Intelligent Algorithms in Ambient and Biomedical Computing

Philips Research

VOLUME 7

Editor-in-Chief
Dr. Frank Toolenaar
Philips Research Laboratories, Eindhoven, The Netherlands

SCOPE TO THE *'PHILIPS RESEARCH BOOK SERIES'*

As one of the largest private sector research establishments in the world, Philips Research is shaping the future with technology inventions that meet peoples' needs and desires in the digital age. While the ultimate user benefits of these inventions end up on the high-street shelves, the often pioneering scientific and technological basis usually remains less visible.

This 'Philips Research Book Series' has been set up as a way for Philips researchers to contribute to the scientific community by publishing their comprehensive results and theories in book form.

Dr. Rick Harwig

Intelligent Algorithms in Ambient and Biomedical Computing

Edited by

Wim Verhaegh
Philips Research Laboratories, Eindhoven,
The Netherlands

Emile Aarts
Philips Research Laboratories, Eindhoven,
The Netherlands

and

Jan Korst
Philips Research Laboratories, Eindhoven,
The Netherlands

Springer

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN-10 1-4020-4953-6 (HB)
ISBN-13 978-1-4020-4953-8 (HB)
ISBN-10 1-4020-4995-1 (e-book)
ISBN-13 978-1-4020-4995-8 (e-book)

Published by Springer,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

www.springer.com

Printed on acid-free paper

Printed in the Netherlands.

Contents

Part III Technology

Contributing Authors

Emile Aarts
Philips Research Laboratories Eindhoven
High Tech Campus 36, 5656 AE Eindhoven, The Netherlands
emile.aarts@philips.com

Lalitha Agnihotri
Philips Research Laboratories USA
345 Scarborough Rd., Briarcliff Manor, NY 10510, USA
lalitha.agnihotri@philips.com

Zharko Aleksovski
Philips Research Laboratories Eindhoven
High Tech Campus 31, 5656 AE Eindhoven, The Netherlands
zharko@cs.vu.nl

Mauro Barbieri
Philips Research Laboratories Eindhoven
High Tech Campus 34, 5656 AE Eindhoven, The Netherlands
mauro.barbieri@philips.com

Richard Y. Chen
Philips Research Laboratories USA
345 Scarborough Rd., Briarcliff Manor, NY 10510, USA
richard.chen@philips.com

Chun-Ting Chou
University of Michigan
1301 Beal Avenue, Ann Arbor, MI 48109, USA
choujt@umich.edu

Christopher D. Clack
University College London
Gower Street, London WC1E 6BT, United Kingdom
clack@cs.ucl.ac.uk

Nevenka Dimitrova
Philips Research Laboratories USA
345 Scarborough Rd., Briarcliff Manor, NY 10510, USA
nevenka.dimitrova@philips.com

Aukje E.M. van Duijnhoven
Numerando Groep
Amersfoortseweg 10e, 3705 GJ Zeist, The Netherlands
aukjevanduijnhoven@hotmail.com

Gijs Geleijnse
Philips Research Laboratories Eindhoven
High Tech Campus 34, 5656 AE Eindhoven, The Netherlands
gijs.geleijnse@philips.com

Frank van Harmelen
Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
frank.van.harmelen@cs.vu.nl

Herman J. ter Horst
Philips Research Laboratories Eindhoven
High Tech Campus 31, 5656 AE Eindhoven, The Netherlands
herman.ter.horst@philips.com

Nick de Jong
Tiobe Software B.V.
De Zaale 11, 5612 AJ Eindhoven, The Netherlands
nick.de.jong@tiobe.com

Warner ten Kate
Philips Research Laboratories Eindhoven
High Tech Campus 31, 5656 AE Eindhoven, The Netherlands
warner.ten.kate@philips.com

Martin L. Kersten
Centrum voor Wiskunde en Informatica
Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
martin.kersten@cwi.nl

Jan Korst
Philips Research Laboratories Eindhoven
High Tech Campus 34, 5656 AE Eindhoven, The Netherlands
jan.korst@philips.com

Ben J.A. Kröse
University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
krose@science.uva.nl

Akash Kumar
Philips Research Laboratories Eindhoven
High Tech Campus 45, 5656 AE Eindhoven, The Netherlands
akakumar@natlab.research.philips.com

Martin McKinney
Philips Research Laboratories Eindhoven
High Tech Campus 36, 5656 AE Eindhoven, The Netherlands
martin.mckinney@philips.com

Steffen Pauws
Philips Research Laboratories Eindhoven
High Tech Campus 34, 5656 AE Eindhoven, The Netherlands
steffen.pauws@philips.com

