DSO Cognitive Architecture: Implementation and Validation of the Global Workspace Enhancement

Khin Hua Ng(✉), Zhiyuan Du, and Gee Wah Ng

Cognition and Fusion Lab 2, DSO National Laboratories, Singapore, Singapore
{nkhinhua,dzhiyuan,ngeewah}@dso.org.sg

Abstract. An enhanced DSO Cognitive Architecture design was recently introduced to augment its cognitive functions by incorporating the Global Workspace Theory. A computational implementation of this new design is described in detail in this paper. The implementation is built as a distributed system with parallel pipelines of specialised processes, executing asynchronously. Competition initiated by these processes, and facilitated by the attention mechanism and global broadcast mechanism, leads to pipelines being dynamically created and allows disconnected pipelines to influence the processing of others. To validate the implementation, it was applied to a traffic control problem and experimental results showed increase in performance gain using the enhanced cognitive architecture.

Keywords: Cognitive architecture · Global Workspace Theory
Adaptive traffic control

1 Introduction

The DSO Cognitive Architecture (DSO-CA) [4] is a top-level cognitive architecture that incorporates the design principles of parallelism, distributed memory and hierarchical structure to model how the human brain processes information. It has been successfully used to develop Artificial Intelligence (AI) solutions to problems in applications like scene understanding [6] and mobile surveillance [5]. More recently, an enhanced design of the DSO-CA has been proposed [7] with the goal of enabling more human-like general intelligence and dynamic reasoning in AI systems. The design extension makes use of the Global Workspace Theory (GWT) [1] to enable Unified Reasoning — a process that permits reasoning across different knowledge domains and representations.

The motivation for unified reasoning is inspired by a cognitive architecture design problem known as the diversity dilemma [9] by which there is a need to blend diversity of different cognitive functions with uniformity of structure for efficiency, integrability, extensibility, and maintainability. The Global Workspace Theory is a neuro-cognitive theory of consciousness developed by

M. Iklé et al. (Eds.): AGI 2018, LNAI 10999, pp. 151–161, 2018.
https://doi.org/10.1007/978-3-319-97676-1_15

Bernard Baars [1]. It advances a model of information flow in which multiple, parallel, specialised processes compete and co-operate for access to a global workspace, which permits the winning coalition to broadcast to the rest of the specialist. By making use of an integrative memory system and applying the GWT, the newer DSO-CA with GWT design is able to facilitate collaboration among different cognitive functions and therefore indirectly provides a resolution to the diversity dilemma. Due to space constraints, we refer the reader to [7], for details on the design of the enhanced architecture, the inspirations drawn from the GWT, the principles behind the unified reasoning process using an integrative memory system, and the discussions on related cognitive architectures that had influenced the design. Nevertheless, to paint a clearer picture to the motivation behind the newer DSO-CA design, we will highlight two related work here, where the detailed comparisons are also given in the original paper. First, the diversity dilemma was discussed by Paul Rosenbloom and his answer to the dilemma is the SIGMA cognitive architecture that attempts to merge all the cognitive functions using a language representation which can be compiled into a common representation [10]. Second, Ben Goertzel formalised the concept of cognitive synergy [2], a framework that measures the compatibility and interaction between different cognitive functions (defined as knowledge creation mechanism that acts on a specific memory type), and how a cognitive function can help another when one gets 'stuck' if both functions have high compatibility. The point here is, these related works share a similar approach towards producing more general intelligence in AI systems, and the key is to work out how meaningful fusion and interaction among different cognitive functions can be achieved. The approach adopted by the DSO-CA is akin to creating a small-world network where cognitive processes that contribute to similar functionalities with respect to either agent's environment or it's task, form cliques amongst themselves due to frequent interactions. Communications between these cliques of disparate functionalities happen when the agent is met with a new or infrequent task. To solve this, the agent needs to chain different processes or cliques together dynamically through a GWT-inspired implementation. With that, it can create a platform for emergent, adaptive behaviours by allowing different pathways (learned or not) to communicate with one another through a common global workspace.

