THE AUTONOMOUS SYSTEM

A Foundational Synthesis of the Sciences of the Mind

Szabolcs Michael de Gyurky
with Mark A. Tarbell

WILEY

The Autonomous System

A Foundational Synthesis of the Sciences of the Mind

Szabolcs Michael de Gyurky

with **Mark A. Tarbell**

WILEY

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

de Gyurky, Szabolcs Michael.
The autonomous system : a foundational synthesis of the sciences of the mind / Szabolcs Michael de Gyurky, Mark A. Tarbell.
pages cm
ISBN 978-1-118-29424-6 (hardback)
1. Artificial intelligence. 2. Thought and thinking. I. de Gyurky, Szabolcs Michael. II. Title.
Q335.T37 2013
006.3–dc23

2013012064

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

The Autonomous System

*To the Voyager Project at the Jet
Propulsion Laboratory in Pasadena, California.*

*To its unforgettable crew members—the engineers,
scientists, and administrative personnel who facilitated
the encounters with the great planets of our solar
system—and our great captain, Dr. Edward C. Stone.*

*Above all, to the two Voyager spacecraft now
entering interstellar space, on their way into infinity.
Be well and safe, and remember us when we are long gone.*

Preface

> It can scarcely be denied that the supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate representation of a single datum of experience.
>
> —Albert Einstein[1]

WHY DESIGN AND BUILD AN AUTONOMOUS SYSTEM?

The subject of this book has long been designated the "holy grail" of computer science by researchers in academia, research institutions, and industry. Progress toward this goal has been pursued laboriously by a great number of distinguished colleagues and acquaintances since 1955, when the term Artificial Intelligence (AI) was coined. This goal of a truly autonomous system, capable of human-like thinking, contemplation, learning, making rational decisions, and acting on them has not been achieved to date. The reasons for the failure to achieve autonomy in a man-made system will become clear in this book; essentially, this failure is due to the need of a long-overdue paradigm shift in computer science.

The fundamental science referred to in the term "computer science" is not the science of computability; it is not the science of programmability; nor is it the science of abstraction, or of modeling, or of analysis, or of data processing.

The fundamental science in "computer science" is the *science of thinking*. And the fundamental product of computer science, *software*, is an abstraction of the human thought system. Without the full and complete understanding of these two fundamental ideas, an autonomous system can never become a reality.

ORIGINATION OF THE IDEA

Certainly the past efforts expended by our colleagues in Artificial Intelligence require appreciation and recognition. Their efforts are, to use a military analogy, as important as a route reconnaissance that has found that the area reconnoitered was not passable. This intelligence is as important to a field commander as finding the correct avenue of approach to the key terrain.

The other important recognition is for the paradigm shift in thought that came about by an open challenge in 1991 to the Jet Propulsion Laboratory's science and engineering divisions by its new director, Dr. Edward Stone. Before he became director, Dr. Stone was chief scientist of the Voyager Project at JPL, and our colleague. Upon his promotion, one of the first requests he made of the employees of our laboratory was to "think outside the box" and come up with a "paradigm shift" for JPL. This serious request inspired the thought process in the minds of several JPL employees. They developed a rational, well-developed concept for the architectural design of an autonomous system.

As the idea evolved under stress into a concept, it became clear that it was an achievable system. The realization that an autonomous system could be achieved came about through the intense thought exercises described by Immanuel Kant, Georg Wilhelm Friedrich Hegel, and Arthur Schopenhauer in their masterworks. Surprisingly, the architecture for the autonomous system is implied in their works in sufficient detail to be inferentially defined and articulated. Thus, the architecture presented in this book is the result of the paradigm shift requested by Dr. Stone. The system's fully functional cognitive attribute is purely software—the *mind* of the autonomous system.

That the mind is mathematically very intense may be intimidating at first glance. However, this will pose no problems that cannot be resolved with the appropriate mathematical tools. Hegel in his *Wissenschaft der Logik* points the way by his discourse on dynamic logic and quantum using integral and differential calculus.[2] The other essential mathematical tools needed for implementing complex operations and functions are readily available in Gradstein and Ryshik's beautifully compiled *Tafeln: Tables of Series, Products, and Integrals*.[3] This book has been a sheer delight to the mathematicians we know.

Hierarchical-sequential binary logic using algebra is a part of the mathematics of the architecture of the Thought system. However, Boolean logic[4] is only a part of the overall mathematics that form the operation of the human thought system; *it alone cannot lead to an autonomous system*. An autonomous system is no simple system; in fact, it is not a single system. A thinking machine capable of making rational decisions using the human mind as the model requires most of the equations listed in Gradstein and Ryshik.