Sai Shankar N.
Qualcomm Standards Engineering Dept.
5775 Morehouse Drive, San Diego, CA 92121, USA
nsai@qualcomm.com

Sergei Sawitzki
Philips Research Laboratories Eindhoven
High Tech Campus 31, 5656 AE Eindhoven, The Netherlands
sergei.sawitzki@philips.com

Ruediger Schmitt
Philips Research Laboratories USA
345 Scarborough Rd., Briarcliff Manor, NY 10510, USA
ruediger.schmitt@philips.com

Berry Schoenmakers
Technische Universiteit Eindhoven
Den Dolech 2, 5612 AZ Eindhoven, The Netherlands
berry@win.tue.nl

Kang G. Shin
University of Michigan
1301 Beal Avenue, Ann Arbor, MI 48109, USA
kgshin@umich.edu

Arno P.J.M. Siebes
Universiteit Utrecht
Padualaan 14, 3584 CH Utrecht, The Netherlands
arno.siebes@cs.uu.nl

Janto Skowronek
Philips Research Laboratories Eindhoven
High Tech Campus 36, 5656 AE Eindhoven, The Netherlands
janto.skowronek@philips.com

Pim Tuyls
Philips Research Laboratories Eindhoven
High Tech Campus 34, 5656 AE Eindhoven, The Netherlands
pim.tuyls@philips.com

Wim F.J. Verhaegh
Philips Research Laboratories Eindhoven
High Tech Campus 34, 5656 AE Eindhoven, The Netherlands
wim.verhaegh@philips.com

Michael Verschoor
Technische Universiteit Eindhoven
Den Dolech 2, 5612 AZ Eindhoven, The Netherlands
m.p.f.verschoor@tm.tue.nl

Nikos Vlassis
University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
vlassis@science.uva.nl

Wojciech Zajdel
University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
wzajdel@science.uva.nl

Kees van Zon
Philips Research Laboratories USA
345 Scarborough Rd., Briarcliff Manor, NY 10510, USA
kees.van.zon@philips.com

Preface

The rapid growth in electronic systems in the past decade has boosted research in the area of computational intelligence. As it has become increasingly easy to generate, collect, transport, process, and store huge amounts of data, the role of intelligent algorithms has become prominent in order to visualize, manipulate, retrieve, and interpret the data. For instance, intelligent search techniques have been developed to search for relevant items in huge collections of web pages, and data mining and interpretation techniques play a very important role in making sense out of huge amounts of biomolecular measurements. As a result, the added value of many modern systems is no longer determined by hardware only, but increasingly by the intelligent software that supports and facilitates the user in realizing his or her objectives.

Over the past years, considerable progress has been made in the area of computational intelligence, which can be positioned at the intersection of computer science, discrete mathematics, and cognitive science. This has led to a growing community of practitioners within Philips Research that develop, analyze, and apply intelligent algorithms. The Symposium on Intelligent Algorithms (SOIA) intends to provide this community of practitioners with a platform to exchange information. The first edition of SOIA, held in 2002, addressed the topic of intelligent algorithms in ambient intelligence. To share the output of the symposium with a larger audience, a selection of papers was edited and published by Kluwer in the Philips Research Book Series under the title "Algorithms in Ambient Intelligence." For the second edition, held in 2004, the scope of the symposium was broadened so as to comply with the three main topics of the Philips company strategy, i.e., Healthcare, Lifestyle and Technology. Again a selection of papers was edited, resulting in the present book. It consists of 17 chapters, divided over three parts corresponding to the strategic topics mentioned above. The main topic in Healthcare is the understanding of biological processes, for Lifestyle the main topic is content retrieval and manipulation, and finally for Technology most contributions relate to media processing. Below we present more detailed information about the individual chapters.

Part I consists of four chapters. In Chapter 1, Chris Clack discusses the topic of modeling biological systems, thus allowing to perform in-silico experiments by means of computer simulation, to formulate hypotheses. In Chapter 2, Nevenka Dimitrova gives an overview of the reverse approach, where one does not use computers to simulate biological processes, but where one uses biology to perform computations, in DNA computing and synthetic biology. In Chapter 3, Martin Kersten and Arno Siebes discuss data management inspired by biology, resulting in an organic database system. In Chapter 4, Kees van Zon discusses how to achieve machine consciousness, and how it can be applied.

