

SOFTWARE AND MIND

Andrei Sorin

EXTRACT

Chapter 5: *Language as Weapon*
Section *The Slogan “Technology”*

**This extract includes the book’s front matter
and part of chapter 5.**

Copyright © 2013, 2019 Andrei Sorin

**The free digital book and extracts are licensed under the
Creative Commons Attribution-NoDerivatives
International License 4.0.**

This section explains how the abstract term “technology” is being misused in order to make concepts, products, and activities appear more important than they actually are.

The entire book, each chapter separately, and also selected sections, can be viewed and downloaded free at the book’s website.

www.softwareandmind.com

SOFTWARE
AND
MIND

The Mechanistic Myth
and Its Consequences

Andrei Sorin

ANDSOR BOOKS

Copyright © 2013, 2019 Andrei Sorin
Published by Andsor Books, Toronto, Canada (www.andsorbooks.com)
First edition 2013. Revised 2019.

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, without the prior written permission of the publisher. However, excerpts totaling up to 300 words may be used for quotations or similar functions without specific permission.

The free digital book is a complete copy of the print book, and is licensed under the Creative Commons Attribution-NoDerivatives International License 4.0. You may download it and share it, but you may not distribute modified versions.

For disclaimers see pp. vii, xvi.

Designed and typeset by the author with text management software developed by the author and with Adobe FrameMaker 6.0. Printed and bound in the United States of America.

Acknowledgements

Excerpts from the works of Karl Popper: reprinted by permission of the University of Klagenfurt/Karl Popper Library.

Excerpts from *The Origins of Totalitarian Democracy* by J. L. Talmon: published by Secker & Warburg, reprinted by permission of The Random House Group Ltd.

Excerpts from *Nineteen Eighty-Four* by George Orwell: Copyright ©1949 George Orwell, reprinted by permission of Bill Hamilton as the Literary Executor of the Estate of the Late Sonia Brownell Orwell and Secker & Warburg Ltd.; Copyright ©1949 Harcourt, Inc. and renewed 1977 by Sonia Brownell Orwell, reprinted by permission of Houghton Mifflin Harcourt Publishing Company.

Excerpts from *The Collected Essays, Journalism and Letters of George Orwell*: Copyright ©1968 Sonia Brownell Orwell, reprinted by permission of Bill Hamilton as the Literary Executor of the Estate of the Late Sonia Brownell Orwell and Secker & Warburg Ltd.; Copyright ©1968 Sonia Brownell Orwell and renewed 1996 by Mark Hamilton, reprinted by permission of Houghton Mifflin Harcourt Publishing Company.

Excerpts from *Doublespeak* by William Lutz: Copyright ©1989 William Lutz, reprinted by permission of the author in care of the Jean V. Naggar Literary Agency.

Excerpts from *Four Essays on Liberty* by Isaiah Berlin: Copyright ©1969 Isaiah Berlin, reprinted by permission of Curtis Brown Group Ltd., London, on behalf of the Estate of Isaiah Berlin.

Library and Archives Canada Cataloguing in Publication

Sorin, Andrei

Software and mind : the mechanistic myth and its consequences / Andrei Sorin.

Includes index.

ISBN 978-0-9869389-0-0

1. Computers and civilization.
2. Computer software – Social aspects.
3. Computer software – Philosophy. I. Title.

QA76.9.C66S67 2013

303.48'34

C2012-906666-4

Don't you see that the whole aim of Newspeak is to narrow the range of thought?... Has it ever occurred to you ... that by the year 2050, at the very latest, not a single human being will be alive who could understand such a conversation as we are having now?

George Orwell, *Nineteen Eighty-Four*

Disclaimer

This book attacks the mechanistic myth, not persons. Myths, however, manifest themselves through the acts of persons, so it is impossible to discuss the mechanistic myth without also referring to the persons affected by it. Thus, all references to individuals, groups of individuals, corporations, institutions, or other organizations are intended solely as examples of mechanistic beliefs, ideas, claims, or practices. To repeat, they do not constitute an attack on those individuals or organizations, but on the mechanistic myth.

Except where supported with citations, the discussions in this book reflect the author's personal views, and the author does not claim or suggest that anyone else holds these views.

The arguments advanced in this book are founded, ultimately, on the principles of demarcation between science and pseudoscience developed by philosopher Karl Popper (as explained in "Popper's Principles of Demarcation" in chapter 3). In particular, the author maintains that theories which attempt to explain non-mechanistic phenomena mechanistically are pseudoscientific. Consequently, terms like "ignorance," "incompetence," "dishonesty," "fraud," "corruption," "charlatanism," and "irresponsibility," in reference to individuals, groups of individuals, corporations, institutions, or other organizations, are used in a precise, technical sense; namely, to indicate beliefs, ideas, claims, or practices that are mechanistic though applied to non-mechanistic phenomena, and hence pseudoscientific according to Popper's principles of demarcation. In other words, these derogatory terms are used solely in order to contrast our world to a hypothetical, ideal world, where the mechanistic myth and the pseudoscientific notions it engenders would not exist. The meaning of these terms, therefore, must not be confused with their informal meaning in general discourse, nor with their formal meaning in various moral, professional, or legal definitions. Moreover, the use of these terms expresses strictly the personal opinion of the author – an opinion based, as already stated, on the principles of demarcation.

This book aims to expose the corruptive effect of the mechanistic myth. This myth, especially as manifested through our software-related pursuits, is the greatest danger we are facing today. Thus, no criticism can be too strong. However, since we are all affected by it, a criticism of the myth may cast a negative light on many individuals and organizations who are practising it unwittingly. To them, the author wishes to apologize in advance.

