

# Vedic Computation: Redefining Computer Science in the Light of Maharishi Vedic Science

Gregory Guthrie

Maharishi University of Management  
Fairfield, Iowa

## *Abstract*

*Computer science has a unique and important role in today's world. With its powerful synthesis of new tools and mechanisms through which to express our growing knowledge of natural law, it has extended and redefined many areas of life to create new disciplines, and new possibilities for the application and development of existing fields.*

*This rapid development is based on progress in two fundamental areas; the discipline of computer science, and the increasing knowledge of natural law in the application areas. Together these have combined to give us new methods, and new depths and extents of inquiry and achievement. With this phenomenal growth in the capabilities and applications of computing, technology has expanded into areas critical to the safety and stability of our society. This raises important questions about the soundness of these foundations, and therefore the future development and applications of computer science and computing technologies. Issues of the correctness, limits, and applications of computation and computing systems are now vital to our future, yet cannot be resolved within the current scope of the discipline. This is because these fundamental issues require a clear understanding and precise description of the structures of intelligence expressed in computing systems, and this is not present in computer science, or any other modern scientific discipline.*

*Maharishi Vedic Science, a science of the full range of intelligence including its representations in objective forms and its realizations in human awareness, provides a unique context for understanding and resolving these problems. Because it provides both a theoretical model and practical techniques for investigating the abstract structure of intelligence and principles of Natural Law common to all application areas, it encompasses and integrates the areas of application knowledge and computational representations. Reconsidering the fundamental structure of the discipline of computer science and its applications in computing systems in this context, results in new and more comprehensive understanding of the status and future of the entire field of computing.*

## **Contents**

Introduction .....	192
The Science of Computation .....	193
Computer Science .....	194
Science, Knowledge, and Veda.....	195
Scientific Models .....	196

Maharishi Vedic Science .....	198
A New Paradigm.....	199
Computational Structures .....	200
Abstractions .....	203
Software and Hardware .....	203
Models and Programming .....	205
Computational Universality .....	206
Computation.....	207
Symbolic Computation .....	208
Vedic Computation.....	210
Veda as the Programs of Natural Law .....	213
Cosmic Computing .....	214
Machine Intelligence .....	216
The Hazards of Computation .....	218
The Future of Computing .....	219

### **Introduction**

Computer science is perhaps the quintessential modern science; it merges the most advanced applied technologies of electronics and semiconductors with the abstract areas of theoretical mathematics, to form a new capability which has become the hallmark of modern technology. These two divergent foundations are combined through the principle innovation of computing, the field of software, which has become the ubiquitous tool of modern society. In its applications, computer science provides the tools which are at the forefront of every field of modern endeavor, supporting, extending, and often redefining the nature of our lives.

Computing has risen to be the universal technology of our modern age. It permeates all aspects of modern life, from applied areas of entertainment, economics, commerce, and business, to the more foundational areas of science, research, and education. Computing has extended the methods and capabilities of every area in such a manner as to redefine the possibilities, and often the approaches, of that area. New disciplines have been created and old disciplines re-defined.

The result is an extension of our senses, actions, and intellect into realms which were hitherto inaccessible, both conceptually and physically. Computing systems and technologies have literally transformed life on our planet and become a key ingredient for future progress and technology.

This rapid development is based on developments in two fundamental areas, the discipline of computer science, and the increasing knowledge of Natural Law in the application areas. Together these have combined to give us new tools, new methods, and new depths and extents of inquiry and achievement. Each succeeding new level of expansion yields new knowledge, which then recursively enables even greater capabilities. It is this self enabling loop of development that has created the tremendous growth in all areas, including that of computing itself.

The pace of the development and application of computing technology has progressed at a pace unknown to other fields. Moore's law (Moore, 1964), which successfully predicted the doubling of the capacity of integrated circuits every 18 months from 1964 (an increase of over 200,000) is one striking example of the ongoing and long-term nature of this progress. This exponential growth of computing technology has allowed us

to build computing systems that are the most complex constructions ever created by man. Today's programs may consist of many millions of lines of code, and run on very sophisticated desktop processors. These small systems with millions of transistors in their microprocessor controllers execute instructions at several million instructions per second, exceeding the capacity of multimillion-dollar supercomputers from only a few years ago. Individual computers are connected to the global Internet, with millions of communicating systems and even more millions of individual users. And yet, this emerging computational technology, connectivity, and capability is still in its early years, and still growing rapidly. While there is speculation that we will soon reach the physical limits of growth on the hardware level, architectural advances within and between systems promise to sustain the growth for the foreseeable future.

With this explosive growth of technology, it is vital that we locate a sound and complete theoretical foundation to guide its development and application. This is the liability of current computing technology. The entire field of computer science is still in its infancy, and is attempting to define its theoretical foundations, and create from them appropriate and effective methodologies for its application areas in the computing technologies.

This paper begins with an overview of the nature and components of the discipline of computer science, which presents the fundamental concepts and issues of the discipline. Then these same principles are revisited in the broader context of the Science of Creative Intelligence® and Maharishi Vedic Science<sup>SM</sup>. The impact of this expanded scope on the methods and limitations of traditional computer science is examined, and a vision of the future of computing presented.

