Analysis and Expansion of Conscious Artificial Intelligence

Computation and the Brain - Project

Adrian Koegl `aak2229`

Yasmin Farhan `yf2600`

December 24, 2021

Contents

We are at the beginning of an exciting journey to implement consciousness in a machine. After successful psychological and neuroscientific insights, combined with steady progress in artificial intelligence, many avenues are open for us to explore the groundbreaking possibilities of *Conscious Artificial Intelligence*.

Blum and Blum have developed the *Conscious Turing Machine* (CTM), a formal model of a conscious AI. Their approach takes the architectural developments by Alan Turing one step closer to consciousness by incorporating an early technical model of human consciousness, called the *Global Workspace Hypothesis*. With abstractions of human brains dynamics, the authors argue that their machine possesses a *feeling of consciousness*. Nevertheless, we argue that missing elements in the difference between abstraction and empirical findings in the human brain constitute essential functions of consciousness.

In this paper, we present our implementation of the most critical aspects of this model and propose an expanded model of their CTM. For the implementation, we use SystemC, a high-level hardware design language that combines hardware and software design elements. To explore further ways of abstracting the human mechanism of consciousness, we use empirical findings from psychological and neuroscience research and formulate our results in the form of an extended, sophisticated model of short-term memory. While there is no formal way to prove a heightened feeling of consciousness, our result brings the CTM closer to a human form of consciousness.

1 Introduction

The advancement of research on consciousness in psychology and neuroscience does not leave the artificial intelligence community untouched. Already in 1997, Baars has formalized conscious processes in the brain into the *Global Workspace Theory*, allowing for computational models of consciousness. Blum and Blum have provided such a formal definition of a Conscious AI, which they call *Conscious Turing Machine* (CTM). Additionally, they argue that their model would result in a conscious machine being aware of its feelings.

To practically investigate the applicability of these models, we implemented the most important dynamics of the formal CTM description in SystemC.

As the authors state that the "reasonableness of the definitions and explanations can be judged by how well they agree with commonly accepted intuitive concepts" [5]. Therefore, while implementing an abstract CTM in SystemC, we tried to embed their modeling into current neuroscientific findings on consciousness and psychological evidence for selection of attention. We have used the deviations found in their model and the current research progress to expand their notion of consciousness, bringing their model closer to a human form of consciousness. In this paper, we focus on the modeling of Short Term Memory (STM).

Leading the way to show our implementation and model expansion results, we first summarize Blum & Blum's conscious Turing machine based on the Global Workspace Theory. Thereby, we specifically investigate their argument of what makes the CTM conscious. Additional theoretical background is provided by wrapping up the relevant research in psychology and neuroscience, which will be essential to discuss improvable parts of the CTM model and introduce our own STM model based on empirical evidence. For the implementation part, we first introduce the SystemC language and argue why we have used it for a first dynamic implementation of the

CTM.

In the methodology, we then discuss which aspects of the CTM can be designed more closely to human consciousness and stress the importance of the STM's ability to pursue parallel processing. We then present our results of implementing the CTM and our extended STM model. Concludingly, we discuss the soundness of the CTM's formal definition and which parts have to be further improved to make a more sophisticated argument in favor of its feeling of consciousness.

2 Theoretical Background

2.1 Global Workspace Theory

The Global Workspace Theory (GWT) describes an architecture of distributed knowledge sources and processors that cooperatively solve problems, which each of these sources couldn't solve alone. The core of this architecture is a fleeting memory capacity that enables access between otherwise separate brain functions [4]. In a system of specialized parallel processors, it makes sense that coordination and control are exercised by central information exchange. Such a central coordinator is the main actor giving rise to consciousness as perceptual contents only become conscious when broadcast to various processors across the brain [13]. According to Baars, consciousness is the primary agent of such global access functions in humans and other mammals [2].

The "conscious access hypothesis" implies that conscious cognition provides a gateway to various capacities in the brain with the primary function of integrating, providing access, and coordinating the functioning of specialized networks [3]. In terms of the *theater metaphor*, consciousness resembles a bright spot on the stage of immediate memory. The spotlight directed by attention represents conscious awareness, while the dark is unconscious. After conscious sensory content is established, it is distributed to decentralized functions lying in the "dark."

GWT is used to model computational and neural net models, as it seems to have reasonable brain interpretations that allow for specific and testable brain hypotheses. It cannot be proven that such architectures exist in the brain, but the validity of models gives existence proof of the functionality [4].

2.2 Conscious Turing Machine

The Conscious Turing Machine (CTM) proposed by Manuel and Lenore Blum, is, in essence a formalization of Bernard Baar's Global Workspace Theory of Consciousness (GWT) described previously. We were successful in establishing a high level hardware design framework for the CTM, but before going into the details, it is necessary to give an overview of what exactly it is we set about implementing. It is also important to note that the overview given below as described as the CTM is proposed in the original paper, but is not all encompassing of what we ultimately implemented or made modifications to.

2.2.1 Technical Details

Prior to deconstructing the CTM, it is necessary to establish that the notion of time is of great importance, given the particular ordering in which various events take place and the dependencies which exist between them. Time in the CTM is discrete, beginning at 0, and all components of the CTM possess the same perception of what the time is at any particular moment.

The primary members possessing some level of consciousness from the theater analogy proposed by Baar's in the context of the CTM are the Short-term Memory (STM) and Long Term Memory (LTM) processors. All other components of the CTM can be framed as the structures necessary to uphold the functionality of these main memory components.

The 'Brainish' language is also defined as being the inner language used by and between the LTM processors to communicate their inner worlds to other components in the CTM. Given the scope of this project and the absence of any Brainish language, however, we make the assumption that any information received from the outside world by the CTM is understood as is, without the need to translate it into a Brainish language.

