A Machine Consciousness Approach to Urban Traffic Control

André Luis O. Paraense¹, Klaus Raizer¹, and Ricardo R. Gudwin¹
University of Campinas, School of Electrical and Computer Engineering, Campinas, SP, Brazil

Abstract
In this work, we present a distributed cognitive architecture used to control the traffic in an urban network. This architecture relies on a machine consciousness approach - Global Workspace Theory - in order to use competition and broadcast, allowing a group of local traffic controllers to interact, resulting in a better group performance. The main idea is that the local controllers usually perform a purely reactive behavior, defining the times of red and green lights, according just to local information. These local controllers compete in order to define which of them is experiencing the most critical traffic situation. The controller in the worst condition gains access to the global workspace, further broadcasting its condition (and its location) to all other controllers, asking for their help in dealing with its situation. This call from the controller accessing the global workspace will cause an interference in the reactive local behavior, for those local controllers with some chance in helping the controller in a critical condition, by containing traffic in its direction. This group behavior, coordinated by the global workspace strategy, turns the once reactive behavior into a kind of deliberative one. We show that this strategy is capable of improving the overall mean travel time of vehicles flowing through the urban network. A consistent gain in performance with the “Artificial Consciousness” traffic signal controller during all simulation time, throughout different simulated scenarios, could be observed, ranging from around 13.8% to more than 21%.

Keywords: Global Workspace Theory, Traffic Lights Control, Machine Consciousness, Codelets

1 Introduction
Traffic is one of the biggest problems faced by many big cities. One approach to reduce this problem is the use of adaptive traffic light controllers, able to change its control policy based on local information. Even in the case when they are not able to completely solve traffic problems, they produce significant improvements without the need to change current infrastructure or transportation models.

*This is a preprint. For the published version, please access http://dx.doi.org/10.1016/j.bica.2015.10.001
In this scenario, the control and optimization of traffic lights phases is a key topic in improving cities traffic conditions (Brockfeld, Barlovic, Schadschneider, & Schreckenberg, 2001; Sánchez-Medina, Galán-Moreno, & Rubio-Royo, 2010; Srinivasan, Choy, & Cheu, 2006). For large urban networks, though, there is a prohibitive number of variables, states, stochastic aspects, uncertainty, interactions between subsystems, mutually exclusive goals, and other issues (Guberinic, Senborn, & Lazic, 2008), which make scenarios like these nearly impossible to be solved with conventional strategies.

Some approaches consider the problem only locally, restricted only to a small number of traffic lights (Sik, Soo, Kwang, & Kug, 1999). However, when dealing with the whole urban network, because of the non-linear and stochastic events which happen in the network and their inter-dependencies, the actual state of traffic becomes hard to assess and the effects of changes in traffic control becomes almost impossible to forecast (Srinivasan et al., 2006). Recent works investigated different approaches to this problem, such as dynamic programming (Heydecker, Cai, & Wong, 2007; Heung, Ho, & Fung, 2005), neuro-fuzzy networks (Choy, Srinivasan, & Cheu, 2003) and reinforcement learning (Cai, Wong, & Heydecker, 2009). Box & Waterson (Box & Waterson, 2013) developed one traffic light controller which learns strategies based on previous experience. They used human experts to control a single microscopic traffic simulation of an area in Southampton’s urban road network. The researchers used the experts’ decisions to train a neural network, which was later used to control the simulation and achieved better results than earlier applied algorithms and benchmarks.

This paper is organized as follows. In Section 2 we include some background about Global Workspace Theory, and the CST - the Cognitive System Toolkit - which is being developed by our research group, and is used as the main basis for the construction of the cognitive architecture which will be controlling our traffic lights. In Section 3 we present the materials and methods for our experiments, describing the traffic simulation tool we used and some details about how it models urban traffic networks. In this section, we also start the description of the cognitive architecture controlling the traffic lights by using Global Workspace Theory. In Section 4 we present the main results we obtained with our simulations, and in Section 5 we provide a discussion for these results and the main conclusions.

