A Modern Integration of Cognitive and Computer Sciences

G. Susanne Bahr, Matthew G. Bell, Jason Metz, Sarah Sowle, and Elizabeth Beasley

Florida Institute of Technology, Melbourne, Florida 32901, U.S.A.
gbahr@fit.edu

Abstract. Cognitive and Computer sciences have a long history of shared concepts and shared terminology. This paper explores a radical way of interdisciplinary thinking that ventures beyond loosely modeled metaphorical applications of computer systems and the use of terminology with mere face validity. Our focus is on interdisciplinary conceptual, structure and process commonalities. We provide an example of the discovery of shared concepts, knowledge structures and a common mental model using semantic memory organization in humans and object oriented programming, in particular the principle of inheritance. We discuss whether JAVA applications forget and suggest further research topics.

Keywords: Cognitive Science, Computer Sciences, Interdisciplinary, Crosscutting, Common Concepts, Hierarchical Network Model, Inheritance, Memory, Forgetting.A Modern Integration of Cognitive and Computer Sciences.

1 Brief Historical Context

The concept of the "brain as a computational system" has been part of popular culture for decades. See for instance the 1960s U.S. television series Star Treck and its alien but human like, Vulcan life form Mr. Spock. His persona became exemplary for equating cognitive processing with computational system's activity: In [22] Captain Kirk comments, 'You'd make a splendid computer, Mr. Spock.', to which a flattered Spock replies, 'Thank you, Captain!' Ten years later, in the classic British television series "The Sweeney" [15] and "Open all hours" [5], the characters refer to *human* cognitive activity as "running things through the biological computer".The human mind as an information processing system whose performance is measureable dates back to the Dutch physiologist, and perhaps the first psychometrician, Franciscus Cornelius Donders (1818–89) [12]. Donders measured reaction times and decision making latencies, and interpreted them as processing times of human cognition [12]; Based his model of additive processing times he developed the subtraction method which allowed him to compute the time it takes to make a decision about visual stimuli; (For contemporary evidence see [14].) Similarly Wilhelm Wundt, founder of the first psychology laboratory (1879), measured awareness (attention) as a serial process using his thought meter paradigm [12], [17]. The core research concepts of information processing and their emphases on understanding mechanisms and measuring performance were displaced temporarily in the first half of the 20th century by behaviorist

C. Stephanidis (Ed.): Universal Access in HCI, Part I, HCII 2009, LNCS 5614, pp. 441–449, 2009.

dogma, which delegated perception and cognition to the black box. The renaissance of cognitive sciences began in the 1950s with a re-evaluation of human information processing in parallel with the growing discipline of computer sciences [3]. Regardless of one's acceptance of the computer metaphor [9], [16], the synergy between cognitive psychology and computer sciences acted as a catalyst to advance our understanding of human cognitive performance: For instance, Miller's seminal paper on short term memory provides us with generalized finding about its capacity [20], [34]; Broadbent's work established the serial nature of attention [4] which later informed Simon's seminal work on attention economics [1]; and Baddeley's model of working memory discovered the modular nature and parallel processing of distinct modality inputs in working memory [1], [2].

2 Surface Commonalities: A Not so Common Vocabulary

Based on the shared concept of information processing, the analogy between neurological webs and logic circuits has guided the development of constructs, measurements and vocabulary in cognitive science. For instance, cognitive psychologists have been generously employing the terms of input, output, storage, capacity, modules, buffers, primary memory, secondary memory, prototype, etc. While one might surmise that word sharing provides a basis for interdisciplinary integration, the operational definitions of the same terms in computer and cognitive science vary and potentially lead to mutual confusion. For example in 2007, Unsworth and Engle [34] introduced the terms primary and secondary memory to investigate individual differences in working memory capacity. They refer to primary memory as an active workspace that holds approximately four items that can be replaced by incoming information (new input). Items in primary memory can also be replaced by retrieving items from secondary memory which is considered permanent, long term and searchable. Their terminology is derived from primary and secondary memory or storage in computers: Here, primary memory refers to a temporary and quickly accessible data that is directly linked to the central processing unit [21]; secondary memory is often referred to as *storage* and it is not directly accessible by the central processing unit. Data access and retrieval are comparatively slower than accessing data in primary memory. Unlike primary memory, secondary storage is not volatile: one does not lose one's data when the device is powered down [18]. The similarity between cognitive and computer science uses of these memory related terms is at the surface level and lacks conceptual integration which is necessary for interdisciplinary exchange.