In this paper, we present a computational implementation of the DSO-CA [7] with GWT. The implementation is centred on the same distributed system principle whereby every specialised processor is executed as an independent parallel process with inter-process communication achieved by a message-oriented middleware (MOM). This means the system will have many pipelines executed in parallel, with some of them disconnected from one another. Competitions from these processes will either lead to pipelines being dynamically created, or allow disconnected pipelines to influence the processing of others. Full details of the implementation will be presented in the next section. In the section after that, we will discuss a successful validation of the implemented cognitive architecture applied to an urban traffic control problem. We will conclude the paper with

future work where we discuss about learning pathways between the different specialised processors.

2 Design and Implementation

An overview of the enhanced DSO-CA is shown in Fig. 1. There are three design aspects with respect to incorporating the GWT: (1) parallelised, specialised processes, (2) competition, (3) inhibitory function to suppress competition after a broadcast. For aspect (1), there are Cognitive Codelets, which are specialised functions as described in the GWT (e.g the Reasoners in Executive group in Fig. 1a). Communications between them are done through the Reference Memory Cells (RMCs) which act as interfaces to the Working Memory (Fig. 1b). These communications can be considered as pathways and each pathway can be independent from one another. Aspect (2) is initiated via bottom-up attention which starts with Cognitive Codelets and RMCs sending salient information to compete for global broadcast access. This is realised through the attention mechanism, which comprises of Triggers and Attention Codelets, and Global Broadcast Mechanism (GBM). The competition is multi-tiered and it starts with candidates competing at a localised, contextual level in each Attention Codelet, and finally competing in the GBM, the winner thereby gaining global broadcast access. After which, it is propagated through the system allowing it to influence relevant pathways. These pathways form a coalition which is a group of processes that are dynamically formed to address contextual matters within the system. Finally aspect (3) is achieved via suppressing competition at the attention mechanism, preventing any local competition from taking place. This results in a cooldown period for other Cognitive Codelets to process the global broadcast before the next round of competition is allowed.

To implement the design aspects and information flow, the DSO-CA is implemented as a distributed system consisting of parallel processes. An MOM facilitates the inter-process communications based on a publish-subscribe pattern — a

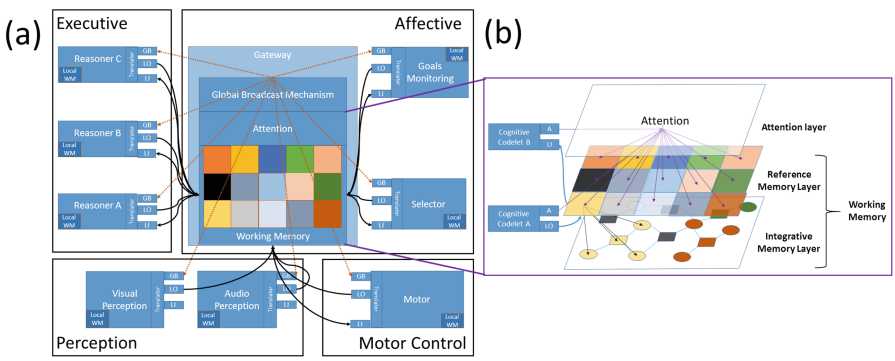

Fig. 1. (a) An overview of the DSO-CA. (b) The Working Memory zoomed in, with the Reference Memory Layer and Integrative Memory Layer.

process publishes its message to a topic and processes subscribed to it will receive the message. Additionally, all codelets are standardised to a multithreaded setup with threads following a producer-consumer pattern in concurrency design. Each thread is either a listener, processor, or sender; listener receives inputs and pre-processes them for the processor; processor represents a specialised function of which the codelet is designed for; and sender post-processes and sends the processor's result to designated codelets. The reasons for such a workflow are to be MOM-agnostic, and decouple preprocessing and post-processing from the main process so as to maximise time on it. Another benefit is code-reuse if the listener and sender threads are applicable to different codelets.