2 Background

2.1 Global Workspace Theory

Global Workspace Theory (GWT) (B. J. Baars, 1988) is a cognitive theory which tries to explain the phenomena of consciousness in the human brain, and is largely inspired by the “blackboard model” from the beginning of artificial intelligence (Nilsson, 1986). Due to its computational origins, it is, among other consciousness theories, one which is particularly interesting for deriving computational models, being very popular in the field of “machine consciousness”. GWT suggests that only one integrated sensory content can be dominant in the brain in a given moment. Potentially conscious content compete for access to the limited capacity of this workspace. This dominant content is then broadcasted to other regions of the brain, in a nervous system seen as a set of massive distributed small networks with specialized purpose. In such a system, coordination, control and resolution of new problems take place with the exchange of central information. This theory tries to conciliate the limited capacity of conscious content with the vast repertoire of long term memories. It states that the limited capacity of conscious content brings advantages to animal survival, because it helps the animal to focus in what is most important in a given critical situation. The concept of a dynamic core provides a
mechanism for events in the Global Workspace, as it projects brain signals in the cortex in a reentrant manner (Edelman, Gally, & Baars, 2011).

In vertebrate mammals consciousness is a dynamic, integrated, multimodal mental process (Fabbro, Aglioti, Bergamasco, Clarici, & Panksepp, 2015). The scientific hypothesis for neural correlates sufficient for this process is that they were naturally selected during animal evolution because they permitted animals to plan for future events and deal with unexpected situations they had never experienced before, in a complex and ever changing environment (Crick & Koch, 2003; Edelman et al., 2011; B. Baars & Franklin, 2009). The main advantages brought by such a mechanism are twofold. First, there is an executive summary of perceptions, generating an unique and integrated content from all perceptual information at a given moment. The most relevant information for the animal survival becomes conscious, enabling it to better deal with unexpected and novel situations that differ from an original plan. Second, there is automation and deautomation of behaviours. In automation, as novel situations become more and more frequent, conscious content is stored in long term memory, becoming accessible for planning and making predictions. Once automated, action selection can happen without conscious interference. However, if an automated behaviour produces unexpected results, consciousness regains control of action selection and information processing, which might result in the deautomatization of this previous behavior. This mechanism makes animal behavior extremely adaptive to changing environments.

Traffic lights control in an urban network can be seen as a set of subsystems operating in parallel, where each subsystem is a single junction composed of $n$ traffic lights influencing and being influenced by its neighbor subsystems. In most cases, each subsystem is operated in isolation. However, for the network to function properly, it would be interesting to have these subsystems interacting in a way that critical situations might be avoided, such as in deadlocks and big traffic jams. This scenario is similar to what is found in the animal body, where different isolated subsystems are coordinated by an executive control nervous system. In this central executive mechanism, consciousness can be viewed as a supervisor process that takes care of many semi-autonomous subsystems.

The scientific hypothesis in this work is that an artificial consciousness mechanism, inspired by GWT, can bring advantages to automatic processes, such as urban traffic lights control.

### 2.2 The CST Cognitive Systems Toolkit

With improvements in the development of intelligent agents, the reuse of specific cognitively inspired strategies and designs has brought about the development of the field of Cognitive Architectures (Langley, Laird, & Rogers, 2009; Samsonovich, 2012), a special kind of control systems using inspiration on how cognitive functions from the human (or animal) mind might be computationally implemented and used to provide intelligence to an artificial agent. There are mainly two kinds of cognitive architectures: frameworks and toolkits. Frameworks provide reuse of previous designs by reapplying the same code in new applications. Toolkits provide a more flexible kind of reuse, when particular cognitive functions and strategies can be chosen in order to build up customized architectures. In this work, we used the Cognitive Systems Toolkit (CST), a toolkit being developed by our research group, and used previously for the development of a neuroscience inspired cognitive architecture (Raizer, Paraense, & Gudwin, 2012), with applications to robotics and assistive technology (Raizer, Rohmer, Paraense, & Gudwin, 2013). In the present work, we extended CST by including a GWT-based consciousness subsystem, and applying it in the implementation of a traffic light controller.

