# A Cognitive Neuroscience Perspective on Memory for Programming Tasks

Chris Parnin[1]

College of Computing
Georgia Tech Institute of Technology
`chris.parnin@gatech.edu`

**Abstract.** When faced with frequent interruptions and task-switching, programmers have difficulty keeping relevant task knowledge in their mind. An understanding of how programmers actively manage this knowledge provides a foundation for evaluating cognitive theories and building better tools. Recently, advances in cognitive neuroscience and brain imaging technology has provided new insight into the inner workings of the mind; unfortunately, theories such as program understanding have not been accordingly advanced. In this paper, we review recent findings in cognitive neuroscience and examine the impacts on our theories of how programmers work and the design of programming environments.

## 1 Introduction

Researchers have long been perplexed in understanding how programmers can make sense of millions of lines of source code text, extract meaningful representations, and then perform complex programming tasks, all within the limited means of human memory and cognition. To perform a programming task, a programmer must have the ability to read code, investigate connections, formulate goals and hypotheses, and finally distill relevant information into transient representations that are maintained long enough to execute the task. Amazingly, programmers routinely perform these mental feats across several active programming projects and tasks in fragmented work sessions fraught with interruptions and external workplace demands.

In coping with these demands and limitations, the programmer must have mental capacity for dealing with large workloads for short periods of time and cognitive mechanisms for maintaining and coordinating transient representations. As of yet, we have no cognitive model that adequately explains how programmers perform difficult programming tasks in the face of constant interruption. As a consequence, we have a limited basis for predicting the effects of interruption or evaluating different tools that may support task-switching for programmers.

Perhaps, new perspectives on memory and programmers are needed. Early models of memory, which we review below, have identified several key processes and provided many fruitful predictions. However, when pressed with more strenuous tasks, such as dealing with an interruption, these models have difficulty accounting for sustained performance [20]. Further, new results continue to emerge from studies of patients with novel brain lesions (injuries to specific brain regions after a stroke or accident) who display behaviors that undermine many of the assumptions of early memory models [55]. Likewise, early perspectives on programmers now seem dated. Shneiderman, who has published several influential articles on programmer memory and comprehension, once likened the ability of musicians to memorize every note of thousands of songs or long symphonies to that of programmers and suggested programmers would obtain the same ability to commit entire programs to memory in exact detail [50]. Rather the opposite has seemed to occur: Programs are not untouched sacred tomes, but organic and social documents that are understood and navigated with the assistance of abstract memory cues such as search keywords and spatial memory within a tree view of documents or scrollbars [27].

The methods available to researchers have expanded greatly. For example, it is now possible to administer drugs that interfere with memory formation or genetically engineer rats, whose basic brain structure for memory is remarkably similar, without the genes for neurotransmitters necessary for consolidating short-term memories into long-term memories. Additionally, fMRI machines provide the ability to measure changes in blood oxygenation levels associated with increased brain activity within 1-2 seconds to regions of brain with 1-3 $mm^3$ precision [62]. These methods have not been previously available have lead to the founding of a new interdisciplinary field: *Cognitive neuroscience*, coined by George Miller and Michael Gazzaniga, is "understanding how the functions of the physical brain can yield the thoughts and ideas of an intangible mind" [24]. For researchers studying the cognitive aspects of programmers, never have more opportunities been available to expand our understanding of the inner workings of the programmer's mind.

In this paper, we review perspectives on memory from the cognitive neuroscience literature to gain insight into how a programmer maintains and remembers knowledge used during a programming task. The perspectives on memory offered by classical psychology have difficulty accounting for programmers in practice and have followed us in our formation of theories of program comprehension. Following our review, we discuss implications for the design of programming environments and comprehension theories as well as remaining issues.

## 2 Memory and Theories of Program Comprehension

### 2.1 Psychological Studies of Memory

Memory research has had a long and rich history in the psychology community. Here, we briefly attempt to cover some of the key findings.

One of the earliest contributions to memory was Miller's work in 1956 on limitations on information processing. Regardless of what item a participant was being asked to memorize, Miller observed that the capacity for short-term memory appeared to be 5-9 items [35]. Recent research has suggested the actual limit is closer to 4 items [16].

In 1968, Atkinson and Shiffrin presented an influential model of memory called the *modal model of memory* [7]. In the modal model, information is first stored in sensory memory. Attentional processes select items from sensory memory and hold them in short-term storage. With rehearsal, the items can then be moved into long-term storage. The model characterizes the process of obtaining long-term memory as a serial and intentional process with many opportunities to lose information along the way via decay or interference from newly formed memories.

Attempting to refine the modal model's account of short-term memory, in 1974 Baddeley and Hitch introduced the idea of working memory [8] to help explain how items could be manipulated and processed in separate modalities (*e.g.*, visual versus verbal). The original model included separate storage of verbal (phonological loop) and visual-spatial memory with a central executive process that guided attention and retrieval from the stores. In 2000, Baddeley added an episodic buffer which allowed temporary binding of items.

Chase and Simon proposed that experts such as chess players can manage larger mental workloads by learning how to effectively *chunk* information after extensive practice and study [14]. The chunking theory proposes that it takes about 8 seconds to learn a new chunk, and that only about seven chunks can be held in short-term memory. For example, a chess master can outmaneuver an expert player because they can store and recall larger amounts of plausible moves and better assess positions of the chess board.

Several researchers have raised concerns about limitations with the chunking theory. First, information for tasks such as playing chess did not appear to be stored in short-term or working memory (or at least was transfered to long-term memory faster than predicted by chunking theory). Charness found when chess players interpolated playing chess with other tasks long enough to eliminate short-term memory, no or minimal effect on recall was found [12]. Second, chunking theory has a hard time explaining how people performing everyday tasks [19] or experts [20] could handle unpacking and shifting between multiple chunks with such a limited store.

An important alternative to the chunking theory was articulated over a series of papers by Chase, Ericsson and Staszewski [13, 21], who observed mental strategies used by mnemonists and experts. The resulting skilled memory theory identifies two key strategies experts use to achieve their remarkable memory and problem-solving ability: (a) Information is encoded with numerous and elaborated cues related to prior knowledge (similar to Tulving's encoding specificity principle [61]; and (b) experts develop a retrieval structure for indexing information in long-term memory (for example, experts might associate locations within a room with material to memorize – by mentally visiting locations within the room, the expert could retrieve associated items from those locations).

