Natural Computing Series

Peter R. Lewis
Marco Platzner
Bernhard Rinner
Jim Tørresen
Xin Yao *Editors*

Self-aware Computing Systems

An Engineering Approach

Springer

Copyrighted material

Editors
Peter R. Lewis
School of Engineering & Applied Science
Aston University
Birmingham, United Kingdom

Marco Platzner
Department of Computer Science
Paderborn University
Paderborn, North Rhine-Westphalia
Germany

Bernhard Rinner
Institute of Networked and Embedded Systems
Alpen-Adria-Universität Klagenfurt
Klagenfurt am Wörthersee
Austria

Jim Tørresen
Department of Informatics
University of Oslo
Oslo, Norway

Xin Yao
School of Computer Science
University of Birmingham
Birmingham, United Kingdom

ISSN 1619-7127
Natural Computing Series
ISBN 978-3-319-39674-3 ISBN 978-3-319-39675-0 (eBook)
DOI 10.1007/978-3-319-39675-0

Library of Congress Control Number: 2016942574

© Springer International Publishing Switzerland 2016
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Copyrighted material

Contents

Copyrighted material

Copyrighted material

Copyrighted material

Copyrighted material

Copyrighted material

Copyrighted material

List of Contributors

Andreas Agne
Paderborn University, Germany. e-mail: agne@upb.de

Rami Bahsoon
University of Birmingham, UK. e-mail: r.bahsoon@cs.bham.ac.uk

Tobias Becker
Imperial College London, UK. e-mail: tobias.becker@imperial.ac.uk

Arjun Chandra
Studix, Norway. e-mail: arjun@studix.com

Renzhi Chen
University of Birmingham, UK. e-mail: rxc332@cs.bham.ac.uk

Tao Chen
University of Birmingham, UK. e-mail: t.chen@cs.bham.ac.uk

Stewart Denholm
Imperial College London, UK. e-mail: stewart.denholm10@imperial.ac.uk

Bernhard Dieber
Alpen-Adria-Universität Klagenfurt, Austria. e-mail: bernhard.dieber@counity.at

Lukas Esterle
Alpen-Adria-Universität Klagenfurt, Austria. e-mail: lukas.esterle@aau.at

Gustavo Fernández Domínguez
Austrian Institute of Technology, Austria. e-mail: gustavo.fernandez@ait.ac.at

Funmilade Faniyi
University of Birmingham, UK. e-mail: f.faniyi@gmail.com

Andreea-Ingrid Funie
Imperial College London, UK. e-mail: andreea.funie09@imperial.ac.uk

Copyrighted material

Kyrre Glette
University of Oslo, Norway. e-mail: kyrrehg@ifi.uio.no

Ce Guo
Imperial College London, UK. e-mail: ce.guo10@imperial.ac.uk

Markus Happe
ETH Zurich, Switzerland. e-mail: markus.happe@alumni.ethz.ch

Maciej Kurek
Imperial College London, UK. e-mail: mk306@imperial.ac.uk

Peter R. Lewis
Aston University, UK. e-mail: p.lewis@aston.ac.uk

Achim Loesch
Paderborn University, Germany, e-mail: achim.loesch@upb.de

Wayne Luk
Imperial College London, UK. e-mail: w.luk@imperial.ac.uk

Leandro L. Minku
University of Leicester, UK. e-mail: leandro.minku@leicester.ac.uk

Georg Nebehay
Austrian Institute of Technology, Austria. e-mail: gnebehay@gmail.com

Xinyu Niu
Imperial College London, UK. e-mail: niu.xinyu10@imperial.ac.uk

Kristian Nymoen
University of Oslo, Norway. e-mail: kristian.nymoen@imv.uio.no

Roman Pflugfelder
Austrian Institute of Technology, Austria. e-mail: roman.pflugfelder@ait.ac.at

Marco Platzner
Paderborn University, Germany. e-mail: platzner@upb.de

Christian Plessl
Paderborn University, Germany. e-mail: christian.plessl@uni-paderborn.de

Bernhard Rinner
Alpen-Adria-Universität Klagenfurt, Austria. e-mail: bernhard.rinner@aau.at