Figure [1] illustrates the core of the CST toolkit. Basically, the CST is a toolkit for the
construction of specialized cognitive architectures. However, all cognitive architectures built with the help of CST share a common core of structures and concepts. Even though different strategies might be chosen for implementing cognitive functions like perception, memory, action selection, etc, these cognitive functions necessarily will be constructed with the help of this common core. We will be referring to this generic kind of cognitive architecture, being constructed using CST, as the CST Architecture. The two basic concepts, fundamental for understanding the CST Architecture, are the concepts of “Memory Object” and “Codelet”. Memory objects are any kind of data structure used to store information and/or knowledge. Using a semiotic terminology, we might refer generically to a memory object as being a “sign”. A memory object has a type $T$, and an encoding of information $I$. This information can be a single measurement, expressed by a number, or a complex data container, which structure is completely defined by the definition of its type $T$. Codelets are small pieces of non-blocking code, executing a well defined and simple task. The prototype of a codelet is a piece of code which ideally shall be executed continuously and cyclically, time after time, being responsible for the behavior of an independent component of a system running in parallel. The notion of codelet was introduced originally by Hofstadter and Mitchell [Hofstadter & Mitchell, 1994] and further enhanced by Franklin [Franklin, Kelemen, & McCauley, 1998]. The CST architecture is “codelet oriented”, since all main cognitive functions are implemented as codelets. This means that from a conceptual point of view, any CST-implemented system is a fully parallel asynchronous multi-agent system, where each agent is modeled by a codelet. CST’s codelets are implemented much in the same manner as in the LIDA cognitive architecture [Franklin, Madl, D’mello, & Snaider, 2014] and largely correspond to the special-purpose processes described by Baar’s Global Workspace Theory [B. J. Baars & Franklin, 2007]. Nevertheless, for the system to work, a kind of coordination must exist among codelets, forming coalitions which by means of a coordinated interaction, are able to implement the cognitive functions ascribed to the architecture. This coordination constraint imposes special restrictions while implementing codelets in a serial computer. In a real parallel system, a codelet would simply be called in a loop, being responsible to implement the behavior of a parallel component.

A codelet has two main inputs (which are characterized as $In$ and $B$ in Figure 1), a local input ($In$) and a global input ($B$). The local input is used for the codelet to get information
from memory objects, which are available at Raw Memory. The global input is used for the codelet to get information from the global workspace mechanism \cite{Baars&Franklin2007}. Information coming from global workspace is variable at each instant of time, and is usually in the form of a summary, which works as an executive filter to select the most relevant pieces of information available in memory at each time-step. The two outputs of a codelet are a standard output ($Out$), which is used to change or create new information in the Raw Memory, and the activation level ($A$), which indicates the relevance of the information provided at the output. This activation level is also used by the consciousness mechanism in order to select information to be destined to the global workspace.

Using this Core, the CST toolkit provides different kinds of codelets to perform most of the cognitive functions available at a cognitive architecture, as indicated in Figure 2. Also, memory objects can be part of different kinds of memories. Raw Memory is therefore split into many different memory systems, which are used to store and access different kinds of knowledge. With specific codelets for perception, attention, learning, planning and behavior generation, one can design different cognitive architectures to address specific control problems. These codelets are constructed according to different techniques from the realm of intelligent systems, such as neural networks, fuzzy systems, evolutionary computation, rule-based systems, Bayesian networks, etc. These are in turn integrated into a cohesive control and monitoring system. The definition and choice of a particular cognitive architecture is constructed using a composition of different kinds of codelets, according to the control problem under analysis. Depending on the problem to be addressed, different strategies might be necessary or useful, depending on the problem constraints.
3 Materials and Method

3.1 Materials

3.1.1 Traffic Simulator

Experiments took place in a computationally simulated environment using the “Simulation of Urban MObility” (SUMO) traffic simulator (Krajzewicz, Erdmann, Behrisch, & Bieker, 2012). SUMO is a free and open traffic simulation suite which is available since 2001. It allows modelling of intermodal traffic systems including road vehicles, public transportation and pedestrians. In this work we used microscopic simulation of vehicles and online interaction through the TraCI API to remotely control the time schedules of traffic lights during simulations.

