Available online at www.sciencedirect.com

ScienceDirect

Cognitive Systems Research 27 (2014) 42–49

Cognitive Systems RESEARCH

www.elsevier.com/locate/cogsys

Implementations are not specifications: Specification, replication and experimentation in computational cognitive modeling

Action editor: Ron Sun

Richard P. Cooper*, Olivia Guest

Centre for Cognition, Computation and Modelling, Department of Psychological Sciences, Birkbeck, University of London, UK

Received 19 March 2013; accepted 14 May 2013
Available online 5 June 2013

Abstract

Contemporary methods of computational cognitive modeling have recently been criticized by Addyman and French (2012) on the grounds that they have not kept up with developments in computer technology and human–computer interaction. They present a manifesto for change according to which, it is argued, modelers should devote more effort to making their models accessible, both to non-modelers (with an appropriate easy-to-use user interface) and modelers alike. We agree that models, like data, should be freely available according to the normal standards of science, but caution against confusing implementations with specifications. Models may embody theories, but they generally also include implementation assumptions. Cognitive modeling methodology needs to be sensitive to this. We argue that specification, replication and experimentation are methodological approaches that can address this issue.
© 2013 Elsevier B.V. All rights reserved.

Keywords: Theory specification; Implementation detail; Replication; Sensitivity analysis; Computational experimentation

1. Introduction

Computational modeling has been a central method in cognitive science since Newell, Shaw and Simon's seminal work on human problem solving (Newell, Shaw, & Simon, 1958). However, a recent debate initiated by McClelland (2009) on the place of computational modeling within the discipline has brought into focus some limitations of the modeling enterprise as currently practiced. Perhaps most critically, one subtext of McClelland's discussion is that many aspects of modeling are not well understood by those who are not trained in its methods. This includes fundamental aspects such as the purpose of modeling and the relation between a model and any theoretical assumptions it might be held to embody, as well as issues such as whether models can be falsified and, if not, whether lack of falsification is critical to the modeling endeavor as a scientific enterprise. Addyman and French (2012) add to this list of limitations the claim that current modeling practices make it "almost impossible to access, check, explore, re-use, or continue to develop" models (p. 332). They argue for a step-change in modeling methodology, suggesting that some of the limitations of modeling might be overcome if modelers were routinely to make their models publicly available, accompanying them with easy-to-use user interfaces.

These are not the only issues of concern to contemporary cognitive modeling, and different issues will be of concern to different audiences. Thus, those who are concerned with understanding theoretical implications, as we are in this paper, may have different requirements from those

* Corresponding author. Address: Department of Psychological Sciences, Birkbeck, University of London, Malet Street, London WC1E 7HX, UK. Tel.: +44 20 7631 6211; fax: +44 20 7631 6312.
E-mail address: r.cooper@bbk.ac.uk (R.P. Cooper).

1389-0417/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cogsys.2013.05.001

concerned with more applied implications. Moreover there are other issues, such as that discussed by Gray (2010), who highlights the difficulty of mastering alternative modeling approaches and the consequent tendency for modelers themselves to be paradigm bound. He notes an "emerging constellation" of modeling systems and tools that spans temporal scales and provides the prospect of environments for modeling that can be used by novices to develop models at different scales. This is certainly an encouraging development, but it does not fully address the issue raised by Addyman and French (2012). Equally, however, the solution offered by Addyman and French falls short in terms of meeting the methodological requirements of cognitive modeling as a scientific endeavor. In this paper we argue that more attention within cognitive modeling needs to be directed to distinguishing between theories and their associated computational implementations. Specification, replication and experimentation are three methodologies that, we argue, may help to draw this distinction.

2. Models versus theory specifications

We do not dispute any of the above concerns with contemporary cognitive modeling, or any of the above methodological prescriptions. Modeling is a potentially powerful technique in helping to understand any complex process, whether it be flow in fluid dynamics, star formation in astrophysics, or the mechanisms or processes underlying human (or animal) behavior. This is the aspect of modeling that McClelland (2009) focuses on when he suggests that "[t]he essential purpose of cognitive modeling is to allow investigation of the implications of ideas, beyond the limits of human thinking" (p. 16).

However, modeling has more to offer cognitive science than just the investigation of the implications of complex ideas. A central aim of cognitive science is the development of theories of the computational processes underpinning intelligent behavior, whether the behavior is human, animal or artificial in origin. Computational methods have the potential to play a central role in the specification of such theories, as the development of implementations of theories allows the simulation of the behaviors of interest (over and above the exploration of implications of sets of ideas or assumptions). In some cases it may be appropriate to equate a model (i.e., a specific implementation) with a theory (cf. Sun, 2009, and references therein). However, in general, implementations are not theory specifications (see McCloskey, 1991, for similar comments specifically with respect to connectionist models), and it is for this reason, we suggest, that the prescriptions of Addyman and French (2012) are at best insufficient for advancing computational modeling. At worst the prescriptions could undermine modeling by promoting a culture outside of the modeling community of accepting implementations at face value.