Cognitive Codelets are implemented as parallel processes that serve as specialised functions within the DSO-CA. Each has its own memory representation, which can differ from the integrative memory's. For example, Cognitive Codelets in Perception (Fig. 1a) can be different deep learning algorithms with different deep networks as their memory representations. The listener thread takes input from other RMCs and global broadcast (Fig. 1), it is also here where translation from integrative memory representation to local representation can take place. When a Cognitive Codelet receives a broadcast, it becomes a prioritised input which will be processed first even if the Cognitive Codelet receives a local input earlier. How the broadcast is processed depends on its relevance and the function of the Cognitive Codelet. For example, a Bayesian reasoner receiving a broadcast from a First Order Logic reasoner can use the conclusion within the result as an input to a what-if situation, i.e. diagnostic reasoning.

Reference Memory Cells constitute the Reference Memory Layer (Fig. 1b), with each RMC holding memory references to the integrative memory. In Fig. 2b, a RMC will merge the inputs into the underlying integrative memory by executing the transaction defined by the sender. This transaction includes adding, removing, refreshing (removing all references and adding new ones), or executing custom transactions. With regards to the references, different RMCs can refer to the same elements, thus changes to these elements are reflected to those RMCs referring to it. With that, a Cognitive Codelet can indirectly influence other Cognitive Codelets by modifying shared elements without needing a pathway. This ties in with the output where RMC will only send its reference memory to other Cognitive Codelets, meaning it only shares the relevant part of the integrative memory without the need to filter. To follow the design principle of distributed memory and parallelism, each RMC also executes in parallel and asynchronous manner for simultaneous transactions.

Attention Codelets represent contexts either abstracted from the goals, environment, or internal states of the system, e.g. survival for an embodied agent; external threats; imminent, and critical failure of other Cognitive Codelets. Each Attention Codelet's main purpose is to oversee competition within its context, and this translates to unique, localised competition that executes in parallel and asynchronously from one another. Each winner is the best representative for a context and is sent to the GBM for the final competition. The Attention