Mark Salmon
University of Cambridge, UK. e-mail: mhs39@cam.ac.uk

Jennifer Simonjan
Alpen-Adria-Universität Klagenfurt, Austria. e-mail: jennifer.simonjan@aau.at

Stephan C. Stilkerich
Airbus Group Innovation, Germany. e-mail: stephan.stilkerich@airbus.com

Copyrighted material

Tim Todman
Imperial College London, UK. e-mail: timothy.todman@imperial.ac.uk

Jim Tørresen
University of Oslo, Norway. e-mail: jimtoer@ifi.uio.no

Ariane Trammell-Keller
ETH Zurich, Switzerland. e-mail: ariane.trammell@alumni.ethz.ch

Shuo Wang
University of Birmingham, UK. e-mail: s.wang@cs.bham.ac.uk

Xin Yao
University of Birmingham, UK. e-mail: x.yao@cs.bham.ac.uk

Copyrighted material

Copyrighted material

Acronyms

ACO	Ant Colony Optimisation
AES	Advanced Encryption Standard
ALA	Ant Learning Algorithm
API	Application Programming Interface
BSD	Berkeley Software Distribution
CDC	Concept Drift Committee
CDT	Correct Detected Track
CMT	Consensus-Based Matching and Tracking
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
CV	Computer Vision
DDD	Diversity for Dealing with Drifts
DDM	Drift Detection Method
DPS	Dynamic Protocol Stack
DWM	Dynamic Weight Majority
EA	Evolutionary Algorithm
EDDM	Early Drift Detection Method
EGO	Efficient Global Optimisation
FAT	False Alarm Track
FB	Functional Block
FF	Flip-Flop
FPGA	Field-Programmable Gate Array
FPS	Frames per Second
FMC	FPGA Mezzanine Card
FOV	Field of View
FSR	Force-Sensitive Resistor
GA	Genetic Algorithm
GPSP	General Purpose Sensor Platform
GP	Gaussian Process
GPU	Graphics Processing Unit
H2S	Hardware-to-Software

Copyrighted material

HLS	High Level Synthesis
HMM	Hidden Markov Models
HPC	High Performance Computing
ICAP	Internal Configuration Access Port
IDP	Information Dispatch Point
ILP	Integer Linear Programming
IP	Internet Protocol
IPC	Inter-process Communication
LUT	Look-up Table
MAC	Media Access Protocol
MLO	Machine Learning Optimiser
MOEA/D	Multi-objective Evolutionary Algorithm Based on Decomposition
MOP	Multi-objective Optimisation Problem
MPI	Message Passing Interface
MTBF	Mean Time Between Failures
NoC	Network-on-Chip
OSC	Open Sound Control
OT	Object Tracking
PE	Processing Element
RAP	Redundancy Allocation Problem
RAM	Random Access Memory
RTM	Reverse Time Migration
S2H	Software-to-Hardware
SA	Self-aware
SACS	Self-aware Computing Systems
SDRAM	Synchronous Dynamic Random Access Memory
SE	Self-expression
SIMD	Single Instruction, Multiple Data
SMT	Satisfiability Modulo Theories
SVM	Support Vector Machine
SoC	System-on-Chip
SOP	Single-Objective Optimisation Problem
SSE	Streaming SIMD Extensions
STEPD	Statistical Test of Equal Proportions
TCP	Transmission Control Protocol
TDF	Track Detection Failure
TPOT-RL	Team-Partitioned Opaque-Transition Reinforcement Learning
Todi	Two Online Classifiers for Learning and Detecting Concept Drift
TLD	Tracking-Learning-Detection
UDP	User Datagram Protocol
VHDL	Very High Speed Integrated Circuit Hardware Description Language

Copyrighted material

Glossary

This glossary lists important terms used in this book, in particular in Part I "Concepts and Fundamentals", with accompanying descriptions or definitions. The glossary is organised into four sections: concepts of self-awareness and self-expression, engineering self-aware systems, related approaches, and general terms. The terms in each of the sections are listed alphabetically.

