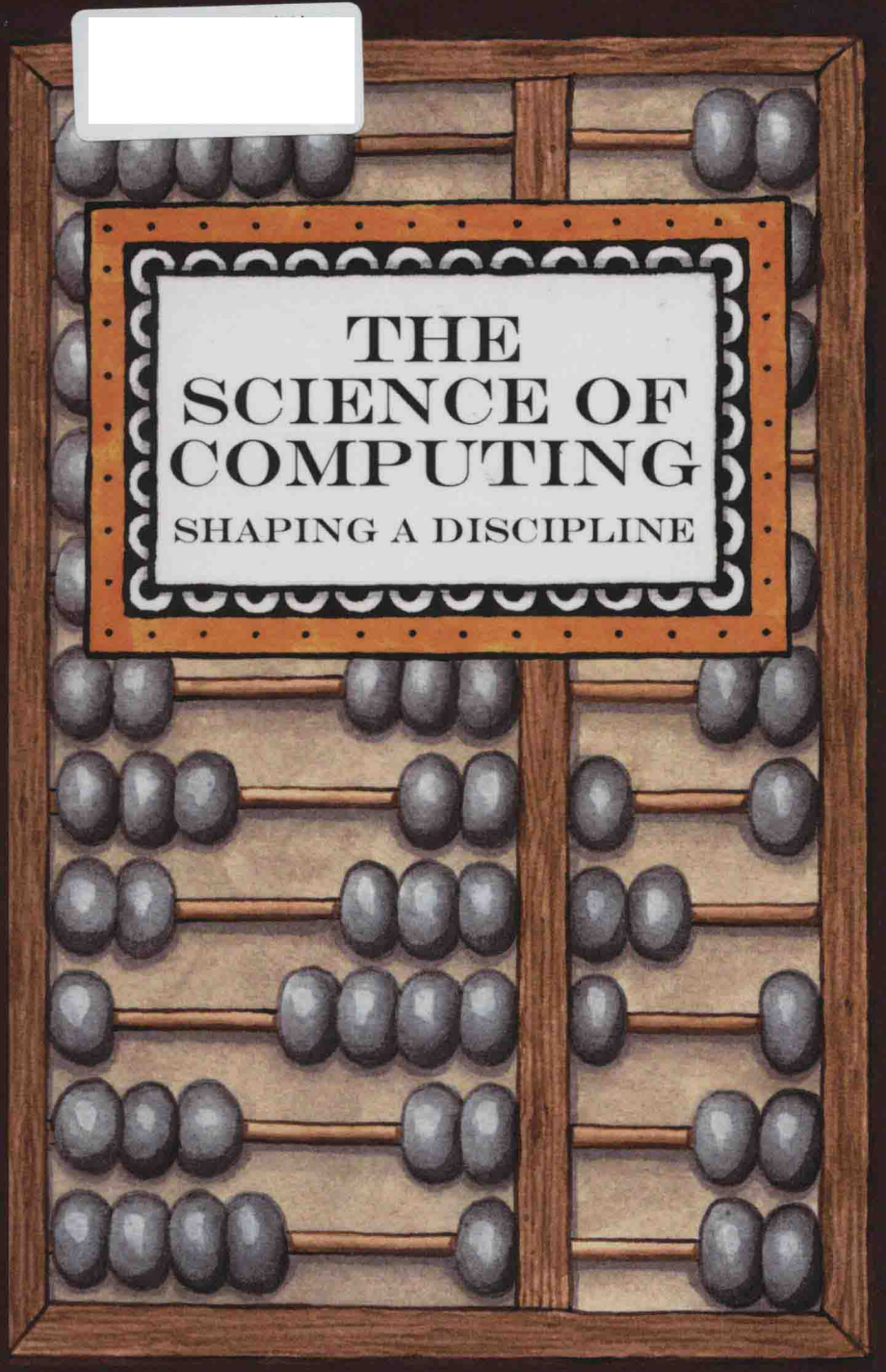
MATTI TEDRE



THE SCIENCE OF COMPUTING

SHAPING A DISCIPLINE

CRC Press
Taylor & Francis Group

A CHAPMAN & HALL BOOK

THE SCIENCE OF COMPUTING

SHAPING A DISCIPLINE

MATTI TEDRE, PH.D.