We are of course not suggesting that modelers should restrict availability of their code, nor that they should not adhere to good practice in coding. If models are to have an evidential role in support of a theoretical position then those models must be freely available for public inspection. This is just as true for models as for empirical data, and just as true in cognitive science as any scientific enterprise. Moreover sharing of models has additional benefits as it supports cumulative model development and is likely to result in the minimisation of programming errors. Thus we are also not suggesting that scrutinizing code is necessarily bad. However, if a model is to be made freely available then it must come with a clear statement of the relation between the model and the theory behind the model. For this reason it should not be necessary to scrutinize code in order to understand the model's assumptions. More specifically, as McCloskey (1991) notes, "the characterization of [a model] should provide a basis for assessing the extent to which successes or failures in simulating particular phenomena reflect theory-relevant or theory-irrelevant features [of the model]" (p. 389). Providing computationally naïve researchers with an implementation without also providing them with the means to discriminate between the consequences of theory-relevant or theory-irrelevant assumptions raises the possibility that such researchers will over-interpret the model's behavior. This can work in two ways: being excessively impressed with some positive aspect of the model's behavior that is not due to the theoretically critical elements of the model, or being unduly critical of some negative aspect of the model's behavior that is a consequence of a minor implementation decision rather than a theoretically critical element.

Addyman and French (2012) do not just argue for increased availability of the implemented code. They argue also for the simultaneous provision of a graphical user interface to support exploration of the model by the wider research community. The use of such interfaces is not novel, and like many modelers, we have relied on such interfaces in developing our models since the beginnings of our careers. However, this too is not straightforward. Contrary to the claims of Addyman and French, user interface design is not straightforward given modern programming languages and environments. In particular what is considered to be easy to use by one person may not be easy to use for another, and "good" user-interface design requires training and discipline (as illustrated by the existence of the field of Human–Computer Interaction). In fact, it is highly plausible that an easy-to-use graphical user interface will only exacerbate the problem of distinguishing theoretical consequences from the implications of a specific implementation, as computationally naïve users are unlikely to go beyond the interface.

McClelland (2009) downplays the role of modeling in theory specification, and in particular the possibility that a model might embody a set of assumptions. He recites the history of the Parallel Distributed Processing (PDP) models of past tense verb formation (Rumelhart & McClelland, 1986). The models transform a representation of the root form of a verb to a representation of its past tense.

Rumelhart and McClelland adopted a specific representation of phonology for the input and output of their original model, so-called Wickelphones and Wickelfeatures. Substantial debate ensued about the descriptive adequacy and psychological plausibility of this representational format (e.g., Pinker & Prince, 1988), and alternative PDP models of the task were developed that used alternative input/output representations in place of the Wickelphone/Wickelfeature representation (e.g., MacWhinney & Leinbach, 1991). The alternative input/output representations shared with the original representation the property that they provide distinct yet overlapping representations of the input and output items. The critical point, McClelland claims, is not the specific form of representations used in the original model, but what he refers to as the *property* of the representations – that they provide distinct yet overlapping representations. That input and output representations have this property, it is argued, is the key to the model's claimed success.

In our view nothing hangs on this distinction between properties of representations and assumptions. The more critical issue is that implementations must be computationally complete, in the sense that they must be specified to a level that allows the implementation to be run. Cognitive-level theories do not specify all aspects of an implementation, so cognitive models, as implementations, include implementation details – details of the implementation that are necessary for the implementation to be run but that are not part of the cognitive-level theory (Cooper, Fox, Farringdon, & Shallice, 1996; McCloskey, 1991; Newell, 1990). This does not mean that an implemented model cannot embody a theory's assumptions. The original past tense model did embody the assumption of distinct yet overlapping representations. It embodied that assumption within a specific implementation – the Wickelphone/Wickelfeature implementation. MacWhinney and Leinbach (1991) provided an alternative implementation of the assumption. Pinker and Prince's (1988) mistake, perhaps engendered by Rumelhart and McClelland's (1986) justifiable focus on other issues, was to conflate the relatively abstract assumption with its specific implementation in the initial model.

Models must therefore be distinguished from specifications of theory. How can computational modeling assist in the promotion of this distinction and in the clarification of cognitive theory? We propose three methodological approaches that modeling might adopt to achieve this goal: abstract specification, reimplementation, and sensitivity analyses via computational experimentation.

3. Specification

As discussed above, psychological theory operates at a more abstract level than computational implementation. For the Rumelhart and McClelland (1986) past tense model, the theoretically critical assumption (i.e., the assumption to which they were committed as theoreticians) was that phonological representations of words were distinct yet overlapping.[1] To develop a working computational model of the task it was necessary to flesh this assumption out into a specific representational scheme. The misinterpretation by Pinker and Prince (1988) of this representational scheme as an element of the Rumelhart and McClelland theory would not have been prevented if Pinker and Prince had had access to Rumelhart and McClelland's original implementation. What is actually required to prevent such misinterpretations is a specification of a theory, or the theory's assumptions, at a level distinct from any specific implementation.