Par II consists of eight chapters, addressing problems from the area of content management and retrieval. In Chapter 5, Wim Verhaegh discusses the problem of making a schedule of preferred TV programs, while at the same time selecting TV programs for recording, under the assumption of a limited number of tuners. In Chapter 6, Mauro Barbieri, Nevenka Dimitrova, and Lalitha Agnihotri present a technique to automatically summarize video into a condensed preview, allowing one to quickly browse and access large amounts of stored programs. Chapters 7–9 concerns audio applications. First, Janto Skowronek and Martin McKinney discuss in Chapter 7 the topic of automatic classification of audio and music, for which they developed the automatic extraction of the higher-level feature of percussiveness. In Chapter 8, Steffen Pauws presents a technique to automatically extract the key from a piece of music, providing an emotional connotation to it, and making it possible to build well-sounding music mixes. In Chapter 9, Zharko Aleksovski, Warner ten Kate, and Frank van Harmelen address the problem of combining multiple databases of music data in a semantic way, by approximating matches of music classes. Next, Jan Korst, Gijs Geleijnse, Nick de Jong, and Michael Verschoor discuss in Chapter 10 the possibilities to fill a knowledge database, using an ontology to collect and structure data from web pages. In the last chapter of part II, which Wim Verhaegh, Aukje van Duijnhoven, Pim Tuyls, and Jan Korst resolve the privacy issue of population-based recommenders by encrypting the users' profiles and performing the required algorithms on encrypted data.

Part III consists of six chapters, focusing on the technology underlying intelligent algorithms and intelligent systems. The first two chapters discuss theoretical aspects of intelligent algorithms. In Chapter 12, Peter Grünwald gives an overview on the minimum description length principle to resolve the problem of model selection, based on the fundamental idea to see learning as a form of data compression. In Chapter 13, Herman ter Horst discusses the computational complexity of reasoning with semantic web ontologies, such as RDF Schema and OWL. Next, Wojciech Zajdel, Ben Kröse, and Nikos Vlassis present in Chapter 14 an introduction to dynamic Bayesian networks, and show their application in robot localization and multiple-person tracking. In

Chapter 15, Berry Schoenmaker and Pim Tuyls discuss efficient protocols for securely matching two user profiles, without leaking information on the details of the profiles. Finally, Chapters 16 and 17 address resource issues in intelligent systems. In Chapter 16, Sai Shankar N., Richard Chen, Ruediger Schmitt, Chun-Ting Chou, and Kang Shin revisit fairness in multi-rate wireless networks, and present a solution to fairly schedule airtime. Finally, in Chapter 17, Akash Kumar and Sergei Sawitzki discuss the design alternatives of Reed Solomon decoders, and address the problem of making optimal design decisions to obtain a high-throughput, low-power solution.

We are convinced that the chapters presented in this book comprise an interesting collection of examples of the use of intelligent algorithms in different settings, and that the book reconfirms that the area of computational intelligence is a truly challenging field of research.

WIM F.J. VERHAEGH, EMILE AARTS, AND JAN KORST
Philips Research Laboratories Eindhoven

Acknowledgments

We would like to thank the following people who helped us to review the contributed chapters: Dee Denteneer, Nevenka Dimitrova, Hans van Gageldonk, Srinivas Gutta, Herman ter Horst, Jan Nesvadba, Dave Schaffer, and Peter van der Stok.

Part I

HEALTHCARE

Chapter 1

BIOSCIENCE COMPUTING AND THE ROLE OF COMPUTATIONAL SIMULATION IN BIOLOGY

Christopher D. Clack

Abstract Bioscience computing exploits the synergy of challenges facing both computer science and biology, drawing inspiration from biology to solve computer science challenges and simultaneously using new bio-inspired adaptive software to model and simulate biological systems. This chapter first provides an introduction to bioscience computing — discussing the role of computational simulation in terms of hypothesis formulation and prototyping for biologists and medics, and explaining how bioscience computing is both timely and well-suited to systems biology. A concrete example of computational simulation is then provided — the artificial cytoskeleton, which utilises swarm agents and a cellular automaton to model cell morphogenesis. Morphological adaptation for tasks such as chemotaxis and phagocytosis are presented, and the role of the artificial cytoskeleton and its swarm-based techniques in both computer science and biology is explained.

Keywords Bioscience computing, systems biology, computational simulation, morphogenesis, adaptive systems, agent based modelling, swarm agents.

1.1 Introduction to bioscience computing

Bioscience computing exploits the synergy of challenges facing both computer science and biology, drawing inspiration from biology to solve problems in computer science and simultaneously using new bio-inspired adaptive software to model and simulate self-organising, adaptive, biological systems.

