AI-Board - An Artificial Intelligence Problem Resolution and Learning System

Pedro Ferreira

Jan/2015

ABSTRACT

This paper describes the AI-Board System, composed of Khalid, AIDL and Hasina, respectively an AI-Library, an AI Problem description language and a Mind System. This describes the plan of work.

DISCLAIMER

This work is intended for mind and cognitive simulation in animations and games mainly. Some of the real-world applications that go outside the simulated environment don't aim on a real robot factoring.

INTRODUCTION

1. Khalid

Khalid is basically an AI Library that encapsulates all state-of-the-art Artificial Intelligence data structures and algorithms. It has implementation of simple agents, search agents, planning agents, neural network agents, and much more. It's designed to have two implementations: one in Java, and other in C. The C version shall be able to join the kernel space, so that AI-Board can can be encapsulated into a kernel module.

The Java version

The Java version of the library is under development.

The C version

The C version of the library is under development.

2. Hasina

Hasina is a Knowledge Representation system based on ontologies, implemented through a logical system.

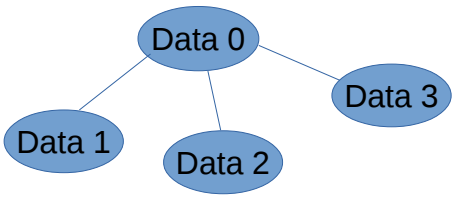
Before starting to display figures about Hasina architecture, let's discuss some elements that belong to it, to understand their real objective and need.

Data Storage Subsystem

This subsystem is responsible to save brain data for use by the reasoning process. It's a three-layer architecture. It has the short-term memory, that stores data that was or is under utilization currently. It also has the medium term memory. Data that is stored in the medium term memory can be pulled from it, and then saved into the short-term memory for utilization by the brain (Remembering process). To finish, there is the long-term memory, that stores data that wasn't being used for a very long time. This data is kind of compressed, and the recovery of this data depends of many factories, such as concentration, relaxation and meditation.

On Hasina, information is atomic with the Node data structure. Each node has two basic components: Data and Connections.

- The data part are designed to store the information that is stored into it.
- The connections part are designed to build connections among different nodes. They can be edges or multiedges, representing all the connections.



Besides the node structure, there are the places where the information is built on.

STM	MTM	LTM

We basically have:

•**Short-term memory (STM)** – In this area of the memory, it's stored the information over with agent is working and reasoning on.

•**Medium-term memory (MTM)** – This area is a few less accessible than Short-term memory. Agent doesn't reason over this data.

•**Long-Term memory (LTM) –** This area is a few less accessible than Medium-term memory. Agent doesn't reason over this data. This data is compressed and classified. Any attempt to access this memory through usual ways will take some time, and will require agent concentration.

(1) Remembering Subsystem

Remembering Subsystem is responsible for bringing data up from medium-term memory and long-term memory into the short-term memory, so that this data can be used by the agent.

Remembering is generally a consequence of re-heard of some subject. When some sensor brings a data to the agent, it will search through its boundary to retrieve all related information originated from the data's relationship.

Sometimes, data is remembered spontaneously, without any reasonable reason for the knowledge appear. It's a kind of entropy. The module responsible for that is called Reflex Subsystem.

The reflex subsystem remember information a kind of randomly. The data that will be remembered depends on the location where the data is, together with its age / utilisation.

(2) Reasoning Subsystem

The reasoning Subsystem is responsible for gathering new information based on the existing information (in the short-term memory). This information is generated from the information that is in the short term memory. The short memory is mainly used for computation, while the other memories are for organization and storage.

3. AIDL

AIDL is Artificial Intelligence Description Language, and it's a language that describe programs, problems and data of Artificial Intelligence. It covers distinct main areas of AI [1]:

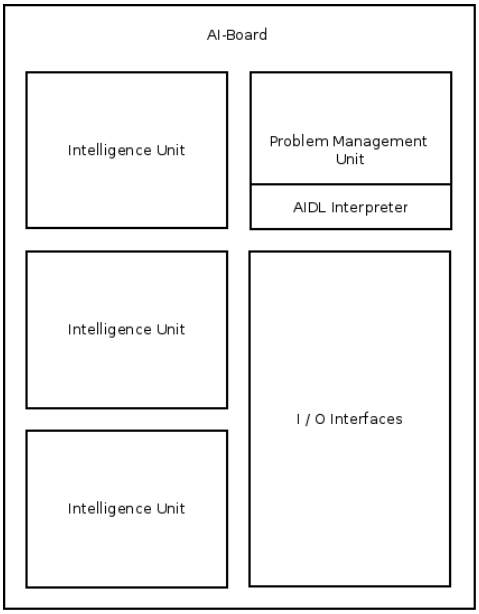
- **Problem Solving** – In which see how an agent can find a sequence of actions that achieves its goals when no single action will do.
- **Knowledge and Reasoning** – In which we design agents that can form representations of a complex world, and use these new representations to deduce what to do.
- **Uncertain Knowledge and Reasoning** – In which we see how an agent can tame uncertainty with degrees of belief.
- **Learning** – In which we describe agents that can improve their behavior through diligent study of their own experiences.
- **Communicating, Perceiving and Acting** – In which we see how to make use of the copious knowledge that is expressed in natural language.

4. AI-Board

AI-Board is a complex software system that is based mainly on the three technologies above to be able to understand and solve Artificial Intelligence problems.

It has basically four main submodules: Intelligence Unities – Problem Solving units based on the Khalid library; Problem Management Unit – A central unit that distribute problems among the different Intelligence Unities and is also capable to join solutions for complex problems (divide and conquer); AIDL Interpreter – An interpreter for the AIDL Language; I/O Interfaces – A set of interfaces that play the roles of sensors and actors in the system.