Specification of a cognitive-level theory independent of an implementation is non-trivial. Cognitive science lacks appropriate accepted specification tools or languages. Over the years, however, several approaches to specification have been proposed within the cognitive sciences and there are lessons that can be learned from closer examination of these. The approaches include mathematical descriptions, information flow (or box and arrow) diagrams, and formal specification languages. While such approaches tend to be appropriate only for specific classes of theory, more widespread use of specification techniques has the potential to clarify the distinction between model and theory.

Consider first process-oriented theories. Here, information flow diagrams capture a very abstract level of specification, as with Broadbent's (1958) filter model of attention (see Fig. 1). The diagram consists of labeled boxes with arrows indicating the flow of information between them. While the verbal descriptors on the boxes do not specify the precise function (in terms of inputs and consequent outputs) of each box, the diagram still conveys a hypothesis about the organization of attentional subprocesses, and Broadbent was able to reason over the diagram and thereby give accounts of attentional phenomena. While those accounts might be considered weak by modern standards, the phenomena could not be explained within the alternative theoretical approach, behaviorism, that was dominant at the time.

As a specification, the filter model lacks precise descriptions of the functions of its components, but those components can be made more concrete without resorting to a full implementation. One could, for example, specify the interface and processing properties of each component, perhaps supplemented with specific examples of inputs and outputs, or with equations for deriving outputs from inputs. This was effectively the approach taken in describing early models of short-term memory (e.g., Atkinson & Shiffrin, 1968; Waugh & Norman, 1965). More formally, one might adopt mathematical methods to specify the functional decomposition of processes and their interactions, such as combinatorial logic (e.g., Hindley & Seldin, 1986), calculi for concurrent communicating systems (e.g., Milner, 1989),

[1] Actually, the assumption is probably more specific than this. Phonologically similar words presumably need to have similar representations.

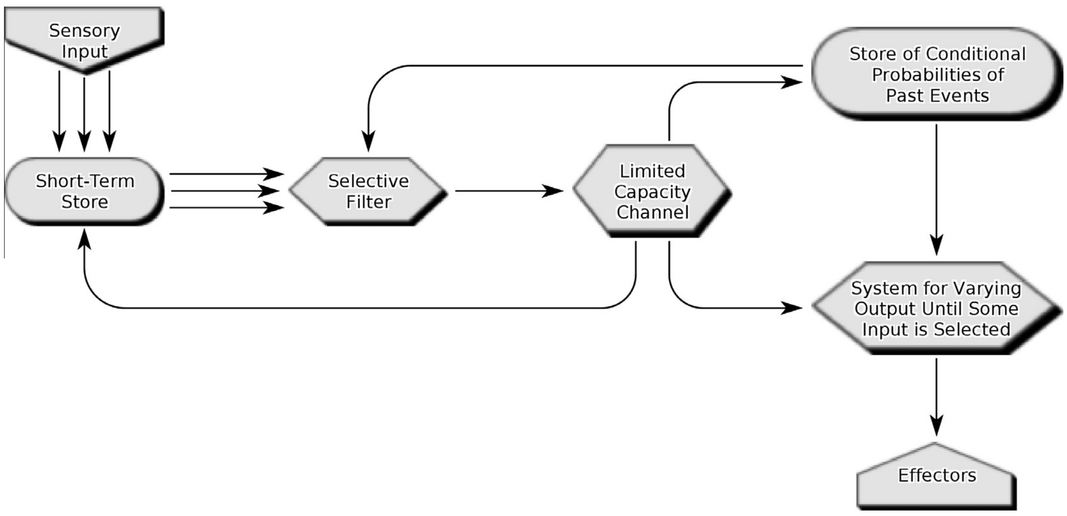
Sensory Input
Short-Term Store
Selective Filter
Limited Capacity Channel
Store of Conditional Probabilities of Past Events
System for Varying Output Until Some Input is Selected
Effectors

Fig. 1. Broadbent's (1958) filter model of attentional processes. (Redrawn from Broadbent, 1958.)

or formal specification languages (e.g., Spivey, 1989). Such approaches have additional benefits, beyond implementation-independent specification of theory. They allow reasoning over theories and proof of properties of theories (e.g., of convergence, or of computational complexity). They also allow one to prove the "correctness" of a specific algorithm (i.e., that an algorithm's output satisfies the specification). While there is increasing use of formal methods in the Bayesian literatures, such approaches remain rare in process-level computational cognitive modeling (though see the analysis of memory retrieval latency and accuracy in ACT-R, by Anderson (2007), and the analysis of connection strengths in interactive activation models, by Mirman, Bolger, Khaitan, & McClelland, in press, for exceptions). Anecdotally, cognitive modelers appear reluctant to invest the time and energy required to develop abstract theory specifications. This may be because it is often faster to develop a model than to specify the underlying cognitive-level theory in rigorous terms, or because the benefits of such specification are not always immediately apparent. We suggest, however, that their increased use within the cognitive sciences would go some considerable way towards the clarification of statements of cognitive theory.[2]

This form of specification would not, however, address the specific representational dispute highlighted by McClelland's (2009) discussion of the past tense model. For this, cognitive science would seem to need a language for characterising properties of representations. The use of natural language suffers from its inherent vagueness, but mathematical approaches may provide a suitably abstract solution.