Contents

	Preface	xiii
Introduction	Belief and Software	1
	Modern Myths	2
	The Mechanistic Myth	8
	The Software Myth	26
	Anthropology and Software	42
	Software Magic	42
	Software Power	57
Chapter 1	Mechanism and Mechanistic Delusions	68
	The Mechanistic Philosophy	68
	Reductionism and Atomism	73
	Simple Structures	90
	Complex Structures	96
	Abstraction and Reification	111
	Scientism	125
Chapter 2	The Mind	140
	Mind Mechanism	141
	Models of Mind	145

	Tacit Knowledge	155
	Creativity	170
	Replacing Minds with Software	188
Chapter 3	Pseudoscience	200
	The Problem of Pseudoscience	201
	Popper's Principles of Demarcation	206
	The New Pseudosciences	231
	The Mechanistic Roots	231
	Behaviourism	233
	Structuralism	240
	Universal Grammar	249
	Consequences	271
	Academic Corruption	271
	The Traditional Theories	275
	The Software Theories	284
Chapter 4	Language and Software	296
	The Common Fallacies	297
	The Search for the Perfect Language	304
	Wittgenstein and Software	326
	Software Structures	345
Chapter 5	Language as Weapon	366
	Mechanistic Communication	366
	The Practice of Deceit	369
	The Slogan "Technology"	383
	Orwell's Newspeak	396
Chapter 6	Software as Weapon	406
	A New Form of Domination	407
	The Risks of Software Dependence	407
	The Prevention of Expertise	411
	The Lure of Software Expedients	419
	Software Charlatanism	434
	The Delusion of High Levels	434
	The Delusion of Methodologies	456
	The Spread of Software Mechanism	469
Chapter 7	Software Engineering	478
	Introduction	478
	The Fallacy of Software Engineering	480
	Software Engineering as Pseudoscience	494

Structured Programming	501
The Theory	503
The Promise	515
The Contradictions	523
The First Delusion	536
The Second Delusion	538
The Third Delusion	548
The Fourth Delusion	566
The <i>GOTO</i> Delusion	586
The Legacy	611
Object-Oriented Programming	614
The Quest for Higher Levels	614
The Promise	616
The Theory	622
The Contradictions	626
The First Delusion	637
The Second Delusion	639
The Third Delusion	641
The Fourth Delusion	643
The Fifth Delusion	648
The Final Degradation	655
The Relational Database Model	662
The Promise	663
The Basic File Operations	672
The Lost Integration	687
The Theory	693
The Contradictions	707
The First Delusion	714
The Second Delusion	728
The Third Delusion	769
The Verdict	801
Chapter 8 From Mechanism to Totalitarianism	804
The End of Responsibility	804
Software Irresponsibility	804
Determinism versus Responsibility	809
Totalitarian Democracy	829
The Totalitarian Elites	829
Talmon's Model of Totalitarianism	834
Orwell's Model of Totalitarianism	844
Software Totalitarianism	852
Index	863

Preface

This revised version (currently available only in digital format) incorporates many small changes made in the six years since the book was published. It is also an opportunity to expand on an issue that was mentioned only briefly in the original preface.

Software and Mind is, in effect, several books in one, and its size reflects this. Most chapters could form the basis of individual volumes. Their topics, however, are closely related and cannot be properly explained if separated. They support each other and contribute together to the book's main argument.

For example, the use of simple and complex structures to model mechanistic and non-mechanistic phenomena is explained in chapter 1; Popper's principles of demarcation between science and pseudoscience are explained in chapter 3; and these notions are used together throughout the book to show how the attempts to represent non-mechanistic phenomena mechanistically end up as worthless, pseudoscientific theories. Similarly, the non-mechanistic capabilities of the mind are explained in chapter 2; the non-mechanistic nature of software is explained in chapter 4; and these notions are used in chapter 7 to show that software engineering is a futile attempt to replace human programming expertise with mechanistic theories.

A second reason for the book's size is the detailed analysis of the various topics. This is necessary because most topics are new: they involve either

entirely new concepts, or the interpretation of concepts in ways that contradict the accepted views. Thorough and rigorous arguments are essential if the reader is to appreciate the significance of these concepts. Moreover, the book addresses a broad audience, people with different backgrounds and interests; so a safe assumption is that each reader needs detailed explanations in at least some areas.

There is some deliberate repetitiveness in the book, which adds only a little to its size but may be objectionable to some readers. For each important concept introduced somewhere in the book, there are summaries later, in various discussions where that concept is applied. This helps to make the individual chapters, and even the individual sections, reasonably independent: while the book is intended to be read from the beginning, a reader can select almost any portion and still follow the discussion. In addition, the summaries are tailored for each occasion, and this further explains that concept, by presenting it from different perspectives.



The book's subtitle, *The Mechanistic Myth and Its Consequences*, captures its essence. This phrase is deliberately ambiguous: if read in conjunction with the title, it can be interpreted in two ways. In one interpretation, the mechanistic myth is the universal mechanistic belief of the last three centuries, and the consequences are today's software fallacies. In the second interpretation, the mechanistic myth is specifically today's mechanistic *software* myth, and the consequences are the fallacies *it* engenders. Thus, the first interpretation says that the past delusions have caused the current software delusions; and the second one says that the current software delusions are causing further delusions. Taken together, the two interpretations say that the mechanistic myth, with its current manifestation in the software myth, is fostering a process of continuous intellectual degradation – despite the great advances it made possible.

The book's epigraph, about Newspeak, will become clear when we discuss the similarity of language and software (see, for example, pp. 409–411).