THE NEGATIVE ASPECTS OF AN AUTONOMOUS SYSTEM

We must use caution here because we are dealing with a design object that will become a reality. There arises a serious question of ethics, reflecting whether we should build it or not. Consider: Does the design team require the autonomous system to be honest in its dealings with those who designed, built, and programmed it? Should it be obedient and forthright with the entire design team, or only with the architect or project manager? Should it be obedient to all instructions and commands received therefrom? This issue of forthrightness with the senior engineer or project manager is very important. Deciding who the single point of responsibility is for the design, implementation, fabrication, assembly, and testing of the autonomous system will play a crucial role.

This question is a valid and important issue and must be decided after careful consideration by the design team and by the sponsoring authority long before implementation begins. We, the human designers and builders, must determine if we want the autonomous system to be truthful and not have hidden agendas the way human beings do, fueled by omissions or outright falsehoods.

We must keep in mind also that an autonomous system initially will be, by design, either deterministic or nondeterministic. It will, from the point of initialization and actuation, be programmed in its intelligence to reflect the mental capacity of the finest human minds in academic and professional vocations. It will therefore begin to learn and gain experience with an IQ many times higher than ours without the hindrance of a *tabula rasa* to overcome. Thus, in the functional analysis and design, we are dealing in the first instance with *thinking*. An autonomous system must be able to think and to be self-aware. This too is a bridge between classical philosophy and modern computer science.

The negative aspects posed by a man-made, autonomous system could easily outweigh the benefits it may give humanity; hence, in this incarnation, it is designed to be deterministic, at least initially, and the application envisioned for it is oriented to interplanetary or interstellar exploration. Thus, the writing of this book and the contemplation of all that it implies must contain not only the architecture of the autonomous system, but relevant information bearing on the subject. As will become clear, the material must be presented in a format usable for the follow-up work in the design process and the development of a Class A spacecraft software system. In this instance, it is similar to the document format referred to as the Concept of Operations (CONOPS), which describes what a system must be able to do in support of its mission objectives.

The "holy grail" of computer science is not a "mere" autonomous system; it is a man-made system capable of thinking rationally, capable of learning everything the brightest among us can. It is capable of contemplation, making sound decisions, and acting on those decisions—on its own, without human intervention. The autonomous system is presented here as an achievable design object, and technically within our reach.

THE DIFFICULTY OF THE UNDERTAKING

The actual construction of an autonomous system is not a trivial matter, as the architecture outlined in this book will show. It cannot be built by an individual or even by a small, expert team of computer scientists. It can only be achieved as a large-scale project, the complexity and size of which will exceed the Voyager, Galileo, and Cassini flight projects combined and all the general and direct support systems they require, to use military terms. Such general and direct support systems include navigation, tracking, and sequencing.

Much thought has gone into this concept since the Voyager spacecraft encountered Jupiter in 1979–1980. It is certainly an achievable goal within the framework of the mental energy, work discipline, intellectual depth, creativity, and the engineering and science experience present at JPL during the era of the great spacecraft flight pioneering projects. During those days, only JPL and its superb minds were able to do it. Today, space exploration is still challenging, but is gradually transitioning to become more of a commodity, with numerous ready launch vehicles to choose from, and more on the way.

POINTS OF DEPARTURE

There are a number of ideas that have evolved into solid concepts that serve as the cornerstones for the architecture presented in this book. The first point of departure is the idea that computer software is an abstraction of the human thought system.[5]

The second and immediate point of departure is based on the famous dictum of René Descartes: "*Cogito, ergo sum*"—I think, therefore I am. Immanuel Kant went one better on this theme by stating: "*Cogito, ergo est*"—I think, therefore *it is*. These Latin *dicta* have many important implications to the computer scientist who is concerned with building an autonomous system. As mentioned earlier, any truly autonomous system, whether it be deterministic or nondeterministic, must by definition be able to think and be self-aware. The property of self-awareness means being able to think utilizing the attribute of self and nonself-discrimination. Therefore, *a machine is considered autonomous only when it can think, contemplate, speculate, and make rational decisions on its own without human intervention*. An autonomous machine must be able to think and act on its own, like an intelligent human being—an average human mind will not suffice. Thus, the only functional model we can use for such an autonomous cognitive capability is the *human* mind—the mind, not the brain. It is the mind we must use as the model for the design.