There is a further problem that abstract theory specification must confront: how can one derive concrete predictions from an abstract specification? One approach is to reason over properties of the specification, as was done informally by Broadbent (1958) and more formally in the context of a well-specified cognitive model by Barnard and colleagues (e.g., Duke, Barnard, May, & Duce, 1995). This kind of reasoning is one of the key motivations for computer science specification languages such as *Z* (Spivey, 1989).

A significant limitation of these formal approaches is the mathematical sophistication required to produce and/or comprehend them, or to reason over them. One minimal step towards improved model description would simply be to distinguish between theoretical assumptions and implementation details in the verbal description of theory (as was done by Cooper & Shallice, 2000, in their model of the control of routine sequential action).[3] Alternatively, one might employ a so-called "executable specification language", which attempts to address this difficulty by providing a formal language that is amenable to mathematical manipulation but also executable.[4] Specifications produced in such languages may be subject to formal analysis to determine their properties, or be executed to determine their predictions. Cooper et al. (1996) proposed the Sceptic language (Hajnal, Fox, & Krause, 1989) as a possible executable specification language for cognitive modeling, illustrating its use with what the authors called a "rational reconstruction" of an early version of the Soar cognitive architecture (Laird, Swedlow, Altmann, Congdon, & Weismeyer, 1989; Newell, 1990). Sceptic allows one to specify processes at a relatively abstract level in terms of rewrite rules (Post, 1943). Thus, one can use the language to

[2] McClelland (2009) also stresses the importance of mathematical methods in computational modeling.

[3] Cooper and Shallice (2000) went further than this. Following Lakatos (1970) they drew a distinction between core and peripheral theoretical assumptions in the verbal description of their model. Implementation

[4] Executable specifications serve as both specifications and implementations. We do not view them as a counter-example to our proposal that "implementations are not specifications", since their primary purpose is specification. That they are executable is an added bonus.

specify, for example, that a process operates on a sorted list, without specifying the algorithm for sorting the list. More generally, like other specification languages Sceptic allows one to specify in logical terms a relation that must hold between input and output representations without specifying how that relation is computed. In the case of the Sceptic specification of Soar, the authors were able to specify that the Soar decision cycle selected the best option based on the available preferences without specifying the algorithm by which preference ordering was done. Given the Soar theory at the time, this seemed to capture the relevant processing assumption of the theory while avoiding a commitment to any specific implementation (though a specific implementation was also offered). The argument here is not that Sceptic is an ideal specification language for describing process models of cognition.[5] Rather, we are suggesting that modelers should embrace some form of theory specification technique that prevents theoretical assumptions from being conflated with implementation details. An added benefit of such deconflation would be the separation of theories from the media in which they are implemented, meaning that discussion can focus on theories rather than on, for example, the strengths and weaknesses of symbolic, connectionist, dynamical or even architectural approaches.

4. Replication

Given a specification and an implementation, how might we determine whether properties or behaviors of the implementation are specific to the implementation or a consequence of the underlying theory (i.e., the specification)? Replication, that is, independent reimplementation, provides an appropriate methodology. In experimental sciences replication is a standard method to ensure that empirical results are robust and not dependent upon specific characteristics of, for example, the lab in which the original experiment was conducted. Failure to replicate, as in recent high profile studies related to cold fusion (Jones et al., 1989) or neutrino transmission at speeds of greater than that of light (Adam et al., 2011), clearly indicate that the original results were flawed, due, perhaps to idiosyncrasies of the experimental apparatus, assumed irrelevant aspects of the initial experimental design, or measurement error. In cognitive psychology this might include characteristics of the specific stimuli or of the specific participants. In cognitive modeling it is likely to include details thought by the originator of a model to be irrelevant.

Replication is infrequently attempted within experimental psychology, though it is becoming more common following widespread concern within some areas (cf. Pashler & Wagenmakers, 2012; Shanks et al., 2013). However, it is even more uncommon within computational cognitive modeling. As Addyman and French (2012) observe, "[a] few diehards will attempt to program the model themselves, based on the verbal description in the article in which it appeared, but they will invariably be stymied by details that the authors left out" (p. 333). Their comments echo those of Lane and Gobet (2003), who argue that "the complexity of computational models can make the task of replication all but impossible" (p. 251).[6] Yet replication has the same role in cognitive modeling as in other sciences. If a published model description fails to replicate (i.e., the reported results are not reproduced by an independent reimplementation of the model based on its published description), then this implies either that some critical aspect of the original model is not adequately specified in the original publication, or that there is a bug in either the original model or the reimplementation. Reimplementation bugs are unlikely given that failed reimplementation will lead to great care in checking the reimplementation code. This leaves the former two possibilities, both of which are critical for scientific progress.