Throughout the book, the software-related arguments are also supported with ideas from other disciplines – from the philosophies of science, of mind, and of language, in particular. These discussions are important, because they show that our software-related problems are similar, ultimately, to problems that have been studied for a long time in other domains. And the fact that the software theorists are ignoring this accumulated knowledge demonstrates their incompetence.

Chapter 7, on software engineering, is not just for programmers. Many parts

(the first three sections, and some of the subsections in each theory) discuss the software fallacies in general, and should be read by everyone. But even the more detailed discussions require no previous programming knowledge. The whole chapter, in fact, is not so much about programming as about the delusions that pervade our programming practices, and their long history. So this chapter can be seen as a special introduction to software and programming; namely, comparing their true nature with the pseudoscientific notions promoted by the software elite. This study can help both programmers and laymen to understand why the incompetence that characterizes this profession is an inevitable consequence of the mechanistic software ideology.

The book is divided into chapters, the chapters into sections, and some sections into subsections. These parts have titles, so I will refer to them here as *titled* parts. Since not all sections have subsections, the lowest-level titled part in a given place may be either a section or a subsection. This part is, usually, further divided into *numbered* parts. The table of contents shows the titled parts. The running heads show the current titled parts: on the right page the lowest-level part, on the left page the higher-level one (or the same as the right page if there is no higher level). Since there are more than two hundred numbered parts, it was impractical to include them in the table of contents. Also, contriving a short title for each one would have been more misleading than informative. Instead, the first sentence or two in a numbered part serve also as a hint of its subject, and hence as title.

Figures are numbered within chapters, but footnotes are numbered within the lowest-level titled parts. The reference in a footnote is shown in full only the first time it is mentioned within such a part. If mentioned more than once, in the subsequent footnotes it is abbreviated. For these abbreviations, then, the full reference can be found by searching the previous footnotes no further back than the beginning of the current titled part.

The statement “*italics added*” in a footnote indicates that the emphasis is only in the quotation. Nothing is stated in the footnote when the italics are present in the original text.

In an Internet reference, only the site’s main page is shown, even when the quoted text is from a secondary page. When undated, the quotations reflect the content of these pages in 2010 or later.

When referring to certain individuals (software theorists, for instance), the term “expert” is often used mockingly. This term, though, is also used in its normal sense, to denote the possession of true expertise. The context makes it clear which sense is meant.

The term “elite” is used to describe a body of companies, organizations, and individuals (for example, the software elite). The plural, “elites,” is used when referring to several entities within such a body.

The issues discussed in this book concern all humanity. Thus, terms like “we” and “our society” (used when discussing such topics as programming incompetence, corruption of the elites, and drift toward totalitarianism) do not refer to a particular nation, but to the whole world.

Some discussions in this book may be interpreted as professional advice on programming and software use. While the ideas advanced in these discussions derive from many years of practice and from extensive research, and represent in the author’s view the best way to program and use computers, readers must remember that they assume all responsibility if deciding to follow these ideas. In particular, to apply these ideas they may need the kind of knowledge that, in our mechanistic culture, few programmers and software users possess. Therefore, the author and the publisher disclaim any liability for risks or losses, personal, financial, or other, incurred directly or indirectly in connection with, or as a consequence of, applying the ideas discussed in this book.

The pronouns “he,” “his,” “him,” and “himself,” when referring to a gender-neutral word, are used in this book in their universal, gender-neutral sense. (Example: “If an individual restricts himself to mechanistic knowledge, his performance cannot advance past the level of a novice.”) This usage, then, aims solely to simplify the language. Since their antecedent is gender-neutral (“everyone,” “person,” “programmer,” “scientist,” “manager,” etc.), the neutral sense of the pronouns is established grammatically, and there is no need for awkward phrases like “he or she.” Such phrases are used in this book only when the neutrality or the universality needs to be emphasized.

It is impossible, in a book discussing many new and perhaps difficult concepts, to anticipate all the problems that readers may face when studying these concepts. So the issues that require further discussion will be addressed online, at www.softwareandmind.com. In addition, I plan to publish there material that could not be included in the book, as well as new ideas that may emerge in the future. Finally, in order to complement the arguments about traditional programming found in the book, I have published, in source form, some of the software I developed over the years. The website, then, must be seen as an extension to the book: any idea, claim, or explanation that must be clarified or enhanced will be discussed there.

The Slogan “Technology”

1

I mentioned in the previous section the use of slogans to deceive and to prevent thought. Slogans are expressions representing high levels of abstraction but used in a way that tempts us to perceive them as *low-level* linguistic entities. To illustrate the power of slogans to shape knowledge, and hence the power that an elite can attain through language, let us analyze what may well be the greatest slogan of all time – the term “technology.”

Everyone agrees that technology has acquired in our culture the kind of prestige and aura formerly held only by such notions as God. It will be interesting, therefore, to see how much of its authority is due in fact to something as simple as linguistic manipulation. We will find that, as in all forms of sloganeering, the purpose of the term “technology” is to raise the level of abstraction of a phrase; specifically, to alter its meaning so that the *high-level* elements of a knowledge structure become *starting* elements. Ultimately, its purpose is to prevent us from judging critically a particular matter by forcing us to think instead of a much broader subject.



“Technology” is an abstract term denoting the body of concepts, means, and methods employed in the pursuit of some practical goals. In its most general sense, it refers to the totality of knowledge and techniques used by a society to satisfy its material needs. In a more restricted sense, it refers to the application of a body of knowledge and methods in a specific domain: information technology, automotive technology, communications technology, mining technology, space technology, metal-processing technology, prosthesis technology, etc. In its narrowest sense, “technology” can refer to a particular set of concepts and procedures *within* a field: digital technology within the field of communications, cold-forging technology within the field of metal processing, etc.