The principal question therefore is: What is the human thought system? No one has seriously attacked this problem since Immanuel Kant in 1792, Georg Wilhelm Friedrich Hegel in 1820, and Arthur Schopenhauer in 1830. Kant began this line of inquiry by conducting what he termed *Gedankenexperimente*—thought experiments—in the classrooms of the university where he taught, the University of Königsberg, known today as Kaliningrad, located in the former Kingdom of Prussia.

The third point of departure concerns the process of selecting the architectural model to be used for the cognitive component of the autonomous system. The only practical model for this is described in the collective works of Kant, Hegel, and Schopenhauer. When reading the works of these philosophers, one must be able to imagine or visualize the model of the mind presented therein, for if an engineer cannot imagine in detail the object to be designed, he or she cannot design it, much less build it. Considering the construction of this model from the viewpoint of modern computer science, it should be clear that the computer hardware equivalent in the human being is the brain. The computer software equivalent is the human mind, for it is the mind—not the physical structure of the brain—that comprises the various operating systems, processes, and suites of applications software, although the physical structures of the human brain are designed to support the functions it executes. So it is with the autonomous system.

THE SCOPE OF THIS BOOK

We have chosen an interplanetary/interstellar spacecraft as the implementation design object of a deterministic autonomous system. The reasons for this are relatively straightforward. The first is the familiarity of the authors with very large spacecraft systems. The second is that a system capable of flying an interstellar mission successfully must be able to do what a human being can do, or better. The third is to isolate a potentially unlimited system from us and our environment.

This book provides sufficient design detail to be presented to a project, and details how the design fits into an overall architecture for implementation. Any greater detail than what we present here would make this book too large to be manageable by a design team. This book, therefore, is the bridge between the Idea and the Concept for the Functional Design (FDD) and Functional Requirements (FRD) documents. More precisely, this is the Concept of Operations for the autonomous system, without which it would be impossible to accomplish.

Szabolcs Michael de Gyurky
Mark A. Tarbell

Introduction

ARCHITECTURAL DESIGN OF THE AUTONOMOUS SYSTEM

This book is neither theoretical nor algorithmic; this is a book of *design*. We restrict ourselves to the preproject design phase of a cognitive software architecture of a true autonomous system. Because of the huge scope of the material, care must be taken not to wander off the ordered path to the goal, which is the implementation of the design object. The design object is the mind of the autonomous system, which is modeled on the human mind, the computer software equivalent. This principal issue of differentiation between the brain (hardware) and the mind (software) is but the first point of departure from the architecture of most artificial intelligence software. The software architecture of the mind—in the case of the autonomous system—includes the operating system software, the application software, the sensory interfaces, and data conversion processing software, compartmentalized into discrete entities known as *systems* in an overarching *constellation*.

THE OBJECTIVE

The first and most important issue of this book is that of presenting the architecture of the cognitive functions of an autonomous system in sufficient detail for an engineering team to begin the requirements and design process required for the building of the design object. This is the goal. The concept presented here therefore is not theoretical. This concept is firmly anchored in both philosophy and applied technology in Immanuel Kant's *Kritik der reinen Vernunft*. Kant's obsession with how human beings think was based on his love of Isaac Newton's *Philosophiae Naturalis Principia Mathematica* and the works of Wolff, Leibniz, Laplace, Shaftesbury, and the other giants of thought of his time, under the guidance of his mentor and professor, Martin Knutzen.

Note that this work is heavily influenced by classical cognitive philosophy. Nonetheless, one must keep in mind what changes in languages and their usage have done to the term "philosophy." The word originally meant "science and knowledge." Take for example the era in which Arthur Schopenhauer worked and wrote. He translated the title of Sir Isaac Newton's masterpiece into 19th-century German as *Naturwissenschaft, Mathematische Prinzipien*, which in today's English translates as "Natural Sciences: The Principles of Mathematics." So when we are discussing *Philosophiae Naturalis*—"the philosophy of nature"—we are discussing natural science.