One may take exception to the term "diehards" – contemporary connectionist modeling environments such as Emergent (Aisa, Mingus, & O'Reilly, 2008), PDPTool (McClelland, 2013) and OxLearn (Ruh & Westermann, 2009) make it feasible in some instances for novice modelers to produce replications. In any case, we are amongst the diehards to which Addyman and French (2012) refer. In four major reimplementation attempts we have sometimes, but not always, been stymied by "details that the authors left out". Nevertheless our experience suggests that the reimplementation exercise is an instructive one. For a start, if authors have left out details that prevent reimplementation, then those details would seem to be critical, and it is scientifically important to know those details and understand their role in the model's behavior (i.e., why they are critical). Otherwise the theory behind the model is, at best, incomplete.

The first reimplementation we attempted was the rational reconstruction of Soar version 4.5 discussed above

[5] Indeed, much of our subsequent work has focussed on COGENT (Cooper & Fox, 1998), a domain-specific graphical object-oriented modeling environment in which models are specified in box and arrow terms, with boxes corresponding to objects within a hierarchy of well-defined object types. Boxes can be configured to produce an executable model by specifying relevant properties. COGENT inherits some of Sceptic's specification features. More generally, object-oriented approaches, in which the components of a model are parameterised encapsulated objects, have much to offer the specification of cognitive models of both symbolic and connectionist varieties (see, for example, Aisa, Mingus, & O'Reilly, 2008; Cooper, 2001), just as they have in other disciplines where modeling plays an important role (see, for example, the Unified Modeling Language: Booch, Rumbaugh, & Jacobson, 1999). The specification of a model in an object-oriented connectionist modeling framework should not, however, be confused with the specification of the cognitive theory underlying the model.

[6] To support replication, Lane and Gobet (2003, 2012) propose a methodological approach based on the explicit provision of different types of test case (algorithmic, functional and behavioral) in order to create "reproducible" cognitive models. This allows the researcher(s) developing the replication to more readily determine whether their replication is accurate.

(Cooper et al., 1996). While a Lisp implementation of Soar was freely available at the time, the purpose of this work was to understand the theoretical assumptions of Soar, separate from any specific implementation, and the Lisp code was not consulted during the reimplementation work. The reimplementation was successful in that the rational reconstruction was able to take preprocessed Soar productions and generate output that was indistinguishable from the Lisp implementation. Moreover the reimplementation consisted of a relatively small set of Sceptic rewrite rules (28) which, we argued, captured the essence of Soar version 4.5. Soar has developed considerably since this work (cf. Laird, 2012), but we believe the reconstruction is both more complete than verbal descriptions of Soar 4.5 (e.g., the user manual: Laird et al., 1989) and more comprehensible than the Lisp implementation, even to non-modelers.

Cooper and Shallice (2006) report a replication of Botvinick and Plaut's (2004) simple recurrent network model of the control of sequential action. Botvinick and Plaut argued that their model demonstrated that a recurrent associationist account of sequential behavior could, when impaired through the addition of internal noise, give rise to order errors in the sequential selection of action (e.g., the sequence ABC being generated as ACB, or as AC, or as ABBC). The existence of such errors across a range of human behaviors had, since the arguments of Lashley (1951), been taken to suggest that sequential behavior was controlled by a mechanism in which multiple upcoming action representations were simultaneously activated, with a selection mechanism choosing from the activated representations. The published model description was sufficiently complete to allow reimplementation and replication of all simulations reported in the original work. The key lessons from this reimplementation concerned a clearer appreciation of the strengths and limitations of the model. For example, as a result of the reimplementation work it became clear that order errors within the simple recurrent network model's output were more properly characterized as errors of sequence blends (where one learned sequence blends into another), and hence that the model's ability to produce such errors was a function of its experience with subsequences that were consistent with the error. In other words, to produce a transposition error such as ACBD (when attempting to ABCD), the model must have learned sequences including AC, CB and BD (in addition to ABCD).

More recently, Guest and Cooper (2012) report an attempted replication of the Rogers et al. (2004) model of the representation and organisation of semantic knowledge. The model attempts to account for normal performance on a range of tasks that tap semantic knowledge, as well as the breakdown of performance on those tasks following temporal lobe damage. Certain aspects of the model were not fully described in the original report (such as the precise training algorithm), and reimplementation proved to be troublesome. At the time of writing, we are able to replicate the Rogers et al. results for the fully trained, unimpaired, model, but when lesioned our model does not reproduce the lesioned results reported by Rogers et al., and the reason for this eludes us.[7] Conceivably the issue relates to some under-appreciated aspect of the original implementation. If this is the case then the published verbal description of the theory is incomplete.