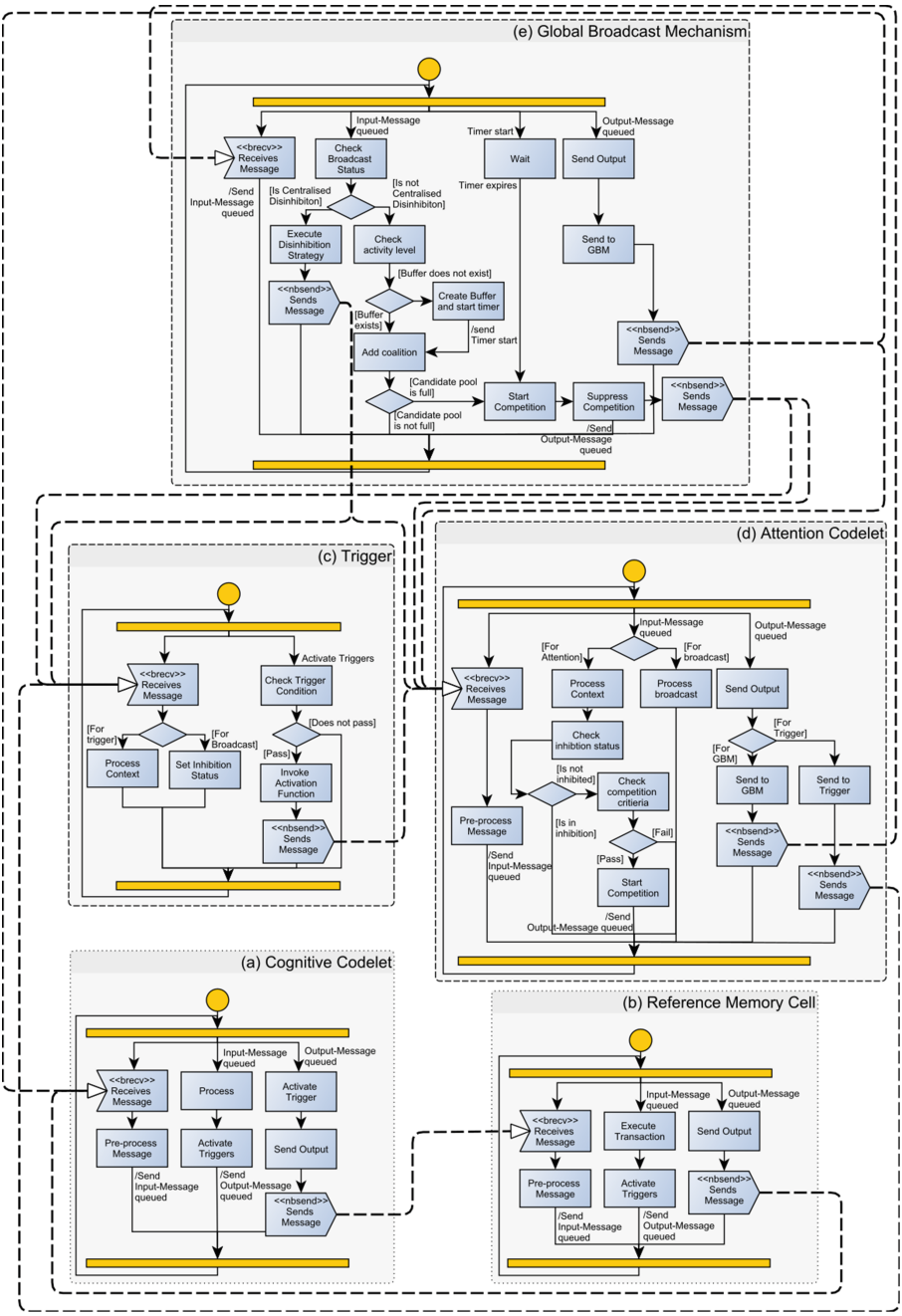

Fig. 2. Summarised activity diagrams of the DSO-CA with GWT. Dotted arrows indicate interprocess communication. Text with '/Send' prefix indicates the start of parallel thread e.g. '/Send Input-Message Queued' indicates the start of activity after 'Input-Message Queued'.

Codelet can also be affected by the GBM in the form of either inhibition signals or global broadcasts (Fig. 2d). An inhibition signal disables competition however, it can still monitor changes to its context by disinhibiting Triggers related to it. Meanwhile, global broadcast can modify the state of the context, for example unsuppressing competition on its own if the context has become critical with respect to the broadcast.

Triggers are special codelets that are part of the processor thread of a Cognitive Codelet or RMC of which both can have multiple Triggers. For bottom-up attention, Triggers compute the novelty or saliency of content sent by their attached codelet ('Activate Trigger' in Fig. 2) using an activation function, and send it along with the metadata to their Attention Codelets for competition. Each Trigger is assigned to only one Attention Codelet. Each Trigger also has a listener thread for either top-down attention messages from the Attention Codelets or inhibition signal from the GBM (Fig. 2c), which will reject any activation attempts until the Trigger is unsuppressed. Examples of top-down attention include Attention Codelet tasking a Trigger to adjust the activation level depending on the winner, or unsuppressing the Trigger's inhibition if the winning context requires special attention to its attached codelets.

Global Broadcast Mechanism (Fig. 2e) serves to broadcast the most salient content after a winner-take-all competition. There are two criteria to start competition: (1) activation level of all candidates must cross a GBM-set threshold, (2) either candidate buffer reached its limit or time to competition is up; the candidate buffer and timer is created when the first candidate is accepted. Following the GWT, the GBM will send inhibition signal to suppress all competitions before the broadcast. With regards to disinhibition, two strategies can be employed: centralised and decentralised disinhibition. In centralised disinhibition, the GBM controls it and maps every broadcast to a set of criteria that must be satisfied before the GBM unsuppresses competition. Thus, Attention Codelets in this scheme will switch to finding candidates that satisfy criteria relevant to their context. Under 'Execute Disinhibition Strategy' in Fig. 2e, the GBM will send disinhibition signals if these candidates met the criteria. For decentralised disinhibition, Attention Codelet determines disinhibition instead. Each Attention Codelet in this scheme will have their criteria to satisfy before continuing competition ('Process Context' in Fig. 2d).