Concepts of Self-awareness and Self-expression

self-awareness Self-awareness is a broad concept which describes the property of a system (typically a human) which has knowledge of "itself", based on its own senses (perceptual) and internal models (conceptual). This knowledge may take different forms (cf. levels of self-awareness), and be based on perceptions of both internal and external phenomena (cf. public vs. private self-awareness). It can be a property of single systems (e.g., agents) and collective systems.

collective self-awareness Collective self-awareness refers to the self-awareness property of a collective system, i.e., as opposed to a single agent. Levels of, and public/private self-awareness apply also at this abstraction. This means that a self-aware system is not required to have a central "knowledge" component (though it may have, if desired).

computational self-awareness Computational self-awareness is a notion we have developed to refer to a computational interpretation of self-awareness. Since much of the literature on self-awareness does not readily make sense to engineers or applies directly to technical systems, aspects of computational self-awareness are designed to describe self-awareness properties of computational systems, inspired by self-awareness in humans.

emergent self-awareness This is a special case of collective self-awareness, when the collective self-awareness properties are present, but it is not obvious how this comes about by simply examining the behaviour of individual nodes within a collective.

Copyrighted material

methodology for engineering self-aware systems We developed a methodology for engineering self-aware systems, based on the reference architecture and the derived architectural patterns.

primitive A primitive is a particular block in the reference architecture, representing, for example, a level of self-awareness, self-expression and a sensor. They are instantiated for particular applications.

reference architecture We developed a reference architecture which captures the core aspects of computational self-awareness. The aim is to provide a common, principled basis on which researchers and practitioners can structure their work. We have argued that the psychological foundations, while not strictly necessary, can provide a means of channelling a wide range of ideas, which would perhaps otherwise not have occurred to engineers, acting to inspire the design of future computing systems. The architecture can also be used as a template for identifying common ways of implementing self-awareness capabilities. Different implementations of the same capability can thereby be compared and evaluated. Further, we have derived a set of architectural patterns from the reference architecture.

(self-aware) node We use the term self-aware node to refer to various types of system that are self-aware, e.g., an agent, a robot and a camera. Agent is an alternative term, but node can be used when not wanting to be specific about a particular system being an agent. We also claim that self-aware collectives (see next entry) can be viewed as self-aware nodes, at a higher level of abstraction. A node may or may not correspond to a physical system—this is not a requirement, but it may often make sense to make it correspond.

tactic/algorithm/technique A tactic is a particular instantiation of a primitive in the reference architecture, typically referred to as a particular algorithm, technique, etc. These are application specific. Multiple tactics may be suitable for a particular primitive, and some tactics may implement multiple primitives simultaneously.

Related Approaches

autonomic (computing) Autonomic computing is a vision originally pioneered by IBM, of engineered systems which manage themselves. This self-management is stated to include: self-configuration, self-optimisation, self-healing and self-protection. The aim is to reduce the need for human involvement in the management of complex computing systems. Some autonomic computing literature mentions the need for self-awareness as a characteristic to support self-management, though the literature on autonomic computing does not significantly expand on this. (Not to be confused with autonomous.)

autonomous (system) Autonomy is a broad notion with much disagreement surrounding it. However, in general, an autonomous system is one which acts without any external direction. Examples include robots, vehicles and software agents. In many cases, this ability to make decisions is based on a method of decision making pre-programmed into the system, in other cases it is learnt online at run time. The

Copyrighted material

types of systems we are concerned with in this book are ones which would typically be considered to be autonomous to a greater or lesser extent. (Not to be confused with autonomic.)

metacognition/metareasoning Metareasoning is reasoning about reasoning, and has been the topic of a significant amount of research primarily in the US, where it has been primarily led by DARPA. Metareasoning relies on meta-self-awareness, and again the metareasoning community has discussed self-awareness as being important, but not expanded on the notion significantly.

organic computing This is a vision from a long-running (primarily) German research project to create "life-like" engineered systems, in which self-organising emergent behaviour is controlled (by an observer/controller component), to ensure desirability in the self-organisation. The Organic Computing literature also mentioned self-awareness as beneficial, but again does not expand on this significantly.