Lastly, we have developed a reimplementation of Tyler, Moss, Durrant-Peatfield, and Levy's (2000) model of the representation of semantic knowledge. The model attempts to account for the kinds of category specific semantic deficits sometimes observed in patients following stroke or other organic neural injury. The theoretical claim that the authors advance is that apparent category-specific deficits, where patients show a selective impairment in knowledge of, for example, living things compared to artefacts, are a result of differing patterns of feature correlations within different domains. Living things are held to be characterized by shared functional features that are correlated with shared perceptual features but not with distinctive perceptual features, while artefacts are characterized more by distinctive functional features that are correlated with distinctive perceptual features. Given such representations, it is claimed, neural damage will affect different types of features (i.e., functional or perceptual) differentially for living things and artefacts. The position is encapsulated by a set of six explicit assumptions and supported by a connectionist model (a simple feed-forward auto-associator) that is trained on an abstract set of patterns having the claimed structure. Following varying degrees of damage (modeled by ablating a proportion of connections), the model is shown by Tyler et al. to reproduce the theoretically relevant effects. The basic model is straightforward and, while not all parameters were fully specified, reimplementation was successful and we were able to replicate the six key effects discussed by the original authors.[8] However, the reimplementation demonstrated that the key effects were not all *necessary* consequences of the assumptions. Some effects were only present when parameters that might be assumed to be irrelevant (e.g., the distribution of the values of weights prior to training) took values within specific ranges.[9]

[7] Personal communication with McClelland (January, 2011), Rogers (August, 2012) and Lambon Ralph (November, 2012) has clarified some issues, and McClelland has provided us with an updated version of the model. However, due to changes in input/output representations, the updated version cannot be applied to some of the semantic tasks considered in the original research of Rogers et al. (2004).

[8] In fact, one effect discussed by the Tyler et al. (2000) – the relative preservation of functional features of artifacts as compared to living things following damage to the network – was not fully present in the original published results (cf. Tyler et al., 2000, figure 6). Our reimplementation reproduced the desired effect, suggesting that the failure of the original simulation to produce the effect may have been due to a bug either in the

[9] In these studies the model was trained to the same criterion, regardless of the values of initialisation or other learning parameters. The failure to replicate these effects is therefore not due to differences in baseline network performance.

These four case studies demonstrate that replication in computational cognitive modeling does have value. It is not straightforward but in each of the above cases it led to further insight into the functioning, strengths and limitations of the models (and underlying cognitive theory) concerned. More broadly it addresses the theory/implementation issue by allowing one to explore the relevance or otherwise of putative implementation assumptions.

5. Experimentation

A further way in which the relevance of implementation details may be assessed is through computational experimentation – systematically varying aspects of an implementation to determine their effects on the implementation's behavior. Such sensitivity analyses are relatively rare, but they are potentially very powerful, particularly given that implementation details are necessary for computational completeness. For example, in their study of acquired dyslexia, Plaut and Shallice (1993) explored a space of connectionist models, all of which had four key properties related to representation and learning. Six different network architectures were considered (varying in the topography of the connections between orthographic units and semantic units) and two different learning algorithms (back-propagation through time and contrastive Hebbian learning). All networks with the four key properties were found to exhibit the target effects. This provides a much stronger theoretical account of the domain (in terms of abstract properties, rather than a specific implementation) than is common in computational cognitive modeling.

The Plaut and Shallice (1993) approach to demonstrating sensitivity (or lack of it) reflects extreme diligence on the part of the authors. Systematic variation of parameters is simpler and allows one at least to understand whether a model's behavior is robust to some aspects of implementation. Cooper and Shallice (1997), for example, demonstrated that their interactive activation model of routine sequential action selection, which contained eight numeric parameters, produced well-structured action sequences over a wide range of all parameters. Each parameter was first varied independently to determine its viable range. Five key parameters were then varied simultaneously to determine whether behavior was sensitive to interactions between parameter values. Acceptable performance was recorded over a large segment of the multidimensional parameter space, indicating that the model's behavior was not the product of judicious parameter settings (and thereby defending against the well-known criticism of Roberts & Pashler, 2000, concerning the drawing of inappropriate conclusions following parameter fitting).

A third example of sensitivity analysis is given by the work described above on the reimplementation of the Tyler et al. (2000) model. The model's ability to capture the theoretically critical effects was found to depend on its parameters in unforeseen ways. These sensitivity analyses therefore demonstrate not that the theory is wrong, but that it is incomplete – the critical effects are *possible* given the claimed distributional characteristics of shared and distinctive perceptual and functional features, but they are not *necessary*. Whether the effects follow from the characteristics of the representations also depends on some, as yet unspecified, aspect of the learning mechanism.