3 Experiment

In this section, we present validation results from applying the DSO-CA implementation to an urban traffic control problem discussed in [8] whereby the CST group showcased the gain in performance using their cognitive architecture which also incorporated the GWT. It is clear that we have chosen to validate using the same problem because both architectures share a commonality on incorporating the GWT to enhance their architectures. The availability of the data and results presented in [8] also forms the baseline for our experiment. The experiment is

conducted on a simulation platform known as Simulation of Urban Mobility (SUMO) [3]. We used the same experiment data made available by the CST group which includes the road network (Fig. 3) and the routes of all vehicles, each route dictating the start, destination and time of insertion; and the activation function which will be elaborated below. The aim of the experiment is to reduce mean traveling time of each vehicle via controlling phases of a traffic controller which consists of all traffic lights in a junction. Phase in this case means the lights of the traffic controller and each light presides over an incoming lane, for example `GGrrGr` means green light for incoming lane 0, 1 and 4. For more details about the experiment, please refer to [8]. In addition to the original three phase selection schemes reported in the paper, we have designed two additional schemes made possible using our GWT implementation, which will further improve the performance gains.

Fixed Timing: Fixed phase cycle in which the traffic controller goes through a cyclic timed sequence of phases. Phases are predefined within the network.

Parallel Reactive (PR): The activation function of an incoming lane is as followed: $AT_l(t) = \sum_{c \in C}(1 - \alpha V_c(t) - \beta X_c(t))$ where l is the lane, t is the time, c is a vehicle, V_c and X_c are velocity and distance from the traffic light of the vehicle respectively, $\alpha = 0.01\,\mathrm{m}^{-1}$, and $\beta = 0.001\,\mathrm{m}^{-1}$. The activation value is an indication of how congested the junction is — the higher it is, the more congested it gets. Phases are from Fixed Timing and the phase selected is the highest activation value summed from the green lights of that phase [8].

Artificial Consciousness (PR-GWT): The junction activation value is calculated ($\frac{\sum_{l \in L} AT_l(t)}{|L|}$) [8] and this serves as the metric for competition. Junction with the highest activation value that passes a threshold will be selected as the critical junction by the GWT and be broadcasted to other traffic controllers. The traffic controllers whose lanes are within range to the critical junction will form a coalition and their phases will be generated based on the following rules: (1) critical junction's outgoing lane to any incoming lane will be given the green light. (2) critical junction's incoming lanes connected to any outgoing lanes are given red light. We shall call this generated phase, forced phase.

Projection Scheme: Built upon PR-GWT, this new scheme allows coalition of traffic controllers to compromise between the critical junction and their traffic by selecting a phase based on projected activation. Given a candidate phase, compare each light in it to the corresponding current phase's light (could be any of the phases above) and subtract ϵ to that lane's activation if the transition is red→green. Add ϵ for the inverse. If the projected value crosses a threshold, permutate all possible phases constrained by only flipping the red lights in the current phase e.g. `GrrG` will yield 4 different phases. The permuted phase whose projected activation value is closest to the threshold, will be selected.

Reactive Scheme: Similar to Projection Scheme except traffic controller initially follow the PR-GWT scheme. Once its current activation value crosses a threshold, it will change its phase. First, it inverts the lights in the forced phase

e.g `GrGGrrG` → `rGrrGGr`, this allows vehicles at the red light to move. Next, it selects a PR phase e.g `GrrrrrG`. This is to preserve green lights to those lanes that may still have significant traffic. Lastly, an OR operation is performed on the two phases e.g `GrrrrrG` ∨ `rGrrGGr` → `GGrrGGG`.