Another example of cosmetic use of computer science derived terminology is Baddeley's [2] use of the term *buffer* for the concept of episodic buffer. The episodic buffer links information across long term memory and working memory module to form integrated units of visual, spatial, and verbal information with chronological ordering; This use of buffer coveys more sophisticated functions than that the computer science buffer concept, which is limited to data being held temporarily after they have been retrieved from an input device (such as a mouse) or before they are sent to an output device (such as a printer) [18]. Data integration at the buffer level is not accomplished but the passing of data between different units that require synchronization.

This section has been intended to pique interdisciplinary awareness and to evaluate the utility of cosmetic word usage. So far the state of the sciences seems to indicate that modern commonality is limited to the same words pointing to different models and definitions. However, further examination in the next section provides an example of an existing, common conceptual architecture that has given rise to a divergent terminology.

3 Below the Surface: Common Concept

This section provides an example of a modern integration based on the commonality of two mature areas in cognitive and computer sciences, respectively: Semantic memory organization and object oriented programming.

3.1 Semantic Memory Organization

Tulving [33] defines semantic memory as the storage of generalized world knowledge that is not attached to autobiographical or temporal codes. The Hierarchical Network Model [33] conceptualizes that within memory, categorical information is stored though its associations. In this model, categories are arranged according to their relations. This is done so that general categories (like food) are stored at higher levels, more specific categories (like fruit) are stored at intermediate levels, and highly specific items (like tangerine) are stored at lower levels.

A group of these related items makes up a network. The associations are represented by arrows which show the relation between the category represented by nodes. For example, the item tangerine may point towards the category fruit, because tangerines are an example of fruit. However, this relationship is unidirectional, and an arrow would not point from fruit towards tangerine, because fruit are not an example of tangerines. Characteristics that apply to all the categories would be stored at the highest level of the hierarchy, while lower level characteristics would only apply to that particular item and not to all in the hierarchy. For example, "plant life" may be a high level characteristic applying to all fruit, but "segmented", "round", "orange in color", "easy to peel" may be more lower level characteristics applying specific fruits.

Collins & Quillian [6] developed the hierarchical network model to conceptualize human semantic memory. They proposed two main features: The first is that moving from one level to another in the hierarchy takes time. The second is that retrieving features stored at another level also takes time. The data collected in their studies supported both feature assumptions about human memory. Conrad [7] noted that this model is economical because all characteristics do not need to be stored at each category, but can be stored at the most general category that applies. Further, these characteristics are "inherited" by the lower level categories that apply.

Related to hierarchical network modeling, Rosch et al. [23] hypothesized that there are at least three hierarchical semantic categories: the largest and most general are the superordinate categories, the intermediate categories are the basic-level categories, and the highly specific are the subordinate categories. Differentiation of categories is difficult at the superordinate and subordinate levels because they represent items that have either very few attributes, or that have very specific attributes, respectively. The basic level categories provide an intermediate level of categorization. The reason for their frequent is likely the result of their high degree of differentiation, their prominence in language and early acquisition during cognitive development [23]. However, with increasing domain expertise, differentiation and classification abilities are enhanced and refined and it appears that subordinate categories may rise to the level of basic categories; this hypothesis was confirmed by subsequent research [28]: Experts classified birds and dogs and then vice versa; the subordinate level classifications were made just as quickly as the basic-level classifications by experts compared to novices who exhibited increased latency for subordinate classification.

In summary, hierarchical network models employ at least three levels of semantic categories, they rely on at least two semantic relationships (category level membership and attribute relation) and they store attributes parsimoniously at the highest level applicable without redundancy at lower levels in the hierarchy.

3.2 What Is Object-Oriented Programming?

The object-oriented programming (OOP) paradigm involves the categorization of data and code and can be conceptualized as programming with taxonomically organized data [13]. Object-oriented programming first appeared in the programming language Simula and was focused towards the simulation of real-world phenomena [29].

OOP divides the data being processed into objects with both “static aspects”, the object’s characteristics, and “dynamic aspects”, the object’s behaviors [10], [13]. A “class” is a blueprint for a set of objects that share some characteristics [10], [13]. Each of an object’s static characteristics has a value and a type; According to [13] a type is a set of values, and a value is a mathematical abstraction. A class is required to create an instance of an object. The object instances created from the same class are distinct from one another. An object instance is considered a labeled set of labeled constants [13] and can accept parameters unique to the instance [10]. Thus, an instance of a class may contain the same or different values as another instance while having a different label.