Recently, the skilled memory theory has been extended into the long-term working memory theory, which claims many of the problems with previous theories can be explained if working memory actually involves immediate storage and activation of long-term memories [20].

### 2.2 Cognitive Theories in the Psychology of Programmers

In studying the psychology of programmers, many researchers devised theories based on notions of memory that were available at the time. For example, many theories use the concept of *chunking* to build cognitive models of programming. Despite the problems noted by other psychologists, many of these notions still persist today. Here, we briefly review current theories of programmer cognition and comprehension.

In top-down comprehension [11] the programmers formulate a hypothesis about the program that is refined by expanding the code hierarchy. The programmers are guided by using cues called *beacons* that are similar to *information scent* in information foraging theory [43]. In bottom-up comprehension [49, 42], the programmer gradually understands code by *chunking* the source code into syntactic and semantic knowledge units. In opportunistic and systematic strategies [28], programmers either systematically examine the program behavior or seek boundaries to limit their scope of comprehension on an as-needed basis. Von Mayrhauser and Vans offered an integrated metamodel [63] to situate the different comprehension strategies in one model.

## 3 Memory in Cognitive Neuroscience

### 3.1 Building Blocks of Memory: Long-term Potentiation (LTP)

Like physicists who seek to understand the building blocks of atoms to understand the world, we seek to understand the building blocks of the brain, especially those that contribute to memory. Nearly a century after scientists recognized the atom as a fundamental unit of matter, neuroscientists followed by recognizing that the neurons play a similar role. Certainly, when examining the neuron in depth today, the picture has much changed from the simple view of passive integration of incoming signals, into the view of a complex interplay of voltage-gated ion channels with local synaptic regulation. Here, we focus on the fundamental aspects of a neuron that explains how a brief stimulus from the world can have long-lasting effects on the brain.

The neurological basis for memory is widely believed to be the long-term potentiation (LTP) of neuron synapses. After a synapse undergoes LTP, subsequent stimulus of the synapse will display a stronger response than prior to undergoing LTP. In 1973, Bliss and Lomo [10] first observed LTP after repeatedly stimulating rabbit brain cells and found responses to increase 2-3 times and persist for several hours. Some researchers consider LTP to be a neurobiological codification of the Hebbian learning process: Neurons that fire together, wire together [26].

An interesting aspect of LTP is its various forms of persistence and its connection with memory consolidation. It is now understood that LTP occurs in at least two stages: *early LTP* and *late LTP*. In early LTP, increased response is achieved for a few hours by temporarily increasing the sensitivity and number of receptors at a given synapse occurring within 1-2 seconds [29].

In late LTP, more long-lasting changes involve production of proteins to signal changes to the synapse's surface area and additional dendritic spines associated with stimulation [29].

But how long are these long-lasting changes? In general, synapses undergoing early LTP will return to baseline within three hours. Late LTP, however, has a much longer duration: lasting from several hours or days to months or over years (See Abraham's review on LTP duration [2] for a more in-depth coverage). LTP in the rat hippocampus lasting months and in one instance, one year, has been observed in the laboratory simply after applying four instances of high frequency stimulation spaced by five minutes [3]. In the human brain, newly formed memories are only expected to persist in the hippocampus for a few months or years until system memory consolidation into the neocortex is complete. This is consistent with amnesia patients who have difficulty recalling long-term memories a few months or years prior to their accident [55].

Further neurological processes of memory are of interest such as long-term depression (LTD) and neurogenesis. Whereas LTP increases the efficacy of synaptic transmission, LTD unravels those improvements to make it more difficult for two neurons two fire. The interactions between LTD and LTP are not yet entirely understood; however, it is known that during initial phases of LTP, reversal is more easily accomplished but becomes less so as time passes. If LTP is a mechanism for rapid memorization, are there other possible mechanisms for changes in the brain to occur? In short, yes, with neurogenesis it is possible to grow new neurons and form new growths of white matter. Brain cells were once considered to be like teeth, once lost we could not regrow new brain cells. It has been demonstrated that brain cells routinely die and new ones grow throughout our lives [51]. One of the most striking examples is a study of taxi drivers in London (who need to know very detailed spatial and contextual representations such as street intersections, routes, and traffic conditions of the city) that found when comparing the size of the hippocampus (an area of the brain responsible for remembering associations and spatial memory) of taxi drivers with that of the general population, a significant increase in size was observed and was correlated with time on the job [30].

### 3.2 Role of Hippocampus in Rapid Memorization

Few medical cases both arrest the imagination and have made a profound impact on memory research as has the story of H.M. [48]. H.M. was a man suffering from severe seizures who elected to have most of his medial temporal lobe bilaterally removed in an attempt to reduce the occurrence of the seizures. Although the surgery was successfully in reducing the seizures, an unforeseen consequence was that H.M. now suffered from anterograde amnesia, a condition where a patient cannot recall or form new memories but can otherwise recall past life events and facts and operate normally. H.M., with very few exceptions, could not learn new semantic facts such as new words or remember recent events such as meeting a person or having a meal. For H.M., retention of new memories generally only lasted a few minutes. If H.M. was having a conversation with a person for the first time, who then left the room and then reentered after a few minutes, afterward H.M. would not have recollection of having met the person or even having a conversation. A detailed analysis of the surgery performed on H.M. indicates that virtually all of the entorhinal cortex and perhinal cortex were removed, about half of hippocampal cortex remained although severely atrophied, and a largely intact parahippocampal cortex [15]. Since H.M., numerous cases have emerged demonstrating how different lesions result in different loss memory abilities; however, the case of H.M. illustrates the essential role of the hippocampus in forming long-lasting memories.

Morris and Frey postulate that the hippocampus provides the ability for an "automatic recording of attended experience" [38]. They argue that many important events cannot be anticipated nor may not occur again, and therefore traces and features of experiences must be recorded in real-time as they happen. Further, Morris makes the argument based on neuroanatomical studies that the hippocampus does not store sensory stimuli directly, but rather associates indices into other cortical regions [39]. For example, the memory of eating a new food at a restaurant is associated with various stimuli (the visual appearance, aroma, taste), contextual details such as the scuffle and movements of other patrons, and semantic details such as the name of the restaurant. The hippocampus is perfectly situated and equipped for this role of automatic association: with very plastic neurons able to undergo LTP and with connections from numerous regions such as visual and auditory pathways having already performed bottom-up processing, and connections with the prefrontal cortex for top-down processing.

