Cognitive architectures need compliancy, not universality

Richard M. Young

Psychology Department, University of Hertfordshire, Hatfield, Herts AL10 9AB, United Kingdom. **r.m.young@herts.ac.uk**
http://www.psy.herts.ac.uk/pub/r.m.young/

Abstract: The criterion of computational universality for an architecture should be replaced by the notion of compliancy, where a model built within an architecture is compliant to the extent that the model allows the architecture to determine the processing. The test should be that the architecture *does easily* – that is, enables a compliant model to do – what people do easily.

Anderson & Lebiere (A&L) are to be congratulated on advancing the agenda of assessing cognitive architectures (or other cognitive theories of broad scope) as a whole. The inspiration is clearly Newell's, but the authors take a major step towards bringing Newell's criteria down to earth by operationalising them and bringing them closer to objective criteria and tests. This commentary is offered as a minor contribution to that same goal.

In section 2.1, A&L follow Newell in identifying the criterion of *flexible behavior* with computational universality, that is, equivalence to a Turing machine. But Turing computability is inappropriate as a criterion for cognitive architectures. It is by nature an all-or-nothing test: Can, or cannot, the architecture be programmed to compute any Turing-computable function, yes or no? The authors certainly do themselves no favours by adopting Turing universality as the touchstone for flexible behaviour. Indeed, it forces them into a contradiction. Although in section 4.5 they deny that "ACT-R is a general computational system that can be programmed to do anything," that is indeed what being Turing universal means, that the architecture can be "programmed to do anything." What is needed instead is a graded measure, reflecting the reality that, as A&L acknowledge, "some things are much easier for people to learn and do than others." Ideally, the architecture should learn and do easily the things that people learn and do easily and, similarly, find the same things difficult.

Of course, what is meant by an architecture doing or learning something *easily* itself needs careful definition and explication. It is no trivial matter to replace the all-or-nothing concept of Turing computability by a more appropriate measure that both captures and makes precise these important but rather vague ideas about "doing something easily" or doing it by means "in keeping with the spirit of an architecture." However, a start has been made, with the concept of the *compliancy* of models constructed within a cognitive architecture. The idea has been worked through most thoroughly for SOAR, but is applicable in principle to any cognitive architecture.

In Howes and Young (1997), we approach the issue by considering how in practice architectures are used by cognitive modellers, and how credit and blame for the resulting models get assigned in the light of agreement with empirical data (or other evaluative criteria). We note how, in applying an architecture to a particular domain or task, the modeller inherits all the theoretical commitments of the architecture and then adds further commitments, often expressed in the form of production rules, which are specific to the domain or task being modelled. We refer to these additions as *model increments*, by analogy with the *method increments* which Laird (1986) identifies as giving rise to the "weak methods" of problem solving. We are led thereby to pose a methodological question: Given that a model (of this kind) consists of a cognitive architecture together with a specific model increment, in cases where the model does well, that is, provides a good prediction and explanation of the data, where does the credit belong: to the architecture, to the model increment, or somehow to both? And likewise if the model does badly, where lies the blame?

We note too that the extent to which cognitive architectures constrain and shape the models constructed within them, and thereby contribute to their predictions, is not widely recognised by those without first-hand experience of such architectures. Building model increments is not at all like writing programs in a theoretically neutral programming language. An architecture is not simply a programming environment for constructing cognitive models according to the modeller's fancy. Indeed, some architectures, of which SOAR (Newell 1990) is an example, are themselves capable of generating behaviour once they are given a specification of the task to be performed, even without further information about *how* it is to be performed. In such cases, the role of the model increment becomes not so much to generate behaviour, as to modulate or modify the behaviour which the architecture is already advocating.

That observation leads us to introduce the idea of compliancy. A model increment is compliant to the extent that it follows the architecture's lead, that is, takes advantage of the architecture's own tendency, allowing it mostly to do what it wants, intervening just occasionally to keep it on track. A model increment with low compliance, by contrast, disregards or overrules the architecture's own tendencies and simply forces the architecture to do what the model increment wants. (If the architecture is indeed Turing universal, then a model increment can always be written to produce any specified behaviour, but the increment may have to fight against the architecture in order to achieve that behaviour.)

The notion of compliancy allows us to answer the question about credit assignment. To the extent that the model increment is compliant with the architecture, much of the credit or blame attaches to the architecture itself. But to the extent that the model increment is noncompliant, responsibility for the resulting behaviour, whether good or ill, rests mostly with the model increment.

My suggestion is that compliancy also offers promise as a route for explicating what it means for an architecture to perform a task easily or with difficulty. An architecture can be said to find a task easy if a compliant model increment suffices to build a model to do it. Contrariwise, the architecture finds a task difficult if a noncompliant model increment is required, which therefore has to "force" the architecture to do it in a way "not in keeping with its spirit." By utilising compliancy, Newell's criterion of flexible behaviour can be interpreted as a requirement that the architecture does or learns easily (in other words, enables a compliant model to do or learn) what people find easy to do or learn, and finds difficult (in other words, requires a noncompliant model to do or learn) what people find difficult.

Authors' Response

Optimism for the future of unified theories

John R. Anderson and Christian Lebiere

Department of Psychology, Carnegie Mellon University, Pittsburgh, Pa 15213.
ja@cmu.edu **cl@andrew.cmu.edu**

Abstract: The commentaries on our article encourage us to believe that researchers are beginning to take seriously the goal of achieving the broad adequacy that Newell aspired to. The commentators offer useful elaborations to the criteria we suggested for the Newell Test. We agree with many of the commentators that classical connectionism is too restrictive to achieve this broad adequacy, and that other connectionist approaches are not so limited and can deal with the symbolic components of thought. All these approaches, including ACT-R, need to accept the idea that progress in science is a matter of better approximating these goals, and it is premature to be making judgments of true or false.

https://doi.org/10.1017/S0140525X03450139 Published online by Cambridge University Press