There has recently been a substantial increase in inter-disciplinary research interactions between computer science and the life sciences. From the biologist's perspective, the post-genomic era is characterised by huge amounts of data but little understanding of how genes map to physiological functions, and there is an urgent need for the application of intelligent computing techniques to gain increased understanding. From the computer scientist's perspective, the new biological data and expanding understanding of biological processes

Wim F.J. Verhaegh et al. (Eds.), Intelligent Algorithms in Ambient and Biomedical Computing, 3-19.

provide both an excellent driver for new methods in bioinformatics and an increasing source of ideas for new computational techniques in areas such as intelligent systems and artificial life.

The purpose of the first part of this chapter is to provide an introduction to the biological context and to explain the role of bioscience computing within that context.

1.1.1 A change of focus in biology and medicine

The traditional reductionist view of biology is rooted in analysis and biophysics; it is based on a hierarchical perspective where the functioning of the physiome[1] is the deterministic product of a 'one-way upward causation from genes to cells, organs, system and whole organisms' [Noble, 2002], and has been remarkably successful with fundamental achievements such as discovering the structure of DNA and mapping the genome for not one but several organisms. The traditional role of computer science in biology (e.g. of bioinformatics) has been to support this endeavour by providing data-handling, data visualisation, numerical simulation and data-mining services.

However, in the post-genomic era the super-abundance of data and relative paucity of understanding, coupled with a clearer perspective of the complexity of living organisms, are causing biologists to question whether the traditional view is sufficient as a basis for a full understanding of nature. The traditional view is giving way to a new biology, often referred to as systems biology.

The rise of systems biology has caused a much closer relationship to develop between biologists and computer scientists. In systems biology, the computer science techniques are no longer merely a data service to the biologists, but are intimately involved in the formulation of biological hypotheses as biologists embrace the process-oriented world of the computer scientist. systems biology considers an organism as a self-organising, adaptive, complex, dynamic system providing an information framework with global constraints and multiple feedback and regulation paths between high and low levels (e.g. controlling gene expression); the sub-modules are too inextricably connected, there are too many interactions between levels, for a one-way hierarchy to be possible [Noble, 2002]. Biologists now experiment not just in-vivo and in-vitro, but increasingly *in-silico*. These in-silico experiments are the basis for what we term bioscience computing.

1.1.2 Modelling and simulation

The primary aims of modelling and simulation in biology are to improve understanding of a process or hypothesis, to highlight gaps in knowledge, and

[1] A glossary of biological terms is provided in Table 1.1 at the end of this chapter.

to make clear, testable predictions [Kirkwood et al., 2003]. Note, however, that an in-silico experiment itself can never truly be used to *test* a biological hypothesis — rather, computational simulation in biology should be viewed as a process of *prototyping to assist hypothesis formulation*.

Wet-lab experimental techniques tend to focus analytic attention on single mechanisms. By contrast, computational simulation can contribute to the activity of synthesis, of integrating many separate elements that form a network of activity. The resultant interaction and synergy can provide a qualitatively much improved experimental framework. These in-silico results may then guide the choice of (more expensive) subsequent wet-lab experiments.

Techniques. There is a wide spectrum of techniques available to support modelling and simulation, ranging from high-level phenomenological approaches which generally represent qualitative features of a system, to low-level mechanistic simulations which typically represent quantitative aspects (though abstraction and quantification need not be mutually exclusive concepts [Ideker & Lauffenburger, 2003]). Examples of available techniques include statistical data-mining, clustering and classification (e.g. support vector machines), Bayesian networks, Markov chains, fractal theory, Boolean logic, and fuzzy logic. At the mechanistic extreme there are cellular automata and agent-based simulations. Differential equations are widely used and capable of capturing detail at varying levels of abstraction. See Figure 1.1.

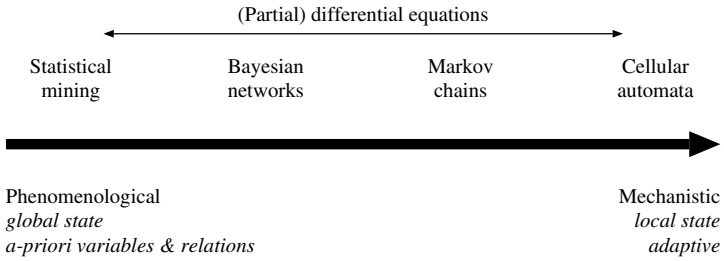

Figure 1.1. Comparative spectrum of available techniques.