General Terms

adaptability In high level terms, this is similar to adaptivity, but describes a system's potential for adaptation, rather than actual realised adaptivity.

adaptivity In high level terms, this concerns the amount to which a system adapts, e.g., in the presence of a changing environment, or as a result of its learning.

collective We use the term collective to refer to various types of distributed systems, typically without central control. Examples include swarms, systems-of-systems, populations, multi-agent systems, interwoven systems, etc. The term can be used when there is a need to talk generally of these types of systems, without restricting the discussion to a specific one.

learnt model A learnt model is a model which has been induced through a process of (typically online) learning, based on data from sensors and other existing models. Learnt models hold the conceptual knowledge a self-aware system has concerning itself, its interactions, history, expectations, goals, etc.

model We use the term model in a very general way, to refer to a conceptual representation of some knowledge, typically obtained through sensors. A model could simply be a direct representation of some data, or could be abstractions of that data, or further data synthesised from sensory input.

online learning Online learning is the process of learning a model from data on an ongoing basis. Typically, not all data is available in advance (e.g., it arrives in a streaming fashion from sensors), and the concept being learnt may change over time (i.e., concept drift). In online learning, models are often used (e.g., through self-expression in this case) before learning "completes", if indeed it ever does. Hence most online learning algorithms also need to be anytime algorithms, implying that models are used and improved continuously as time goes by.

self-adaptive system A system which adapts (typically its behaviour) in response to external or internal changes, but without external control. We have argued that

Copyrighted material

self-awareness is an enabling property for effective self-adaptation. When self-adaptation behaviour is based on self-awareness, it is a form of self-expression.

self-organising system A system which changes its organisation (e.g., its structure, architecture, topology), without external control.

Copyrighted material

Chapter 1
Self-aware Computing: Introduction and Motivation

Peter R. Lewis, Marco Platzner, Bernhard Rinner, Jim Tørresen, and Xin Yao

1.1 Self-aware Computing: A New Paradigm

Designing and operating computing and communication systems are becoming increasingly challenging tasks, due to a multitude of reasons. First, compute nodes are evolving towards parallel and heterogeneous architectures to realise performance gains while minimising their power consumption. Progress in micro(nano)-electronics allows us to integrate more and more functionality on a single compute node, but at the same time requires us to deal with increasing numbers of faulty and unreliable components. Second, distributed systems are growing in the numbers and heterogeneity of nodes and must be able to cope with an increasing level of dynamics. The network topology and the collective resources of a distributed system can vary strongly during runtime since nodes may leave and enter the network dynamically. The position, functionality and available resources of each node may also change dynamically. Third, future challenging application domains have quite divergent requirements with respect to functionality and flexibility, performance, resource usage and costs, reliability and safety, and security. Fuelled by technological progress, applications with exciting levels of user interaction will be possible, and these dynamic socio-technical systems bring with them numerous additional runtime trade-offs to consider. Fourth, the size and complexity of decentralised com-

Peter R. Lewis
Aston University, UK, e-mail: p.lewis@aston.ac.uk

Marco Platzner
Paderborn University, Germany, e-mail: platzner@upb.de

Bernhard Rinner
Alpen-Adria-Universität Klagenfurt, Austria, e-mail: bernhard.rinner@aau.at

Jim Tørresen
University of Oslo, Norway, e-mail: jimtoer@ifi.uio.no

Xin Yao
University of Birmingham, UK, e-mail: x.yao@cs.bham.ac.uk

© Springer International Publishing Switzerland 2016
P.R. Lewis et al. (eds.), *Self-aware Computing Systems*, Natural Computing Series,
DOI 10.1007/978-3-319-39675-0_1

Copyrighted material

puting systems have grown at an increasingly fast rate, posing new challenges in terms of scalability and complexity. Based on our experience, we believe that current design and operation principles and methods will neither be able to scale with future systems, nor efficiently handle the variety and changing nature of requirements and optimisation goals. Novel design and operation principles and methods, such as those incorporating self-awareness (SA) and self-expression (SE), are needed.

Self-aware computing describes a new paradigm for systems and applications that proactively gather information; maintain knowledge about their own internal states and environments; and then use this knowledge to reason about behaviours. This paradigm is well suited for advanced intelligent decision making in dynamic and uncertain environments, which can in turn support effective and explainable autonomy and self-adaptation. Self-expression describes behaviours that are based on the knowledge acquired through system self-awareness, such as self-adaptation and self-explanation. A self-expressive system can adapt to its environment including its users, and thus limits the need for users to adapt to fixed system behaviours [390].

