off-line' / abstract problem solving. Whilst a domain like the TOL has intervening external states between the initial state and goal, the type of problem solving modeled here has no such states (i.e. an equation is presented and then the participant gives an answer). 'Off-line' problem solving presents a significant challenge to grounded / embodied approaches to cognition and it is important to establish that a theory like GLAM-PS can provide a sufficient account.

The problems modeled in this paper are linear equations of the form: Ax + B = C. Where A, B and C are integers and the value of x needs to be found.

GLAM-PS Algebra Problem Solving Model

The structure of the model (see Miles, 2011) can be understood in terms of the productions that are local to each of the six modules. Immediately the complexity of the model is apparent, in total 109 productions are included. Of these productions 37 are in the Visual Input module, 39 in the Speech module, 13 in the Auditory input module, 3 in the Manual module and 17 in the Occulomotor module.

Broadly speaking the sub-processes/steps involved in solving a linear equation in the GLAM-PS model are the same as those taken by Anderson's (2005, p. 319) model of 2-step linear equation solving using the unwind strategy (skilled performance). The difference between them is found in the grain size of each model (the GLAM-PS account is arguably finer grained) and the manner in which control is achieved. In the ACT-r model the sub-processes involved in solving the equation are dependent on goal representations, which must be matched, making each production specific to a sub-process. In the GLAM-PS model the lack of explicitly defined goals necessitates the use of naturally occurring representations as control states.

In the following sections the manner in which control is achieved by the GLAM-PS model is described in detail. The solution to the equation '3x + 5 = 11' will be used as an example. This description is broken into the three main sub-processes performed by the model. Step 1 and Step 2 correspond to the sub-processes described by Anderson (2005, p. 319), with Step 0 covering the encoding of the problem. Figure 2 shows an example of the contents of Module Working Memory in the early stages of Step 1.

Step 0: Reading and Encoding the Equation

The necessarily fine grained nature of GLAM-PS requires the detailed modeling of the initial encoding of a problem. GLAM-PS first encodes the broad structure of the problem. This is initiated by eye movement to the left side of the equation, then eye movement to the right side of the equation. At this point no information is being recoded to the Speech module.

The encoding of the elements within the equation begins following eye movement to the first element of the equation ('3x'). Eye movement to this element creates a Visual Input (VisIn) MME encoding of the '3x'. This includes information about the perceptual category of the MME, the identity of the characters in the different roles within this MME, the vividness of the MME and the spatial relationship between the MME and other objects (both hierarchically and peer-to-peer). It takes 5 cycles (the set value for IMC) before the encoding of this element will be available to modules other than the VisIn module. During this time the MMEs available for matching in the Occulomotor (Occ) module are unchanged and the '3x' remains fixated.

Only after 5 cycles does the VisIn representation of the '3x' become available to the Occ module. This provides the signal for the Occ module to move the focal point to the next element (i.e. '+ 5'). At the same time the VisIn '3x' MME becomes available to the Speech module. Speech module productions then begin to recode the '3x' phonologically. At this point, the next VisIn MME (the '+5') encoded won't become available to the Speech module for another 5 cycles, so the Speech module can use these cycles to further process the phonological encoding of the '3x' VisIn MME. This allows time for Speech module productions to initiate a Speech Plan representation, as well as encoding the '3x' as the first element of the Speech Plan.

When the '+5' VisIn MME does become available to the Speech module, it is encoded as part of this 'open' Speech Plan (hierarchical relationship) and as the item following the '3x' in this Speech Plan (peer-to-peer relationship). The remainder of the equation is read and encoded using a similar pattern of interaction between the Occulomotor, Visual Input and Speech modules.

Control is possible during Step 0 because of the serial nature of the task. Each step is dependent on the previous step. Only when the Occ module notes that an item has been encoded by the VisIn module will eye movement to the next item move control to that item. That is not to say that each term is fully processed before the next begins, as within-task threading of action is implicit within GLAM-PS. In this case the processing of '3x' by the Speech module will occur at the same time as the Occ and VisIn modules begin to process the next item ('+5') (this is consistent with models of eye-movement in reading).