Experimentation in the form of sensitivity analyses is therefore a potentially powerful method for demonstrating the independence of a model's behavior on implementation assumptions. Replication supports experimentation by providing researchers beyond the original team with an implementation to explore, but publication of models *a la* Addyman and French (2012) would also allow this – a point which Addyman and French make as part of their second manifesto principle. Our argument is specifically concerned with the role of experimentation and sensitivity analyses in distinguishing the role of assumptions in the generation of model behavior, and the argument holds regardless of the model's adequacy in other respects.

6. Conclusion

We began this article by noting that Addyman and French lament the fact that current modeling practices within cognitive science make it "almost impossible to access, check, explore, re-use, or continue to develop" computational models (Addyman & French, 2012, p. 332). We do not dispute the problem, but neither do we fully embrace their proposed solution, of producing public code with an associated easy-to-use user interface.[10] In our view more attention must be devoted to model specification as an independent exercise to implementation. This may involve greater use of mathematical frameworks, but clear verbal statements of theoretical assumptions (as in Plaut & Shallice, 1993, or to a lesser extent, Tyler et al., 2000) are an important first step. Moreover replication and experimentation are tools that may allow one to distinguish between specification and implementation.

Acknowledgments

We are grateful to Wayne Gray and four anonymous reviewers for comments on early drafts of this manuscript.

[10] Another issue that Addyman and French (2012) do not consider is that as a consequence of continuing changes in technology code needs to be kept up to date. Models deposited in a repository, for example, need to be curated to ensure that they remain functional as computer systems develop. This is non-trivial, and even more of an issue when user interfaces are involved. The model of Cooper and Shallice (1997) was originally developed with Sun's now defunct XView widget toolkit. It was subsequently updated to use version 1 of the GTK+ widget toolkit and more recently version 2 of that toolkit. Version 3 of GTK+ is now available and it is likely that a further upgrade will soon be required if the code is to remain executable as GTK+ 2 becomes less widely available.

References

Adam, T., Agafonova, N., Aleksandrov, A., Altinok, O., Sanchez, P. A., Anokhina, et al. (2011). Measurement of the neutrino velocity with the OPERA detector in the CNGS beam. *arXiv, preprint arXiv:1109.4897*.

Addyman, C., & French, R. M. (2012). Computational modeling in cognitive science: A manifesto for change. *Topics in Cognitive Science, 4*, 332–341.

Aisa, B., Mingus, B., & O'Reilly, R. (2008). The emergent neural modeling system. *Neural Networks, 21*(8), 1146–1152.

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

Atkinson, R. C., & Shiffrin, R. M. (1968). Human memory: A proposed system and its control processes. *The Psychology of Learning and Motivation: Advances in Research and Theory, 2*, 89–195.

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The unified modeling language user guide*. Addison-Wesley.

Botvinick, M., & Plaut, D. C. (2004). Doing without schema hierarchies: A recurrent connectionist approach to normal and impaired routine sequential action. *Psychological Review, 111*, 395–429.

Broadbent, D. E. (1958). *Perception and communication*. London, UK: Pergamon Press.

Cooper, R. P. (2001). The role of object-oriented concepts in cognitive models. *Trends in Cognitive Sciences, 5*, 333.

Cooper, R. P., & Fox, J. (1998). COGENT: A visual design environment for cognitive modeling. *Behavior Research Methods, Instruments, and Computers, 30*, 553–564.

Cooper, R. P., Fox, J., Farringdon, J., & Shallice, T. (1996). A systematic methodology for cognitive modeling. *Artificial Intelligence, 85*, 3–44.

Cooper, R. P., & Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive Neuropsychology, 17*, 297–338.

Cooper, R. P., & Shallice, T. (2006). Hierarchical schemas and goals in the control of sequential behavior. *Psychological Review, 113*, 887–916.

Cooper, R. P., & Shallice, T. (1997). Modeling the selection of routine action: Exploring the criticality of parameter values. In M. G. Shafto & P. Langley (Eds.), *Proceedings of the 19th annual conference of the cognitive science society* (pp. 131–136). Lawrence Erlbaum Associates.

Duke, D. J., Barnard, P. J., May, J., & Duce, D. A. (1995). Systematic development of the human interface. In *Proceedings of APSEC'95: Second Asia-Pacific software engineering conference* (pp. 313–321). IEEE Computer Society Press.

Gray, W. D. (2010). The shape of things to come: An emerging constellation of interconnected tools for developing the right cognitive model at the right scale. Keynote address delivered to the Behavior Representation in Modeling and Simulation Conference (BRiMS). Charleston, SC.

Guest, O., & Cooper, R. P. (2012). Semantic cognition: A re-examination of the recurrent network "hub" model. In N. Rußwinkel, U. Drewitz, & H. van Rijn (Eds.), *Proceedings of the 11th international conference on cognitive modeling* (pp. 259–264). Germany: Technische Universität Berlin.