Phenomenological models tend to focus on the *global* state of a system. Often they describe an a-priori given set of relations between an a-priori given set of variables [Giavitto et al., 2002]; the two sets cannot evolve jointly with the running system, and very few of these models successfully capture a rich enough semantics to be able to predict complex behaviour [Anderson & Chaplain, 1998]. By contrast, mechanistic models provide local interaction modelling, where cells react (often adaptively) to a *local* environment, not to the state of the system as a whole (thereby supporting heterogeneity). This leads to a rich model of spatiotemporal dynamics, and offers insights into the

parameters and mechanisms responsible for system dynamics [Gatenby & Maini, 2003] and for collective organisational behaviour at the microscopic level [Patel, 2004].

Differential equations and partial differential equations provide an excellent mechanism for detailed expression of behaviours of many kinds, but are unsatisfactory for some highly detailed spatiotemporal behaviours [Araujo & McElwain, 2004]. For example, where precise local effects due to intermolecular interactions and random molecular movement are required, a great number of equations must be generated and solved [Succi et al., 2002]. In practice, the computational limits on solving a large number of related partial differential equations leads to the technique normally being applied only to *abstractions* of internal mechanisms and processes.

An interesting mechanistic approach is the use of cellular automata — e.g. Scalerandi's 2D model of cardiac growth dynamics [Scalerandi et al., 2002]. When coupled with agent-based modelling, using a 'swarm' of thousands of tiny agents (a mechanism itself inspired by nature) each representing a separate macromolecule, this method has the advantages of both mathematical simplicity and that the spatiotemporal fates of individual components (cell, proteins etc.) can be tracked in minute detail. The resulting system is very good at representing spatiotemporal dynamics and organisational behaviour, particularly for the simulation of adaptive behaviour.

Objects and processes. The specific attraction of computational simulation is that the computational approach corresponds more naturally to the way that biologists think about their subject. Biologists (in particular molecular biologists) naturally focus on *objects*, *interactions* and *processes*.

Computational simulation permits biologists to express biological systems in terms of computational objects, interactions and processes that relate directly to their biological counterparts and are therefore far easier to understand and easier to manipulate than differential equations. Computational simulations can be expressed in terms of information networks and can use interaction-centric models (e.g. local-neighbourhood operations within a cellular automaton grid), all of which naturally map onto (for example) cell structure and the interaction of macromolecules.

The experience of systems biology has been that biologists have increasingly adopted the computational systems concepts of computer scientists. This should not come as a surprise, since computer scientists have extensive experience of building, modelling, and simulating complex systems that require analysis and synthesis at many different levels of abstraction.

1.1.3 A computational approach to biological complexity

The computational approach to biology enables simulations as dynamic emergent hierarchies of biological complexity, with interactions and feedback between the levels, for example as illustrated in Figure 1.2. At the lowest level, system components are lightweight agents governed by local-neighbourhood rules. The rules provide the system of dynamic interaction between agents, and from this comes the self-organising properties of the simulated organism (threshold parameters may need to be derived via automatic search methods). The emergent behaviour of the system is dependent on a combination of the competitive and co-operative interactions of the underlying local-neighbourhood rules, the regulatory effects that arise from the self-organising properties of those rules, and sets of global constraints (which may be derived from experimental observation). The result is a complex, dynamic system, which can itself be considered as an agent in a larger network of agents of similar complexity, each undergoing interactions according to local-neighbourhood rules at a higher level, and from which yet more complex behaviour emerges.

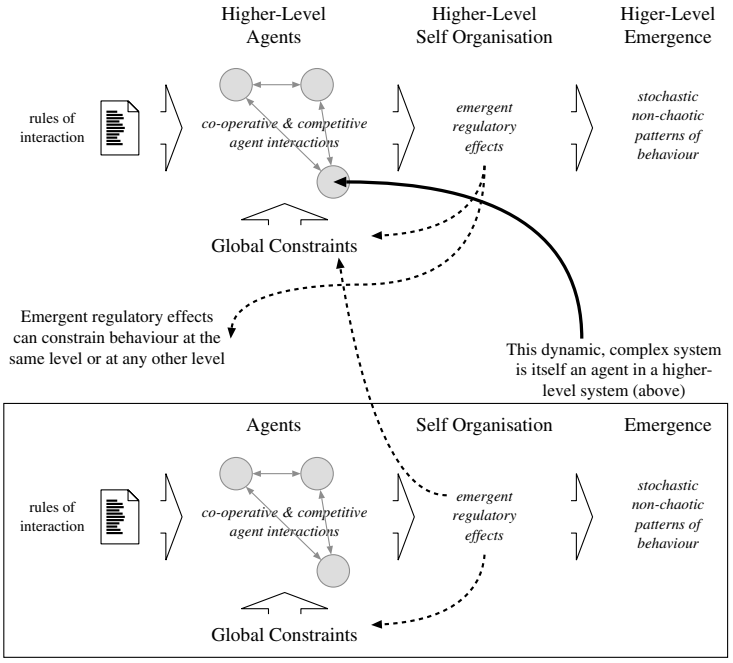

Figure 1.2. The dynamic emergence of hierarchies of biological complexity.