An important point is that all the productions that govern the reading of the equation are applicable during the other Steps (i.e. 1 and 2) involved in the solution. The productions governing recoding to the Speech module of VisIn are often fired during other Steps, sometimes as a critical part of that sub-process, but at other times as an incidental by-product. The same is true of the productions governing eye movement during reading.

Step 1: Resolving the Addend

Control in GLAM-PS is achieved through the opportunistic use of MMEs that reliably predict when a process should occur (the MMEs are used as control states). Control over the shift from encoding the equation (Step 0) to solving the equation (Step 1) must be based on a representation that reliably predicts the end of encoding. In the model this representation is the Speech module encoding of the last term in the equation. This is matched to Occ productions that begin to solve the problem by searching for the location of a variable term (if a VisIn representation of the variable isn't active) or fixating on the variable term (if it is active).

A production in the Speech module keeps the controlling last term MME active.

Once the renewed encoding of the variable (‘3x’) is available to all modules then focus is shifted to the addend on the same side of the equation as the variable (‘+5’). Fixation is maintained on the addend. In the next stage of the solution control is dependent on the presence of i) the last element encoding in the Speech module, ii) the Occ encoding of the fixation, iii) the VisIn encoding of the variable and iv) the VisIn encoding of the addend. Together these four conditions act as a highly specific control state that identifies the presence of the necessary conditions for the main sub-process in Step 1 to begin.

Once this control state is true, the VisIn representation of the addend (‘+5’) is inhibited. The inhibited representation then controls the next stage which is to project/simulate the presence of the addend in the VisIn module (initially in the same location as the inhibited representation of the addend and without the sign, i.e. ‘5’). This type of simulated perceptual representation is a characteristic of Grounded Cognition models. Differences in ‘vividness’ allow GLAM-PS to distinguish externally created perceptual objects from simulated ones.

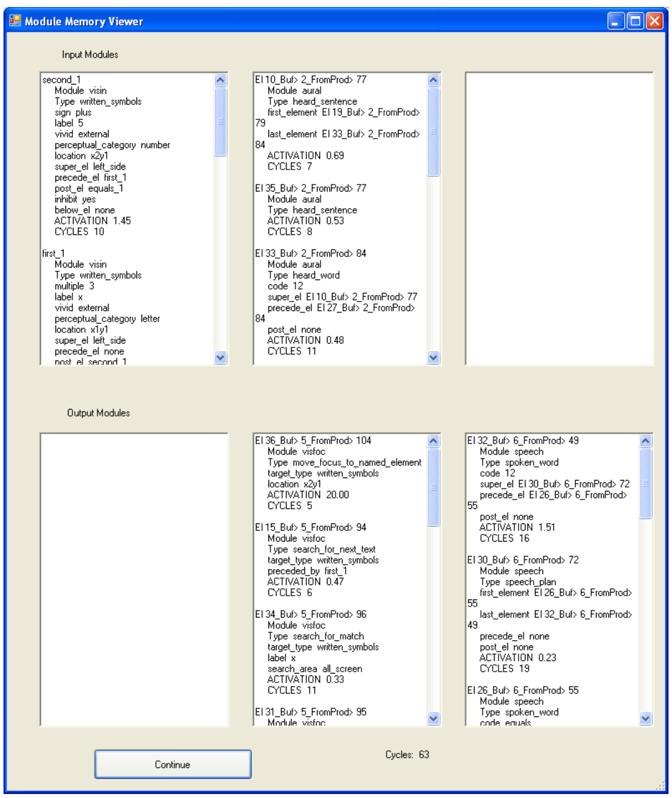

Figure 2: A partial view of the most active elements in each modules working memory early during step 1. (Modules are, clockwise from top left: Visual Input, Auditory Input, Tactile, Speech, Occulomotor and Manual). CYCLES indicates how long a given ME has been active.