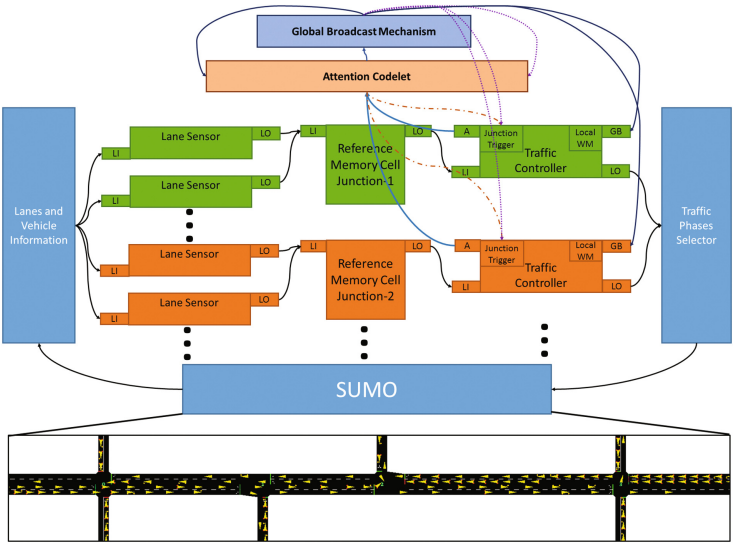

Fig. 3. Overview of the DSO-CA with Global Workspace applied to the traffic experiment.

Figure 3 shows the instantiation of the implementation for this experiment. Each traffic controller independently optimises its own traffic in parallel. When junctions get congested, the traffic controllers will start competing for broadcast access facilitated by the attention mechanism and GBM. Traffic controllers upon receiving the winner, will form a coalition if they are within reach to the critical junction and generate a forced phase to optimise for the critical junction's traffic. They will maintain it until the GBM signals that the activation level of the critical junction has fallen below a threshold, this can be considered as the **goal**. This also means that centralised disinhibition is used. Regarding the codelets in this instantiation, **Lane Sensor codelets** will retrieve the speed and distance of each vehicle to the traffic light on their respective lane, V_c and X_c at time step, t. Each **Reference Memory Cell** represents a junction. They will fuse inputs from Lane Sensors connected to their respective junction, and send to the Traffic Controller, V_c and X_c of each vehicle, c on that junction. When a **Traffic Controller Codelet** receives an update from a RMC, it selects the best phase and sends it back to SUMO. Under normal circumstances, PR is used to optimise local traffic. However if it is in a coalition, it will stick with the forced phase originally generated for PR-GWT. If Projection or Reactive scheme is used, the forced phase is regenerated after every update. The Traffic Controller

Codelet will go back to using PR when the goal is reached. For **Attention Codelet**, the context is the congestion of each junction, thus its competition criteria is to find the most congested junction. Competition is initiated when all Triggers have sent their candidates. When it knows of the critical junction, the Attention Codelet will need to monitor its activation level to meet the goal. To do that, it will disinhibit the Junction Trigger presiding over the critical junction (dashed-dot line in Fig. 3), this is a form of top-down attention. Related to the Attention Codelet, each **Junction Trigger** is attached to a Traffic Controller Codelet. For bottom-up attention, it will send the junction activation value to the Attention Codelet (Fig. 3). The functionality remains the same when one of them is disinhibited by the Attention Codelet. As for the **Global Broadcast Mechanism**, its implementation follows the Design and Implementation section.

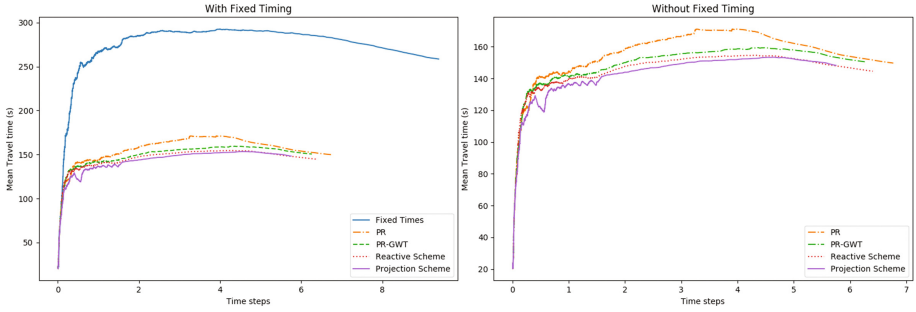

Fig. 4. Experiment results. Graph on the right is zoom-in-view without Fixed Timing.