Like all abstract terms, then, “technology” plays an important linguistic role by subsuming a number of ideas. If the meaning of these ideas is understood, the ability of the word to represent high levels of abstraction helps us to think about or discuss complex matters.

The abundance of the term “technology” in contemporary discourse reflects, undoubtedly, the growing number of occasions when we encounter the application of one technology or another. Much of this abundance, however, springs from a phenomenon that is best described as an *inflation* in the use of this term: “technology” is used to describe narrower and narrower areas. Instead of defining a significant range of activities, or an important body of concepts and methods, the term is increasingly applied to specific situations.

Thus, “we have the technology” to do something may simply mean having a certain device; “we are using an older technology” may mean using an older device; “we are upgrading the technology” may mean buying a new device; “a technology company” may mean an electronics company; “developing new technologies” may mean writing some new software; “using a different technology” may mean using different software; “a technology career” may mean an involvement with software; “a technology investment” may mean

purchasing a computer; "its technologies" may mean a company's products, or services, or capabilities; and so on.

Here are some actual examples of this style: "Adobe InDesign includes technology for exporting files directly to Adobe Portable Document Format."¹ "Can be used as the ideal technology for backup or storage."² "The intelligent technology in our electrical calculation software"³ "Canada's banks [and other organizations] expect to have their technology fully prepared."⁴ "Five bottom-line technologies."⁵ "Older engines can benefit from using Slick 50 Plus, fortified by unique technology"⁶ "Many [mid-sized firms] apply technology to virtually every part of their business."⁷ "Our books are a simple way to learn from the experts about the latest technologies from Intel."⁸ "Governments can get into [trouble] when they rush to embrace technology they don't really understand.... [One province] so far has spent \$185-million developing new technologies under the flag of the Health Canada Infoway.... The objective is to create a national network of electronic medical records and other, related technology.... Without the in-house expertise to develop new technology, the provinces have relied upon contractors 'My biggest concern has always been technology investments.' ... 'This is highly sophisticated technology.'"⁹

Just as common is the use of "technology" to describe individual notions or products. The following expressions, taken from the thousands encountered in brochures, periodicals, catalogues, and websites, demonstrate this practice: desktop technology, RISC technology, relational technology, C++ technology, CASE technology, Windows technology, point and click technology, plug and play technology, call center technology, client/server technology, data warehouse technology, object technology, document management technology, cloud technology, ebook technology, text-to-speech technology, web-to-host technology, dual monitor technology, 90 nanometer technology, optical image stabilization technology, perpendicular recording technology, retina display technology, 2.4 GHz technology, V.90 technology, IntelliSense technology, Complete-Compare technology, ColorSmart technology, Q-Fan2 technology, CrossFire technology, WhisperDrive technology, iTips technology, Senseye

¹ Adobe Systems, *Adobe InDesign 2.0 User Guide*, p. 375.

² <http://www.ahinc.com/>.

³ <http://solutionselectricalsoftware.com/>.

⁴ Government of Canada, year 2000 preparedness, adv.

⁵ Article title, *Momentum: The Microsoft Magazine for Midsize Business* (Oct. 2005).

⁶ Slick 50 Plus engine treatment for older engines, package text.

⁷ "Firms See Link Between Innovation and Technology," *Computing Canada* (Oct. 6, 2006), p. 20.

⁸ <http://noggin.intel.com/>.

⁹ "Technology in health care: big trouble when mishandled," <http://www.globeandmail.com/> (Oct. 9, 2009).

imaging technology, AMD64 technology, Data Lifeguard technology, cPVA technology, Flash Scan technology, ClearType technology.

Anyone, thus, can take a device, or a method, or a feature, and confidently call it a technology. A sentence will always appear more authoritative if it includes the word “technology,” and as everyone is trying to take advantage of its mystique, we encounter this word now in almost any context. So we see “technology” in expressions where it is obviously spurious – expressions where we were content previously with such terms as “system,” “feature,” “method,” “technique,” “procedure,” or “process”; or, we see it in expressions where neither “technology” nor any other term is necessary, as the thing being described can stand alone, on the strength of its own meaning.

2

To confirm this inflation, let us analyze the phrase “MMX technology,” coined by Intel Corporation in 1997 for a new feature of its Pentium processor. Intel is best known as the maker of the processors used in IBM-compatible personal computers. And, continually since 1979, Intel has been introducing new versions of these processors, each time adding new features. The feature called MMX (multimedia extension) includes special data types and instructions, and its purpose is to improve the performance of applications that require intensive computations with graphics or sound data. These computations often involve the repeated execution of one simple operation with several simple operands. The new instructions take advantage of this fact and speed up the computations by executing several operations in parallel; for example, they add at the same time four related values to four others.

Now, Intel had introduced many enhancements before MMX; and, if compared with those enhancements, the novelty, complexity, or scope of MMX, or its impact on the application’s performance, can be described as average. So why did Intel decide to call MMX a technology, while the previous enhancements – many of which were broader and more significant – were simply called features, or improvements? The most likely answer is that Intel succumbed to the “technology” inflation.

This can be demonstrated by comparing MMX with another enhancement: the numeric processing feature, which greatly speeds up mathematical operations. This feature had been available since the earliest processors as a separate device, called NPX (numeric processor extension). And, starting with the i486 processor in 1989, the feature became the FPU (floating-point unit), an internal and faster element. But, even though the FPU and the NPX were much more complex than MMX, and much more important, Intel never referred to them

as “FPU technology” or “NPX technology.” More than that, MMX uses the FPU registers, and the MMX instructions can even be seen as nothing but an enhancement of the FPU.

