CASYS2005 talk

General Principles for Brain Design

Brian D. Josephson
Department of Physics, University of Cambridge,
Cambridge CB3 0HE, UK.

bdj10@cam.ac.uk
http://www.tcm.phy.cam.ac.uk/~bdj10

Thanks to:

Hermann Hauser
David Blair
Nils Baas
Andrée Ehresmann

What would count as ‘explaining the brain’?

models of important phenomena required
including enough detail, or we can’t correlate architecture with capabilities

would like to be able to discuss e.g. language in action

problem of complexity

possibility of ‘general understanding’, supported by many specific examples: e.g. chemical reactions fundamental to biosystems

similarly, general understanding of working of computers

•explanation = principles + details

‘Mechanisms that work’

How do ‘mechanisms that work’ and ‘mechanisms that don’t work’ differ?

In general, only details tell
But we can still understand underlying principles

re ‘general understanding’ for the brain, our main line of argument is:

- Start with idea that mechanisms that help computer programs work effectively are relevant
- Adaptation to neural hardware
- Additional factors relevant in brain context

Key principles:

1. Programs make use of hierarchy of functions; relevant *local models* enable sequential checking of code (i.e. that it achieves aims) in a bottom-up manner

2. Classes + objects make programs more straightforward (economy):
•basic models are extended in scope through use of parameters
•models encompass collection of functions ('methods') in a class and variables, ensure that they work together properly, each being informed about the actions of all the others
•example of coordination: airline reservation program: the class reservation (collection of reservations in system) AcceptPayment sets PaymentMade, IssueTicket reads it.

- parameter sets for a given situation/entity are stored in their own block of memory

- new blocks allocated whenever a new instance of a particular class is encountered

- the identical code, applied to the block of memory (object) appropriate to the given instance of the class, defines the way a given function is implemented for that instance

How to adapt to nervous system?

[NB: Minsky worked with the idea of designing networks equivalent to programs, but did not discuss in a fruitful way the fundamental principles needed for the idea to work effectively]

Start from source code
•Arrangement of symbols that indicates what should happen when program is run
•Compiler translates it into code that instructs the computer's microprocessor
•Source code explains, executable code implements

For the NS, we would need something similar to source code to characterise and explain behaviour, plus corresponding implementation in hardware

Source code language involves ideas, which need to be made explicit in the translation

Representation is the key here. In computing:

•data described by source code is linked to a physical representation, in accord with specific rules

•processes are defined, whose effect is equivalent to the source code directives

In the NS case: again, specific links of various kinds between entities in a model and neural systems
•precise details in general not innate
•hence use ‘dynamic allocation of memory/dynamically organised representation’

A range of features is needed:

- memory block > neural system, with organisational (formatting) conventions
- content > local parameter or connection (in cases of value and address type content resp.)
- processing done by neurons using signals to represent information, with specific circuits for specific processes
- each variable has its own system and signals
- classes are associated with different neural regions (different things happen for different classes, processes need to be kept separate from each other)
- learning rather than preprogramming
- more need to cope with problems
- subject to such provisos, programming concepts carry over

learning

both biological processes and typical environments display more variability than with artificial mechanisms, hence need to learn by feedback rather than everything being innate/preprogrammed

•our assumption is that in a given class of situations learning can be modelled in a uniform way; here preprogramming is possible
•learning involves very specific adaptations, doing very specific things
•model of learning should reflect the abilities of actual neural nets
•learning = learning of relevant relationships (recalling is a part of an activity)

•specialised network (initiator for the type) controls learning process, looking for and staying in situations of desired type, assigning memory/neural system and making interconnections to mediate processes associated with class (*can we identify these?*)

Effect of successful learning: applicability of a new local model reflecting success (Baas hyperstructure concept, cumulative learning)

(effect of failure: inhibition of exploration in the given context)

repair

much of the design is concerned with good ways of handling
situations where the default process fails
(example: use of language to convey missing information)
or simply to develop a process that one has not yet developed

Top down aspect

In biological realm, high-level behaviour is important (natural selection)

source code defines whole process in terms of parts; parts in terms of subparts … [great fleas have little fleas …]

developing systems find the parts that fit the requirements of the various models

problem-solving makes the parts that will construct the whole from a specification of the whole (top-down aspect)

Inhibition and excitation mechanisms

Excitatory connections: activate what we know

Inhibitory connections: block what we do not want to know; inhibit till what is left is does the job

Features of biological complexity (plus creativity):

•case of biology generally: many different types of parts (biomolecules)
•but number of significant processes is many fewer than theoretically possible with this number of parts
•biodesign selects preferentially subset of processes of value in regard to overall scheme

•similarly, with NS, there are very many parts but particular circuit possibilities, having value in regard to overall scheme, are preferred

Creativity

activities in general develop through well-specified steps
model-justified activities develop the steps

e.g. steps to skill of walking
new possibilities are achieved by systematic organisation of existing means, as models specify and networks carry out

'whatever is found and is good' is added
the existence of good things (of a given kind) makes this work

Cognitive effectiveness depends on *reliable subsystems* having been developed.