While emergent behaviour has the potential for chaotic results, in a hierarchy of levels each can constrain the realisable solutions of the other

levels — thus, an understanding of dynamic emergence in complex hierarchies is a fundamental step in understanding the underlying mechanisms of biology.

1.1.4 Summary: The role of bioscience computing

The first part of this chapter has explored the role of bioscience computing in biology and argued that it is both timely and well-suited to the emergence of systems biology: it provides *in-silico* experiments; focuses on interactions and integration of concurrent mechanisms; is intimately involved in the formulation of biological hypotheses; manipulates objects, processes and interactions; is mathematically straightforward, with a low barrier to uptake; and captures rich spatiotemporal detail at low computational cost.

1.2 Simulating adaptive behaviour

This second part provides a concrete example of the bioscience computing techniques discussed in the first part of this chapter, and presents the *artificial cytoskeleton*, a computational simulation of the development and adaptation of the shape and form of an organism: *morphogenesis*. The work is more fully described by Bentley & Clack [2004; 2005].

Organisms in nature exhibit complex adaptive behaviour that far surpasses the ability of current state-of-the-art autonomous software and robotics. Our research focuses on morphological adaptation, the continuous lifetime reconfiguration of phenotypic form (shape) exhibited by natural systems in order to continue to survive in a changing environment. Many unicell organisms exhibit complex adaptations of their shape in rapid response to environmental changes — e.g. fibroblast cells change shape to assist movement during wound healing, and immune system cells change shape to eat invading bacteria — even though they have no centralized control system. We aim to understand the underlying mechanisms and principles that govern this adaptive behaviour, to explore the concept of morphological adaptation as a mapping from environment to phenotype rather than merely from genotype to phenotype, and to draw inspiration from those mechanisms to improve the adaptive behaviour of artificial systems.

The detailed spatiotemporal aspects of morphogenesis are difficult to compute using partial differential equations and so we turned to a bioscience computing technique; a cellular automaton and agent-based computing using a very large number of simple agents (‘swarm’ agents).

1.2.1 The artificial cytoskeleton

Our mechanism, the ‘artificial cytoskeleton’, is closely modelled on the eukaryotic cytoskeleton, a complex, dynamic network of protein filaments which extends throughout the cytoplasm and which gives the cell dynamic structure

and function. In particular *actin* cytoskeleton microfilaments are involved in rapid changes to membrane shape in response to environmental signals [Alberts et al., 1994]. We use agent-based swarm techniques combined with 3D cellular automaton (CA) rules to allow proteins to exist and interact with their 26 nearest neighbours in a 3D voxellated environment. The agent-based swarm technique permits the modelling and tracking of individual components and their interactions. The CA simplifies visualization, supports 3D spatial placement and movement, and reduces system complexity. The combination of the two techniques (agent-based swarm and CA) provides opportunities for optimizing computational overhead (e.g. it is not always necessary to compute interactions for all cells in the CA — only those that contain or abut an agent). The CA rules for chemical diffusion and agent interactions can be checked against current understanding of the biology.

The artificial cytoskeleton resides within a membrane-bound 'cell' and receptors (sensors) in the membrane relay external signals to the artificial cytoskeleton via a pathway of protein reactions: the transduction pathway (TP). See Figure 1.3. For efficiency, the artificial cytoskeleton and transduction pathway comprise only a small selection of proteins — just those necessary for a particular experiment. The artificial cytoskeleton's non-rigid form permits it to disassemble rapidly and re-form in a more advantageous distribution; it constantly responds to environmental cues by reorganizing, i.e. altering the cell's internal topography and the membrane morphology.

The underlying mechanism. The artificial cytoskeleton consists of *structural* proteins (actin and a nucleator), which make up the filaments, and several *accessory* proteins, which regulate a filament's behaviour (e.g. inhibiting, activating, severing, bundling). Environmental signals filter into the cell via the transduction pathway, affecting concentrations of accessory proteins and structural protein behaviour. The cooperative and competitive interactions of these structural and accessory proteins can dramatically alter the cytoskeleton's filamentous structure, affecting the shape and structure of the cell as a whole, and resulting in rich diversity in cell shape [Alberts et al., 1994].