We are witnessing, thus, an absurd situation: while the FPU (with its great impact on *many* types of applications, including multimedia, and with a broader scope and complexity) is merely a *feature*, MMX (intended *mainly* for multimedia applications, and logically just part of the FPU) is a *technology*. The term “technology” – a high level of abstraction, which must describe a whole domain – is applied here to an entity that is, however we look at it, at a *lower* level than a level that is *too low* to be called a technology.

This absurdity reflects the effect of the “technology” inflation over a period of ten years. As a result, some of Intel’s technical manuals started to look quite silly: while dozens of important and impressive features of the Pentium processor were mentioned simply by their names or acronyms, MMX was regularly followed by “technology.” (Example: “The MMX technology intrinsics are based on a new `_m64` data type to represent the specific contents of an MMX technology register.”¹⁰ Twice in one sentence, “MMX” is used adjectivally to modify the noun “technology,” and then the whole phrase, “MMX technology,” is used adjectivally to modify another noun. To comprehend this sentence, we must read it by omitting the word “technology.”)

But Intel did not call MMX a technology just to use this expression in its manuals. Now it could coin the famous slogan “with MMX technology,” which was displayed everywhere the latest Pentium processors were mentioned. And this slogan was taken over by every computer maker that used these processors, and by every dealer that sold the computers, and was repeated ad nauseam in advertising and sales literature.

The phrase “MMX technology” also exemplifies what is the most common method of presenting something – a particular concept, or process, or feature – as a technology: instead of simply allowing an appropriate term to describe that thing, the sloganeers construct an expression out of that term and the word “technology.” Since we perceive “technology” as a whole domain, this usage makes a specific thing appear bigger and more important than it actually is. Thus, the expression “with MMX technology” means exactly the same thing as does “with MMX,” but it tempts us to perceive MMX as a broader, and hence more important, notion.

The inflation is also demonstrated by the fact that, while in the *many years* preceding MMX it is hard to find a *single* use of “technology” with these

¹⁰ Intel Corporation, *IA-32 Intel Architecture Software Developer’s Manual*, vol. 2, *Instruction Set Reference* (2001), p. 3-9. (The “intrinsics” are C language extensions that provide access to the MMX features.)

processors and the related innovations, Intel has resorted to this practice *many times* in the *few years* since. Some examples:¹¹ Hyper-Threading technology, vPro technology, Viiv technology, Centrino mobile technology, Memory Pipeline technology, Extended Memory 64 technology, Flex Memory technology, Matrix Storage technology, Virtualization technology, Quiet System technology, Active Management technology, I/O Acceleration technology, Performance Acceleration technology, Clear Video technology, GMA 900 graphics technology, Zone Rendering technology, LaGrande technology, SpeedStep technology, Trusted Execution technology, QuickData technology.

3

Let us see now how the deception is achieved. Grammatically, the term describing the concept, or process, or feature is demoted to the role of qualifier: it becomes an adjectival element modifying the noun “technology.” Since what is being described is fully defined by the original term, “technology” is always superfluous. But this word has become such a familiar and striking slogan that it is invariably *it* that claims our attention. Thus, from an unnecessary element, this usage turns “technology” into the most important part of the expression.

Logically, the altered phrase deceives us by forcing our thoughts to a higher level of abstraction. Instead of allowing us to create a rich knowledge structure in the mind, starting with low-level elements, the expression shifts the emphasis to a high-level element – “technology.” Instead of thinking of the term describing the particular concept, process, or feature, and all the facts associated with it, we are tempted to use the abstract term “technology” (which suggests a whole domain) as the *starting* element of the new knowledge structure. Being forced to create in the mind an impoverished structure, we are prevented from gaining any real knowledge. The expression appears to describe something important, when in fact it is just a slogan.

It is senseless to use “technology” when referring to a *specific* thing, and yet this usage is now widespread. The term “technology,” when qualified by the name of a thing, defines a body of principles or techniques that is reflected *entirely* in that thing; so it defines a technology that is, essentially, that thing alone. But then, if one thing can be a technology, why not everything else? If one specific concept, process, or feature is a technology, why not *every* concept, process, and feature? We reach the absurd conclusion that there are as many technologies as there are concepts, processes, features, methods, techniques, procedures, systems, and so forth. Clearly, if we agree to call *specific* things

¹¹ Terms used on <http://www.intel.com/> (Dec. 2006).

“technology,” the term cannot also retain its *abstract* sense; that is, a body of concepts, means, and methods that defines a whole domain, and hence subsumes *many* things. We are deceived precisely because we continue to perceive “technology” as a global term, referring to a large body of things, even as we see it applied to only one thing.

Let us analyze some of these expressions. “Java technology”¹² refers presumably to everything that is related to the Java programming language – definitions, principles, methods, and so forth. But simply “Java” or “Java language” would suggest exactly the same thing. There does not exist a body of principles or techniques that are part of the technology of Java, but are not also part of what is encompassed by the programming language Java. The very existence of this language implies the definitions, principles, methods, etc., related to it; in other words, what I have just listed as its technology. The language Java and a technology called Java must be one and the same thing.

But “technology” is used for even narrower areas. For example, “Oracle relational technology”¹³ refers to the particular implementation of relational database principles found in the system called Oracle. The technology of the Oracle relational database system subsumes, presumably, all the principles, methods, software, etc., related to this system. But the phrase “Oracle relational system” describes the same thing, since it implies the principles, methods, software, etc., related to this system. There cannot exist two different domains – the Oracle relational system, and the Oracle relational technology; one is the same as the other.