Such systems have specific mathematical properties (e.g. a phrase being able to define an object) that can be concatenated to explain effectiveness of the whole (good parts make a good whole).

[the issue defining goodness is, does the physical system behave in accord with the associated model?]

Levels of abstraction

Karmiloff-Smith argues for different (discrete) levels of abstraction with different roles. This fits well with our scheme

illustration: special processes with regard to the abstraction of classification:

- start with class with just one member
- characterise its properties
- try to extend the class

as AI tells us, these are very specific procedures

appropriate procedures are needed to develop rapidly

Research strategy

We have a very complicated situation, as is typical in biology

•as in biology generally, we do not have to find a total solution, but can find bits at a time

•we may get clear insights into language, planning, etc., by seeing how well 'specific classes of situation handled by specific means' fits the reality; the observed complexity may be comprehensible as a cumulation of specific elaborations (cf. Jackendoff's evolutionary approach, or Arbib's).

•e.g. 'this use of language develops out of this one, thus' (e.g. given sign-meaning relationships, develop applications of strings of signs); 'one property leads to another'.

•Finding the specific models and mechanisms will lead to a precise theory, a precise design like any precise human design.

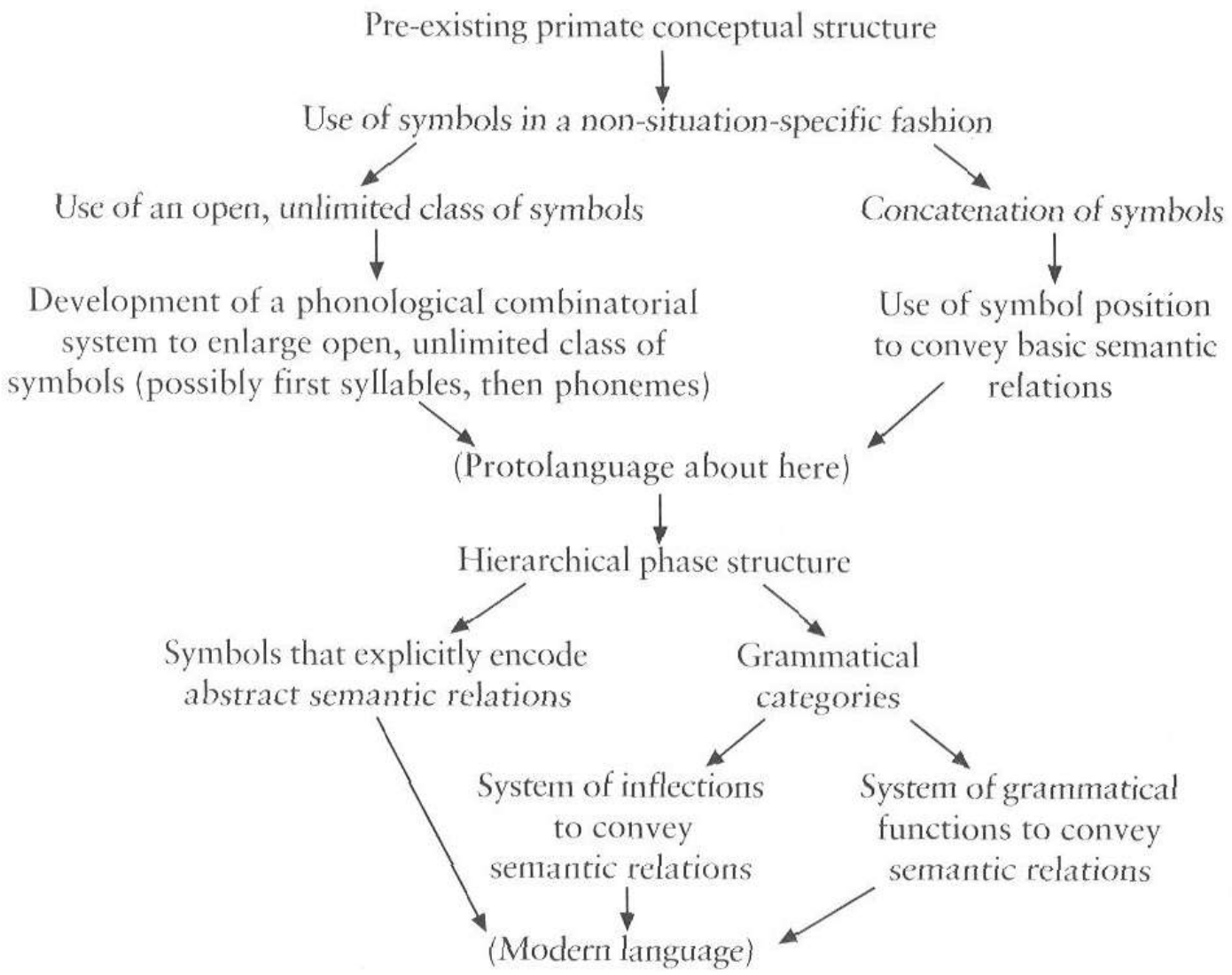

Fig. 8.1. Summary of incremental evolutionary steps

From Jackendoff, *Foundations of Language*

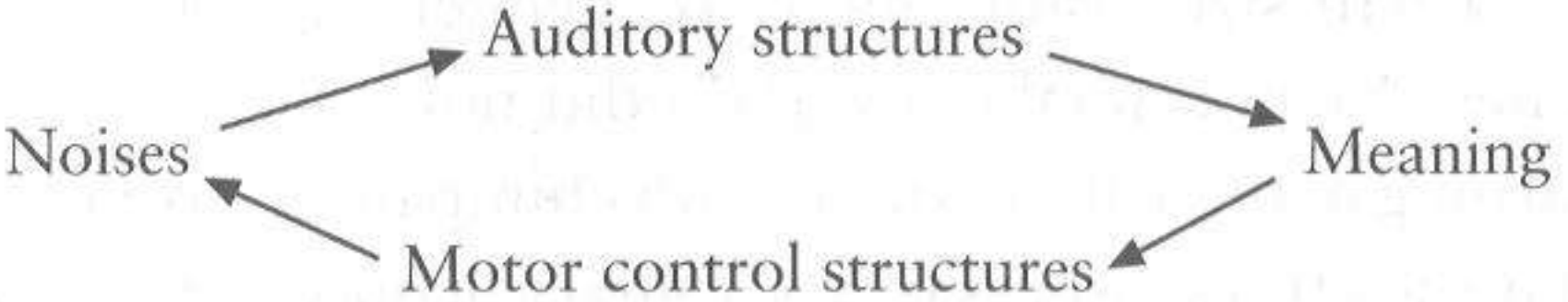

Fig 8.2. Architecture of early single-symbol stage

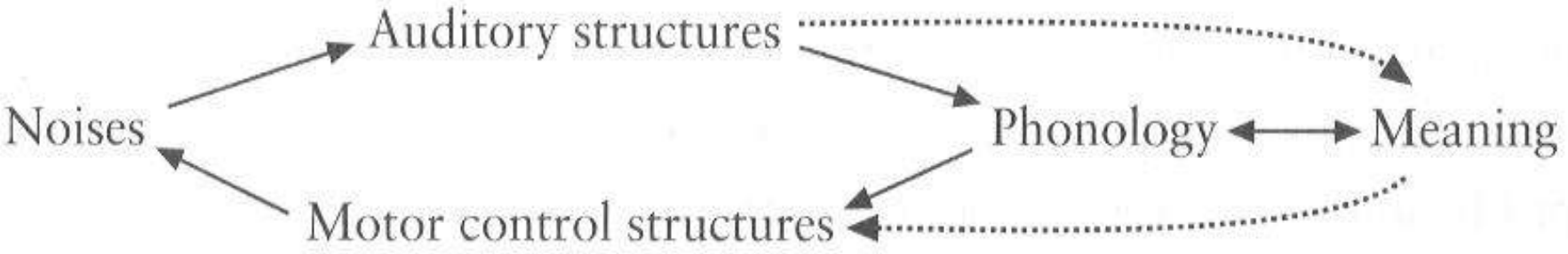

Fig. 8.3. Architecture of protolanguage/Basic Variety

The nature of explanations in this picture:

•it is like any explanation of a mechanism, computer program, etc.

•we describe all the pieces at the quasi-source code level, reasoning about what they do, explaining their implementation in terms of circuitry
•the (essentially mathematical) accounts at source code will encompass all the types of things that are involved, e.g. in language and applications of language, relating their coming into being in terms of specific processes and mechanisms