The CTM is a seven-tuple, $< STM, LTM, Down-Tree, Up-Tree, Links, Input, Output >$, and a short overview of each of these components which the CTM comprises, and which are shown in Figure 1 is given below.

1. STM - Short Term Memory
   The STM is described as being the one actor on the central stage of consciousness, holding one 'chunk' of information at every time tick/clock cycle. This chunk of information represents the entirety of the CTM's conscious content at time t. Any chunk possessed by the STM will have been received by an LTM through the UpTree competition that will be detailed in later sections.

2. LTM - Long Term Memory
   The LTM is a collection of N initially unlinked processors representing the unconscious working memory. All processors work and process information in parallel, and each has their own address and dedicated function. At every clock tick, each processor generates a chunk of information which is placed in the UpTree competition, a single chunk of which will either make it deterministically or probabilistically into the STM.

   A chunk can be defined as a six-tuple - $< address, t, gist, weight, intensity, mood >$, with the different elements described as follows:

   2.1. *address*: Denotes the processor which generated the chunk

   2.2. *t*: The time tick during which this chunk was generated

   2.3. *gist*: A multimodal thought that encapsulates essence of the chunk in Brainish

   2.4. *weight*: A real number that can be positive or negative, it is representative of the processor's estimate of how important it is to get the chunk into the STM

   2.5. *intensity*: The cumulative sum of absolute values of the weight after moving up each level of the UpTree

   2.6. *mood*: The cumulative sum of real values of the weight after moving up each level of the UpTree

3. $Down-Tree$
   While dubbed a tree, the DownTree is better thought of as a broadcast mechanism which

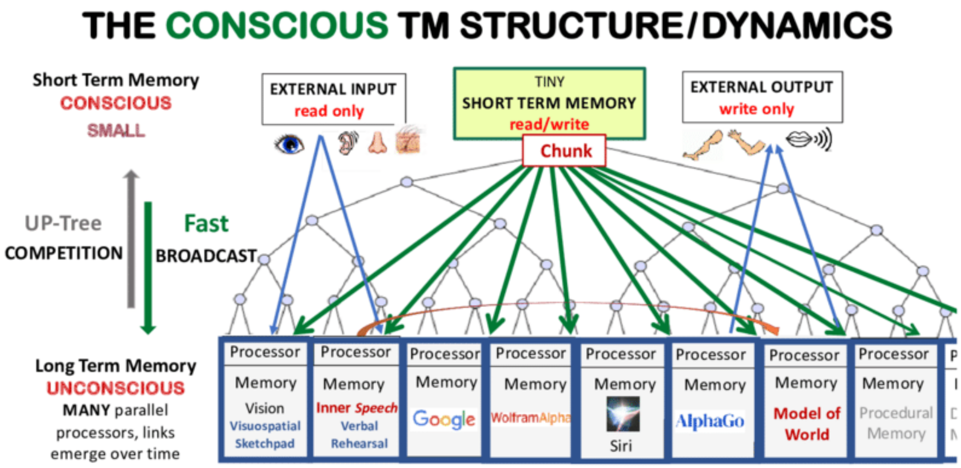


Figure 1: Conscious Turing Machine Model [5, Figure 4]

the STM uses to deliver the chunk at any given time t to all processors, to be received at time t+1. The point at which all processors receive the broadcast chunk from the STM is defined as being that which the processors gain *consciousawareness*. This is noted as being the formal definition in the context of the CTM but not meant to be offered as an explanation for why the CTM is argued to actually exhibit the feeling of consciousness.

4. $Up - Tree$
   An up directed binary tree of height h, the purpose of which is to allow the processors to compete amongst themselves to determine which chunk will make it into the STM. At time 0, all processors place their chunk onto a leaf of the UpTree, and at every subsequent time step from $t+1$ to $t+h$, each chunk will either move up or disappear depending on the competition algorithm and function used. Thus at each time step from $t = 0$ to $t + h$, the number of competing chunks is effectively halved due to the local competitions occurring at every node until a single chunk wins the competition globally at time $t + h$.

5. *Links*
   These are bidirectional edges between processors which represent a channel of communication between them. A link between two processors develops every time a transaction that establishes the pair of processors' usefulness to each other occurs.

6. *Input*
   This is the information from the outside world acquired by the CTM's sensors which are converted to Brainish coded gists using input mappings, and subsequently sent to designated LTM processors.

7. *Output*
   Output maps are used to convert Brainish gists from the LTM processors into commands for the CTM's actuators to be communicated with the outside world.

2.2.2 Feeling of Consciousness

The whole formal definition of consciousness made by the authors relies on the assumption that the GWT ”captures the essence of consciousness” [5]. Nevertheless, the whole model is kept relatively simple as Turing has suggested [17]. There are several aspects of a CTM which make Blum & Blum argue in favor of a feeling of consciousness, of which one crucial factor is to implement a *continuous narrative* experience into the CTM model. This is formally defined by the STM holding exactly one chunk at any time t with *mood* and *intensity* influencing which chunk arrives in the STM. Mood is a parameter that ranges from positive or happy to negative or sad. Intensity describes how strong this mood is expressed. It should be noted that the authors argue that the machine does not possess these feelings - these parameters are merely a formal definition. This model, in theory, makes the STM *consciously aware* of one narrative and influences the LTM's on what mood and intensity to assign to the following chunks by feedback. Therefore, conscious awareness affects the mood of the CTM. In addition, the expressiveness of brainish language is supposed to bring the capabilities of a CTM close to human thoughts.

One important modeling decision by the authors is to have only one brainish chunk present in the STM at any time so all the processors can focus on the same information. According to the authors, all processors being aware of the same content gives rise to a *feeling of consciousness.* A *stream of consciousness* is modeled by the process of the competition and broadcasting of chunks to the STM and LTM's, respectively.