A printer is said to incorporate “straight paper path technology”¹⁴ – a feature of the paper-feeding mechanism. This technology subsumes, presumably, all the issues related to a straight paper path. But the fact that the printer has a straight paper path already implies all the issues related to a straight paper path. So, when saying that the printer incorporates straight paper path technology, we cannot mean more than what we mean when simply saying that it has a straight paper path. The domain known as straight paper path technology is the same as the domain of the issues related to a straight paper path.

The same argument could be repeated for the other expressions. Thus, desktop technology is the same as desktop computers, 2.4 GHz technology is the same as 2.4 GHz telephones, WhisperDrive technology is the same as the WhisperDrive feature, data warehouse technology is the same as data warehouse software, 90 nanometer technology is the same as the 90 nanometer process, and so on.

¹² For example, Sun Microsystems training course, adv. pamphlet.

¹³ For example, “Oracle object technology is a layer of abstraction built on Oracle relational technology,” *Oracle Database Application Developer’s Guide*, <http://www.download.oracle.com/>.

¹⁴ Brother HL-660 laser printer, package text.



An indication of the trend to use “technology” to denote almost anything is the frequent use of the plural, “technologies.” Logically, it is senseless to use the plural: since “technology” already means an indefinite number of principles, methods, etc., employed in a particular pursuit, the plural can add nothing. And indeed, in the past the plural was used only in the rare situations where several domains of technology had to be mentioned together (as in, “use of capital cost allowance ... to allow companies to write down equipment used in information, energy, and environmental technologies”¹⁵). But now that “technology” is used for small and specific things, we encounter its plural very frequently, as a pompous substitute for “systems,” “methods,” “techniques,” “processes,” “concepts,” or “features.”

Some examples: “The MSDN Library is an essential resource for developers using Microsoft tools, products, and technologies.”¹⁶ “On this page you can browse technologies currently available on Adobe Labs.... You can find technologies that may interest you by reviewing related technologies.”¹⁷ “Small to medium-sized suppliers [will not require] an expensive investment in traditional EDI technologies.”¹⁸ “Discover solutions that leverage the newest cyber-security techniques and technologies.”¹⁹ “HR suite of tips, tactics and technologies to attract, retain and train skilled workers.”²⁰ “A new generation of methods and technologies has arrived.”²¹ “Now includes Service Pack 2 with advanced security technologies.”²² “Businesses can take advantage of Internet technologies without sacrificing performance or security.”²³ “A guide to the technologies frequently used in Web-enabled teaching and learning activities.”²⁴ “An overview of some different computer cooling technologies.”²⁵ “See the latest technologies.”²⁶

As part of the inflation, we note also the large number of companies whose *name* includes “technology,” or “technologies.” There are probably thousands of such companies, with names varying from the simple XYZ Technology Ltd. to wordy ones like Exquisys Software Technology Ltd., Photo Violation Technologies Corp., and Critical Outcome Technologies Inc. In reality, “technology”

¹⁵ “The \$10-billion plan to help manufacturing compete globally,” <http://www.globeandmail.com/> (Feb. 6, 2007). ¹⁶ <http://msdn.microsoft.com/>.

¹⁷ <http://labs.adobe.com/>.

¹⁸ <https://delphi.portal.covisint.com/>.

¹⁹ Infosecurity Canada conference and exhibition (2003), adv. pamphlet.

²⁰ CATA conference (1999), adv. pamphlet.

²¹ Database and Client/Server World conference (1997), adv. pamphlet.

²² Microsoft Windows XP upgrade CD, package text.

²³ “Surviving the Unexpected,” *Computing Canada* (Nov. 3, 2006), p. 10.

²⁴ <http://www.umuc.edu/>.

²⁵ <http://www.windowsnetworking.com/>.

²⁶ Solution City exhibition (2006), adv. pamphlet.

hardly ever serves to identify the type of business. Its purpose is to mislead us, by forcing us to associate a specific product or service with a universal and glamorous concept.

And it is not just in advertising and propaganda that we find this style; more and more *individuals* are now using it, in order to enhance their own discourse. Since calling things “technology” imparts a tone of authority to any statement, people everywhere have learned to take advantage of this inflation. Thus, when mentioning a particular product or concept, if we refer to it as a technology we can more effectively impress our listeners. In addition, we can delude ourselves that what we are saying is more important than it actually is.

Also, while this slogan is found mostly in the area vaguely known as high technology, we increasingly see it everywhere. Some examples: People watching instant replay in a tennis event on television “had access to replay technology.”²⁷ To reduce referee mistakes, soccer officials are discussing “the possibility of using goal-line technology.”²⁸ A type of motor oil uses “SuperSyn technology.”²⁹ A ball pen refill “contains advanced ink technology.”³⁰ A scrub sponge uses “unique antimicrobial Stayfresh technology.”³¹ An air conditioner uses “dripless technology.”³² A fitness device “has air power technology to help you work out.”³³ Some winter tires use “Microbit technology, which incorporates thousands of crushed walnut shells into the tread compound.”³⁴ An adjustable wrench uses “gripping technology far superior to standard wrenches.”³⁵ Some windshield wiper blades use “flex shell technology,” while others use “special water repellent technology.”³⁶ Some vacuum cleaners use “WindTunnel technology,” while others use “Root Cyclone technology.”³⁷ An office paper punch uses “One-Touch technology.”³⁸ A cooking device uses “Vapor technology.”³⁹ A kettle uses “quiet boil technology.”⁴⁰ A clothes dryer uses “a new vacuum technology.”⁴¹

²⁷ “Instant replay makes U.S. Open debut,” <http://www.globeandmail.com/> (July 18, 2006).