Although studies of amnesia patients provide insight into loss of ability, they cannot account for how these systems operate for healthy people. Imaging studies of people performing memorization tasks have provided even more understanding of the hippocampus. In one study, subjects memorized a list of words, and then were asked to recall the studied words [18]. What was unique about this study was that fMRI images were taken while the subjects where studying and recalling the words. The researchers found that failure to recall a word was linked to weaker activity in the hippocampus during memorization; in contrast, success of recalled words was linked to stronger activity. From this study, one could conclude that if a stimulus failed to induce LTP in hippocampal cells at the time of the event, then no conscious memory is likely to remain. Another study has found a similar effect in the entorhinal cortex for items judged to be familiar but not recalled [37].

Research has also found evidence suggesting that specific subareas (*e.g.*, perirhinal or parahippocampal cortices) and specific lateralization (left or right) appear to be associated with different functions (*e.g.*, familiarity recognition or encoding) and different modalities (*e.g.*, spatial vs. verbal). However, it is not still not entirely clear how well we can localize function. For example, the parahippocampus was associated with encoding and recall of spatial memories [44], but activity in the parahippocampus was also found to be highly associated with recognizing objects with unique contextual associations: A hardhat invokes a specific context of dusty construction yards and therefore is associated with higher parahippocampal response; whereas a book has a less specific context and thus less activity [9]. One view put forward by Mayes, proposes that rather than operating on specific modalities of a hard-coded domain, such as verbal specific processing, the hippocampus supports different types of associations —inter-item, within-domain, and between-domain associations —and requires different computations for these associations types [31]. This domain dichotomy view explains why a process such as familiarity recognition may be associated with different regions because recognizing a familiar object would require different processing (and thus different regions) than recognizing a familiar object and location association.

### 3.3 Memory Organization and Architectures

**Memory Types** As previously mentioned, researchers distinguish between sensory, short-term, working memory, and long-term memory. For long-term memory, Squire proposed a taxonomy [56] that divides types of long-term memory hierarchically starting with a distinction between non-declarative (implicit) and declarative (explicit) memories. Non-declarative memory includes priming and muscle memory whereas declarative memory includes knowledge of facts and events. Tulving, an influential memory researcher publishing for over 50 years, describes semantic memory as knowledge of facts and episodic memory as a recollection of past events. Tulving's experience with an amnesiac patient E.P., who could learn new facts but not remember how he came to learn about them, lead Tulving to distinguish between our ability to *know* (to recall that the sky is blue) and *remember* (to relieve a past experience via mental time travel) [59].

Studies of patients with newly acquired amnesia have revealed further subtypes of memories. This includes *familiarity*, *recency*, and *source* memories. *Familiarity memory* involves the "feeling of knowing" that an object in a particular context has been encountered before without necessarily recalling the context (*e.g.*, seeing a face in the crowd that seems familiar but does not trigger a name). Familiarity memory is not to be confused with priming. First, in priming, a person previously exposed to an item is more likely to recall that item in the future; however, without a conscious recollection of having been primed. With familiarity, a person is aware that something seems familiarity. Second, priming and familiarity have doubly dissociated brain regions: Familiarity is supported in the entorhinal cortex; priming is believed to involve *modification* of the object representations within perceptual memory (for example, H.M. could be primed only for words he had learned prior to his accident) [45].

Some tasks involve recalling how long ago an event occurred, called *recency memory*. Milner studied patients who underwent surgery affecting the frontal lobes and found certain patients would have difficulty recalling how recently they have seen a word [36]. This suggests the prefrontal cortex plays a role in maintaining and binding a temporal context to memories. Further research has uncovered the importance of top-down involvement of the prefrontal cortex in episodic memory. Although many associations can be remembered in a bottom-up fashion as part of episodic memory, certain types of memories require top-down control and thus direct involvement of the prefrontal cortex.

Often when we learn facts we can associate the initial experience where we learned that fact; these types of memories are called source memory. Activation of the prefrontal cortex is necessary for forming source memories [25].

**Memory Systems** Since the modal model of memory was proposed in 1968, numerous findings have challenged many of basic premises of the model and, accordingly, several researchers have sought to put forth their own account. Here, we review a few of these models.

In Tulving's serial-parallel-independent (SPI) model [60], rather then providing a mechanistic model of memory, Tulving provides a few guiding principles or generalizations of memory. Simply, he believes the process of encoding a memory to be a serial process (output of one system provides the input for another), the process of storage to be distributed and in parallel (traces of a single stimulus exists in multiple regions of the brain with the potential for later access), and the retrieval of memory to be independent (different systems do not depend on others – once a memory is formed it is available within that system even if others are damaged). This view highlights the importance of viewing memory as a network of coordinating systems rather than an unified store.

In Fuster's account of memory systems [22], he uses a two-stage model derived from anatomical and neuropsychological studies of the brain. In general, raw senses are processed at progressively higher and higher levels of analysis starting from the most posterior region of the brain to the most anterior region of the brain. Fuster divides this journey into two major components: perceptual memory and executive memory. Within the perceptual region, processes and memory start from phyletic sensory memory, then integrating into polysensory, forming into episodic memory, generalizing into semantic memory, and abstracting into conceptual memory. After a brief hop over the motor system, the executive memory region involves concept, plan, program, act and phyletic motor memories. Fuster's account again highlights the specialization and localization of memory, but highlights the importance of top-down and bottom-up processes in memory, and how processing of an stimuli is collocated with its memory.

In Anderson (of ACT-R fame [6]) and colleagues' account of memory [5], a cognitive architecture is composed of several modules responsible for specialized processing of information. Modules have access to buffers which include: goal buffer, a retrieval buffer, a visual buffer, and a motor buffer. Interestingly, the model also includes a production system for learning based on the basal ganglia. The basal ganglia is mainly responsible for motor control; however, it also has been "recruited" by the prefrontal cortex for reward-based learning of rules [34]. As such, the strength the ACT-R model is in simulating learning and problem solving; however, the model is less effective in modeling memory retention (for an exception, see Altmann and Trafton's work on memory for goals [4]).