Self-awareness is not a new concept and has been studied for a long time in the fields of psychology and cognitive science [275]. However, a clear understanding and interpretation of self-awareness in computer science and engineering is lacking. There has been no universally agreed and accepted definition of self-aware computing in spite of frequent use of the word "self-aware" in different contexts. Although there are systems that are declared to be self-aware in one sense or another, little has been said about engineering methodologies that can help to build such systems. It has not previously been clear what properties a self-aware system could and should have, and what capabilities such a system might have.

This book attempts to bridge the gaps in the literature related to self-aware computing, in the spirit of recent work to translate concepts of self-awareness from psychology to computing [236]. It focuses on key ideas from self-awareness theory, leading to working definitions and pragmatic principles of *computational self-awareness* that can be used in engineering self-aware systems and understanding their behaviours. While it surveys a range of views concerning self-awareness, it does not attempt to engage in more philosophical debates on the ability of machines to achieve so-called "true" self-awareness, or what that might mean. As an example of the pragmatic approach taken in this book, building on notions from psychology, different levels of self-awareness as they apply to computing systems are described, examples are then used to illustrate what capabilities a system could have with which level(s) of self-awareness, how these might be implemented, and what benefits and costs are associated with such functionality. An engineering methodology is then introduced to facilitate the design of self-aware systems with different required capabilities.

This book builds on and extends earlier work in the related fields including autonomic computing [105] and organic computing [277] and architectures like MAPE-K and Kramer and Magee's three-layered architecture [225]. It takes an engineering approach to the design of self-aware computing systems, that includes considering different levels of self-awareness and the introduction of a reference architecture for designing self-aware and self-expressive computing systems. As a more con-

Copyrighted material

in distributed applications and briefly describes a dedicated middleware implementation.

Part IV demonstrates how self-awareness and self-expression are useful in the three widely different application domains of hardware acceleration of financial computation, object tracking in multi-camera networks, and active music systems, respectively. Chapter 12 demonstrates how complex financial models can be speeded up using reconfigurable hardware combined with optimisation algorithms. Object tracking in multi-camera networks is the topic of Chapter 13, where autonomous monitoring of each camera in a network is combined with learning mechanisms to adapt its behaviour to changing conditions. Finally, Chapter 14 illustrates how persons without musical skills can influence music in interactive music systems using nature and socially-inspired methods.

The book is a result of extensive teamwork through the EU-funded research project “Engineering Proprioception in Computing Systems (EPiCS)”. Eight research groups in five different countries collaborated to develop concepts and foundations for self-awareness and self-expression in computing systems and tested and demonstrated their usefulness in highly different domains. While the book includes some of the research results, it also, and more importantly, serves the purpose of stimulating further research into the field of self-aware and self-expressive computing systems.

Copyrighted material

Part I
Concepts and Fundamentals

Copyrighted material

Part I motivates the concepts of self-awareness and self-expression for engineering computing systems by looking into other disciplines and related concepts. It introduces a reference architecture for describing and engineering computational self-awareness and self-expression in computing systems. The architecture provides a common language which paves a way for identifying architectural patterns influencing the engineering of computational self-awareness and self-expression capabilities across a range of applications. Chapter 2 translates concepts from psychology to the domain of computing, introducing key ideas in self-aware computing. Chapter 3 relates our concepts of computational self-awareness and self-expression to other efforts in computer science and engineering under the self-awareness label. Depending on the fields, the term self-awareness may have different meanings. Chapter 4 concludes the first part by presenting our reference architecture for describing self-aware and self-expressive computing systems.

Copyrighted material

Chapter 2
Self-awareness and Self-expression: Inspiration from Psychology

Peter R. Lewis, Arjun Chandra, and Kyrre Glette

Abstract Self-awareness concepts from psychology are inspiring new approaches for engineering computing systems which operate in complex dynamic environments. There has been a broad and long-standing interest in self-awareness for computing, but only recently has a systematic understanding of self-awareness and how it can be used and evaluated been developed. In this chapter, we take inspiration from human self-awareness to develop new notions of computational self-awareness and self-expression. We translate concepts from psychology to the domain of computing, introducing key ideas in self-aware computing. In doing so, this chapter therefore paves the way for subsequent work in this book.