The next stage in solving the algebra problem is to apply the inverse effect of the current addend to the other side of the equation (to unwind it). In the model the critical thread of control is now governed by the presence of the projected addend (‘5’). Eye fixation is now moved to the last term of the equation. Once fixation is on this location, and the VisIn representation of both the projected addend (‘5’) and the last (‘11’) are active, then the projected element is ‘moved’ to a location to the right of the last term (i.e. its projected location is altered). Other VisIn productions adjust information about item order and add an operator to the projected addend (the inverse of the one in the inhibited representation, i.e. ‘-5’).

The model at this point retrieves the number fact describing the addition of the projected addend (‘-5’) and the right-side term (‘11’), in the case of the example ’11 – 5 = 6’. It is perhaps conceivable that this fact could be encoded entirely within the VisIn module (especially if experienced enough times), however the algebra GLAM-PS model retrieves number facts using Speech representations (quite literally GLAM-PS’s declarative memory). The VisIn representations of the right-side term and the addend are matched by a production that recodes the sum/subtraction phonologically (‘eleven minus five’) to the Speech module. Other productions structure this Speech representation within a Speech Plan. The phonological code and ‘open’ Speech Plan then acts as the condition for a Speech module production that adds the answer to the end this speech plan (‘equals six’).

Once the Speech modules ‘answer’ becomes available to the Visual Input module (5 cycles) then a trio of VisIn productions inhibit both the right-side term and the projected addend (i.e. the components of the sum that has just been retrieved) and in their place the answer is projected/simulated visually (again, other VisIn productions adjust order information).

At this point Ax + B = C has been reordered as Ax = D, where D = C – B, in our example we now have 3x = 6.

Step 2: Resolving the Coefficient

The transition to the final stage of the solution requires a relatively complex control state. The model must establish that a variable term is alone on the left side of the equation and a numeric term alone on the other side of the equation. This requires the variable term, the equals sign and the numeric term to act as a combinatorial control state. Each is represented separately and only two MMEs from each module may match any given production instance. Hence, in the model both VisIn and Speech representations of the transformed state of the equation (‘3x = 6’) are required to establish the necessary control state. The VisIn representation is already active. The Speech representation is established by re-reading the current state of the equation (using a combination of external representations and projected representations). The re-read itself is initiated only once a complex control state is true (the presence of a projected right-side numeric term on its own next to the equals sign).

Once the re-read has taken place it is possible to establish the control state for resolving the coefficient. The rest of the process then proceeds in a relatively straight forward manner. As was the case with the added, the term in question (the variable term here) has its external representation inhibited and then replaced by a projected/simulated copy (‘3x’). The coefficient is then separately projected below the right-side numeric term (‘3’). Other visual input productions then remove the coefficient element from the projected variable term (leaving the x on its own) and update the relationship between the terms on the right-side (describing the ‘6’ as being above the ‘3’).

The retrieval of the number fact (‘6 / 3 = 2’) is then accomplished by recoding the visually represented division sum into the Speech module (‘six divided by three’). The retrieval of the answer to this division sum proceeds as per the retrieval of the subtraction fact in Step 1. The answer is added first phonologically then after 5 cycles for IMC it is projected in place of the visually represented division sum (which is inhibited). Hence the variable (‘x’) is now alone on the left-side of the equation and our solution is alone on the right-side (‘2’). The equation is solved.

The presence of the variable alone on the left-side of the equation triggers the creation of a Manual response representation. This incomplete Manual response representation is then after 5 cycles (IMC) used to move fixation to the right-side term (if isn’t already active). Finally Manual productions add the identity of the keys to be pressed (’2 then <ENTER>’) and execute the action.

General Observations on Control in the Model

The paper describes the workings of a computational model of simple algebra problem solving. The issue of interest within these workings is the way in which control is achieved without the use of explicit goal representations. This is best understood by observing how the model achieves control at each given point in the solution path (see previous sections). However it is possible to make some general observations about how control is achieved in this model:

1) The control states used varied in complexity. In some cases four or five MMEs were needed in coalition for an adequate control state to be described.

