Biologically Inspired Cognitive Architectures (2015) 12, 9–19

Available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/bica

RESEARCH ARTICLE

A comparison of simple agents implemented in simulated neurons

Christian Huyck *, Carl Evans, Ian Mitchell

Middlesex University, Dept. of Computer Science, London, UK

Received 5 February 2015; received in revised form 23 March 2015; accepted 23 March 2015

KEYWORDS
Cell Assembly;
PyNN;
Embodied agent;
Neural simulation

Abstract
Neuromorphic embodied Cell Assembly agents that learn are one application being developed for the Human Brain Proj (HBP). The HBP is building tools, available for all researchers, for building brain simulations. Existing simulated neural Cell Assembly agents are being translated to the platforms provided by the HBP; these agents run on neuromorphic chips in addition to von Neumann based computers. Whilst translation of the agents to the software technology demanded by the HBP platforms is relatively straightforward, porting to the neuromorphic chips is a non-trivial software engineering task. Versions of the simple agent, CABot1, have been developed in fatiguing leaky integrate and fire neurons, Izhikevich neurons and leaky integrate and fire neurons. These have been developed to run in Java, PyNN, NEST and Neuromorphic hardware. All variants are roughly equivalent. The agents view a picture, implement simple commands, and respond to a context sensitive directive involving the content of the picture. By running variants of these agents on different platforms, and with the different simulated neural models, implicit assumptions in these models can be revealed. Once these Cell Assembly agents have been translated and embodied in a virtual environment, they will be extended to learn more effectively. The use of neural hardware supports the real time simulation of many more neurons, providing a platform for exploration of more complex simulated neural systems.

1. Introduction

The Cell Assembly (CA), initially proposed in 1949 (Hebb, 1949; Huyck & Passmore, 2013), is a key component in brain function linking neural behaviour to psychological behaviour. A CA is a set of neurons that are able to support firing

* Corresponding author.
E-mail addresses: c.huyck@mdx.ac.uk (C. Huyck), c.evans@mdx.ac.uk (C. Evans), i.mitchell@mdx.ac.uk (I. Mitchell).

http://dx.doi.org/10.1016/j.bica.2015.03.001
2212-683X/

amongst themselves; when activated, the firing CA becomes a short-term memory that persists while many of the neurons continue to fire at a rate much higher than the background base neural firing rate. Most typical concepts humans have, for instance *dog*, *mother*, or *foot*, have a CA that represents them.

CAs are central to a series of embodied agents, the Cell Assembly robots (CABots) (Huyck et al., 2011), implemented entirely in simulated neurons. CABots are interactive agents operating within a virtual environment that explore, plan and learn from sensory information. The agent can be directed by a user with natural language commands.

Three Java versions of the CABot agent were developed. The first version of the agent, CABot1, included a component for natural language processing (NLP) and could process user input for goal setting via a planning component. It also had a vision system which could detect lines within its environment. The second version, CABot2, made some advances on the earlier version. In particular, it included a verb learning component and had improved vision which could detect edges rather than simply lines. The third version of the agent, CABot3 (see Fig. 1), is composed of four major sub-systems, and a simple control system to manage some inter-system behaviour. The major systems, each consisting of hundreds of CAs, are:

- Natural Language Processing (NLP): process user input for goal setting;
- Visual: identification and location of objects;
- Plan: goal-oriented, communication and action; and
- Spatial Cognitive Map: long-term mapping of the environment.

When commanded, the CABot3 agent explores the environment, building a spatial cognitive map from information gathered via its visual field. The planning mechanism is a type of Maes network (Maes, 1989). A Maes net is a spreading activation system with goals, modules, facts, and actions. For example, facts support goals and modules that are related to them.

The virtual environment has four rooms and each room has a different object (either a pyramid or stalactite) within it. Testing the system relies on the agent using the cognitive map to find an object.

The agent is managed by sub-networks composed of simulated neurons, a technical possibility thought too complex in 1988 (Smolensky, 1988). There are some compromises made with regard to biological plausibility, but the agent is composed of, and controlled entirely by, simulated neurons.

Recently, work has begun translating the CABots from their Java based neural simulation and porting them, via PyNN (Davison et al., 2008), to neuromorphic chips (Indiveri et al., 2011). This work has commenced with CABot1, but ultimately the intention is to complete this re-engineering process for all three versions. This is one small component of the Human Brain Project (HBP) (HBP, 2015). Porting the existing CABots to the chips is a non-trivial software engineering task. Once done, the agent will be extended to learn visual objects, learn plans, and will be tested on new cognitive modelling tasks. The HBP is in its early stages, but will include a series of platforms that researchers, including those not affiliated with the project, can use for large scale neural simulations. CABots are one type of simulation.

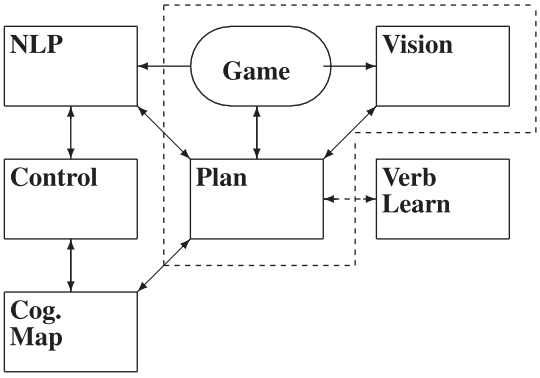

Fig. 1 Gross Topology of CABot3. Boxes represent subsystems of subnets. The oval represents the environment. The dashed box represents subsystems for CABot1; note that the environment is static for the simulations described in this paper. The vision system is introduced in Section 4.2, and the planning system is most fully described in Section 4.3.

This paper presents several initial prototypes based on several different types of neurons, on different simulators, on standard hardware and on the SpiNNaker neuromorphic chip. The paper presents a qualitative comparison of these systems. There is a system with the authors' bespoke Fatiguing Leaky Integrate and Fire neural model implemented in Java, and systems using the HBPs typical PyNN middleware. PyNN simulations specify the topology and then run that topology in a standard simulator. There is a PyNN system using Izhikevich (Izhikevich, 2004) neurons in the NEST simulator; a system using Integrate and Fire (Brette & Gerstner, 2005) neurons on NEST; a system using Integrate and Fire neurons on the SpiNNaker emulator; and a system using Integrate and Fire on SpiNNaker chips. These systems have roughly the same functionality, however there are some differences that make software engineering in some systems simpler, at least for this project. This provides evidence for a rough equivalence of these point neural models.

All of these prototype agents have both vision and planning. While vision uses almost half of the neurons in CABot3, for example, planning is much smaller; none the less, this is a solid test of the basic agent technology. Whilst the focus for this paper is the translation and porting of the CABot1 agent prototype, the first major version of the FLIF Java agent (Huyck, 2008), the longer term aims of the Neurmorphic Embodied Agents that Learn (NEAL) project include an effort to reproduce CABot3 (Huyck et al., 2011) using PyNN. This project aims to execute the code on neuromorphic chips. This will then be extended to perform more tasks in richer environments aiming to provide further understanding of general neural cognitive