Hajnal, S., Fox, J., & Krause, P. (1989). The Sceptic user manual. *Advanced computation laboratory technical report, Imperial Cancer Research Fund, London*.

Hindley, R. J., & Seldin, J. P. (1986). *Introduction to combinators and λ-calculus*. Cambridge, UK: Cambridge University Press.

Jones, S. E., Palmer, E. P., Czirr, J. B., Decker, D. L., Jensen, G. L., Thorne, J. M., et al. (1989). Observation of cold nuclear fusion in condensed matter. *Nature, 338*(6218), 737–740.

Laird, J. E. (2012). *The soar cognitive architecture*. Cambridge, MA: MIT Press.

Laird, J. E., Swedlow, K., Altmann, E., Congdon, C. B., & Weismeyer, M. (1989). Soar user's manual: Version 4.5, Tech. Rept., University of Michigan, Ann Arbor, MI.

Lakatos, I. (1970). Falsification and the methodology of scientific research programmes. In I. Lakatos & A. Musgrave (Eds.), *Criticism and the growth of knowledge* (pp. 91–196). Cambridge, UK: Cambridge University Press.

Lane, P. C. R., & Gobet, F. (2003). Developing reproducible and comprehensible computational models. *Artificial Intelligence, 144*, 251–263.

Lane, P. C. R., & Gobet, F. (2012). A theory-driven testing methodology for developing scientific software. *Journal of Experimental and Theoretical Artificial Intelligence, 24*, 421–456.

Lashley, K. S. (1951). The problem of serial order in behavior. In L. A. Jeffress (Ed.), *Cerebral mechanisms in behavior* (pp. 112–131). New York: Wiley.

MacWhinney, B., & Leinbach, J. (1991). Implementations are not conceptualizations: Revising the verb learning model. *Cognition, 40*, 121–157.

McClelland, J. L. (2009). The place of modeling in cognitive science. *Topics in Cognitive Science, 1*(1), 11–38.

McClelland, J. L. (2013). *Explorations in parallel distributed processing: A handbook of models, programs, and exercises* (2nd ed.). <http://www.stanford.edu/group/pdplab/pdphandbook/> Retrieved 18.03.13.

McCloskey, M. (1991). Networks and theories: The place of connectionism in cognitive science. *Psychological Science, 2*(6), 387–395.

Milner, R. (1989). *Communication and concurrency*. New York: Prentice Hall.

Mirman, D., Bolger, D. J., Khaitan, P., & McClelland, J. L. (in press). Interactive activation and mutual constraint satisfaction in perception and cognition. *Cognitive Science* (in press).

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

Newell, A., Shaw, J. C., & Simon, H. A. (1958). Elements of a theory of human problem solving. *Psychological Review, 65*, 151–166.

Pashler, H., & Wagenmakers, E. J. (2012). Editors' introduction to the special section on replicability in psychological science: A crisis of confidence? *Perspectives on Psychological Science, 7*(6), 528–530.

Pinker, S., & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition, 28*, 73–193.

Plaut, D. C., & Shallice, T. (1993). Deep dyslexia: A case study of connectionist neuropsychology. *Cognitive Neuropsychology, 10*, 377–500.

Post, E. (1943). Formal reductions of the general combinatorial decision problem. *American Journal of Mathematics, 65*, 197–215.

Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. *Psychological Review, 107*, 358–367.

Rogers, T. T., Lambon Ralph, M. A., Garrard, P., Bozeat, S., McClelland, J. L., Hodges, J. R., et al. (2004). Structure and deterioration of semantic memory: A neuropsychological and computational investigation. *Psychological Review, 111*(1), 205.

Ruh, N., & Westermann, G. (2009). OXlearn: A new MATLAB-based simulation tool for connectionist models. *Behavior Research Methods, 41*(4), 1138–1143.

Rumelhart, D. E., & McClelland, J. L. (1986). On learning the past tenses of English verbs. In J. L. McClelland & D. E. Rumelhart (Eds.). *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. II, pp. 216–270). Cambridge, MA: MIT Press.

Shanks, D. R., Newell, B. R., Lee, E. H., Balakrishnan, D., Ekelund, L., Cenac, Z., et al. (2013). Priming intelligent behavior: An elusive phenomenon. *PLoS One, 8*(4), 1–10.

Spivey, J. M. (1989). *The Z notation: A reference manual*. New York: Prentice Hall.

Sun, R. (2009). Theoretical status of computational cognitive modeling. *Cognitive Systems Research, 10*(2), 124–140.

Tyler, L. K., Moss, H. E., Durrant-Peatfield, M. R., & Levy, J. P. (2000). Conceptual structure and the structure of concepts: A distributed account of category-specific deficits. *Brain and Language, 75*(2), 195–231.

Waugh, N. C., & Norman, D. A. (1965). Primary memory. *Psychological Review, 72*(2), 89–104.