### 3.4 PFC: Goal Memory and Executive Processes

Humans fluidly perform and seamlessly switch among different tasks in a day. Routine activities, such as ordering a cup of coffee, can be performed without much cognitive effort. The rules are readily apparent: selecting a cup size, specifying a brew, paying the cashier – yet routine activities can be highly dynamic and even arbitrary. People have no problem adapting the rule to buy a cup of tea instead, or, given a rule never before encountered, *clap if you hear a phone ring*, most people would have no problem performing the task. However, if the rule was instead, *clap if you hear a phone ring in a coffee shop*, then people may fail to remember to apply the rule. Such forgetting would be a failure of *prospective memory*, remembering to remember. How does the brain store and manage prospective memories and other supporting memories needed for performing tasks?

The prefrontal cortex (PFC) is a region situated in the most anterior (toward forehead) portion of the frontal lobe. The PFC, a recent evolutionary addition, extends from the motor control regions of the frontal lobe to provide cognitive and executive control. E. Miller and Cohen [33] provide a compelling and influential account of the PFC. They argue the PFC provides the ability to bias a particular response from many possible choices. For example, when crossing a street, a person may be accustomed to looking left to check for oncoming traffic. However, if that person were an American tourist visiting London, then top-down control would be required to override the typical response and bias it toward a response for looking right first. In this theory, rules, plans, and representations for tasks are learned via highly plastic PFC neurons (a view also shared by Fuster [22]), but may migrate over time. One apt metaphor offered by Miller and Cohen is a railroad switch:

> “The hippocampus is responsible for laying down new tracks and the PFC is responsible for flexibly switching between them.”

The PFC also plays an important role in top-down attention: In early studies of monkey brains, when a food reward was shown to a monkey and then subsequently hidden for a delay period, persistent firing of neurons in the PFC was sustained during the delay period. Despite distracting stimuli, the monkey could recall the location of the food reward. However, monkeys with PFC lesions could not maintain attention and performed poorly at recalling the food [23]. More recent work has uncovered a possible mechanism for how the PFC can simultaneously maintain several active items in mind. When examining the firing patterns of ensembles of neurons, rhythmic oscillations can be observed. These oscillations are believed to encode attributes of an attended item. Siegel and colleagues [52] observed when multiple items need to be attended to, distinct items were maintained in distinct phase orientations of the oscillating signal. Like our ability to wave a string tied to a door knob, our limit to attend multiple objects may be simply bound to a limit of speed and space for separating items within a frequency spectrum (a problem well known in telephone and ethernet communications). An interesting benefit emerging from phase coding of items is “free” temporal order of those items. In the same experiment, when the order of items were misremembered, there was a correlation with inadequate phase separation of the encoded items: The signal still preserved enough information to represent the items, but not enough information was available to determine order. This view offers an interesting alternative to the concept of working memory. The prefrontal cortex can maintain many representations for tasks (especially if the representations refer to associations within the hippocampus), but can only attend to a few at a time.

Understanding how cognitive control occurs in the prefrontal cortex is still an ongoing research question. However, researchers have been successful in understanding how the prefrontal cortex supports one type of process that is important for suspension of tasks —prospective memory. *Prospective memory* is remembering to remember to perform an action in the future under a specific context (*e.g.*, setting up a mental reminder to buy milk on the way home from work) [64]. Often intentions appear to spontaneously pop into mind prior to an important event or sometimes unfortunately later than intended. Researchers have sought to understand the underlying mechanisms for prospective memory. Essentially, some researchers believe prospective memory requires some form of attentional resources [54]; whereas other researchers believe if reminder cues are readily available then the process could be automatic [32]. A recent fMRI study has found that depending on the nature of the intention, prospective memory could involve both strategic monitoring and automatic retrieval from cues [46].

## 4 Task Memory Model

Understanding both brain structures and the associated jargon (*e.g.*, *dorsolateral prefrontal cortex*) can be a daunting endeavor for anyone. Here, we present the task memory model in part to summarize insights from cognitive neuroscience literature on memory but also to abstract from the intricacies of the brain and nomenclature. Our model shares similar goals with the *stores model* [17], where Douce argues multiple modalities, such as spatial memory, play a crucial role in code cognition. Our model intends to go further, by first accounting for the underlying constraints of different memory, and then reconnecting these constraints to memory requirements in programming tasks and design of programming environments.

For the purpose of the forthcoming discussion, we introduce the term *task memory*, which is the set of constructs (such as goals) and processes (such as suspension) needed to perform tasks. Our goal is to explain how people such as programmers can maintain representations for complex and long-running tasks (over the course of many hours or several days) despite interruptions or task-switches. In defining task memory and its corresponding model, not only do we want to avoid the ambiguity of a term such as working memory, we also want to go further by specifying task related concepts such as suspension or goals and relate them to specific processes and localized function within the brain.

### 4.1 Memory Pathways

As we perceive sensations from the world, those sensations flow along pathways that actively process and interpret perceptions of our world. The impression of these perceptions is what we understand as memory. Even purely internal events, such as our inner thoughts, will activate the same motor speech areas and auditory comprehension pathways (*i.e.*, subvocalization) as listening to ourselves talking. Therefore, to speak strictly in terms of storage, would be to misunderstand memory —the storage of memory is interleaved with the same pathways that process and and later recognize past sensations.

We divide the storage pathways of task constructs into three regions: frontal region, associative region, and perceptual region (see Figure 1).

**Perceptual Region** The perceptual region contains both primitive and salient representations of stimuli. This region is segmented into visual, spatial, and semantic (including auditory) areas. Each area is responsible for interpreting and storing representations of stimuli. These representations are linked so that spreading activation is possible for learned concepts.

There are short-term effects of perceiving a stimulus. Short-term retention occurs locally, allowing for example same/different comparisons to be made. In addition, a stimulus will prime representations, but only at the level of which a person has previous experience (*e.g.*, a person can be not be primed for the semantic meaning of a word they have never learned, but under the right conditions they can be primed for the visual perception of the word).

