Types of Computational Self-awareness And How We Might Implement Them

[Extended Abstract] *

Peter R. Lewis
Aston Lab for Intelligent Collectives Engineering (ALICE)
Aston University, Birmingham, United Kingdom
p.lewis@aston.ac.uk

1. INTRODUCTION

Computing systems increasingly comprise large numbers of heterogeneous sub-systems, each with their own local perspective and goals, connected in dynamic networks, and interacting with each other and humans in ways which are difficult to predict. Nevertheless, users engaging with different parts of the system still expect high performance, reliability, security and other qualities, provided in a way that is robust or adaptive in the presence of unforeseen changes (including to users, the network, physical environment or the system itself). Examples of systems which are facing this challenge are wide-ranging and include robot swarms, personal devices, web services and sensor networks. In all these cases, advanced levels of autonomous behaviour can enable the system to adapt itself at run time, by learning behaviours in real time, appropriate to changing conditions.

To meet this challenge, one approach is to engineer systems to possess greater awareness, of the world around them, of themselves, and of the interactions between the two. In the context of ongoing adaptation to changing environments, this requires them to i) monitor aspects of themself and their environment, ii) learn online, to continue to maintain models and build awareness while in operation, and iii) to reason and adapt, based on this awareness, on an ongoing basis.

2. COMPUTATIONAL SELF-AWARENESS

Self-awareness has long been argued as a potential beneficial property of complex computing systems (e.g., [3, 1]). While there have been several efforts in computing relating to self-awareness, historically, an understanding of what self-aware computing might mean, and how it might be achieved in a principled way, has been lacking. Reviews [5, 9] have clustered contributions either by community or thematically, and have identified that the terms involved have often lacked definition. This has led to different implied meanings in these different communities, and has not aided the development of a fundamental understanding. In response, we have developed a generic conceptual framework that provides a psychologically-grounded generic way to describe and design the self-awareness capabilities of computing systems.

In exploring self-awareness properties in computing, it is tempting to engage in philosophical discussions about the extent to which a machine might truly be (or be able to be) self-aware. From an engineering point of view, however, this does little to progress us toward the technology vision outlined above. In our recent work [4, 7], we have instead proposed that human self-awareness can serve as a source of inspiration for a new notion of *computational self-awareness* that captures a "stack" of self-awareness-related levels. These levels, inspired directly by theories in human self-awareness [8], refer to fundamental aspects of a system's potential self-awareness, such as its awareness of time, social interactions, causality, goals, etc. One such set of levels is presented in Table 1. A self-aware system might implement some or all of these levels, and this becomes an explicit design decision, based on the requirements and context of the application. Multiple algorithms for capturing self-knowledge are then used by the system to realise the required levels. In all but the simplest cases, these algorithms perform online learning [7], providing the ability to learn self-knowledge on the fly from streaming data. Such algorithms adapt the concept being learnt while learning it, and as it changes in the world, as sensed by the system.

We call this approach *computational self-awareness*, first to highlight that there may be differences from human self-awareness, and second since self-awareness arises from computational processes. In our work, we have begun to explore what some of these computational processes could be. We then combine self-awareness with *self-expression*, a further set of run-time processes that generate actions, possibly adaptations or reconfigurations of the system or the environment, based on the system's learnt self-knowledge.

Further, we have developed a general approach to the design of self-aware computing systems, that includes a reference architecture (figure 1) capturing self-awareness concerns. We derived a series of derived architectural patterns [4] from the reference architecture, which were then refined through the experience of applying them to a range of application domains [7]. By following an architecture selection method, the patterns can be used to determine whether, how, and to what extent to build self-awareness capabilities into a system, given a set of goals and considered scenarios.

CODES/ISSS '16, October 01-07 2016, Pittsburgh, PA, USA

ISBN 978-1-4503-4483-8/16/10...$15.00
DOI: http://dx.doi.org/10.1145/2968456.2973273

Table 1: Neisser's levels of self-awareness, and their corresponding computational levels, based on [4].

Neisser's level	Computational level	Description
Ecological self	Stimulus awareness	Experience of self in relation to the environment.
Interpersonal self	Interaction awareness	Awareness includes conceptualisation of interactions between oneself and other individuals / the world. Awareness of causal relationships emerges.
Extended self	Time awareness	Awareness of experiences / concepts over time: memories & anticipations.
Private self	Goal awareness	Awareness that experiences, thoughts, preferences, etc. relate specifically to individuals. Awareness of the effect of phenomena on different individuals, and how they relate to these thoughts and preferences, etc.
Conceptual self	Meta-self-awareness	Awareness of self-awareness at this or other levels. Ability to reason about perceptions, knowledge, and self-awareness processes. Can be iterative.