3.1.2 Network Models - Test Bed

Three network models were tested in this work. One is called “Simple T”, which is composed by a single junction linking three main roads, as shown in Figure 3.

The second model, called “Corridor”, is a little more complex than the “Simple T”, being composed by four junctions connecting many roads, as shown in Figure 4.

The third model is the most complex, called ”Manhattan”, a grid composed by 9 junctions, as shown in Figure 5.

3.1.3 Traffic Controllers

We tested three different traffic controllers in our experiments. The first one, called “Fixed Times”, is the simplest and most common traffic controller. It has a fixed traffic signal plan,
Figure 4: Network model “Corridor”, where the experiments took place.

Figure 5: Network model “Manhattan”, where the experiments took place.
independently from the traffic in its controlled lanes, changing the traffic lights in regular fixed periods called “phases”.

The second and third ones, called “Parallel Reactive” and “Artificial Consciousness”, respectively, are built using CST. Both of them rely in sensory information from vehicles traveling in the crossing lanes in order to choose one phase among \( n \) possible phases with red and green lights for each lane in the junction. Figure 6, which is a zoom of the first two junctions from left to right in Figure 4, details the many sensed regions and the different possible phases which can be chosen by the controller in these junctions.

The controllers built with the CST can be modeled for this simplified situation as shown in Figure 7.

There are four types of codelets in our model:

- **Sensory Codelets**: for each set of inductive sensors in the region in front of the traffic light there is one codelet whose output is the pair of vectors \( X_c \), holding the position of all vehicles in that region, and \( V_c \), holding the velocities of each one of the vehicles.

- **Behavioral Codelets**: for each junction there is one codelet whose inputs are positions and velocities for the vehicles in front of the traffic lights (plus the broadcast from consciousness), and output is one of the traffic lights possible phases for the junction.

- **Consciousness Codelet**: responsible for selecting the sensory content to gain consciousness and for the broadcast of conscious content among all codelets.

- **Motor codelets**: for each junction, there is one codelet whose input is a chosen phase for
The “Parallel Reactive” controller is automatic or “unconscious”, in our analogy to the animal brain. Each sensory codelet captures the vehicles position and velocity in its respective lane continuously, while the junction codelets perform the following algorithm:

1. Calculates its level of activation, which is given by equation (1)

   \[ a(t) = \frac{\sum_{c \in C} (1 - \alpha V_c(t) - \beta X_c(t))}{|C|} \]  

2. Determines the best phase among the possible ones based on a simple calculation as shown in Table 1.

3. Goes back to 1.

In equation (1) \( C \) is the set of all vehicles monitored by the inductive sensors in one junction; \( V_c \) is the vehicle’s velocity and \( X_c \) is the vehicle's distance to the junction; \( \alpha \) and \( \beta \) are constants that can be tuned to adjust the influence in the junction’s codelet activation, based on

Figure 7: Controllers built on top of CST
Table 1: Action selection in the Junction East codelet. In this example, phase number 3 was selected because it gives the best sum of green lanes activations.

<table>
<thead>
<tr>
<th>1. Heuristic activation for each lane</th>
<th>2. Overall Junction East activation</th>
<th>3. Determination of the best phase - sum of activations of green traffic lights</th>
</tr>
</thead>
<tbody>
<tr>
<td>( AT = \sum_{c \in C}(1 - \alpha V_c(t) - \beta X_c(t)) )</td>
<td>( \sum_{i=1}^{n} \frac{AT(i)}{n} )</td>
<td>( \sum_{tl \in G} AT(tl) ) (4)</td>
</tr>
<tr>
<td>AT(g) = 0.09</td>
<td>ATJe = 0.858</td>
<td>Possible phases</td>
</tr>
<tr>
<td>AT(h) = 0.3</td>
<td></td>
<td>1. G,G,R,G,G = 0.57</td>
</tr>
<tr>
<td>AT(i) = 0.85</td>
<td></td>
<td>2. G,R,G,R,R = 0.94</td>
</tr>
<tr>
<td>AT(j) = 0.05</td>
<td></td>
<td>3. R,R,G,R,G = 0.98</td>
</tr>
<tr>
<td>AT(k) = 0.13</td>
<td></td>
<td>Best phase</td>
</tr>
</tbody>
</table>

the number of vehicles, velocities and distances from the given junction. The closer to the junction and the slower the vehicles, the greater the codelet’s activation representing the junction. According to Box & Waterson (Box & Waterson, 2013), \( \alpha = 0.01 \text{m}^{-1} \) and \( \beta = 0.001 \text{m}^{-1} \) are values which result in a good balance between these influences and were, therefore, used in this work.