The protein interactions are defined by a set of functions; these functions encapsulate the complete mapping from environmental cues to cell morphology (which in turn may affect the environment). We call this function set the 'environment-phenotype map' (or 'E-P map'). Different cell behaviours may require different E-P maps. The following explanation of the underlying mechanism will focus on the E-P map for chemotaxis; see Figures 1.3 and 1.4.

Each voxel in the cellular automaton contains one of the following units:

1. environment which may contain concentrations of a chemoattractant '*C*'.

2. cytoplasm which may contain concentrations of the protein profilin;

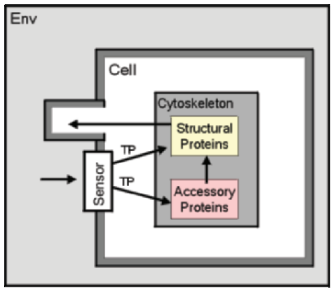

Figure 1.3. A generalized environment-phenotyope map. The cytoskeleton is affected by input from the environment (Env) via the transduction pathway (TP) and can affect the shape of the cell, and thereby also the environment.

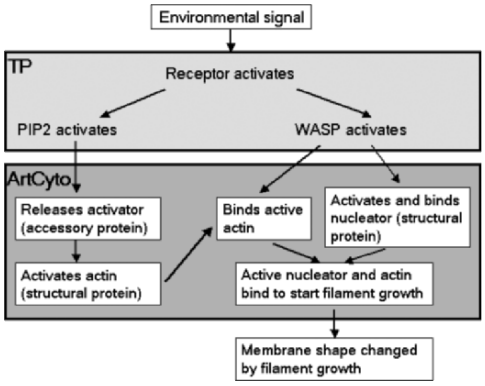

Figure 1.4. The environment-phenotype map as by Bentley & Clack [2004] abstracted from the biological pathway for fibroblast chemotaxis. The simplified transduction pathway (TP) contains a receptor and two macromolecules PIP2 and WASP, which convey information to the artificial cytoskeleton (ArtCyto).

3. an agent which may be either:

actin: which may be in the states S-actin (inactive), P1, P2, or F-actin (in a filament) and which has 2 opposing binding sites ('+','–'); or

a nucleator: the protein complex 'Arp 2/3', which may be switched on or off and has one binding site.

The interactions of these two agents drive the creation, growth and dissociation of actin filaments. The growth of actin filaments forces local membrane shape changes, therefore altering the cell's overall shape.

4. cell membrane which may contain a receptor and/or the two transduction pathway proteins WASP and PIP_2.

The membrane separates the cell from the environment. Initially, no membrane units contain WASP or PIP_2 but each has a probability of containing a receptor.

Cell surface receptors are embedded in the membrane and mediate signals from the external environment to the cytoskeleton. Membrane units containing receptors sum the concentration of C in their adjacent environment voxels. If the sum exceeds a threshold, a cascade reaction inside the cell is triggered; WASP and PIP2 are activated for the receptor and for its adjacent membrane voxels. If the receptor deactivates, WASP and PIP_2 deactivate. See Figure 1.5.

The **WASP** proteins, when activated by a receptor, recruit agents *nucleator* and *P1 actin* to the membrane (see below for a further explanation of recruitment). A recruited nucleator agent will switch on and recruited P1 actin changes state to P2 actin. Activated **PIP_2** releases a one-off plume of protein profilin which diffuses through cytoplasm units. Deactivated PIP_2 causes *removal* of all profilin in the membrane unit's adjacent cytoplasm voxels [Holt & Koffer, 2001].

Protein behaviour is governed by both general rules and specific rules of interaction. The general rules are:

1. *Diffusion:* accessory proteins are represented as concentration gradients which diffuse through cytoplasm voxels. Diffusion is calculated as by Glazier & Graner [1993]; each cytoplasm voxel has a protein threshold, the excess being evenly distributed to its cytoplasm neighbours.

2. *Random movement:* when not bound or stuck, an agent moves randomly. When it moves to a new position, the protein concentration currently in that position is diffused away and the voxel acquires the agent's identifier; the agent's previous voxel becomes cytoplasm.