To design and build the design object as a formal project is certainly achievable. What will be difficult to build is the organization of the design/project team and to enforce the strict discipline required to adhere—to the letter—to a mature software management standard. For our purposes, the software management standard chosen as the most practical is JPL-STD-D-4000,[1] because of the application for which it was designed—space flight projects—and our familiarity with it. It is a mature product and has repeatedly proven its value for controlling the design process and keeping the cost low and economically efficient, and the development time unfailingly on schedule.[2]

With respect to the presentation of this material, there were a number of important approaches that had to be followed in its scope and format. The foremost of these concerns the compromise between the styles used in writing a textbook and the writing of a design document. The former is a book from which to teach a subject; the latter is a technical document conveying engineering design material to be built or fabricated. The skills required for writing the Functional Requirements Document (FRD) or Functional Design Document (FDD) for a large and complex system differ from those used to teach in a classroom. The material in an FRD or FDD is formulated not to teach but to describe in detail what is to be built to individuals who are already accomplished scientists or software engineering professionals. The process of integrating the two styles is critical in computer software. The FRD is the first volume of many in the discipline of a serious software management standard. Especially in software engineering, we are dealing with human language, the understanding of and command of which may vary significantly between individuals, even among native speakers. Without a flawlessly articulated FRD, which is the result of synthesizing huge amounts of material represented by the volumes of important and very relevant reference works, the other design documents will be flawed and useless. Thus, human language in computer science plays a critical role, and the problems stemming from languages are usually manifested in the excessive overall cost of software, schedule slips in delivery, and high failure rates.

THE DISCOVERY PROCESS

It is important that the reader be made familiar with the historical sequence of events as they occurred and led to the definition of the architecture of the autonomous system. These are important events of human perception, insight, and understanding, acquired through reading and listening to design presentations and reviews during mission planning phases and mission operations. The abstract information is synthesized into knowledge through hands-on work and participation in the design and operations processes.

The discovery process forms the logical baseline of thought. Without a thorough understanding of how we think and learn and its importance to modern computer science, the software engineering and science associated with true autonomy is not possible. To begin with, it must be noted that without the works of the great cognitive philosophers, in particular Kant, Hegel, and Schopenhauer, the implementation of the architecture presented here would not have been possible. The impact of these great philosopher-scientists on the autonomous system will become very clear as one progresses through this book. The works of Immanuel Kant—specifically the three "Critiques": *The Critique of Pure Reason, The Critique of Practical Reason*, and *The Critique of Decision and Judgment*—form an inseparable part of the foundation of the autonomous system. These are followed by Arthur Schopenhauer's *The World as Will and Presentation*, and finally G.F.W. Hegel's *The Science of Logic* and *The Science of Being*.[3]

One must also accept the unenviable task of reading through such daunting works as Kant's *Critique of Pure Reason*, which is an absolute must because it stands by itself in cognitive philosophical achievement. The reason is best described by Professor August Messer, who published a reprint of the *Critique of Pure Reason* circa 1924.[4]

> *Kant's singular place in the development of our philosophical thinking is based in this, that it was he who took on the old schools of opposing philosophical positions: Naturalism, Idealism, Empiricism, and Rationalism in a wide and deeply thought-out Synthesis of reconciliation. Far from all prejudices, and preferences, he invites these opposite and contentious parties as equals before the seat of judgment of his "critical philosophy." He examines without taking side or partisanship the correctness or incorrectness of their positions and arrives at the results that they are not bounded by irreconcilable differences, but that there is a higher Entity which can reconcile all sides.*

The works of the nineteenth-century scientist-philosophers and thinkers are much more serious and relevant than we had anticipated. In the prologue of Arthur Schopenhauer's book, *The World as Will and Presentation*, the requirements he sets are as true today as they were in his day. He says without hesitation or qualification:

> *You cannot read my books without being completely versed in Immanuel Kant's* Critiques. *I'm a Kantian scholar. Furthermore, you cannot read my work,* The World as Will and Presentation, *only once. You must read it thoroughly twice. Once for familiarization, and then to mark it up and study it!*[5]

Schopenhauer was absolutely correct in this: One must read these works numerous times to connect them professionally to modern computer science. This process requires years of discipline, commitment, repetition, and practice. With this effort comes the benefit of expanding one's mind, forcing the brain to support and accommodate the mind's work to be able to view and process the monumental works from the perspective of an engineer, a computer scientist, or an architect, instead of as a student of philosophy. The modern computer scientist will learn the important things hidden in these texts that have been ignored by contemporary scholars. The ignorance of these treasured discoveries is due mainly to the fact that these books have been treated as "mere philosophy" and not as a part of science and engineering. It is the discovery of important vignettes of information that forces one to read every line.

The newly embraced empirical sciences, such as physics, biology, and chemistry, separated themselves a few hundred years ago from the metaphysical sciences, such as speculative logic and ontology.