2.1 Introduction to Self-awareness

The *Oxford English Dictionary* defines *awareness* as "knowledge or perception of a situation or fact." Informally, we might typically consider that humans build up knowledge, or become aware of things, by perceiving the world around them. We observe interactions, listen to other people, watch television, read books, and, particularly in early life, learn through play. When considering awareness in humans, it is common to consider that all the knowledge we possess, all of our awareness, is acquired through perception. This idea was first postulated by Hume [187], who argued that all human knowledge is induced from experience. What then does it mean for a human to be *self*-aware? For Hume, the "self" is not a defined physical entity, but instead describes the bundle of experiences or perceptions unique to an

Peter Lewis
Aston University, UK, e-mail: p.lewis@aston.ac.uk

Arjun Chandra
Studix, Norway, e-mail: arjun@studix.com

Kyrre Glette
University of Oslo, Norway, e-mail: kyrrehg@ifi.uio.no

© Springer International Publishing Switzerland 2016

P.R. Lewis et al. (eds.), *Self-aware Computing Systems*, Natural Computing Series,
DOI 10.1007/978-3-319-39675-0_2

Copyrighted material

individual. A Humean form of self-awareness might then be considered to consist of an individual's knowledge of its experiences. Kant [210] criticised Hume's view, extending the scope of the self significantly, arguing that there is some entity which is the subject of these experiences, and is common through space and time. This Kantian self synthesises information from experiences with concepts held in the mind and with the imagination. Kant further argued that as an individual performs actions within the world, since its actions are based on its synthesised knowledge, they represent its self, giving rise to the self also as an object. This object in turn is something which can be perceived and experienced.

Though there is a long history of analysis of the nature of the self in philosophy, more recently, psychology has made a more pragmatic attempt to develop an understanding of the varieties of knowledge individuals possess concerning themselves. The notion of self-awareness first appears in the literature around the turn of the twentieth century [25, 382], perhaps most importantly with James [197] making the distinction between two forms of self based on the differences between the Humean and Kantian views described above. First, the *implicit* self, often referred to as the self-as-subject, or the "I", is the self which is the subject of experiences. These experiences are unique to the individual, and they are from the individual's own point of view, determined by factors such as their sensing apparatus, their situation within the world, and other factors associated with their own state. Second, the *explicit* self, or self-as-object, can be discerned. Here the self is an object of knowledge. It is a thing which can be recognised, modelled and reasoned about, including in relation to other objects in the world. An individual's awareness of its explicit self is often considered the more advanced form of self-awareness in this distinction, building on implicit self-awareness. Indeed, implicit self-awareness emerges much earlier in the lives of human infants than its explicit counterpart does [231].

One commonly considered form of self-awareness is that as measured by the so-called *mirror test* [140]. A subject being evaluated is presented with a mirror, to which it is then allowed to get accustomed. The subject is then distracted and, without its knowledge, a visible change is made to its appearance. This is usually done by marking its face, e.g., putting a spot on its cheek or forehead. The subject is then presented with the mirror again. Any behaviour directed towards this marker by the subject implies self-recognition, which is seen as being enabled by a mental representation of oneself (also known as a secondary representation). As Asendorpf et al. [18] put it:

> *"[secondary representation] is not a perception of oneself but rather a constructed mental model of oneself that can be manipulated in fantasy. Therefore, the ability to recognise oneself in a mirror that requires linking a mirror image (a primary representation) with one's self marks the capacity for secondary representation."*

Explicit self-awareness requires a subject to possess the capacity to construct such a secondary conceptual representation of itself. What then does the mirror test tell us about self-awareness? Humans, primates and some other animals have "passed" the mirror test [140, 18], however, Haikonen [155] showed that very little sophistication in computing machinery can enable a computational system with visual sensors to also pass. Haikonen therefore goes on to suggest that the ability or