**Associative Region** The associative region receives inputs upstream from each area of the perceptual region. The associative region has several interesting capabilities. The associative region is capable of receiving several distinguishing features (such as visual and semantic feature) and can create a resulting association. The formation of the association is fast and automatic —an autoassociative encoding of perceptual features —but the features are not stored, but rather indices into representation sites in the perceptual region. Associations are not formed for every stimuli but instead are selectively formed based on stimuli strength (attention and and novelty detection play a large role).

These associations are formed in such a way that activation of any one of the features will activate the indices of other associated features, which will in turn activate the representations within the perceptual region. The duration of an association can last several hours, but if strengthened can last days and in some cases years. However, associations can be overturned, or may not form in the first place if similar associations already exist.

Finally, the associative region is capable of encoding familiarity. Encoding familiarity allows stimuli to be identified more readily without requiring representations of the stimuli to be well formed. This will allow a feature to be recognizable (*e.g.*, a face) but not associated with other features (*e.g.*, a name).

**Frontal Region** The frontal region contains important pathways for interpreting perceptions, selecting responses, forming and attending to representations and goals, and directing learning and memory. Pathways in the frontal region are well connected the perceptual and associative regions, allowing multiple pathways to accessing representations and imposing top-down influence. Within the frontal region, important pathways exist for managing tasks such as monitoring and switching tasks.

Memory supported by the frontal region includes prospective, source, recency memory. The frontal region provides the primary infrastructure for holding a task's plans, goals, and task-relevant bindings. The duration of these task elements do not fade like short-term memory, but persist for hours or days. Task-relevant bindings do not store items directly, but rather refer to long-term memory or stores within the associative and perceptual regions. Finally, the frontal region provides infrastructure for reminders to "pop into the mind" in the presence of appropriate cues.

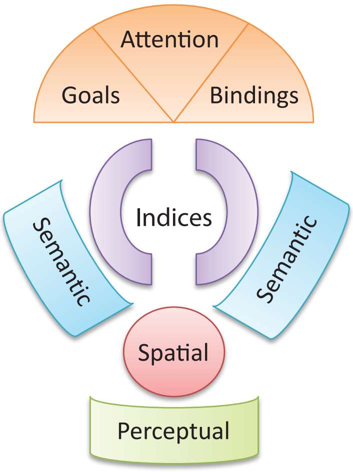

**Fig. 1.** Task memory model.

## 5 Considerations

We have presented several possible mechanisms underlying memory and associated cognitive control processes. In this section, we consider possible ramifications and speculate on impact on programming environments and theories of program comprehension. Design elements for programming environments has been discussed before [57]; here, we describe elements not previously covered in light of new findings in memory and considerations such as interruptions and multitasking. Remember, these ideas are not claims but a line of inquiry.

### 5.1 Programming Environment Support

Development tasks typically require coordinating software changes across multiple locations in a programs source code. Programming environments have provided limited support for managing the active artifacts relevant to the programming task. How a programmer represents these items in memory can inform how to better design programming environments.

**Names and Notes** Programs are comprised almost entirely of names. People forget or have difficulty recalling names on a daily basis. Some names in programs are for common operations such as iteration or familiar concepts such as sorting. But for many program elements, we may know the face, but not the name.

To find or write code, programming environments require either knowing the name precisely or partially, in the case of name completion system or class names. But unlike everyday objects, we do not have other aspects to assist in recall. We cannot call upon temporal or contextual clues, "what was that object I saw yesterday when I was debugging?". We cannot easily ask spatially "what was the object near the parsing code", nor semantically "what object was for checking security". Programmers still try asking these questions, they just have to find creative (but costly) ways of answering them.

For written notes, we often invent our own names to represent a current thought. We might write down, "labels" or "security". Presumingly, we want to record a prospective reminder to perform some action for a programming task. Amazingly, when looking at an old note, we can often recall the purpose of the note and the circumstance in which we wrote it. However, other times we do not even remember writing down the note or the note only triggers a vague recollection. For programmers, although notes have benefits in low overhead and conciseness, they are deficient when capturing detailed and delocalized knowledge. When notes fail to capture appropriate detail, programmers have to resort to costly information-seeking activities such as navigating source code or viewing source code history to rebuild their working context. Ultimately, neither notes nor environmental cues fully utilize program structure or state within programming environments, and more importantly, neither notes nor environmental cues digitally link together. Support for easily attaching notes to cues could create quick and powerful reminders: For example, pinning down a virtual sticky note on a code document or on a file within the document treeview.

**Levels of Support: Memory and Time** In Table 1, we consider different levels of support for interruption recovery based on decay of task memory. When suspending a task for a few minutes, what is at most risk is the loss of an ensemble of well-crafted thought. Humans are limited by the ability to simultaneously maintain attention to mental thoughts. Thus, a short-term interruption may not necessarily erase the memory of those thoughts, but we may never again find that insightful combination of those thoughts attended simultaneously with the same active top-down representations.

When suspending a task for a few hours, many newly formed associations and representations may still be intact. Upon return to the task, the programmer may need a brief reminder to reactivate suspended task goals and representations; it is not likely they would have forgotten these yet. In support of this process, programmers may need a quick refresh of the artifacts to help restore the details of the representations. During the task suspension, weak associations may have faded. Programmers may forget a relationship they discovered between code items or not recall where items are located.

Programmers returning to a task after several days require a different level of support. After such a delay, details such as new names of identifiers may have faded, and many representations used for the task may no longer be active. Traces of memories will guide the programmer in returning to work: Some code sections will feel more familiar than others. Further, external cues, such as jotted down goals, will help guide navigation and jump-start work. Finally, episodic recall of activity will help restore plans and potentially identify what actions to perform next.

| INTERVAL | SUPPORT |
|---|---|
| minutes | Support for managing attention. |
| hours | Brief reminder to restore top-level goals. Support for restoring artifacts. Simple associative cues such as words from familiar code symbols effective. |
| days | Support for restoring representations. Semantic-based interfaces less effective, use episodic-based interfaces. |
| weeks | Most representations have faded. Focus on restoring goals and plans. |

**Table 1.** Different length intervals of task suspension require different types of support from the programming environment when resuming.