•not ‘multilayer networks’, but layers of specialised mechanisms for creating functionality as the various models permit.

Example of abstract model: the hierarchy/tree structure

Hierarchies (with parent child relation, with each child having only one parent) are implicit in many contexts

In a computer program, we'd define a hierarchy/tree class, and various variables and operations characteristic of members of the class.

Operations with this type would be done via these variables and operations.

Arguably, such specifics can all be translated into neural circuitry, giving the NS an ability to 'work with trees', parallelling that of computer programs, which could evolve over time and, e.g., be put to use ultimately by the language system.

Further example relevant to language: primitive distinctions between types of signs (e.g. between signs pointing to things and signs indicative of actions) can be implemented in the hardware, and the corresponding circuitry adapted to very different uses: hence aspects of UG.

UG would thus be linked to very specific mechanisms

Categories?

Categories are specific collections of ‘objects’ and ‘interconnections’ (morphisms)

- special patterns and corresponding inferences
- used by Ehresmann and Vanbremeersch to characterise systematics of brain processes

Comments:

- can be useful adjunct to conventional analyses but cannot replace them (cf. group theory)
- nice maths, but, like string theory, too abstract to reflect the reality accurately?
- More reliable to base thinking on tried and tested technology (programming)?

The most common approach says “this is the network, and this is what it does”, without explaining why it does (the pieces do particular things, but it this enough to explain the whole?).

Our approach asserts that there is a logical explanation for the brain, with two parts, description and implementation. By hypothesising that the brain does it in ways that parallel the ways that computers do it, but also addressing those features that are specific to the brain, and the specific issues it has to face, the approach shows how the parts can fit together to make the working whole expert in complicated ways in different areas.