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper
Version Date: 20141023

International Standard Book Number-13: 978-1-4822-1769-8 (Paperback)

Library of Congress Cataloging-in-Publication Data

Tedre, Matti.
The science of computing : shaping a discipline / author, Matti Tedre.
pages cm
Includes bibliographical references and index.
ISBN 978-1-4822-1769-8 (pbk.)
1. Computer science. I. Title.

QA76.T4116 2014
004--dc23 2014039070

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Preface

"That's not computer science," a professor told me when I abandoned the traditional computer science and software engineering study tracks to pursue computing topics that I thought to be more societally valuable. Very quickly I learned that the best way to respond to such remarks was with a series of counter questions about what exactly is computer science and why. The difficulties that many brilliant people had responding to those questions led me to suspect that there's something deeper about the topic, yet the more I read about it, the more confused I got. Over the years I've heard the same reason—"That's not computer science"—used to turn down tenure, to reject doctoral theses, and to decline funding. Eventually I became convinced that the nature of computing as a discipline is something worth studying and writing about.

Fortunately enough, the word "no" doesn't belong to the vocabulary of Professor Erkki Sutinen, who became my supervisor, mentor, colleague, and friend. Throughout my studies in his group I worked on a broad variety of applied computing topics, ranging from unconventional to eccentric, yet in the meanwhile Erkki encouraged me to continue to study computing's disciplinary identity, and I ended up writing, in a great rush, a thesis on the topic. When I moved from the University of Eastern Finland to Asia and then to Africa for the better half of a decade, I kept on writing small practice essays on computing's identity. And as decent journals kept on publishing those essays, I continued to work on the topic. Many years' worth of evenings in the quiet African town of Iringa, one full day's drive away from the bustling Dar es Salaam, gave me the time and mental space to finally read enough about what computing's pioneers over the years have said about computing. I continued that work at Stockholm University, where my department's management encourages the researchers to do whatever they want to do. That, and a nine-month research break in 2013, enabled me to put it all on paper. Eventually, the hardest part about writing this book was putting an end to it: I still have hundreds of bits and pieces that would amend the book in important ways. I guess that, like dissertations, books like this are never finished, but abandoned when stress grows unbearable.

Over the many years that it has taken to finish this book, I have accumulated a great debt of gratitude. During the past year, many people read parts of this manuscript at different stages of its development. I wish to thank Peter Denning for amending the story with his vision and experiences as well as for sharing many behind-the-scenes stories of events described in this book;

Jan van Leeuwen for pages after pages of insights and suggestions on improving the work; Gordana Dodig-Crnkovic for her enthusiastic support and careful reading of the manuscript; Edgar Daylight for pointing out problems and oversimplifications; Johannes Cronjé for research vocabulary; and Taylor & Francis's anonymous reviewer for a great many suggestions for improvement (and what must have been two penfuls of red ink). I wish to thank Jorma Sajaniemi and Viola Schiaffonati for many constructive comments on the manuscript, and Thomas Haigh for pointing out references that I should not miss. As always, although I received a great amount of help, I alone bear responsibility for incorrect interpretations and factual errors.

Over the years, numerous colleagues have shared their insights and ideas; colleagues with whom I worked at University of Eastern Finland, Ajou University in South Korea, Tumaini University in Tanzania, University of Pretoria and Cape Peninsula University of Technology in South Africa, and Stockholm University in Sweden. I apologize that I have included in the above list only those people who had direct influence on this particular manuscript and not those who have influenced my work or thinking in other ways. I particularly apologize to those colleagues whose articles have not received adequate attention from me during the writing of this book (especially Mikko, Henrik, and Jyri). I thank my closest gene pool for my existence and brotherhood. This work received funding from the Academy of Finland grant #132572, the Finnish Association of Non-fiction Writers, the Ella and Georg Ehrnrooth Foundation, and Kauppaneuvos Otto Malmin Lahjoitusrahasto. The artwork is by Lasse Meriläinen.

The greatest gratitude I owe is to Nella for the mornings, for the evenings, and for all the moments between.

Matti Tedre
Stockholm, Sweden

Contents

List of Figures

List of Tables

I

Introduction

Introduction

COMPUTING is an ancient activity. There is evidence of computing, starting from counting and calculation, that dates back thousands of years. Many early civilizations developed their own, unique means for storing and processing numerical information, such as the Quipu of Incas and the Chinese counting rods.[1] Various tools and aids for computing—such as analog astronomical computers and mechanical computing machinery—have existed for millennia.[2] The modern computer is the latest addition to that continuum.