Figure 4 shows the results of running the experiment using the "Corridor" traffic network model [8] with the setup of vehicles added every 0.1 s in SUMO. Using PR as a baseline, the performance gain for PR-GWT has an average of 3.8% with maximum value of 8.6%. This percentage improvement is comparable to that reported in [8] and it serves as validation for the correctness of our implementation. In addition, our proposed Reactive Scheme reduced the mean travel time even further by 5.9% on average, with up to a maximum of 10.9%. Furthermore, the proposed Projection Scheme has the best result: 7.1% average reduction, with maximum value of 15%. The experiment has successfully demonstrated that a dynamic, collaborative interaction can emerge through incorporating the GWT — pathways leading to Traffic Controller Codelets never interact with each other, however they still form a coalition to address critical context through the competition and broadcast mechanism.

4 Conclusion

In this paper, we have presented the implementation details of the enhanced design of the DSO-CA. The implementation is a distributed system of parallel

processes that communicates with each other via an MOM. Each parallel process represents the specialised function of the GWT. Competitions are initiated by these processes and they compete for global broadcast via the attention mechanism and the GBM. The winner thereby, will either lead to pipelines being dynamically created, or allow parallel pipelines to influence the processing of others. For validation, the implementation was applied to the traffic control problem and experimental results showed increase in performance gain using methods that are enabled by our implementation of the enhanced DSO-CA.

For future work, we will be looking into the learning aspects of the design and implementation. Currently, pathways between codelets are predetermined but in general the pathways should also be learned. This can be done by leveraging on the competition design aspect. Intuitively, a pathway is formed between two Cognitive Codelets, CC_a and CC_b if CC_a frequently accepts the broadcast sourced from CC_b. There are two criteria for acceptance: firstly, translation of CC_b outputs into CC_a inputs must be coherent; secondly, output from CC_a based on CC_b input should be beneficial to the system as a whole. These criteria require feedback loops between the environment and the system, which will be propagated down to individual Cognitive Codelets, and also between codelets because incoherent inputs should lead to feedback to CC_b so it could make correction. To implement the feedback loops, we may leverage on the representativeness of the integrative memory and competition. Thus, future work is to study how these feedback loops can be designed around the competition mechanism and how representation learning can be implemented within the integrative memory.

References

1. Baars, B.J.: A Cognitive Theory of Consciousness. Cambridge University Press, Cambridge (1993)
2. Goertzel, B.: A formal model of cognitive synergy. In: Everitt, T., Goertzel, B., Potapov, A. (eds.) AGI 2017. LNCS (LNAI), vol. 10414, pp. 13–22. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63703-7_2
3. Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L.: Recent development and applications of sumo-simulation of urban mobility. Int. J. Adv. Syst. Meas. **5**(3&4), 128–138 (2012)
4. Ng, G.W., Tan, Y.S., Teow, L.N., Ng, K.H., Tan, K.H., Chan, R.Z.: A cognitive architecture for knowledge exploitation. In: 3rd Conference on Artificial General Intelligence (AGI-2010). Atlantis Press (2010)
5. Ng, G.W., Tan, Y.S., Xiao, X.H., Chan, R.Z.: DSO cognitive architecture in mobile surveillance. In: 2012 Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF), pp. 111–115. IEEE (2012)
6. Ng, G.W., Xiao, X., Chan, R.Z., Tan, Y.S.: Scene understanding using DSO cognitive architecture. In: 2012 15th International Conference on Information Fusion (FUSION), pp. 2277–2284. IEEE (2012)
7. Ng, K.H., Du, Z., Ng, G.W.: DSO cognitive architecture: unified reasoning with integrative memory using global workspace theory. In: Everitt, T., Goertzel, B., Potapov, A. (eds.) AGI 2017. LNCS (LNAI), vol. 10414, pp. 44–53. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63703-7_5

8. Paraense, A.L.O., Raizer, K., Gudwin, R.R.: A machine consciousness approach to urban traffic control. Biol. Inspired Cogn. Archit. **15**, 61–73 (2016)
9. Rosenbloom, P.S.: Towards uniform implementation of architectural diversity. Artif. Intell. **20**, 197–218 (2009)
10. Rosenbloom, P.S., Demski, A., Ustun, V.: The sigma cognitive architecture and system: towards functionally elegant grand unification. J. Artif. Gen. Intell. **7**(1), 1–103 (2016)