Another argument about the feeling of consciousness is made through the participating special processors. A *model of the world* processor constructs the inner and outer world and assigns labels to different objects and actions. This allows the machine to distinguish the self from the non-self in this world model. Additionally, actions of ”self” and ”not-self” can be predicted and planned. The CTM's awareness of its consciousness leads the authors to reason about the machine possessing conscious awareness.

The *inner speech processor* enables the CTM to recollect its past, predict the future and make plans. The *inner vision processor* creates inner images and is used to generate images or dreams. Together with the *inner sensation processor*, these three processors are the special purpose decoders that extract speech, vision, and sensation. The assumption that a brainish chunk in the CTM ”feels” similar to how we see, smell, touch, and feel is supposed to provide the CTM with a better sense of consciousness

2.3 Early vs. late selection of attention

The debate of whether the information or, speaking in technical terms, chunks are selected early or late is a well-known problem in psychology [12]. Broadbent has developed his *selective filter theory* in 1958, in which he hypothesizes that the selection of attention happens early [6]. In this theory, he argues about filters at the beginning of any input processing step, which selects information passing through before higher cognitive processes assign any meaning. It assumes a single serial channel with limited capacity that bases selection exclusively on physical stimulus characteristics. The empirical foundation of this theory lies in the experiments of dichotic hearing performed by Cherry in which participants could only involuntarily focus on specific auditory input [7]. Additional evidence for this theory was given by experiments on the ”psychological refractory period” by Welford, which showed a bottleneck in the processing system: the processing of the first stimulus must be completed before that of the second stimulus can begin [18]. Later, this theory was falsified, one reason being the so-called cocktail-party effect: specific information can still pass through to cognitive processes based on an assigned meaning [14].

Expanding Broadbent's theory, Treisman has developed the *attenuation theory* of attention which describes how information is processed [16]. In contrast to the selective filter theory, she argues that information is not entirely filtered but attenuated. This implies that we can still consciously perceive some information, but we have to actively assign importance to it so that it does not get lost immediately. Therefore, not all filtering is done before the STM, but some chunks simply arrive there attenuated considering the current processing capacity. According to Treisman's theory, it requires some activation threshold to become aware of attenuated information. This can either happen bottom-up by sufficient salience or top-down in experience-dependent meaning.

In 1963, Deutsch and Deutsch formulated an opposing theory: The selection of attention supposedly takes place late in the process [9]. They argue that all incoming stimuli are wholly analyzed, and attention selection only happens late near the reaction. These *late-selection theories* suggest that the most critical stimuli are identified in or near cognitive processes. Nevertheless, these processes also occur before reaching the STM, and without attention from the STM, they will fade quickly without conscious awareness. Such machining requires a parallel processor near the STM, which weights the stimuli according to relevance.

In response to the debate of early vs. late selection theories, Allport has argued that there is no singular answer to this problem [1]. Various studies could provide evidence for both early, and late selective attention [8]. One possible resolution considering both perspectives is the *Perceptual Load Theory* proposed by Lavie [11]. She argues that situational context decides over early vs. late selection of attention. If the problems require low perceptual load, attention will be selected late as capacities are not exhausted, and "distractors" will be processed. On the other hand, if a subject is working with tasks requiring a high perceptual load, selection will happen early and unconsciously because there are no capacities for selection in the STM anymore. One fundamental assumption to Lavie's theory is that the full capacities in the STM are used at any time.

2.4 Neuroscientific view of consciousness

The *Integrated Information Theory* (IIT) is a neuroscientific theory describing the experience of consciousness as the combination of experiential factors which contain intrinsic values — although they are intrinsic only to the specific system in question [15]. Therefore, each experience, a point in consciousness, is imbued with these intrinsic properties that subsequently invoke the feeling of consciousness. According to Tononi, the IIT requires a complex and interconnected system in which a causal nexus exists between such components imbued with the properties mentioned above [15].

The IIT claims that no simulation of the human brain can claim to be conscious as "a simulation of a black hole is not a black hole" [15]. And while this model seeks to take into account methods in which the human brain achieves a sense of consciousness, this point is irrelevant because the goal is not to replicate the human brain but to simulate an experience of consciousness (which is not in itself an experience limited to the human experience). Furthermore, any experience of consciousness has been regulated to areas of the brain involved in what is essentially a simulation of conscious experience (the tempo-parieto-occipital region alongside a frontal thought-experience based region) [10].

Koch offers a ten-point suggestive framework of consciousness given a biological basis [10]. The relevant points are:

1. Zombie Modes and Consciousness: claims that one aspect of consciousness must involve an automatic and unconscious response system.

2. Coalitions of Neurons: involves the idea that various summations neurons can have overall excitatory or inhibitory effects on each other.

3. Higher Levels First: there is a hierarchy of signals in which consciousness exists near the top.

4. Driving and Modulating connections: driven by inputs that focus on either modulating other components or driving them. Driving inputs have to do with back projection.

5. Attention and binding: involves the general division of attention, which comes in two general forms: purposeful attention and automatic or unconscious attention.

This overall framework is provided as a biologically based loose conception of components that contribute to overall consciousness.

3 Methodology

3.1 Implementation

From the onset, we knew it was likely that our implementation would have to be a hardware leaning one, so ultimately SystemC was chosen of the different hardware design options at our disposal for the following reasons.

1. *Concurrency*
   The concurrent nature of the different processes operating throughout the context of the CTM. For example, there is nothing stopping any given LTM from sending information to the outside world at the same time that it is receiving a chunk of information from the STM. In SystemC, a built in event driven simulator is capable of emulating concurrency on a single processor through interleaving. The order in which different processes are issued is decided by a scheduler.

2. *Time*
   The notion of time possessed by all structures in the CTM. In hardware, the clock signal is essential and is considered the heartbeat of a digital system, and any hardware language chosen would be required to distribute all clock signals with a uniform delay across the entire system. In SystemC specifically, the sc_time class is responsible for handling the simulation time (which bears no relation to the wall clock or processor time).