Copyrighted material

2.2.3 Self-awareness in Collective Systems

So far, we have just considered self-awareness in the context of a single individual. However, Mitchell [272] notes that self-awareness can also be observed in collective systems, where there is no central point at which such self-knowledge is located. Examples of these collective systems include those comprised of individuals that might normally be considered either organisms in their own right (e.g., ants in a colony) or constituent cells of a larger organism (e.g., neurons in the brain). In these cases, it appears from an external perspective that such biological collective systems are self-aware at the level of the collective, even though this property may not be present at the level of the individual component. This awareness, Mitchell describes as being concerned with

> *"information about the global state of the system, which feeds back to adaptively control the actions of the system's low-level components. This* ***information about the global state is distributed*** *and statistical in nature, and thus is difficult for observers to tease out. However, the system's components are able, collectively, to use this information in such a way that* ***the entire system appears*** *to have a coherent and useful sense of its own state."* [272]

We have added the emphasis here, in order to highlight that a system which behaves as if it were self-aware is not necessarily required to possess a single "mind-like" component[1]. Indeed, in many cases, the entire system appears self-aware, despite only local knowledge being present at constituent parts of the collective. Self-awareness might be considered the product of emergence.

This is a key observation which can contribute to the design of self-aware systems: one need not require that such a system possess a global omniscient controller. Indeed, many natural systems appear to have been favoured by evolution which do not have such a central point of control, and rely upon relevant knowledge being available at appropriate locations within the system. It is highly likely that this idea can improve the robustness and adaptability of such systems; these are desirable properties for natural and artificial systems alike.

2.2.4 Self-expression

As we have seen, self-awareness is concerned with knowledge synthesised and held by an individual about itself and its experiences. This knowledge may be centrally held, or else distributed in nature. However, in studying the self-awareness properties of natural and computational systems, we have found it advantageous to explicitly and separately consider the related process of an individual determining its behaviour as a result of this knowledge. This process we call self-expression. In

[1] Indeed, while the brain has long been known to be a collective system composed of neurons, consciousness in the human mind is itself thought by some [98] to also be a distributed phenomenon, with nothing like what we might call global knowledge.

Copyrighted material

- **Private self-awareness**: This is a system's ability to obtain knowledge based on phenomena that are internal to itself. A system needs internal sensors to achieve this.

- **Public self-awareness**: This is a system's ability to obtain knowledge based on phenomena external to itself. Such knowledge depends on how the system itself senses/observes/measures aspects of the environment it is situated in, and includes knowledge of its situation and context, as well as (potential) impact and role within its environment.

Some prior work in self-aware computing has considered only private self-awareness, i.e., a system's awareness of its own internal state. However, the importance of the availability of external sources of information to self-awareness processes should be emphasised: self-awareness is not only concerned with sources of information internal to the individual. We argue that a full consideration of computational self-awareness also includes public aspects, where a system's knowledge of itself in relation to its social and physical environment can be synthesised with private self-awareness, in order to produce integrated conceptual models. In this book, we hope to demonstrate that the distinction, inclusion and synthesis of both public and private self-awareness raise many important questions for engineers of self-aware computing systems.

2.3.2 Levels of Computational Self-awareness

We now describe our computational framing of the levels of self-awareness, which were first presented by Faniyi et al. [127] and elaborated upon by Lewis et al. [236]. It is possible to relate the levels of self-awareness to the concepts of private and public self-awareness, and hence the sources of the relevant knowledge (i.e., based on internal or external sensors). The relevance of each level to these concepts is also described. The relationship provides an indication of the architecture that will be required in order to realise each of the levels. In each case, Neisser's level of self-awareness is given on the left, and our corresponding level of computational self-awareness is on the right.

1. **Ecological self $\longrightarrow$ Stimulus-aware**
   A system is stimulus-aware if it is able to obtain knowledge of stimuli acting upon it, enabling the ability to respond to events. It does not have knowledge of past/future stimuli. Since stimuli may originate both internally and externally, stimulus-awareness can be **private**, **public** or **both**.

2. **Interpersonal self $\longrightarrow$ Interaction-aware**
   A system is interaction-aware if it can obtain knowledge that stimuli and its own actions constitute interactions with other systems and the environment. It is able to obtain knowledge via feedback loops that its actions can provoke or cause

Copyrighted material