Architectural Constraints Imposed by Functional Complexity: Implications for the structure and operation of the brain

L. Andrew Coward
School of Information Technology
Murdoch University
Perth, Western Australia
landrewcoward@shaw.ca

Major themes: Robotic and computational models of interaction and computation; Memory; Heuristic learning

A functional system can be defined as a system which acts upon itself and its environment in order to achieve target states (or objectives) for itself and that environment. In a functionally complex system, appropriate behaviour for achieving objectives is determined under tight time constraints by a complex interaction within a high volume of information about current and past environmental conditions, past behaviours, and the internal state of the system.

As the functional complexity of the tasks performed by a system increases, and if the available information recording and processing resources are not unlimited, any need to change functionality without introducing undesirable side effects becomes a severe constraint on the architectural form of the system [3].

Firstly, the functionality of the system will separate into modules, the modules into submodules and so on down to the most detailed operational elements. However, if the functionality is sufficiently complex, resource limitations will mean that it is impractical to achieve a one to one correspondence between modules on any level and system features. Rather, modules will be defined in such a way that any one feature requires the activity of as few modules as possible, and any one module participates in as few features as possible. An equivalent way of stating this is that modules must be defined so that the information exchange needed to coordinate their different functionalities is minimized as far as possible within resource constraints.

Secondly, the system architecture must ensure that adequate meanings are maintained for all information exchanges between modules. From the point of view of the source module, an element of information indicates that some information condition has been detected by that module. From the point of view of the recipient module, the detection of a condition is interpreted to mean that the currently appropriate system behaviour is within a specific subset of the range of behaviours influenced by that module, or not within that range at all. The detection of a condition by one module will therefore have different functional meanings for each recipient module.

The multiple functional meanings assigned to individual elements of information, made necessary by the need to limit information recording and processing resources, is at the root of the difficulty of making functional changes without side effects. If changes to a condition are made for one functional purpose, the change will introduce side effects into all the other functions also making use of the condition.

Thirdly, functional meanings can be of two different types. In one type, recipient modules interpret inputs as limiting the range of behaviours with 100% confidence. Such information exchanges are called unambiguous and can be interpreted as system commands. Detected conditions will correspond precisely with behaviourally significant features and categories. The alternative is that interpretation of inputs is probabilistic, indicating only that one subset of the behaviours influenced by the module is more likely to be appropriate than other behaviours. These information exchanges are called ambiguous and can only be interpreted as system recommendations. None of the conditions detected within such a system will correlate unambiguously with features or categories [1]. Different types of meaning cannot be present in the same system, because with a complex pattern of information exchange the presence of some ambiguous information will render all information ambiguous. Use of ambiguous information to manage the same functionality will require more resources in order to generate a range of recommendations in response to each input state and select the strongest.

Commercial electronic systems use unambiguous information, as can be seen from the instruction syntax of software (e.g. x = a ifTrue: [do: ….]). The need to maintain unambiguous contexts forces such systems into the memory, processing separation often called the von Neumann architecture. If a functional change to an existing function is required, some module or modules must change the conditions detected. In commercial electronic systems using unambiguous information exchange, the effects of such changes are traced intellectually, with the aid of extensive regression testing of all other functions. A major problem is that correction of one side effect may itself introduce additional side effects. Heuristic modification of functionality (i.e. learning) would mean that all side effects would need to be corrected, primarily using consequence feedback. Given the system command interpretation of all information exchanges, such a process would be very unlikely to converge. This impracticality is confirmed by the failure to implement learning in any non-trivial von Neumann system.

If information exchanges are interpreted as recommendations and any accepted behaviour is supported by multiple recommendations, the change to one condition will have less effect on system behaviour. Learning in such a system is therefore possible. However, even if information is partially ambiguous, an adequate level of meaning must be maintained. The need to support an adequate level of meaning in a functionally complex system which heuristically defines its own functionality forces a separation between a modular hierarchy called *clustering* in which modules define and detect portfolios of conditions and indicate the presence of a condition within a portfolio by producing an output, and a *competition* subsystem which uses supervised learning and/or consequence feedback to interpret each condition detected by clustering as a range of behavioural recommendations and to select one recommendation as current system behaviour [1; 4].

In order to maintain adequate meanings, in general once a module on any level has generated an output in response to a condition, an exact repetition of the same condition will always result in a module output including the same output as originally. In other words, the portfolio of conditions detected by a module can expand but not contract. A new condition defined by a module must be similar to existing conditions in the portfolio. Decisions on whether and where to record additional conditions must be carefully managed to prevent excessive use of resources and/or dilution of functional meaning. This change management function is itself functionally complex and must be managed by the modular hierarchy.

The need to define and detect conditions and to use detections to manage both change and external behaviour forces clustering into specific forms. Devices which define and permanently record specific combinations of their inputs are arranged in layers to ensure that conditions are detected in a roughly synchronous set of system inputs. A modular hierarchy of columns and areas is superimposed on these layers. Within a column, some layers generate outputs to competition, others excite or inhibit recording of additional conditions in the column and/or in other columns. This structure strongly resembles the cortex. The need to limit the distribution of information forces competition into three modules which select recommendations determining the sensory domain from which clustering will take inputs, the general type of behaviour, and the specific behaviour within the general type. This separation strongly resembles the thalamus, basal ganglia and cerebellum structures in the brain [3].

Because information is ambiguous, it will not correspond with, for example, cognitive features or categories. There are no structures or activations which correspond with representations of external features or categories within clustering. The response within clustering to the perception of an object will be activation of a population of ambiguous information conditions, some of which were recorded during the presence of other similar objects in the past, and a (generally small) proportion will be recorded at the time of perception. The greater the novelty of the object, the greater the recording of additional conditions. The familiarity of a perceived object can be determined from the degree of recording.

Because information is recorded relatively permanently at the device (or neuron) level in clustering, mechanisms can be established which activate large subsets of the populations corresponding with objects not currently present. Such secondary populations activate conditions often present in the past at the same time as currently present conditions. The functional effect is to supplement the range of recommendations within which behaviour can be selected to include recommendations on the basis of the kind of objects present in the past at the same time as the currently perceived object [2]. The particular set of conditions making up such a secondary population could even correspond with an object never actually perceived.

An implemented electronic version of a system with the recommendation architecture has demonstrated memory and learning phenomena analogous with human cognition, including the activation of behaviourally relevant "mental images".

References

[1] Coward, L. A. (1990). *Pattern Thinking*, New York: Praeger.

[2] Coward, L. A. (1999). A physiologically based theory of consciousness, in Jordan, S. (ed.), *Modeling Consciousness Across the Disciplines*, 113-178, Maryland: UPA.

[3] Coward, L. A. (2000). A Functional Architecture Approach to Neural Systems. *International Journal of Systems Research and Information Systems*, 9, 69 - 120.

[4] Coward, L. A. (2001). The Recommendation Architecture: lessons from the design of large scale electronic systems for cognitive science. *Journal of Cognitive Systems Research* 2(2), 111-156.