3. *Hierarchy*
   There exist different paths of communication between structures of the CTM at varying levels of hierarchy. In SystemC, the design complexity that hierarchy is prone to causing is eased through the use of channels and ports. An SC_MODULE, or component in the system, can be a submodule of a more complex structure.

4. *Level of abstraction*
   SystemC is a high level hardware design framework that is based on C/C++. Using SystemC, designing a hardware implementation is relatively straightforward given the absence of a need for fine-grained design decisions that would be required in other hardware design languages. It takes a top down approach to hardware design, beginning with the high level simulators of the system and injecting concepts of hardware (e.g. clock, communication interfaces) wherever needed.

5. *Software support*
   SystemC, while offering the tools to describe and model hardware, remains fully compatible with software with all the advantages that software design affords, such as non-determinism, which is essential given our later proposed changes to STM. Thus SystemC allows the user to move from working in a simulation environment to offering a means of having a design specification that is mappable to hardware.

3.2 Relating CTM to empirical evidence

Taking into account the argumentation by Tononi [15], it is important to note that we do not actually try to achieve consciousness in the machine. Rather, we are interested in reflecting human consciousness as best we can. Therefore, we will compare and discuss the difference between psychological and neuroscientific evidence previously presented with the CTM, and argue about an extension of the STM based on this comparison.

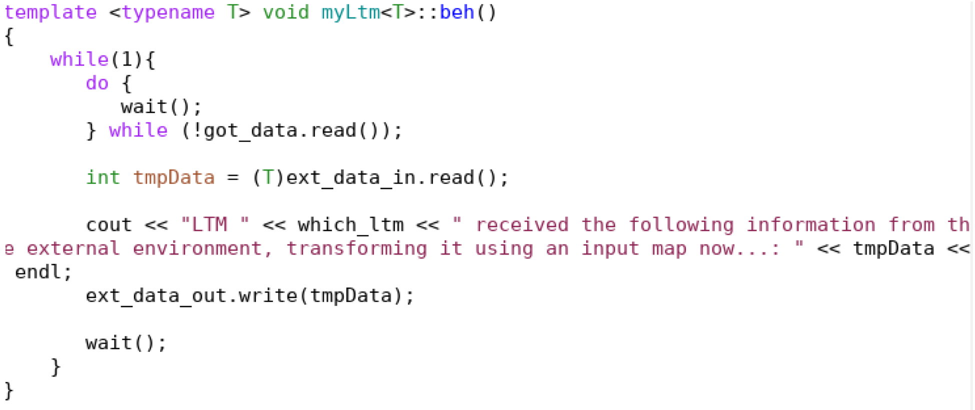
Blum and Blum agree in their introduction of the STM that humans have the capacity of consciously processing 7 ± 2 chunks in the STM in parallel [5]. While we agree that most of the filtering or ”competition” takes place unconsciously, or in the Up-Tree to speak in terms of the model, we argue that the parallel processing of several chunks in the STM would bring the feeling of consciousness closer to human perception. Therefore, we would like to discuss what algorithms and processes take place in the human STM based on psychological evidence and subsequently introduce an extended model of the STM considering these observations.

The current model of an STM assigns the Up-Tree the complete responsibility of filtering the outcome of the LTM processors and providing the STM with only one chunk. In terms of models in psychology, this represents a form of late selection without considering a conscious part in deciding over attention. While Blum and Blum have decided to implement no conscious attenuation, we claim based on research that it does occur in the human brain and potentially contributes to a sense of consciousness. Based on this it is first of all necessary to allow several chunks in the STM which can be processed in parallel based on the assigned attention.

As research has shown, the STM in humans also possesses some capability to prioritize and shift attention. According to Treisman’s attenuation model, several chunks of auditory input can reach the human STM where some chunks will be attenuated rather than previously eliminated [16]. She found that a human can actively assign importance to an auditory chunk present in the STM to which most attention is paid. Additionally, this theory implies that not all filtering is done before the STM. It seems reasonable that such a capability to prioritize conscious content contributes to a feeling of awareness. Based on these empirical evidences, we expand the notion of consciousness in the STM by allowing for parallel processing and active engagement in selecting attention of different chunks. We will call this active involvement in influencing assigned attention *priority*, which does not ultimately decide over attention but allows for some active contribution by the STM.

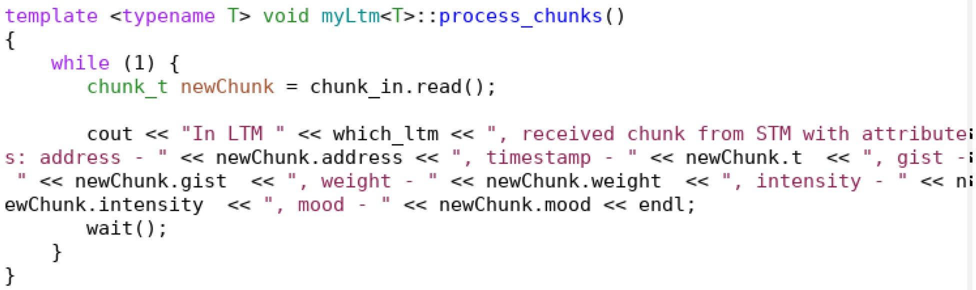
Furthermore, we integrate the assumption of the perceptual load theory that resources for attention are limited and all of these resources are exhausted at any given time. Following this theory, it is necessary to model an attention load assigned to any task. For simplicity, we assume that the sum of all the individual attentions assigned to every chunk at any time t must equal the maximum capacity.