²⁸ “Blatter rules out video replay, but FIFA will discuss new goal technology,” <http://www.globeandmail.com/> (June 29, 2010).

²⁹ Mobil synthetic motor oil, package text.

³⁰ Parker ball pen refill, package text.

³¹ 3M Scotch-Brite all-purpose scrub sponge, package text.

³² Noma air conditioner, Canadian Tire brochure.

³³ AirClimber fitness device, <https://www.airclimbertrial.com/>.

³⁴ <http://www.toyotires.ca/>.

³⁵ HK1 adjustable wrench, Canadian Tire brochure.

³⁶ Reflex, Hybrid and WetTec wiper blades, Canadian Tire brochure.

³⁷ <http://hoover.com/>, <http://www.dyson.com/>.

³⁸ Staples high-capacity 3-hole punch, package text.

³⁹ <http://www.360cookware.com/>.

⁴⁰ KE9200S kettle, <http://www.sunbeam.com.au/>.

⁴¹ DryMate clothes dryer, <http://www.yankodesign.com/>.



To summarize, when applied to a particular thing, “technology” adds nothing to the meaning of the words describing that thing. A specific term – “process,” “method,” “system,” “feature,” etc. – would function equally well; or simply the *name* of that thing would suffice to describe it. Thus, when applied to a particular thing, “technology” is strictly a slogan. Its purpose is to deceive us, to make us perceive an ordinary thing as an important notion – important enough to name a whole domain of technology after it.

Calling things “technology” forces our thoughts to a higher level of abstraction: instead of examining the *details* of a given issue, we are restricted to a broad and vague concept – technology. Also, without the lower levels we cannot link that issue to our previous knowledge, so it remains isolated: it does not enhance our minds the way it would if we faced it through personal experience. Finally, because technology in general is a good thing, we are compelled to perceive anything called “technology” positively. In other words, deprived of the normal means of evaluating a new idea, we end up simply accepting it.

Thus, like all slogans, “technology” impoverishes knowledge by restricting us to mechanistic thinking. When we agree to treat a high-level concept like technology as the *starting* element of a knowledge structure, we are committing the fallacy of abstraction; and when we fail to link this knowledge structure with others, we are committing the fallacy of reification. The new knowledge is impoverished because we are left with only a small fraction of the possible combinations of elements. Our minds have the capacity for complex knowledge structures: we *can* start from low levels, and we *can* link structures. So the purpose of slogans is to neutralize this quality, in order to prevent us from developing in our minds all possible alternatives.

Another fact worth noting is how the guardians of the English language are reacting to the spread of “technology” sloganeering. Some dictionaries, in their entry for the word “technology,” have recently added a definition for its incorrect use (i.e., in specific instances), while listing also its traditional definition (i.e., a global term). Now, it is true that dictionaries must reflect the current use of a language, even if incorrect; so, if the use of “technology” to describe specific things is now prevalent, it must indeed be included. But dictionaries are also educational. This is why certain entries have a qualifier like *archaic*, *slang*, or *substandard*. Similarly, then, the use of “technology” to denote specific things ought to be described as *propagandistic*. By leaving the new definition unqualified, the dictionaries legitimize, in effect, the misuse of this word. “Technology” cannot function as both a global and a specific term, so it is absurd to list both definitions without an explanation.

4

In “technology” sloganeering, the phrase we encounter most frequently is “information technology,” or “IT.” This phrase and its acronym are so widespread, in fact, that they have acquired a reputation of their own. They deserve, therefore, a special analysis.

Information technology is the large domain encompassing computers, software, and related systems; so the phrase itself represents a valid application of the term “technology.” What is wrong, rather, is the *way* in which the phrase is used. It ought to be used only when discussing the *whole* domain, which is what “information technology” stands for. Instead, we encounter it in reference to narrow and specific aspects of this domain – individual computers, programs, people, tasks, etc. The absurdity of this practice is masked by the fact that it is the acronym, “IT,” rather than the whole phrase, “information technology,” that is most often used: IT management, IT department, IT consultant, IT professional, IT staff, IT infrastructure, IT budget, IT job, IT training, IT career, IT problem, IT equipment, IT project, IT spending, IT investment, IT planning, IT initiative, etc.

The key term in this domain is, obviously, “information.” So it is this term alone that ought to be used as qualifier: information worker, information project, information equipment, and so forth. The phrase “information technology” is then merely a particular use of the term, needed when we must describe the whole domain. What the propaganda has achieved, thus, is to substitute this global sense for the original qualifier. And as a result, the whole domain of information technology is invoked every time we discuss a computer, a piece of software, a person, a project, or any other detail from this domain. This forces our thoughts to a higher level of abstraction: we may be discussing small and concrete entities, but we are thinking in fact of a large and abstract concept – the whole domain of information technology. So we end up perceiving ordinary things as more important than they actually are.

We use expressions like “IT manager,” “IT department,” and “IT budget,” for instance, only because we saw them repeated a thousand times in the past. To recognize their absurdity, all we have to do is expand the acronym. Thus, while “IT manager” sounds important, “information technology manager” sounds silly: how can a person manage the universal, abstract concept of information technology? Similarly, “IT department” sounds important, but what is an “information technology department”? How can something be a department of an abstract concept? “IT budget,” too, sounds important; but what is an “information technology budget”? How can a company have

a budget for the abstract concept of information technology? The proper description, again, is “information manager,” “information department,” and “information budget.” It is absurd to use the whole domain as qualifier.