2) The complexity of control states was greatest when transitioning between sub-processes. It is notable that this is the point when control moves from one explicit goal to another in traditional models of problem solving.

3) When transitioning between sub-processes there is often a delay whilst the necessary control state is assembled.

4) Within each sub-process control states tended to be simpler, often being based on the presence of a single MME and/or the evidence that the prior step in the sub-process was completed (or nearly completed).

5) Control states were not just defined by the presence of MMEs, but also by the absence of MMEs.

General Discussion

The current paper demonstrates how control over complex problem solving can be achieved without the use of explicit amodal goal representation. The problem solving modeled was ‘offline’ and abstract, yet it was accounted for by a production system model that had no specific goal representation and was only able to use modal representation (an example of grounded cognition; Barsalou, 2009). The model (of linear equation solving using an unwind strategy) achieves control over action through the influence of naturally occurring control states. These control states are one or more representations (the latter co-occurring) that provide a reliable and stable signal that a particular set of productions are appropriate.

The model is implemented in the GLAM-PS Cognitive Architecture. GLAM-PS is an attempt to computationally implement a ‘strong’ version of Grounded Cognition (c.f. Barsalou, 2009). GLAM-PS does not allow any amodal representation, nor has it any device for goal representation. Hence, demonstrating sufficiency in complex problem solving domains is an important challenge for GLAM-PS. However it is important to note that the conclusions of the current paper are not just applicable to theories suggesting grounded representation.

The current model of algebra problem solving differs from ACT-r models primarily in terms of how imaginable working memory is represented. In ACT-r the ‘Imaginal’ buffer (also known as the ‘Problem State buffer’) is used to hold the internal representations of the interim states of problem solving. In the GLAM-PS model these states are represented directly into the Visual Input module (i.e. they are simulated, c.f. Barsalou, 2009). Whilst there is quite a big theoretical difference between these two approaches, there is a great deal of functional equivalence (i.e. the kind of information ACT-r places in the imaginal buffer is very similar to the information simulated in GLAM-PS’s Visual Input buffer).

If the Grounded approach to knowledge representation used by GLAM-PS is set aside, it is still possible to conclude that internal and external representations can be used (often in combination) to provide control states that could guide behaviour. For instance, in the current model the projected interim results of the solution serve as a control state that indicates the solution to the problem is in progress (indicating the transition from Step 0 to Step 1). In an ACT-r model the contents of the imaginal buffer could be used in much the same way (i.e. to control action).

Whilst the current paper suggests explicit goals might not exist, it paradoxically does not necessarily imply that ACT-r is incorrect in much of its account of how goals control behaviour. Rather, GLAM-PS suggests that ACT-r (and similar theories) have an abstracted view of cognitive control. This view, with its explicit and amodal goals, is useful and allows for tractability in the modeling of complex behaviour. Whilst the GLAM-PS architecture is relatively simple, it requires complex models that are time-consuming to develop. GLAM-PS may potentially provide a finer grained account of behaviour, but a more abstract account using explicit amodal goals may be easier to use and more useful in many situations. Conversely, there will be times when a finer grained account might be desirable (e.g. when attempting neural mappings).

It is still necessary to establish the validity of the model. Future Experimental studies will attempt to do this, as well as playing a formative role in future iterations of the algebra model. Arguably the model inherits at least some validity from the algorithmically similar ACT-r algebra modeling (e.g. Anderson, 2005).

What makes GLAM-PS 'Grounded'?

Neural network or Bayesian formalisms have typically been suggested as the most appropriate way of computationally implementing grounded / embodied cognition. In this regard GLAM-PS is important as it seeks to establish that grounded cognition can be modeled symbolically. Such an approach has many advantages (see Anderson, 2007), but requires consideration of what constitutes a grounded representation.