Since a proposed class may include static and dynamic aspects that closely resemble another concept that requires a class definition, the object-oriented idea of inheritance was developed. Inheritance is the mechanism that combines the specific properties of a newly-defined class with the properties of one of more existing classes from which it inherits; in this way, a programmer only needs to develop the differences between the more specific class and the more general class [2]. For example, the class convertible may inherit the properties from the class car and hence is equipped with four wheels, an engine, a steering wheel, etc. Although different definitions of relationships between classes resulting from inheritance have been defined, in general inheritance seems to be tied to the concept of specialization, meaning that new concepts can be derived from less specific classes [10], [29]. Taivalsaari [29] clarifies that the distinction between the inheritance and specialization: specialization appears to be abstract, while inheritance appears to be a tool of convenience. Formica & Missikoff [13] note that specialization may also be called *subtyping*. An additional distinction can be made that nonstrict inheritance is based on programming language convenience, whereas “strict inheritance” requires that the derived class and the more general class are “behaviorally compatible” [35],, [36]. The latter rely on the implementations of inheritance with restrictions [13].

A restrictive relation supporting strict inheritance is the “is a” relation. The “is a” relation expresses belonging of a class to superclass. For instance, superclass car and subclass convertible are related such that the convertible “is a” car. The “is a” relation appears to create ambiguity in cases of a union between two classes which have different behavior for two properties of the same name. Such ambiguity appears to be solved, however, by “overriding” (redefining the domain of) the conflicting properties [13], [29]. Unlike single inheritance, multiple inheritance allows a subclass to derive from multiple superclasses. Singh [26] describes a few uses for multiple inheritance that vary across their purposes and the relatedness of the classes involved. The main advantage however is that single inheritance might not accurately represent the data for real world simulations because realistic classes tend to share attributes and behaviors.

In summary, class definition in OOP are hierarchically organized blueprints of objects who store a set of attributes at the class level; these sets can be inherited by subclasses without the necessity of duplication and may be further refined or specialized as needed. Strict inheritance requires the “is a” relationship which is a onedirectional relationship pointing from subclass to a superclass: a convertible is (always) a car but not all cars are convertibles.

4 Common Structure and Process

Four points of commonality, the last being initially in the guise of a difference between object oriented programming (OOP) and the hierarchal networks of semantic memory, are the subject of this section. We present the equivalency of classes and categories, semantic relations and attribute storage, and the reverse engineering of their processes.

1. OOP’s classes are equivalent to categories or levels of classification, where superclasses serve as superordinate or basic levels and subclasses as subordinate levels or highly specific categories.
2. Hierarchical network models and OOP make use of relations. Specifically, the “is a” relation is structurally and semantically equivalent to the subordinate to basic to superordinate classification in semantic memory. (See the tangerine example as a subordinate example in semantic memory section.)
3. Both approaches rely on graph structures and traversal time across the network with focus on optimization. In particular, both concepts appear to make use of distributing information across several layers of their network structures. This economical allocation of class or category attributes and behaviors (characteristics) allows both inheritance and the hierarchical network model of semantic memory to reduce processing time for information retrieval.
4. OOP is focused on generating new objects or exemplars; Hierarchical network models were developed to explain classification and organization of existing knowledge. The main difference between OOP class structures and semantic memory networks lies in the directionality of their application and function. Classes are defined in order to create new instances of objects with certain attributes and behaviors that exist, e.g., in a virtual world. Semantic hierarchical networks provide

categories for the classification of the real world. They have been experimentally investigated in sentence verification paradigms collecting response latencies to test whether memory is indeed organized as hypothesized. For example, a participant judges the following statements, one at a time, to be true or false: A robin is a bird; a robin is an animal. The process of using a structure for generating objects is the reversal of testing the structure for its properties. Hence, while the applications in cognitive and computer sciences appear different they have reverse engineered each others processes.

5 Synthesis Example: Forgetting

What appears to be a distinction may well be the impetus for venturing to a new perspective on the utility of OOP class structures and semantic networks. We suggest that concept such as creativity (creating new instances) and forgetting (loss of objects) will be relevant for the growing understanding and development in network structure research with benefits for both fields. For instance, we were reminded that OOP has the express purpose of creating objects. Objects as instances (or class exemplars) consume memory resources. JAVA is a popular OOP language that makes use of automatic garbage collection. Garbage collection is a mechanism that allows for memory that is being consumed by an object and its values, to be given back to the system by destroying the object. While a *destructor* can be invoked deliberately in the code, automated *destruction* is handled by the garbage collector and relies on the absence of pointers. This means that an object is considered unused and marked for garbage collection if it is no longer referenced, referred to, pointed to, or associated (all synonymous). One might infer that JAVA applications *forget*.