These modelling decisions are additionally affirmed by the neuroscientific perspective on consciousness developed by Koch, as he also argues that one aspect of consciousness is the ability to purposefully steer attention [10]. The first four points presented in subsection 2.4 are already sufficiently met by the abstractions made by Blum and Blum.



```
template <typename T> void myLtm<T>::beh()
{
    while(1){
       do {
          wait();
       } while (!got_data.read());

       int tmpData = (T)ext_data_in.read();

       cout << "LTM " << which_ltm << " received the following information from th
e external environment, transforming it using an input map now...: " << tmpData <<
 endl;
       ext_data_out.write(tmpData);

       wait();
    }
}
```

Figure 2: beh() clocked thread in LTM



```
template <typename T> void myLtm<T>::process_chunks()
{
    while (1) {
       chunk_t newChunk = chunk_in.read();

       cout << "In LTM " << which_ltm << ", received chunk from STM with attribute
s: address - " << newChunk.address << ", timestamp - " << newChunk.t  << ", gist -
 " << newChunk.gist  << ", weight - " << newChunk.weight  << ", intensity - " << n
ewChunk.intensity  << ", mood - " << newChunk.mood << endl;
       wait();
    }
}
```

Figure 3: process_chunks() clocked thread in LTM

4 Result

4.1 Foundational implementation of CTM

The CTM implementation can be found here . It contains three directories, each with a different purpose.

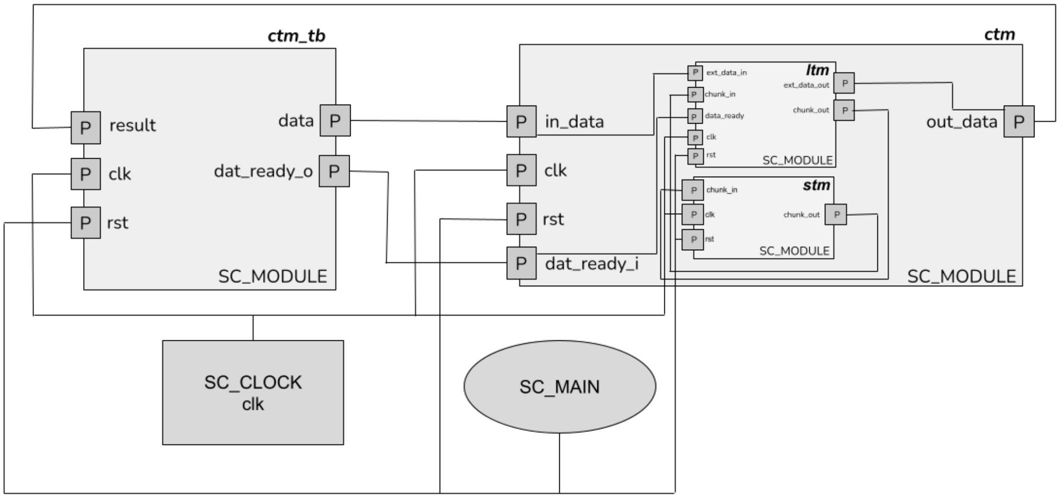


Figure 4: Simplified block diagram of CTM implementation

1. *src* - The src directory contains all of the source code for the CTM implementation, including the header and .cpp files needed for every module (CTM, LTM, STM). It is the code in this directory which would be synthesized in hardware were the implementation made fully ready for such a feat.

2. *tb* - The tb directory contains the testbench code that generates random stimulus to be sent to the CTM and models the outside world from which the CTM and its various internal components receive information. It also contains the sc_main.cpp file, which contains the necessary simulation configuration information such as the global clock period, and is where the main ctm and ctm_tb modules are instantiated and connected to each other for effective communication throughout the life of the simulation.

3. *sim* - The sim directory is the directory from which the src and tb code is compiled and tests are run.

The primary components defined in this implementation are the STM and LTMs. Figure 4 depicts the connections between the CTM and CTM_TB in addition to the hierarchical connections between the inner components of the CTM. For every connection between two sc_modules, there is a pair of end ports (sc_in and sc_out) and a channel (sc_signal) connecting them. The figure shows the connections between only a single LTM and the STM for demonstration purposes, but in practice there are multiple LTMs all connected to the STM and to the outside world in a similar manner.

Given the scope of this paper, we don't attempt to offer a syntactical explanation for every line of code, but rather aim to highlight points of communication between the modules' member SC_CTHREAD (clocked thread) processes.