**Environmental Cues and Beyond** Observations of developers suggest they frequently rely on cues for maintaining context during programming. For example, Ko *et. al* [27] observed programmers using open document tabs and scrollbars as aids for maintaining context during their programming tasks. However environmental cues often do not provide sufficient context to trigger memories: In studies of developer navigation histories, a common finding is that developers frequently visit many locations in rapid succession in a phenomenon known as *navigation jitter* [53]. Navigation jitter has been commonly attributed to developers flipping through open tabs and file lists when trying to recall a location [53, 41]. Environmental cues such as open tabs may be insufficient because *what* a developer remembers may be spatial and textual cues within the code document and not the semantic or structural location of the code element when automatically encoding working state [58].

By enriching environmental cues to take more advantage of the temporal, spatial, and contextual aspects we have previously discussed we would expect improvements to programmer productively. Research comparing development interfaces using names or *content* that a name refers to has shown that names are slower and less accurate than content [47], and content is strongly preferred over names when presented temporally [40]. Cues should enhance both an item's recency and familiarity. Temporal order of visiting an item should be easily discoverable. The context of visiting an element should also be clear: Tabs or files can be more understandable if it was made clear how a programmer visited the item (*e.g.*, indicate if a file was edited, visited from stepping through a debugging session, or found from a search result [with search keyword used to find it]). Other artifacts can be important cues for a programming task: events on calendars, meeting notes, checkins from source control, and emails from colleagues. Exploring how to collect, integrate and present these various cues offers an exciting research challenge.

### 5.2 Theories

Here, we consider some implications to current programming theories of comprehension and provide some concepts for developing richer theories of program comprehension.

**Visual Chunks** In Shneiderman and Mayer's syntax/semantic model [49], programmers do not retain memories of syntax, but only their meanings. This conclusion was reached based on the programmer's ability to *exactly* recreate a program statement: *i.e.*, even changing a symbol from $i$ to $x$ would invalidate that statement. By these measures, programmers tended to perform poorly when exactly reproducing the syntax of statements recently read, but instead retained their meanings.

When the syntax/semantic model was conceived, it was based on a variation of the modal model of memory (items move from sensory memory, short-term memory, and then long-term memory through active rehearsal). For anyone that has read paragraphs of text or lines of code, such a model may seem counter-intuitive. Unlike attempting to rehearse a phone number, when we read text, we do not frequently stop to remember the words or meaning, neither do we pause when engaged in casual conversations.

We propose that semantic meanings of read program statements are retained without intentional rehearsal, but instead with autoassociative support from the hippocampal formation. In contrast to the syntax/semantic model, we suggest memory of syntax is retained —not in an exact memorization of characters of text —but via abstracted perceptual patterns or visual sketches. For example, a certain region of code containing many distinct patterns of for loops and operations with character strings produces an unique signature of text indention and syntax highlighting that would be recognizable when quickly scanning source code. Such an ability would be advantageous to programmers who need to quickly and frequently switch documents and skim through code without having to deeply process the text in order to recognize relevant bits.

We introduce the concept of *visual chunks*, regions of code which may not yet have any strong semantic association, but which have perceptual features that are familiar and recognizable by a programmer. Visual chunks can be associated with temporal and contextual details such as a search term or hypothesis used in finding the visual chunk. Visual chunks can also be associated spatially within each other (*e.g.*, above or below another visual chunk). Finally, visual chunks can associated with subvocalized inner thought, giving it an internal nickname.

**Iterative Comprehension** The syntax/semantic model suggests that programmers use previously learned schemas (programming plans) to interpret text into semantic chunks in a hierarchal process. Alternatively, top-down theories explain that programmers parse code based on their current level of understanding and goals. Neither theory details the structures or mechanisms necessary for partial understanding of code or explain how a programmer can maintain these intermediate representations of unfamiliar code when switching between multiple tasks as observed in our recent experiment [40]. Opportunistic theories do not fare better: As programmers do not necessarily – upon reaching a new understanding – revisit every previously encountered item to update its understanding, but rather a programmer must have some form of intermediate representation in mind. Finally, a failing of all of these theories is their inability to identify exactly when learning occurs.

We also introduce the concept of *iterative comprehension*. With iterative comprehension, a programmer uses autoassociative memory of processed perceptual events to rapidly record many traces and facts about a program, even without having seen the code before. The programmer can draw upon numerous resources —familiarity, spatial, visual, auditory, autoassociative, and prospective memory, each involving distinct parts of the brain —that collectively allow the programmer to maintain partial representations when solving a problem.

For a new program, a programmer initially gathers numerous visual chunks when exploring the program. As the programmer learns more about the program, she iteratively updates previous visual chunks with knowledge of new events or relates with top down concepts and goals. The programmer can take advantage of previously learned schemas to provide strong associations with events and rapidly consolidate new facts. This explains how a programmer can retain memory of semantic properties of code while also associating other visual and spatial properties.

Here, we have only provided a sketch of what iterative comprehension entails. However, we believe iterative comprehension may provide a more compelling account of how programmers manage programming knowledge and can explain how programmers are able to explore and keep track of many items beyond traditional accounts of memory while only having partial knowledge of the code.

## 6 Remaining Issues and Future Questions

Several directions can be taken to move ideas presented in this paper forward. For the most part in our discussions on structures within the brain we have omitted detail on how structures differ based on location within the left or right hemispheres of the brain, also called *lateralization* of the brain. Extending models to include lateralization is both necessary for brain imaging studies and understanding the dual but separate roles that a structure plays (such as differences in encoding and retrieval in the left and right hippocampus).

Computational architectures for cognitive models such as ACT-R [6] or SOAR [1] are steadily improving. Still, these models are dealing with relatively simple tasks. Recently, Altmann and Trafton's work on memory for goals [4] have made modifications of the ACT-R architecture to include the ability to model the effect of interruptions on goal memory. Situating our work within these models would provide a mutual benefit of validating these ideas while suggesting modifications to the computational architectures.

Finally, despite new advances in memory research, there remains numerous unresolved issues. The biological mechanisms for forming memory are still not fully understood: One striking observation has been that spatial memory appears to use distinct processes when compared to those used in the normal associative processing occurring in the hippocampus. We do not yet understand the impact this has on encoding and consolidation of spatial memories. A strength of memory research has been the various lines of evidence used to investigate memory. But this is also a weakness: Some findings have only been established in animal studies, which may not hold in the same manner for humans. Further, these approaches have been effective at finding dissociations between brain regions and memory types but not in understanding how these regions coordinate and what information they carry. Finally, much care must be taken when using the results of fMRI studies; if not carefully guarded, poor statistical designs can allow over broad interpretations.