The same is true of any other expression: Does an IT project encompass the whole domain of information technology? Does an IT course teach the abstract concept of information technology? Is an IT career a career in a philosophical, abstract subject?

Thus, while appearing to be just an abbreviation, “IT” serves to control minds. As acronyms always do, it raises the level of abstraction of an expression, thereby preventing us from interpreting it correctly. Even the whole phrase, “information technology,” forces our thoughts to a level that is too high – because it invokes the whole domain when discussing, in fact, specific things; but the acronym takes us to an even higher level. Although “information technology” is used incorrectly, we still see the words – so we can reflect on their meaning and recognize the mistake, as we did a moment ago; with “IT,” on the other hand, this is no longer possible.

By eliminating the words, and hence the lower levels, acronyms numb the mind. They stand for certain ideas, but they prevent us from linking these ideas to our previous knowledge. Ideas are high levels of abstraction, and we discover their meaning when we understand the meaning of the words at the lower levels. By eliminating the words, acronyms obstruct this process. They turn whole ideas into simple, starting elements. These elements, moreover, come with a ready-made, predefined meaning, which we must accept.

The meaning we accept for IT is “strategic business advantage,” “critical success factor in a changing economy,” “powerful tool in today’s competitive environment,” etc. But we did not discover this meaning on our own, by combining bits of previous knowledge. We acquired it ready-made, through messages encountered in publications, lectures, and advertising – messages that associated IT with those benefits. Instead of treating it as the top element of a particular knowledge structure, we use the acronym “IT” as a *starting* element in *new* knowledge structures. In reality, the domain of information technology is not a phenomenon *within* the other phenomena that make up our existence; it *interacts* with them. Now, however, we perceive it as a building block of those phenomena. So, if the notion of IT is distorted, we will perceive everything associated with it – IT budget, IT department, IT consultant, IT project, IT investment – as more important than it actually is.



We saw that “information technology” and “IT” are used mostly for propaganda. Logically, they should be used only on the rare occasions when the

whole domain of information technology is discussed; instead, we find them in reference to small and specific things. But we can also demonstrate the propagandistic nature of this practice in a different way: by comparing the phrase and the acronym with their counterparts in other technologies.

Automotive technology is the domain of activities related to the design and manufacture of vehicles. But we rarely see the phrase "automotive technology," simply because we rarely need to refer to the whole domain. And we hardly ever see the acronym, "AT"; after all, if the phrase itself is rarely used, there is no need to abbreviate it. The key term in this domain is "automotive." And indeed, this word alone is used as qualifier when referring to specific aspects of the domain: "automotive company," "automotive worker," "automotive industry," "automotive research," "automotive career," and so on. We don't see expressions like "automotive technology company" or "automotive technology worker"; nor do we see "AT company" or "AT worker." To duplicate the usage current in information technology, we would have to refer to our cars as "AT equipment," to car mechanics as "AT specialists," and to a car purchase as "AT investment."

Let us take a specific example. The label of an AC/DC adapter designed to charge the battery of laptop or notebook computers includes this note: "For use with Information Technology Equipment."⁴² The closest equivalent in the automotive field would be a car battery charger carrying the note, "For use with Automotive Technology Equipment." If we ever came across such a charger in a store, we would find the note (and the capitals) ludicrous. In fact, we would probably fail to understand the note, and we would have to ask the salesperson whether the charger worked with a car battery. The note for the computer adapter is, in reality, just as ludicrous; yet we find it perfectly logical. This shows how successful has the "information technology" propaganda been.

Let us examine another area. Space technology is the domain of activities related to the exploration of outer space. The key term now is "space," and this word alone is used as qualifier: "space program," "space research," "space vehicle," and so on. And, although we do encounter the phrase "space technology" more often than we should (as a result of the general "technology" inflation), it is still used mostly to describe the whole domain. It is hard to find expressions like "space technology budget" or "space technology manager." As for "ST," if used at all, it is as a legitimate abbreviation when discussing the whole domain – not in phrases like "ST program" or "ST research." We don't refer to satellites as "ST equipment," nor to astronauts as "ST professionals."

Lastly, medical technology is the domain of activities involving the application of science and engineering in health-related matters. As we would expect,

⁴² Delta Electronics adapter ADP-30JH B.

“medical technology” is used only for the whole domain, and we hardly ever see the acronym, “MT.” The word “medical” alone is used as qualifier: “medical research,” “medical equipment,” “medical personnel,” and so on. We don’t refer to a particular X-ray machine as “MT equipment,” nor to technicians as “MT workers,” nor to a medical laboratory as an “MT company.”

The same arguments could be repeated for any other field: environmental technology, mining technology, farming technology, maritime technology, etc. Only in information technology, then, is language manipulation so widespread. And the explanation is simple: In the other fields we get more or less what we expect, relative to what we invest in them. In our computer-related activities, on the other hand, the inefficiency is so high that the elites must constantly *fool* us into accepting their ideas. This is especially true of software ideas. If we were to judge the importance of their activities objectively, we would find that less than 10 percent of what the software bureaucrats are doing has any value. In fact, the only evidence we have for the effectiveness of software theories, methodologies, tools, and applications is found in “success stories” and “case studies.” (As we learned in “Popper’s Principles of Demarcation” in chapter 3, the very fact that we are asked to rely on this type of evidence proves that the ideas are pseudoscientific.)

Thus, since the software novelties rarely work as claimed, the use of deception is an important factor in their promotion. And the manipulation of language is part of this deception: by encouraging us to misuse the abstract terms “information technology” and “IT,” the elites prevent us from noticing the details; without details we cannot tell the difference between useful and useless, or between good and bad, so our computer-related activities appear more important and more successful than they actually are.