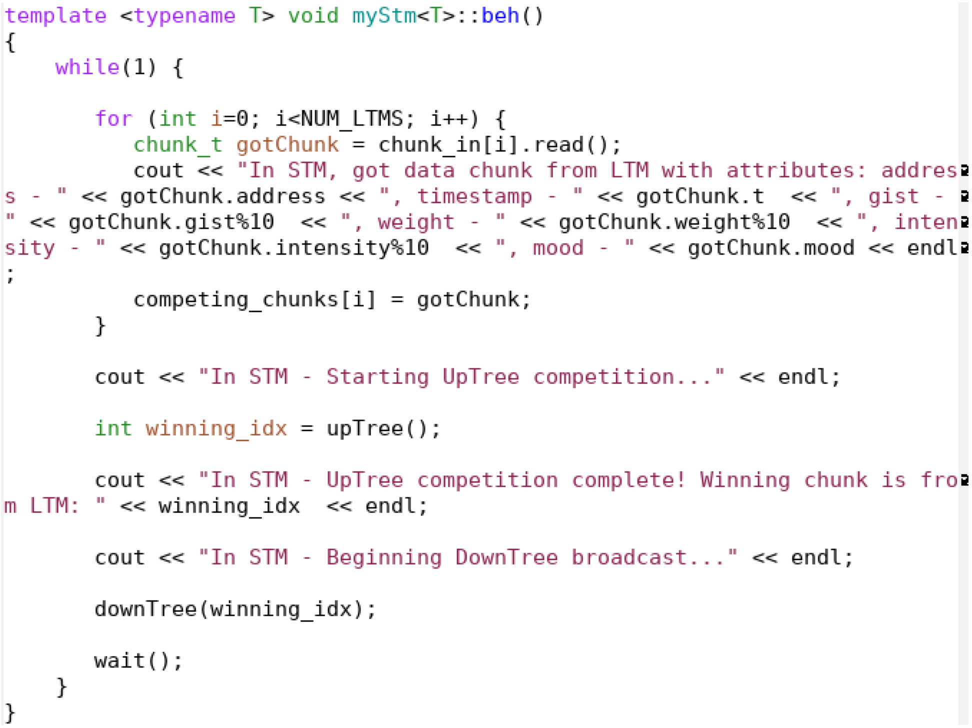
- *LTM*

Figures 2 and 3 depict the SC_THREAD processes defined in the LTM. The beh() thread in 2 is intended as a way of waiting on the got_data event until relevant information can be read in from the external environment through the ext_data_in port. Any subsequent manipulation of the data by the LTM can take place here, and can then be written to the ext_data_out port. Data will only be read by the LTM when there is data available.

The process_chunks() SC_CTHREAD in 3, on the other hand, does not wait on a particular signal, but rather reads in a new chunk of information on every clock cycle. Given the way CTHREADs are defined in SystemC, there is no need to explicitly call on the clock edge before reading in or writing information to a port. It is sufficient to have a while loop that remains active throughout the entire simulation, and the wait() statement will ensure that the process logic will execute again on the next clock cycle.

- *STM*

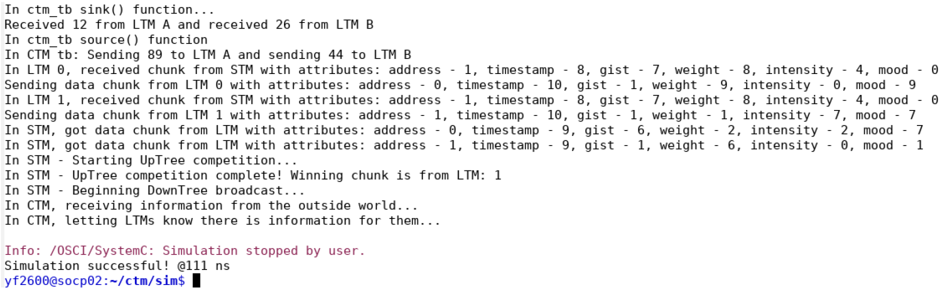
  In the STM, the primary beh() in 5 clocked thread is responsible for reading in the chunks from every LTM, and initiating both the upTree() competition and downTree broadcast mechanism.



```
template <typename T> void myStm<T>::beh()
{
    while(1) {

       for (int i=0; i<NUM_LTMS; i++) {
          chunk_t gotChunk = chunk_in[i].read();
          cout << "In STM, got data chunk from LTM with attributes: addres
s - " << gotChunk.address << ", timestamp - " << gotChunk.t  << ", gist - 
" << gotChunk.gist%10  << ", weight - " << gotChunk.weight%10  << ", inten
sity - " << gotChunk.intensity%10  << ", mood - " << gotChunk.mood << endl
;
          competing_chunks[i] = gotChunk;
       }

       cout << "In STM - Starting UpTree competition..." << endl;

       int winning_idx = upTree();

       cout << "In STM - UpTree competition complete! Winning chunk is fro
m LTM: " << winning_idx  << endl;

       cout << "In STM - Beginning DownTree broadcast..." << endl;

       downTree(winning_idx);

       wait();
    }
}
```

Figure 5: beh() clocked thread in STM

Finally, 6 shows a snippet of the output which flushes to console when the ‘make run’ command is executed. It demonstrates the flow of information from the ctm_tb (the outside world), to the LTMs, in addition to the flow of chunks to and from the STM from the LTMs. One important thing to note is the interleaving between the processes taking place between the LTMs and STMs, and the process which is responsible for communicating information to and from the outside world.



```
In ctm_tb sink() function...
Received 12 from LTM A and received 26 from LTM B
In ctm_tb source() function
In CTM_tb: Sending 89 to LTM A and sending 44 to LTM B
In LTM 0, received chunk from STM with attributes: address - 1, timestamp - 8, gist - 7, weight - 8, intensity - 4, mood - 0
Sending data chunk from LTM 0 with attributes: address - 0, timestamp - 10, gist - 1, weight - 9, intensity - 0, mood - 9
In LTM 1, received chunk from STM with attributes: address - 1, timestamp - 8, gist - 7, weight - 8, intensity - 4, mood - 0
Sending data chunk from LTM 1 with attributes: address - 1, timestamp - 10, gist - 1, weight - 1, intensity - 7, mood - 7
In STM, got data chunk from LTM with attributes: address - 0, timestamp - 9, gist - 6, weight - 2, intensity - 2, mood - 7
In STM, got data chunk from LTM with attributes: address - 1, timestamp - 9, gist - 1, weight - 6, intensity - 0, mood - 1
In STM - Starting UpTree competition...
In STM - UpTree competition complete! Winning chunk is from LTM: 1
In STM - Beginning DownTree broadcast...
In CTM, receiving information from the outside world...
In CTM, letting LTMs know there is information for them...

Info: /OSCI/SystemC: Simulation stopped by user.
Simulation successful! @111 ns
yf2600@socp02:~/ctm/sim$ █
```

Figure 6: CTM logfile snippet

4.2 Extension of the STM model

As we have seen from psychological evidence, some of the selection of attention is made late in the conscious process. Therefore, in expanding the STM model, we have most importantly allowed for several parallel chunks in the STM (see Figure 7). This models the balance of early and late selection of attention, as the Up-Tree still decides which chunks arrive at the STM, but the STM can make a "decision" depending on several parameters on which chunk to assign how much attention. Speaking in terms the authors used, this will provide an improved feeling of consciousness, as the conscious part of the CTM is to a certain extend actively involved in deciding over attention.