Finally, based on the chosen phase for each junction, the respective motor codelet modifies the junction traffic lights and the cycle is repeated. In the “Parallel Reactive” case, there is no Consciousness Codelet, and each junction codelet decides its phase based solely on the information they receive from their respective sensory codelets.

In the case of the “Artificial Consciousness” Controller, the only difference is the presence of the Consciousness codelet, which does the following:

1. Defines the junction codelet with greater activation level, which gains access to conscious global workspace while respecting a minimum threshold. If none of the codelets reaches the threshold, the system works unconsciously and global workspace remains empty.

2. Broadcasts the sensory information of the conscious codelet.

3. Goes back to 1.

Broadcast information contains details about how critical the situation is in the worst junction in the network controlled lanes. Other junctions receiving the broadcast will decide whether or not to use this information to choose its next phase based on the network topology by following two simple rules:

1. If the outgoing lanes of my traffic lights feed the incoming lanes of the conscious junction, I must close it with a red light.

2. If the incoming lanes of my traffic lights are fed by the outgoing lanes of the conscious junction, then I must open it with a green light.
The hypothesis is that these two simple rules should generate a behavior similar to dynamic green waves in the network whenever there is a critical situation, helping to solve the conflict as soon as possible, alleviating the situation until the flow becomes normal again.

For the sake of clarity, the example given in this section modeled only two of the four junctions in Figure 4. Nevertheless, the controller has the ability to model any urban network given in a digital format that can be read by SUMO, such as the Open Street Maps “.OSM” format [Haklay & Weber, 2008], for instance. In order to do so, in the beginning of the simulation, the controller reads the network model and creates the corresponding codelets: one behavioral codelet for each junction in the model, one sensory codelet for each lane controlled by each junction and one singleton consciousness codelet. These codelets are also attached to their corresponding memory objects, following the same architecture principles as the ones given in this section example: outputs of sensory codelets are attached as inputs to behavioral codelets, whose outputs are attached as inputs to motor codelets and the broadcast of the consciousness codelet is attached as an input to the behavioral codelets. The behavioral codelets keep a memory of the topology of the network model surrounding them (not represented in Figure 7), within a predefined radius (in this work we used 1 km), so they can answer whether they are connected or not to the conscious junction, which is important in order to interpret the broadcast information and decide whether or not to act upon it.

3.2 Method

For each network model, two simulated scenarios were considered. The first scenario, called “P = 0.1”, generates vehicles with random routes in a 0.1 second period during a time window of 5400 seconds, generating a very high concentration of vehicles coming from different sources of the network model and flowing to different edges. The edges probability of being assigned as the destination of one vehicle trip is weighted by length and number of lanes, so that larger avenues get more cars. The second scenario, called “P = 0.7” has the same attributes of the first one except for the vehicle generation period, which is 0.7 second, generating a lower traffic outcome. This gives a combination of 6 experimental scenarios, which were run 4 times each, summing up a total of 24 experiments reported in this paper. In each one of these experiments, the simulation was run for each one of three different controllers.

Each simulated experiment is run for as long as necessary, until all vehicles generated during the first 5400 seconds reach their final destination. During the simulation, the mean travel time of all vehicles' trips is measured and later plotted over time to compare the performance of the three controllers.