Any representation, whether amodal or modal can be described as grounded if its origins can be traced to perception and action. Hence, an amodal representation abstracting information from multiple modalities could be described as grounded if its perceptual-motor origins were known. However, First Order Grounding (used exclusively in GLAM-PS) distinguishes representations that are modality specific from grounded representations that integrate information from multiple modalities (Higher Order Grounding).

The modeling work reported here has been guided by a small set of principles that describe First Order Grounding. The Principle of Within-Module Analysis specifies that all representations in perceptual modules should only include information derivable from perceptual inputs to that module. Similarly, the Principle of Within-Module Capability specifies that all representations in motor/gestural modules should only include information describing the actions taken by that module alone. Information about co-occurrence of representations within a module may also be included (e.g. perceptual categories). Whilst it is up to the modeler in the first instance to ensure these principles are followed, the explicit nature of symbolic modeling allows any deviation from these principles to be identified by readers/reviewers.

Mapping GLAM-PS to the Brain

The neural mapping of GLAM-PS is tentative and emerges from the theory (rather than the other way round, as in 4CAPs, Just & Varma, 2007). A key assumption is that the complexity of central / modality-independent areas of the brain reflects the complexity of the inputs and outputs to these areas rather than the complexity of the function being computed (in said areas). On this view, apparent fractionation of function (as observed in FMRI studies) will often reflect differences in the input and outputs to a function rather than differences in the function itself.

The account of Prefrontal function is a potential strength of GLAM-PS. Inter-Module Communication (IMC) is mapped onto the Prefrontal Cortex (PFC). In GLAM-PS IMC provides a signal biasing the action of modality specific modules, this is consistent with the function of the PFC described by Miller & Cohen (2001). Complex control and LTM retrieval are reliant on IMC in GLAM-PS (the productions required will typically match to one or more non-local MMEs). Tellingly, many simpler productions are also IMC dependent because they detect the absence of an inhibitory non-local MME (hence becoming more likely to match when not appropriate if IMC is disrupted). This combination of function (complex control, memory retrieval, inhibition of inappropriate responses) appears a good match for what is known about the PFC.

The Anterior Cingulate Cortex's (ACC) role in cognitive control (Anderson, 2007) is not currently accounted for in GLAM-PS. A potential mapping might focus on the somatic inputs / outputs of the ACC, perhaps utilising the concept of a 'drive'. Indeed, the ACC may be the source of the subjective feeling of intentionality that is currently missing from the theory of cognitive control presented here.

References

Altmann, E. M., & Trafton, J. G. (2002). Memory for goals: An activation-based model. *Cognitive Science, 26,* 39-83.

Anderson, J. R. (2005). Human symbol manipulation within an integrated cognitive architecture. *Cognitive Science, 6,* 287-317.

Anderson, J. R. (2007) *How can the human mind occur in the physical universe?* Oxford: Oxford University Press.

Baddeley, A. D. (1996). Exploring the Central Executive. *Quarterly Journal of Experimental Psychology, 49A,*5-28.

Barsalou, L.W. (2009). Simulation, situated conceptualization, and prediction. *Philosophical Transactions of the Royal Society of London: Biological Sciences, 364*, 1281-1289.

Just, M. A., & Varma, S. (2007). The organization of thinking: What functional brain imaging reveals about the neuroarchitecture of complex cognition. *Cognitive, Affective & Behavioural Neuroscience, 7,* 153-191.

Miles, G. E. (2009). Representing goals modally: A production system model of problem solving in the Tower of London. *Proceedings of the Thirty-First Annual Conference of the Cognitive Science Society,* 469–475. Mahwah, NJ: LEA

Miles, G. E. (2011). *Glamorgan Problem Solver and Algebra Model* [Visual Studio 2005 code]. Retrieved from http://psychology.research.glam.ac.uk/miles/

Miller, E. K., & Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *Annual Review of Neuroscience*, 24, 167-2002.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded Cognition: An Integrated Theory of Concurrent Multitasking. *Psychological Review, 115*(1), 101-130.