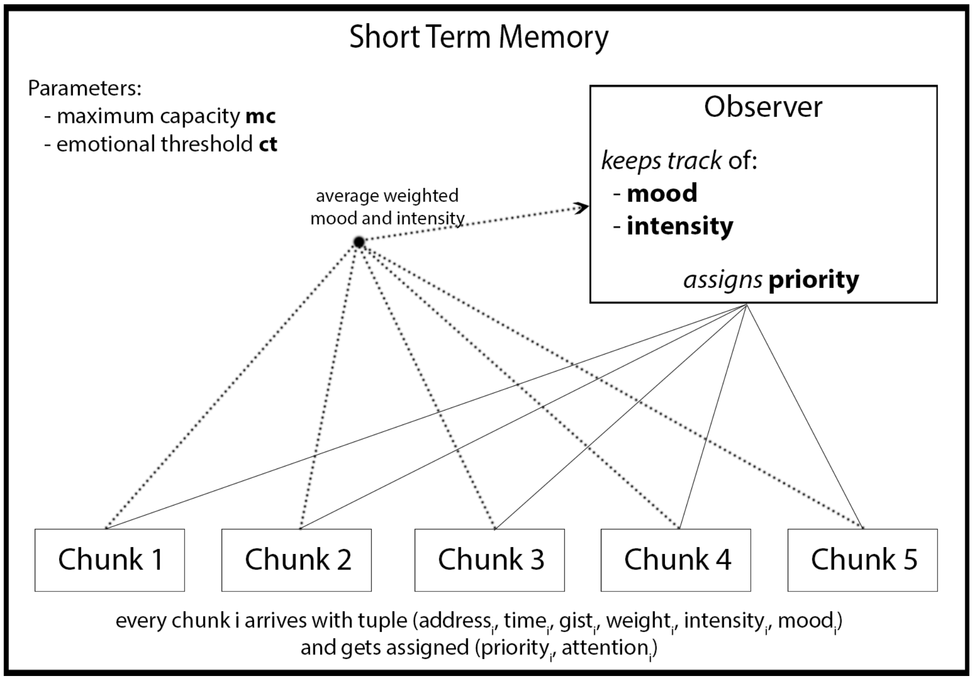

Short Term Memory
Parameters:
- maximum capacity **mc**
- emotional threshold **ct**
Observer
keeps track of:
- **mood**
- **intensity**
assigns **priority**
average weighted mood and intensity
Chunk 1
Chunk 2
Chunk 3
Chunk 4
Chunk 5
every chunk i arrives with tuple (address$_i$, time$_i$, gist$_i$, weight$_i$, intensity$_i$, mood$_i$)
and gets assigned (priority$_i$, attention$_i$)

Figure 7: Extended STM model

Such modeling requires some kind of active algorithm in the STM which takes part in decision making. We call this algorithm the *Observer*. It should be mentioned in advance that this Observer does not have full power over which arrived chunk gets the most attention. It only assigns logically / cognitively evaluated priorities that compete with the emotional importances described by mood and intensity. We introduce two additional parameters that are assigned to every chunk in the STM itself: *priority* and *attention*. Priority is such a cognitive value which is assigned based on current perception and action. Intuitively, it can be regarded as a measurement of what we know we "should" pay attention to given the circumstances. For example, we evaluate that we should study, but this cognitive judgement competes with emotional value of other chunks. The priority is evaluated in combination with other chunks that give sensory input, but we won't define such an algorithm here. Attention is an implicitly assigned parameter of how much attention the STM pays to a chunk dependent on the other attention values and the current *maximum capacity* (mc).

Incorporating the perceptual load theory [11], every individual has limited resources of which all are used at any given time. The maximum capacity (mc) denotes the maximum resources available. We introduce this additional variable as the maximum capacity might change over time. Every chunk is assigned an additional parameter $attention_i$ which describes how much attention is paid to a certain chunk or how many resources it claims. As the maximum capacity is always completely exhausted at any time t, the following holds true:

$$mc = \sum_{i=1}^{\#chunks} attention_i \tag{1}$$

Instead of assigning mood and intensity to chunks and propagating them with the chunks from STM to LTMs and back, we assign the Observer with a current mood and intensity. Not only do we believe that this models the CTM's consciousness closer to the brain, but it is also required when enabling several chunks in the STM as the current mood is ambiguous when still held by the chunks. Therefore, these parameters are continuously changed by the incoming chunks by calculating the weighted average in the following way:

$$mood_O = weight * mood_O + \frac{(1 - weight)}{\#chunks * mc} * \sum_{i=1}^{\#chunks} mood_i * attention_i \quad , weight \in [0,1] \tag{2}$$

$$intensity_O = weight * intensity_O + \frac{(1 - weight)}{\#chunks * mc} * \sum_{i=1}^{\#chunks} intensity_i * attention_i \quad , weight \in [0,1] \tag{3}$$

The weight parameter decides over how much the mood and intensity of the previous discrete time point *t - 1* influences the current parameters. If it turns out that the last mood does not affect the current one, the weight would be 0. This additionally expands the notion of the stream of consciousness. We assume that the current mood and intensity is influenced more by chunks which the STM pays more attention to, which is why the mood and intensity are each multiplied by the attention. To take the average and normalize the multiplication of the attention, we divide the whole sum by the number of chunks and the maximum capacity.

With this model as framework, we have basically incorporated the late selection of attention as well. The only remaining question how it is decided which chunk to pay attention to. One possible implementation we suggest is a competition between the emotional state consisting of (mood, intensity) and assigned priorities through the fixed parameter *emotional threshold (et)*: et is a simple threshold value which defines the point at which the emotional state wins over the priority value. This implicitly determines, without any control by the Observer, how the priorities are assigned to the chunks.