Computing as a discipline—whether one prefers to talk about computer science, informatics, algorithmics, or something else—is a much more recent phenomenon than abstract mathematical concepts or the practice of using mechanical aids for calculation. There is, however, no birth date for computing as a discipline. Looking for such a date, one could point out computational or algorithmic-like concepts from the 1800s, the 1600s, or perhaps 1800 BCE. But it would be untrue to say that George Boole in the 1800s, or Blaise Pascal in the 1600s, or the mathematicians of ancient Babylon around 1800 BCE were early computer scientists or that they worked in the discipline of computing. The discipline simply did not exist at the time. The mathematician and logician George Boole, for instance, was a professor of mathematics at Queen's College in Ireland.

The pieces of a new era in computing emerged in the early 1900s from developments on a broad front, most prominently electrical engineering and mathematical logic. In the 1930s, answers to David Hilbert's decision problem led to formal definitions of an algorithm.[3] Also in the 1930s, Boolean logic was connected with the design of digital circuits, and theoretical definitions of computability were presented. The turn of the 1940s saw the crossing of the Newton–Maxwell gap from mechanical and electromechanical computing—governed by Newton's laws of motion—to fully electronic devices—governed by Maxwell's laws of electromagnetism.[4] In and around the 1940s, visions of modern computing started to form from earlier and newer innovations, and later condensed into a hard core of computing knowledge. The now stable hard core of computing includes ideas like the formalization of computable functions, the idea that instructions can be encoded as data, the idea that

instructions and data reside in the same memory storage, and the separation of memory, the processing unit(s), control unit, and input/output unit(s).

Indeed, the late 1940s and the 1950s must have been an exciting time to work in computing. The emergence of a new and electrifying field with tangible progress and a whole horizon of open problems, attracted young and talented researchers, as well as significant financial resources. The nascent field brought together people from various backgrounds, such as mathematics, physics, and electrical engineering, and this diversity fueled a rapid series of technological and theoretical breakthroughs. Diversity of viewpoints, immediate applications, and a bold pioneer attitude were among the driving forces of the theoretical and technical revolution in computing.

The disciplinary identity of computing professionals began to emerge around the same time as modern computing machinery, but there is no single birth date of computing as an independent discipline. The field gradually developed attributes of an independent discipline. Theoretical foundations—which were first conceived in terms of mathematical logic, and later as theory of computing, saw great advances starting from the 1930s. A number of central technological advances were made in the 1940s. The 1940s also saw the emergence of today's central academic and professional associations for computing. Conferences for computing machinery were around well before the advent of modern computers. The 1950s saw a number of major computing journals and magazines, and the 1960s computing departments, complete curricula for universities, and the first Ph.D. graduates. In the 1970s, a number of major funding agencies made computing a category of its own. By the 1990s the discipline had a rich and unique body of deep theorems and algorithms.[5] There is no longer doubt about computing researchers' ability to independently set and follow a unique research agenda for their trade.

Today computing is a broad, thriving topic for academic research. Computing fields and branches span from information technology and information systems to software engineering, theoretical computer science, and scientific computation. Technological aspects of computing are studied in computer engineering and electrical engineering. There are a vast array of computing branches that are related to each other to various degrees—take, for instance, branches like computer security, human-computer interaction, artificial intelligence, health informatics, and computability theory.

But over the years computing has grown so large that it is sometimes hard to comprehend the shape and size of the discipline as a whole. Hence, it is no wonder that there is no consensus on what computing as a discipline is actually comprised of. Asking ten computing researchers what computing as a discipline is will yield ten different answers. There are a number of concepts—take for instance, the Turing machine and the stored program concept[6]—whose fundamental significance for the field is now rarely disputed. But aside from some central concepts and innovations, there is considerable disagreement with many aspects of computing as a discipline: for instance, opinions diverge

about proper methods, subjects of study, or proper curricula for computing fields.