Human memory as a connectionist semantic organization relies on its associations. In order to retrieve information we step through a series of associated items. Likewise, recall of memories or knowledge in general is triggered by associations. By this explanation, one might concur that in the absence of any association (even those supplied by sensory data) the memory has been destroyed. The implication of the analogy between JAVA destructors and loss of memories is that poorly associated contents in long term memory are more likely to decay. This is not a new proposition: The acquisition of new knowledge requires semantic relation building also referred to as semantic elaboration. The more connected a concept is the easier it is to retrieve. Conversely, we suggest that items without semantic relations can be considered non-existent.

6 Common Research Topics and Conclusions

This paper explored a radical way of interdisciplinary thinking with the emphasis on discovering commonalities. We suggest that a modern integration between computer and cognitive sciences must venture beyond loosely modeled metaphorical applications of computer systems and the use of terminology with mere face validity. In this paper we provided an example of the discovery our shared concepts, knowledge structure and a common mental model of semantic structure and process. A problem space of particular interest for joint research appears to be parallel processing and programming.

The human brain is a sophisticated, albeit poorly understood system built on parallel processes with perceptual and cognitive components [24], [30]. One of the hard problems in perception and cognitive sciences is the binding problem: How are features or data processed by separate neurological pathways and/or in separate areas of the brain *integrated* and perceived as a gestalt? Stryker [27] proposed the synchronization hypotheses where features of the same object, processed separately, are integrated with a synchronized neurological firing rate. Furthermore it appears that attention increases synchronous firing [11]. Related problems dealing with how to synchronize data, how to design data storage (unitary vs. multiple memory stores) and delays resulting from synchronization and communication between multiple processing streams are currently under investigation in computer sciences [24]. An interdisciplinary investigation to address these issues may focus on the phenomenon of illusory conjunctions. Illusory conjunctions illustrate the break down of the binding process where features (color, shape, hairstyle, glasses) are associated incorrect objects (or people) [11], [32]. Illusory conjunctions appear as memory errors resulting form *parallel process integration errors.* It appears reasonable that by investigating how the human brain manages the parallel process we may discover meaningful solutions to the questions currently under investigation in parallel computing.

While processor speed enhancements are asymptotic, the next enhancement is the optimization and development of efficient robust parallel programs. Given the "biological computer concept" and our current set of questions as cognitive and computer scientist, some of our options are to reverse engineer existing phenomena and potential solutions and to advance the state of modern artificial intelligence by continued efforts to model human processing *authentically*. Hence, the inevitable conclusion to this paper is that interdisciplinary exchange on the findings and hypotheses in our respective fields may inspire all parties involved to follow new lines of investigation and to experiment with new ways of problem solving.

References

1. Baddeley, A.D., Hitch, G.: Working memory. In: Bower, G.H. (ed.) The psychology of learning and motivation: Advances in research and theory, vol. 8, pp. 47–89. Academic Press, New York (1974)
2. Baddeley, A.D.: The episodic buffer: A new component of working memory? Trends in Cognitive Science 4, 417–423 (2000)
3. Boden, M.: Mind as Machine: A history of cognitive science, vols. 1 and 2. Oxford University Press, New York (2006)
4. Broadbent, D.E.: A mechanical model for human attention and immediate memory. Psychological Review 64, 205–215 (1957)
5. Clarke, R.(Writer), Lotterby, S.(Director).: A Mattress on Wheels [Television series episde]. In: Lotterby, S. (Producer)(ed.) Open all hours. BBC Two, London (1976)
6. Collins, A.M., Quillian, M.R.: Retrieval time from semantic memory. Journal of Verbal Learning and Verbal Behavior 8, 240–247 (1969)
7. Conrad, C.: Cognitive economy in semantic memory. Journal of Experimental Psychology 92, 149–154 (1972)