5 Discussion

In the process of comprehending Blum & Blum's modeling decisions, we had to face the fact that they did not provide reasoning based on empirical research in psychology and neuroscience. While most of their modeling decisions are based on the GWT, some additional components seem arbitrary in terms of the GWT. According to our research, their abstractions reflect the current state of research to a limited extent. We understand that many abstractions are necessary for a first model of a conscious AI. Still, we are convinced that some crucial aspects constituting the mechanisms of human consciousness are abstracted too much and would be crucial to modeling machine consciousness.

With our changes to the STM of introducing an observing actor which influences late selected attention, the effects do not remain local. This means that it would be necessary to make some changes to other components of the LTM to enable the possibility to process several chunks in the STM in parallel. Blum and Blum have decided to only allow one chunk in the STM for the sake of simplicity that the whole system could focus on only one thought, idea, situation, or the like. With the introduction of several chunks, this simplicity is not given anymore. The algorithms of the LTMs would have to be changed so parallel thoughts can be considered, and some connection or influence between them would model human consciousness more realistic. For example, when the STM broadcasts the chunks back to the LTM to which it had paid at least some attention, the algorithms of the LTMs have to be adapted such that they also incorporate attention. This consideration would have a different, more convoluted effect on the chunks subsequently released for competition in the Up-Tree.

Regarding the SystemC implementation of the CTM, the STM was implemented in such a way that the STM is privy to all the chunks' information prior to commencing the competition process. This is in contrast to the formal model where the STM possesses only the winning chunk which it then broadcasts to all LTMs. It was implemented in such a manner so that that our proposed extensions to the STM could be integrated more easily.

References

[1] Alan Allport. "Visual attention". In: *Foundations of cognitive science*. Cambridge, MA, US: The MIT Press, 1989, pp. 631–682. ISBN: 0-262-16112-5.

[2] Bernard J Baars. "In the theatre of consciousness. Global workspace theory, a rigorous scientific theory of consciousness". In: *Journal of consciousness Studies* 4.4 (1997), pp. 292–309.

[3] Bernard J Baars. “The conscious access hypothesis: origins and recent evidence”. In: *Trends in cognitive sciences* 6.1 (2002), pp. 47–52. DOI: 10.1016/S1364-6613(00)01819-2.

[4] Bernard J. Baars. “Global workspace theory of consciousness: toward a cognitive neuroscience of human experience”. In: *The Boundaries of Consciousness: Neurobiology and Neuropathology.* Vol. 150. Progress in Brain Research. Elsevier, 2005, pp. 45–53. ISBN: 9780444518514. DOI: 10.1016/S0079-6123(05)50004-9.

[5] Manuel Blum and Lenore Blum. “A Theoretical Computer Science Perspective on Consciousness”. In: *CoRR* abs/2011.09850 (2020). arXiv: 2011.09850.

[6] D. E. Broadbent. “Effect of Noise on an “Intellectual” Task”. In: *The Journal of the Acoustical Society of America* 30.9 (1958), pp. 824–827. DOI: 10.1121/1.1909779. eprint: https://doi.org/10.1121/1.1909779.

[7] E Colin Cherry. “Some experiments on the recognition of speech, with one and with two ears”. In: *The Journal of the acoustical society of America* 25.5 (1953), pp. 975–979.

[8] Jan De Fockert. “Beyond perceptual load and dilution: a review of the role of working memory in selective attention”. In: *Frontiers in Psychology* 4 (2013), p. 287. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2013.00287.

[9] J. A. Deutsch and D. Deutsch. “Attention: Some theoretical considerations”. In: *Psychological Review* 70.1 (1963), pp. 80–90. DOI: 10.1037/h0039515.

[10] Christof Koch et al. “Neural correlates of consciousness: progress and problems”. In: *Nature Reviews Neuroscience* 17.5 (2016), pp. 307–321. ISSN: 1471-0048. DOI: 10.1038/nrn.2016.22.

[11] Nilli Lavie. “Perceptual load as a necessary condition for selective attention”. In: *Journal of Experimental Psychology: Human Perception and Performance* 21.3 (1995), pp. 451–468. DOI: 10.1037/0096-1523.21.3.451.

[12] Karina Linnell and Serge Caparos. “Perceptual load and early selection: an effect of attentional engagement?” In: *Frontiers in Psychology* 4 (2013), p. 498. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2013.00498.

[13] George A. Mashour et al. “Conscious Processing and the Global Neuronal Workspace Hypothesis”. In: *Neuron* 105.5 (2020), pp. 776–798. ISSN: 0896-6273. DOI: https://doi.org/10.1016/j.neuron.2020.01.026.

[14] Neville Moray. “Attention in Dichotic Listening: Affective Cues and the Influence of Instructions”. In: *Quarterly Journal of Experimental Psychology* 11.1 (1959), pp. 56–60. DOI: 10.1080/17470215908416289. eprint: https://doi.org/10.1080/17470215908416289.

[15] Giulio Tononi. “An information integration theory of consciousness”. In: *BMC Neuroscience* 5.1 (2004), p. 42. ISSN: 1471-2202. DOI: 10.1186/1471-2202-5-42.

[16] Anne M Treisman. “Selective attention in man”. In: *British medical bulletin* 20.1 (1964), pp. 12–16. DOI: 10.1093/oxfordjournals.bmb.a070274.

[17] A. M. Turing. “On Computable Numbers, with an Application to the Entscheidungsproblem”. In: *Proceedings of the London Mathematical Society* s2-42.1 (1937), pp. 230–265. DOI: https://doi.org/10.1112/plms/s2-42.1.230. eprint: https://londmathsoc.onlinelibrary.wiley.com/doi/pdf/10.1112/plms/s2-42.1.230.

[18] A. T. Welford. “The ’psychological refractory period’ and the timing of high-speed performance—a review and a theory”. In: *British Journal of Psychology* 43 (1952), pp. 2–19.