Starting from the 1950s, computing professionals have presented a dizzying number of arguments concerning the essential features of computing as an academic discipline. After almost 60 years of debates, the field seems to be further away from a consensus than ever before. Moreover, most arguments make a strong case for their cause, and they are often based on an intuitively appealing idea of computing as an activity, body of knowledge, or principles. It is easy to get lost amongst the contradictory, yet compelling, arguments for computing as a discipline made by the field's pioneers. Some argue that computing is primarily a technical field that aims at cost-efficient solutions, and others argue that the field's important contributions are theoretical by nature. Yet another school of thought argues that computing is an empirical science of information processes that are found everywhere—numerous accounts of computing try to combine different aspects of computing into a singular comprehensive package.

So, the field's identity has been fiercely debated throughout its short history. But what are the debates about and what is at stake? Why have the debates not ceased over the 60- or 70-year history of the discipline? Why is it still so difficult to define computing as a discipline? What do people mean when they say that computing is a scientific, mathematical, or engineering discipline? More precisely, what is computing, the academic discipline, about?

This book tries to shed some light on those questions by presenting the reader with arguments and debates about the essence of computing as a discipline. The book puts those debates in a broader disciplinary context, clarifies their background, and analyzes the reasoning behind those debates. By doing so, the book aims at presenting a rich picture of computing as a discipline from the viewpoints of the field's champions. Although this book is written by a computing researcher for other people in computing disciplines, it does not require very deep knowledge of any specific branches of computing.

Viewpoint of This Book

This book emphasizes three viewpoints that are helpful for understanding the debates about computing as a discipline: contextual awareness of those debates, a broad perspective of the field, and tolerance towards different uses of terminology. First, in order to understand debates about computing as a discipline, it is crucial to understand the roots, and the context, of those debates. Many arguments about the nature of computing as a discipline are so deeply rooted in ages-old debates about computing that it is nigh impossible to appreciate them without a contextual frame of reference. For instance, a look behind the still-ongoing debates about the scientific nature of computing reveals a wide range of stimuli. Many computing pioneers were originally natural scientists, and academic and public prestige played an important role in university politics. Attracting students, staff, and funding were affected by

the field's image, and there were issues with the burgeoning field's intellectual integrity and progress. Those aspects, and many others like them, shaped the course of discussions about computing's scientific nature.

Contextual understanding is also important because many famous quotes get a different twist in the context where they were first presented. Take, for instance, the oft-cited remarks of Edsger Dijkstra—a computing pioneer and visionary, and a master of metaphors and catchy one-liners. One popular quotation of Dijkstra is his comparison of calling the discipline "computer science" with calling surgery "knife science."[7] Dijkstra's remark was part of a decades-long debate between theoretical, scientific, and engineering approaches to computing. Those debates were fueled by the software crisis, which was manifest in overbudget, poor-quality, unmanageable, and overdue software projects. The crisis with the work force greatly affected academic computing, too. There was a lot to be unhappy about, and a lot of the blame was put on the sloppy practices of software producers. Dijkstra—a recognized practitioner himself—belonged at that time to an influential but at the time already diminishing group of theoretical purists regarding programming, he had his own vision for how computing should develop as a discipline, and his view was that technology is contingent, while formal, abstract theoretical knowledge is of lasting value.

Second, in addition to contextual understanding, it is important to appreciate the breadth of the field. The diversity of the discipline and its dizzying variety of applications have been some of the main driving forces of the field's development. During the past sixty years, computing researchers have brought together a wide variety of scientific disciplines and methodological standpoints. The resulting discipline has its own distinct body of knowledge, but its body of knowledge also intertwines with knowledge from many other fields and it offers a variety of unique means of modeling and simulating phenomena in other fields. The expansion of computational and algorithmic models—"the idiom of modern science"[8]—to all other fields has been dubbed the "algorithmization" of the sciences.[9] The increased investments in research efforts in computing have been paralleled by the growth of the number of branches of computing, such as scientific computation, artificial intelligence, decision support systems, architectural design, and software engineering. Arguments about the content of the field, its methods, and its aims have sometimes been fierce, and the rapid pace of extension of the field has made it even harder to define computing as an intellectual activity faithfully to what happens in the laboratories, offices, and garages.

Although interdisciplinarity made the rapid development of computing possible in the first place, it also gave rise to very real challenges. For example, there never was an agreement over what kinds of topics should be included in the discipline, and it was very difficult to come up with a common understanding of how research in computing should ideally be done. If a generic set of rules for quality research in all of computing were formulated, those rules should cover research in fields such as software engineering, computational