8. Cowan, N.: The magical number 4 in short-term memory: A reconsideration of mental storage capacity. Behavioral and Brain Sciences 24, 87–185 (2001)
9. Crowther-Heyck, H.: George A. Miller, language, and the computer metaphor of mind Dispelling the “mystery” of computational cognitive science. History of Psychology 2, 37–64 (1999)
10. Deitel, P.J., Deitel, H.M.: JAVA: How to program, 7th edn. Pearson, Upper Saddle River (2007)
11. Engel, S., Fries, P., Konig, P., Brecht, M., Singer, W.: Consciousness & Cognition 8, 128–151 (1999)
12. Fancher, R.E.: Pioneers of Psychology, 3rd edn. Norton, New York (1996)
13. Formica, A., Missikoff, M.: Inheritance processing and conflicts in structural generalization hierarchies. ACM Computing Surveys 36, 263–290 (2004)
14. Gottsdanker, R., Shragg, G.P.: Verification of Donders’ subtraction method. Journal of Experimental Psychology Human Perception and Performance 11, 765–776 (1985)
15. Graham, R.(Writer), Ritelis, V.(Director).: Queen’s Pawn [Television series episode]. In: Childs, T. (Producer) (ed.) The Sweeney. Thames Television, London (1976)
16. Green, C.D.: Dispelling the “mystery” of computational cognitive science. History of Psychology 3, 62–70 (2000)
17. Hunt, M.: The Story of Psychology. Anchor, New York (2008)
18. Laplante, P.: Dictionary of Computer Science, Engineering and Technology. CRC Press-Taylor & Francis Group, Boca Raton (2000)
19. Lester, B.P.: The art of parallel programming. First World Publishing, Fairfield, Iowa (2006)
20. Miller, G.A.: The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. Psychological Review 63, 81–97 (1956)
21. Mogle, J.A., Lovett, B.J., Stawski, R.S., Sliwinski, M.J.: What’s So Special About Working Memory? An Examination of the Relationships Among Working Memory, Secondary Memory, and Fluid Intelligence. Psychological Science 19, 71–77 (2008)
22. Roddenberry, G.(Writer), Pevney, J.(Director): Return of the Archons [Television series episode]. In: Roddenberry, G.(Producer) (ed.) Star Treck. Desilu, Los Angeles (1967)
23. Rosch, E., Mervis, C.B., Gray, W.D., Johnson, D.M., Boyes-Braem, P.: Basic objects in natural categories. Cognitive Psychology 8, 382–439 (1976)
24. Salvucci, D.D., Taatgen, N.: Threaded Cognition: An Integrated Theory of Concurrent Multitasking. Psychological Review 115, 101–130 (2008)
25. Simon, A.: Designing organizations for an information rich world. In: Greenberger, M. (ed.) Computers, communications and the public interest, pp. 38–52. Johns Hopkins Press (1971)
26. Singh, G.B.: Single versus multiple inheritance in object oriented programming. ACM SIGPLAN OOPS Messenger 6, 30–39 (1995)
27. Stryker, M.P.: Is grandmother an oscillation? Nature 338, 297–298 (1989)
28. Tanaka, J.W., Taylor, M.: Object Categories and expertise: Is the basic level in the eye of the beholder? Cognitive Psychology 23, 457–482 (1991)
29. Taivalsaari, A.: On the notion of inheritance. ACM Computing Surveys 28, 438–479 (1996)
30. Thornton, T.L., Gilden, D.L.: Parallel and Serial Processes in Visual Search. Psychological Review 114, 71–103 (2007)
31. Treisman, A.M.: Features and objects. Quarterly Journal of Experimental Psychology 40A, 207–237 (1988)

32. Treisman, A.M.: Attention and binding. Presentation to the cognitive science group. University of Arizona (February 5, 2005)
33. Tulving, E.: Episodic and semantic memory. In: Tulving, E., Donaldson, W. (eds.) Organization of memory, pp. 381–403. Academic Press, New York (1972)
34. Unsworth, N., Engle, R.W.: The Nature of Individual Differences in Working Memory Capacity: Active Maintenance in Primary Memory and Controlled Search From Secondary Memory. Psychological Review 114, 104–132 (2007)
35. Wegner, P.: The object-oriented classification paradigm. In: Shriver, B., Wegner, P. (eds.) Research directions in Object-Oriented Programming, pp. 479–560. MIT Press, Cambridge (1987)
36. Wegner, P., Zdonik, S.B.: Inheritance as an incremental modification mechanism or what Like is and isn't like. In: ECOOP 1988: European Conference on Object-Oriented Programming, Oslo, Norway, August 15-17. Lecture Notes in Computer Sci., vol. 276, pp. 55–77. Springer, Heidelberg (1988)