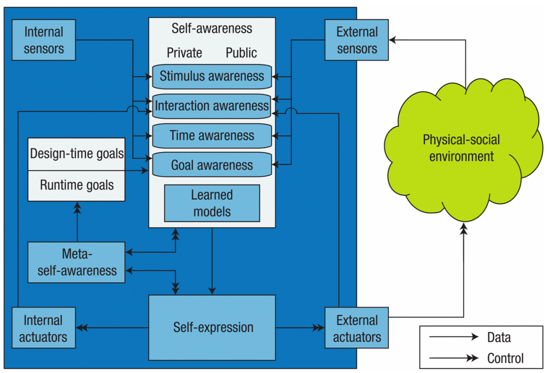

Figure 1: Lewis et al's reference architecture for self-aware computing systems. Source: [4].

3. EXAMPLE APPLICATION: VISUAL SENSOR NETWORKS

We have explored explicitly designing in the computational self-awareness properties of a system, in a number of applications [7]. One such application is visual sensor networks [2, 6]. Here, a network of smart cameras is given the goal of tracking objects that move between their fields of view (FOV). The system coordinates tracking in a decentralised way: each camera makes decisions at run time concerning which other cameras to communicate with, and a market mechanism [2] is used to exchange responsibilities. Self-awareness is realised by two online learning processes sitting above this. First, a pheromone-inspired algorithm [2] monitors trading behaviour in the market, building a graph-based model of beneficial interactions, which adapts to changes over time (i.e., forgetting previously good interactions which are no longer beneficial). Second, a reinforcement learning algorithm [6] learns what communication policy to use, to act on the pheromone information.

The outcome is an efficient and adaptive balance of the trade off between the competing objectives of maximising network-wide tracking performance and minimising communication overhead. Unlike previous approaches to this handover problem, the cameras did not require any *a priori* knowledge of their environment or the camera neighbourhood structure; it is learnt online by the cameras themselves, in a way that is adaptive to changes over time. Figure 2 illustrates the pheromone learning process.

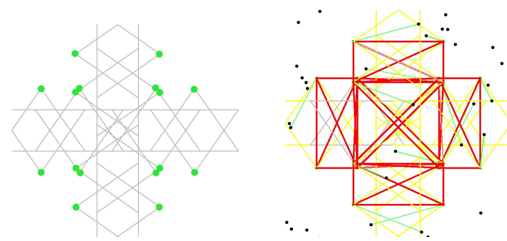

Figure 2: Visual sensor network scenarios. Left: Initially no information is present. Right: The learnt graph after some time (red lines). Cameras (circles), FOVs (triangles) and objects (dots) are shown.

4. CONCLUSIONS

A self-aware system has the ability to learn and maintain knowledge of itself, its experiences, and interactions with its environment, on an ongoing basis. This permits reasoning and decision making to support effective, autonomous adaptive behaviour, in an uncertain and changing environment. This extended abstract describes recent work to develop the concept of *computational self-awareness*, a notion of machine self-awareness inspired by theories of human self-awareness, and that arises through computational processes. Recent work to develop conceptual reference architecture is highlighted, and a key role for online learning is identified.

5. REFERENCES

[1] E. Amir, M. L. Anderson, and V. K. Chaudhri. Report on DARPA workshop on self-aware computer systems. Tech. Rep. UIUCDCS-R-2007-2810, UIUC CS, 2007.

[2] L. Esterle, P. R. Lewis, X. Yao, and B. Rinner. Socio-economic vision graph generation and handover in distributed smart camera networks. *ACM Trans. on Sensor Networks*, 10(2):20:1–20:24, 2014.

[3] P. Horn. Autonomic computing: IBM's perspective on the state of information technology. IBM Corp., 2001.

[4] P. R. Lewis, A. Chandra, F. Faniyi, K. Glette, T. Chen, R. Bahsoon, J. Torresen, and X. Yao. Architectural aspects of self-aware and self-expressive computing systems: From psychology to engineering. *Computer*, 48(8):62–70, 2015.

[5] P. R. Lewis, A. Chandra, S. Parsons, E. Robinson, K. Glette, R. Bahsoon, J. Torresen, and X. Yao. A Survey of Self-Awareness and Its Application in Computing Systems. In *Proc. Int. Conf. on Self-Adaptive and Self-Organizing Systems Workshops (SASOW)*, pages 102–107. IEEE, 2011.

[6] P. R. Lewis, L. Esterle, A. Chandra, B. Rinner, J. Torresen, and X. Yao. Static, Dynamic, and Adaptive Heterogeneity in Distributed Smart Camera Networks. *ACM Trans. on Autonomous and Adaptive Systems*, 10(2):8:1–8:30, 2015.

[7] P. R. Lewis, M. Platzner, B. Rinner, J. Torresen, and X. Yao, editors. *Self-aware Computing Systems: An Engineering Approach.* Springer, 2016.

[8] U. Neisser. The Roots of Self-Knowledge: Perceiving Self, It, and Thou. *Annals of the NY AoS.*, 818:19–33, 1997.

[9] J. Schaumeier, J. Jeremy Pitt, and G. Cabri. A tripartite analytic framework for characterising awareness and self-awareness in autonomic systems research. In *Proc. Int. Conf. on Self-Adaptive and Self-Organizing Systems Workshops (SASOW)*, pages 157–162. IEEE, 2012.