## 7 Conclusion

Nearly 40 years have passed since some of the earliest cognitive models of programmers have been proposed. Both the programming landscape and our understanding of the human brain have dramatically changed. Unfortunately, in the time since, the impact on practicing programmers has been negligible; the predictive power nearly non-existent; and, our understanding of the mind furthered little beyond common sense.

In this paper, we have outlined the background, tools, concepts and vocabulary for a challenging but hopefully rewarding trek forward. By understanding how a programmer manages task memory, especially in the context of multi-tasking and interruptions, we can begin to unravel this mystery.

## References

1. Newell A. *Unified Theories of Cognition.* Harvard University Press, Cambridge, MA, 1990.
2. Wickliffe C. Abraham. How long will long-term potentiation last? *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 358(1432):735–744, April 2003.
3. Wickliffe C. Abraham, Barbara Logan, Jeffrey M. Greenwood, and Michael Dragunow. Induction and Experience-Dependent Consolidation of Stable Long-Term Potentiation Lasting Months in the Hippocampus. *J. Neurosci.*, 22(21):9626–9634, 2002.
4. E. M. Altmann and J. G. Trafton. Memory for goals: An activation-based model. *Cognitive Science*, 26:39–83, 2002.
5. J. R. Anderson, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1050, 2004.
6. John R. Anderson and Christian Lebiere. *The Atomic Components of Thought.* Erlbaum, Mahwah, NJ, June 1998.

7. R. C. Atkinson and R. M. Shiffrin. *The psychology of learning and motivation (Volume 2)*, chapter Human memory: A proposed system and its control processes, pages 89–195. Academic Press, 1968.
8. A.D. Baddeley and G. Hitch. *The psychology of learning and motivation: Advances in research and theory*, chapter Working memory, pages 47–89. Academic Press, New York, 1974.
9. Moshe Bar and Elissa Aminoff. Cortical analysis of visual context. *Neuron*, 38(2):347–358, April 2003.
10. T. V. Bliss and T. Lomo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of physiology*, 232(2):331–356, July 1973.
11. Ruven Brooks. Towards a theory of the comprehension of computer programs. *International Journal of Man-Machine Studies*, 18:543–554, 1983.
12. N. Charness. Memory for chess positions: resistance to interference. *Journal of Experimental Psychology: Human Learning and Memory*, 2:641–653, 1976.
13. W. G. Chase and K. A. Ericsson. *The psychology of learning and motivation*, volume 16, chapter Skill and working memory, pages 1–58. Academic Press, New York, 1982.
14. W.G. Chase and H.A. Simon. Perception in chess. *Cognitive Psychology*, 4:55–81, 1973.
15. S. Corkin, D. G. Amaral, R. G. González, K. A. Johnson, and B. T. Hyman. H. m.'s medial temporal lobe lesion: findings from magnetic resonance imaging. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 17(10):3964–3979, May 1997.
16. N. Cowan. The magical number 4 in short-term memory: a reconsideration of mental storage capacity. *The Behavioral and brain sciences*, 24(1), February 2001.
17. Christopher Douce. The stores model of code cognition. In *Programmer Psychology Interest Group*, 2008.
18. L. L. Eldridge, B. J. Knowlton, C. S. Furmanski, S. Y. Bookheimer, and S. A. Engel. Remembering episodes: a selective role for the hippocampus during retrieval. *Nature neuroscience*, 3(11):1149–1152, November 2000.
19. K. A. Ericsson and P. F. Delaney. *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*, chapter Long-term working memory as an alternative to capacity models of working memory in everyday skilled performance, pages 257–297. Cambridge University Press, Cambridge, UK, 1999.
20. K. A. Ericsson and W. Kintsch. Long-term working memory. *Psychological Review*, 102(2):211–245, 1995.
21. K. A. Ericsson and J. J. Staszewski. *Complex information processing: The impact of Herbert A. Simon*, chapter Skilled memory and expertise: Mechanisms of exceptional performance, pages 235–267. Lawrence Erlbaum, Hillsdale, NJ, 1989.
22. J. M. Fuster. The prefrontal cortex–an update: time is of the essence. *Neuron*, 30(2):319–333, May 2001.
23. J. M. Fuster and G. E. Alexander. Neuron activity related to short-term memory. *Science (New York, N.Y.)*, 173(997):652–654, August 1971.
24. Michael S. Gazzaniga, Richard B. Ivry, and George R. Mangun. *Cognitive neuroscience: the biology of the mind.* Norton, 3rd edition, 2009.
25. E. L. Glisky, M. R. Polster, and B. C. Routhieaux. Double dissociation between item and source memory. *Neuropsychology*, 9:229–235, 1995.
26. D.O. Hebb. *The organization of behavior.* Wiley, New York, 1949.
27. Andrew J. Ko, Brad A. Myers, Michael J. Coblenz, and Htet Htet Aung. An exploratory study of how developers seek, relate, and collect relevant information during software maintenance tasks. *IEEE Trans. Softw. Eng.*, 32(12):971–987, 2006.
28. David C. Littman, Jeannine Pinto, Stanley Letovsky, and Elliot Soloway. Mental models and software maintenance. In *Papers presented at the first workshop on empirical studies of programmers on Empirical studies of programmers*, pages 80–98, Norwood, NJ, USA, 1986. Ablex Publishing Corp.
29. M. A. Lynch. Long-term potentiation and memory. *Physiological reviews*, 84(1):87–136, January 2004.
30. E. A. Maguire, D. G. Gadian, I. S. Johnsrude, C. D. Good, J. Ashburner, R. S. Frackowiak, and C. D. Frith. Navigation-related structural change in the hippocampi of taxi drivers. *Proceedings of the National Academy of Sciences of the United States of America*, 97(8):4398–4403, April 2000.
31. Andrew Mayes, Daniela Montaldi, and Ellen Migo. Associative memory and the medial temporal lobes. *Trends in cognitive sciences*, 11(3):126–135, March 2007.
32. M. A. McDaniel and G. O. Einstein. Strategic and automatic processes in prospective memory retrieval: A multiprocess framework. *Applied Cognitive Psychology*, 14:127–144, 2000.
33. E. K. Miller and J. D. Cohen. An integrative theory of prefrontal cortex function. *Annual review of neuroscience*, 24(1):167–202, 2001.
34. E.K. Miller and T.J. Buschman. *The Neuroscience of Rule-Guided Behavior*, chapter Rules through recursion: How interactions between the frontal cortex and basal ganglia may build abstract, complex, rules from concrete, simple, ones, page (in press). Oxford University Press., 2007.
35. G. A. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. 1956. *Psychological review*, 101(2):343–352, April 1994.
36. B. Milner, P. Corsi, and G. Leonard. Frontal-lobe contribution to recency judgements. *Neuropsychologia*, 29(6):601–618, 1991.
37. Daniela Montaldi, Tom J. Spencer, Neil Roberts, and Andrew R. Mayes. The neural system that mediates familiarity memory. *Hippocampus*, 16(5):504–520, 2006.