The choice of three network models and two concentration scenarios of simulation was intended to provide control groups for the algorithms and controllers being tested. In the case of the Simple T model, as there is only one junction, there is no use for the artificial conscious broadcast, since there are no other junctions to receive it. Hence, it is expected that both controllers, parallel reactive and artificial conscious ones, would behave in the same way. In this sense, “Simple T” represents a control group. The same analogy can be applied to scenarios where “P = 0.7” in the network models because, with lower traffic, codelets’ activations will rarely reach the conscious threshold. It is also expected that the controllers would have similar performances, also representing some sort of control group to evaluate the algorithms’ outputs.
4 Results

4.1 Simple T Model

Figure 8 shows the results of four simulation test cases considering scenario $P = 0.1$. Figure 9 shows the results of four simulation test cases considering scenario $P = 0.7$.

Figure 8: Model Simple T in scenario $P = 0.1$. Mean travel time is given in seconds in the vertical axis, and simulation time is given in hours in the horizontal axis. Panels 8a, 8b, 8c and 8d represent four distinct simulation experiments.
Figure 9: Model Simple T in scenario P = 0.7. Mean travel time is given in seconds in the vertical axis, and simulation time is given in hours in the horizontal axis. Panels (a), (b), (c), and (d) represent four distinct simulation experiments.

4.2 Corridor Model

Figure 10 shows the results of four simulation test cases considering scenario P = 0.1. Figure 11 shows the results of four simulation test cases considering scenario P = 0.7.
Figure 10: Model Corridor in scenario $P = 0.1$. Mean travel time is given in seconds in the vertical axis, and simulation time is given in hours in the horizontal axis. Panels 10a, 10b, 10c and 10d represent four distinct simulation experiments.
4.3 Manhattan Model

Figure [12] shows the results of four simulation test cases considering scenario P = 0.1. Figure [13] shows the results of four simulation test cases considering scenario P = 0.7.
Figure 12: Model Manhattan in scenario $P = 0.1$. Mean travel time is given in seconds in the vertical axis, and simulation time is given in hours in the horizontal axis. Panels (a), (b), (c), and (d) represent four distinct simulation experiments.
Figure 13: Model Manhattan in scenario P = 0.7. Mean travel time is given in seconds in the vertical axis, and simulation time is given in hours in the horizontal axis. Panels 13a, 13b, 13c and 13d represent four distinct simulation experiments.

5 Discussion

5.1 Main Findings

A consistent gain in performance with the “Artificial Consciousness” traffic signal controller during all simulation time, throughout different simulated scenarios, can be observed in Figure 10 and in Figure 12. By the end of the simulation, this gain is almost always around 13.8%, but it can represent more than 21% in specific times of the simulated scenarios. Due to the stochastic nature of simulated experiments, sometimes a smaller gain in performance is observed, as in Figure 10c, which shows a gain around 7%, but even in this case it is a relevant and consistent gain throughout the simulation.

As expected, there was no relevant difference in the behaviour of the “Parallel Reactive” and the “Artificial Consciousness” controllers, in the scenarios represented in Figure 8 and
which is an evidence that the presence of the GWT mechanism not only brings gains in performance in stress situations, but also does not interfere in situations where the network is under a normal flow, resembling the way consciousness interferes in the automatic unconscious processes in animal brain. In the cases of Figure 13a, 13b, 13c and 13d due to the complexity of the network model, even the concentration $P = 0.7$ was enough to produce lower stress situations.

5.2 Limitations and Future Work

The next steps for improving and testing our hypothesis are the following: working with varying conscious thresholds for codelet activation - which could be necessary for a less specialized controller - applying different heuristics in the unconscious automatic codelets, and working with other network models and scenarios, which should include real networks and data.

5.3 Conclusion

This work produced evidence to support the hypothesis that an artificial consciousness mechanism, which serially broadcasts content to automatic processes, can bring advantages to the global task performed by such a society of parallel agents working together for a common goal. A consistent gain in performance with the “Artificial Consciousness” traffic signal controller during all simulation time, throughout different simulated scenarios, could be observed, ranging from around 13.8% to more than 21%. Further work is necessary, though, to evaluate if these results are scalable to more complex traffic networks.

Acknowledgments: The first and second authors acknowledges CAPES, the brazilian research agency supporting graduate studies. The third author acknowledges FAPESP, the São Paulo state agency for funding support.

References