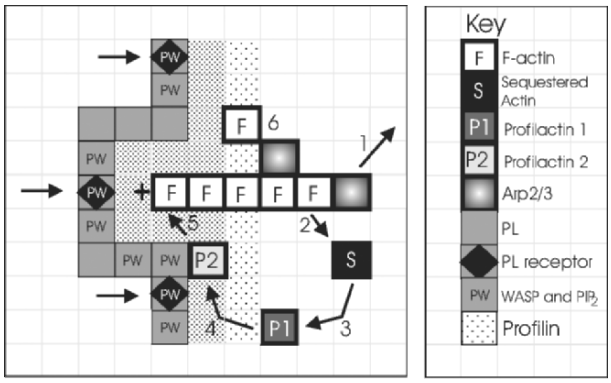

Figure 1.5. Artificial cytoskeleton interactions. Receptors detect chemoattractant, WASP and PIP_2 activate and cause the cytoskeletal behaviours shown in stages 1– 6, see text for details.

3. *Recruitment:* the biological concept of recruitment of proteins, to a specific protein *S*, is modelled as follows: an agent follows random movement until it encounters an *S* in its nearest neighbours. It then can only move such that an *S* is still in its nearest neighbours. Recruitment stops if there is no *S* nearest neighbour.

The specific rules of interaction for the chemotaxis environment-phenotype map consist of rules governing actin filament formation (and destruction) and rules governing modifications to the shape of the cell membrane. These are illustrated in Figure 1.5 and the stages are described in detail below:

An actin filament (AF) is created when a nucleator agent combines with an actin agent. Figure 1.5 illustrates a chain of F-actin agents 'F' and a nucleator ('Arp2/3'). Each F-actin agent has two binding sites ('+'/'–'): filament growth occurs at the end with the exposed '+' binding site. Subsequently other actin agents may join the filament by attaching to an actin agent already in the filament. Over time the nucleator disassociates (and un-sticks) from its AF and deactivates (stage 1). Similarly actin in a filament (F-actin) loses affinity for the filament allowing cofilin (a severing protein) to disassociate it; it then gets sequestered and changes to the inactive S-actin state (stage 2). Disassociation always occurs at the filament's '–' end. The actin or nucleator agent disassociates with a probability that increases with time spent in a filament. As the '+' end of the filament grows, the '–' end shrinks and the filament, as a higher level entity, moves towards the membrane.

Actin agents are initiated in the inactive state S-actin; S-actin units sum the concentration of profilin in their nearest neighbours — if it exceeds the threshold then the actin binds to profilin and changes to state P1, removing an amount of profilin from the surrounding cytoplasm (stage 3). P1 actin is recruited to active WASP to form P2 (stage 4). After recruited movement,

if P2 actin has an actin filament '+' site in its nearest neighbours, it binds to it, changes state to F-actin, releases profilin to the surrounding cytoplasm, and moves to the nearest neighbour cytoplasm voxel that permits its '–' site to directly abut the actin filament '+' site (stage 5).

A nucleator agent activates when recruited by WASP and then can nucleate (start) actin filaments and set their orientation by binding to a P2 actin agent in its nearest neighbours (also see push-out rule below). If there is a fully bound F-actin nearest neighbour, then a nucleator can also 'stick' to it and nucleate a *branch* actin filament (stage 6 [Alberts et al., 1994]).

There are three interactions affecting the cell membrane:

1. A gap must exist or be created between the AF's '+' end and the membrane to allow P2 actin to bind either to F-actin or a nucleator. Adjacent membrane is 'pushed-out' — the membrane voxels become cytoplasm and the adjacent environment voxels become membrane (*C* is diffused away first).[2] The precise biology for this process is unclear [Condeelis, 2001].

2. To keep cytoplasm volume constant following 'push-out', the cytoplasm or agent (but not F-actin) voxel within the cell that is furthest from the newly created cytoplasm is replaced with a membrane voxel (any affected profilin or agent is displaced).

3. If a membrane unit has no contact with inner cellular units, it is removed (becomes an environment unit); this ensures there are no doubled-up layers of membrane.

The combination of the above three interactions contracts the cell at the opposite side to a leading edge and allows the cell's centre of mass to move.

Experiments. *Chemotaxis experiment.* The artificial cytoskeleton was tested in a simple experiment based on animal cell chemotaxis, requiring the cell to undergo transformations in form in response to an external chemical stimulus. The specific E-P map for our chemotaxis experiment [Bentley & Clack, 2004] is given in Figure 1.4. The artificial cytoskeleton's response to the stimulus mimicked that of a real fibroblast cell (Figure 1.6), forming a leading edge with protrusions. It moved towards the chemical source purely by lifetime adapation of shape: see Figure 1.7.

Phagocytosis experiment. In nature, a single adaptive mechanism is able to provide different morphologies in response to different environmental stimuli.

[2] After implementing this rule, a nucleator would switch off as it would no longer have WASP nearest neighbours, so we permit a nucleator to remain switched on if any of its 26 nearest neighbours or any of their surrounding 98 voxels contain WASP.