It has the following conceptual architecture:



Below, the reader can better understand the meaning and function of each box:

- **AI-Board** – This box represents the agent itself, containing all of its components.
- **Intelligence Unit** – This box represents a problem solving unit, that can solve unit problems or can simulate a live agent.
- **Problem Management Unit** – This box represents the problem distribution manager that will allocate problems to each problem.
- **AIDL Interpreter –** This box represents an interpreter for the Artificial Intelligence Description Language. It receives data from the I/O interfaces, and send to the Problem Management Unit Core to rightly distribute the problems or questions to the Intelligence Units
- I**/O Interfaces** – This box represents a set of I/O interfaces that establish communication for the Problem Management Unit (more specifically the AIDL Interpreter) and the outside world..

Regarding communications between the modules, we have:

- **Intelligence Unit – Problem Management Unit –** As we already know, Problem Management Unit is responsible for managing the problems and their distribution and orchestration among all Intelligence Unities. Basically, Problem Management Unit sends problems to the intelligence unities to solve. If the problem is static, then there is only one piece of data back – the problem solution; if the problem is dynamic, then there is opened a data stream between these two components so that data can be sent while the problem environment is running.
- **Problem Management Unit – I/O Interfaces –** I/O Interfaces send and receive data to and from the Problem Management Unit. It always depend on the implementation of the interface. For example, an “eye” always receives data from the environment, but it’s not always that AI-Board cares about this data, once it can be busy processing the problem solution, and ignoring all it sees, because it’s focused on something. Depending on the eye implementation, some part of the data can be cached or simply ignored. All that the AI-Board standard defines is an interface for the I/O interfaces, and the kind of data that it can exchange. Whenever an I/O interface wants to send data to the Problem Management Unit, it normally does raise a signal to it, so that it can open a data stream, or reject the request, because it’s busy.

5. Applications

AI-Board, with it’s kernel-based design, can have a lot of practical applications, both in toy world and in real world problems

Toy World Problems

- **Bast-Universe –** This project is a kind of an aquarium, that comprises an simulated environment physical environment, and a set or robots intended to interact together, as a community. The purpose of this project is to study how the robots can learn alone and in group (socially). This is actually being developed by our group.
- **Vacuum-World Problem** – This problem consists of a MxN grid, with one or more cleaner agents that clean the possible dirty in each cell of the grid. Each cell can be dirty

or clean. There are some variations of the problem, including one that the dirty can be back after being cleaned, also, the agent can clean with an additive, that keeps the cell safe from being re-dirty after T time unities.

Real World Problems

- **Social Network Helper** – Today, the social networks play a great role in the social interaction of people. Sometimes, busy people don't have time to maintain their social networks updated, acting and reacting to the action of others. The idea here is map the behavior of the user, and use this knowledge to automate the posting and privacy protection in the social network profile
- **Intelligent Environments** – Imagine an intelligent house, that automate a lot of tasks for a family. AI-Board could be the kernel of that, in the sense of information, and also in the sense of intelligent environments, for example, detecting that the children are watching TV, and turning on the parental settings on the TV, or detecting that the man woke up, and turn on the TV on news, and prepare the toast. There are a lot of variations of this scenarion, for example, an store with electronic showcase, and it displays what the customer has more probability to buy, based on its buying history in the store, and also in his main characteristics, as age, gender, etc.

References

[1] Russel, Stuart and Norving, Peter – Artificial Intelligence: A Modern Approach, Third Edition, 2010

[2] Ernst, G.W. and Newell, A. (1969). – GPS: a case study in generality and problem solving. Academic Press. (revised version of Ernst's 1966 dissertation, Carnegie Institute of Technology.)

[3] Enderton, Herbert B. (2001) – A Mathematical Introduction to Logic (2nd edition)

[4] Enderton, Herbert B. (1977) – Elements of Set Theory

[5] van Dalen, Dirk (2008) – Logic and Structure (4th edition)

[6] Kernigham, Brian W. and Ritchie, Dennis M. – The C Programming Language, 2nd Edition by (Apr 1, 1988)

[7] Wang, Y. 2006. A Layered Reference Model of the Brain (LRMB), IEEE Trans. on Systems, Man, and Cybernetics (PartC), Vol. 36, No. 2

[8] Bogner, M., Ramamurthy, U., Franklin, S. 2015. “Consciousness” and Conceptual Learning in a Socially Situated agent, Institute of Intelligent Systems, The University of Memphis

[9] Davis, R., Shrobe, H, Szolovits, P. 1993. What is a Knowledge Representation? AI Magazine, 14(1): 17-33, 1993

[10] Barsalou, L. W. 1999. Perceptual symbol systems, Behavioral and Brain Sciences (1999) 22:577-660

[11] Barsalou, L W. 2013. Mirroring as Pattern Completion Inferences withing Situated Conceptualizations. Cortex 49 (2013) 2951-2953

[12] Pezzulo, G., Barsalou, L. W., Cangelosi, A., Fischer, M. H., McRae, K, Spivey, M. J. 2012. Computational Grounded Cognition: a new alliance between grounded cognition and computation modelling, DOI> 10.3389/fpsyg.2012.00612

[13] Powers, T. W., Abbot, B., Carey, T. A., Goldstein, D. M., Mansell, W., Marken, Marken, R. S., Nevin, B., Robertson, R., Taylor, M. 2011. http://www.pctweb.org

[14] Wang, Y. 2010. On Formal and Cognitive Semantics for Semantic Computing, International Journal of Semantic Computing, Vol. 4, No. 2 (2010) 1-34, DOI 10.1142/S1793351X10000833