38. R. G. Morris and U. Frey. Hippocampal synaptic plasticity: role in spatial learning or the automatic recording of attended experience? *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 352(1360):1489–1503, 1997.
39. R. G. M. Morris. Elements of a neurobiological theory of hippocampal function: the role of synaptic plasticity, synaptic tagging and schemas. *European Journal of Neuroscience*, 23(11):2829–2846, 2006.
40. Chris Parnin and Robert DeLine. Evaluating cues for resuming interrupted programming tasks. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 93–102, New York, NY, USA, 2010. ACM.
41. Chris Parnin and Carsten Görg. Building usage contexts during program comprehension. In *ICPC '06: Proceedings of the 14th IEEE International Conference on Program Comprehension*, pages 13–22, 2006.
42. Nancy Pennington. Stimulus structures and mental representation in expert comprehension of computer programs. *Cognitive Psychology*, 19:295–341, 1987.
43. Peter Pirolli and Stuart K. Card. Information foraging. *Psychological Review*, 106:643–675, 1999.
44. C. J. Ploner, B. M. Gaymard, S. Rivaud-Péchoux, M. Baulac, S. Clémenceau, S. Samson, and C. Pierrot-Deseilligny. Lesions affecting the parahippocampal cortex yield spatial memory deficits in humans. *Cerebral cortex (New York, N.Y. : 1991)*, 10(12):1211–1216, December 2000.
45. B. R. Postle and S. Corkin. Impaired word-stem completion priming but intact perceptual identification priming with novel words: evidence from the amnesic patient h.m. *Neuropsychologia*, 36(5):421–440, May 1998.
46. Jeremy R. Reynolds, Robert West, and Todd Braver. Distinct neural circuits support transient and sustained processes in prospective memory and working memory. *Cerebral cortex (New York, N.Y. : 1991)*, 19(5):1208–1221, May 2009.
47. Izzet Safer and Gail C. Murphy. Comparing episodic and semantic interfaces for task boundary identification. In *CASCON '07: Proceedings of the 2007 conference of the center for advanced studies on Collaborative research*, pages 229–243, New York, NY, USA, 2007. ACM.
48. W. B. Scoville and B. Milner. Loss of recent memory after bilateral hippocampal lesions. 1957. *The Journal of neuropsychiatry and clinical neurosciences*, 12(1):103–113, 2000.
49. B. Shneiderman and R. Mayer. Syntactic semantic interactions in programmer behavior: a model and experimental results. *International Journal of Computer and Information Sciences*, 8(3):219–238, June 1979.
50. Ben Shneiderman. *Software psychology: Human factors in computer and information systems (Winthrop computer systems series)*. Winthrop Publishers, 1980.
51. T. J. Shors, G. Miesegaes, A. Beylin, M. Zhao, T. Rydel, and E. Gould. Neurogenesis in the adult is involved in the formation of trace memories. *Nature*, 410(6826):372–376, March 2001.
52. Markus Siegel, Melissa R. Warden, and Earl K. Miller. Phase-dependent neuronal coding of objects in short-term memory. *Proceedings of the National Academy of Sciences of the United States of America*, 106(50):21341–21346, December 2009.
53. Janice Singer, Robert Elves, and Margaret-Anne Storey. Navtracks: Supporting navigation in software maintenance. In *Proceedings of the 21st IEEE International Conference on Software Maintenance*, pages 325–334, 2005.
54. Rebekah E. Smith. The cost of remembering to remember in event-based prospective memory: investigating the capacity demands of delayed intention performance. *Journal of experimental psychology. Learning, memory, and cognition*, 29(3):347–361, May 2003.
55. H.J. Spiers, E.A. Macguire, and N. Burgess. Hippocampal amnesia. *Neurocase*, 7:352–382, 2001.
56. Larry R. Squire. Memory systems of the brain: a brief history and current perspective. *Neurobiology of learning and memory*, 82(3):171–177, November 2004.
57. M.-A. D. Storey, F. D. Fracchia, and H. A. Müller. Cognitive design elements to support the construction of a mental model during software exploration. *J. Syst. Softw.*, 44(3):171–185, 1999.
58. J. Gregory Trafton, Erik M. Altmann, Derek P. Brock, and Farilee E. Mintz. Preparing to resume an interrupted task: effects of prospective goal encoding and retrospective rehearsal. *International Journal of Human-Computer Studies*, 58:583–603, 2003.
59. E. Tulving. *Organization of memory*, chapter Episodic and semantic memory, pages 381–403. Academic Press, New York, 1972.
60. E. Tulving. *The Cognitive Neurosciences*, chapter Organization of memory: Quo vadis?, pages 839–847. MIT Press, Cambridge, MA, 1995.
61. E. Tulving and D. M. Thomson. Encoding specificity and retrieval processes in episodic memory. *Psychological Review*, 80:352–373, 1973.
62. A. Villringer. *Functional MRI*, chapter Physiological Changes During Brain Activation, pages 3–13. Springer, 2000.
63. A. von Mayrhauser and A. M. Vans. From code understanding needs to reverse engineering tools capabilities. In *CASE '93: The Sixth International Conference on Computer-Aided Software Engineering,* (Institute of Systems Science, National University of Singapore, Singapore; July 19-23, 1993), pages 230–239, July 1993.
64. E. Winograd. *Practical Aspects of Memory: Current Research and Issues*, volume 2, chapter Some observations on prospective remembering, pages 348–353. Wiley, Chichester, 1988.