### **The Science of Computation**

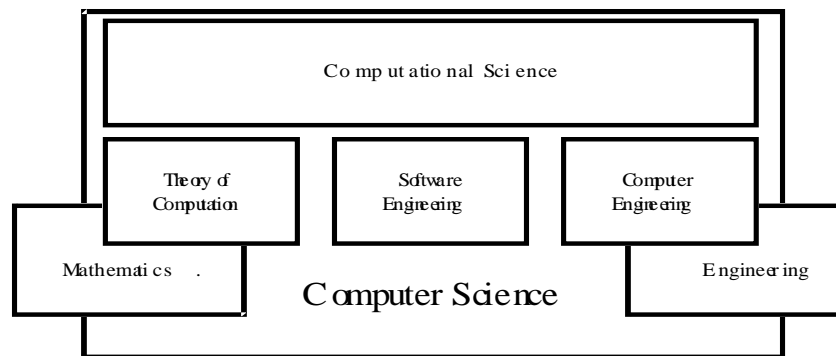
Computer science is a relatively new field, and in its formation has combined aspects of existing disciplines and given rise to new areas of study. The history of computer science is an interesting interplay and, finally, merger of the disparate fields of theoretical mathematics and applied electronics. Theoretical mathematicians were interested in the descriptive power of mathematics, and investigating the limits of such mathematical models. Electronic engineers were simultaneously seeking to discover smaller, faster, and cheaper electronic devices. These explorations converged when it was realized that the discrete states of mathematical logic could be easily represented as the binary modes of an electronic switch. From this basic connection, and the resulting empowerment of both fields, the synergy and success of this combination has created new areas of specialization within the existing disciplines, and completely new areas of study. There is some latitude in the terminology of the related and sub-disciplines of computer science, so an overview is given here.

Computer science studies the structures, dynamics, and methods of representing information, and the limits and extents of these structures to model aspects of natural law. It attempts to discover, describe, and create methods which apply this knowledge to create computational models of the real world. These models can be purely symbolic, similar to the symbolic descriptions of nature created in physics or mathematics, or they can be embodied in physical structures, to create computing machines. Computing

systems combine both through their software and hardware components, and thus also include the interfaces between abstract and concrete computing structures.

Computer science studies the theory, tools, and methods at the basis of all computing systems. From this, several related fields arise, creating technologies that apply this underlying science towards specific practical goals. *Software engineering* investigates pragmatics of constructing large computing models and representative software systems. These large systems are the most complex objects constructed by mankind, and they are created in a new abstract media which provides great freedom of expression and also introduces new challenges. *Computer engineering* considers the creation of physical realizations of these systems, machines which physically reflect and embody the desired computational models. For an effective and practical result, the physical structure of these machines must correlate to the abstract structure of the computation to be run on them, as specified in the software model. The newly delineated discipline of *computational science* looks at combining these technologies and using them to model the most advanced aspects of various application areas.

The distinction between the science and technologies of computing is important, as the nature of the methods, goals, and limits of each is quite different. Figure 1 shows the relationship between computer science and its related and component disciplines. Strengths or weaknesses in the underlying science are naturally reflected in deficiencies in its related technologies, improper or incomplete understanding resulting in errors in application. To strengthen the applications of computing, one must look to its basis in computer science.



**Figure 1.** Computer science combines aspects of the disciplines of mathematics and engineering, and gives rise to several related sub-disciplines which focus on specific aspects of computing systems.

### Computer Science

The goal of computer science is to understand, express, and apply knowledge of natural law in some situation to create computing systems which represent some essential nature of the qualities of the original system. While it could be said that an appreciation and expression of natural law is the goal of all sciences, and indeed arts, there are unique and important aspects of computer science which distinguish it from all others. First, computer science attempts to provide a universal representational model for information. It does not limit its scope to representation of any specific or particular form of information, or description of structure, but strives to provide an ability to describe

understandings of all areas. In this sense computer science is a meta-science; it hopes to discover the common structure(s) of knowledge at the foundation of all expressions of nature. Further, having such a description, it aspires to create machines which can automatically express the implications of this knowledge, to utilize the information in the symbolic structural descriptions to guide the operation of physical computing machines. The symbolic descriptions of system structure are the *software*, and the machines the *hardware*, which together form a computing system.

### Science, Knowledge, and Veda

With computer science as the basis of so many great accomplishments and so much of modern technology, it is important to clearly define and understand the scope and nature of the discipline. If computer science is a science, a science of what? What is the fundamental element studied by the discipline?

The fundamental premises of all science are,

1. there is an underlying order in nature;
2. this orderly structure can be known;
3. this knowledge can be expressed in concrete (symbolic) forms.

Against these scientific premises computer science argues,

4. symbolic forms can be manipulated in systematic ways to represent other structures of knowledge,
5. this symbolic manipulation can itself be represented as a symbolic structure of transformational states,
6. this transformational process can be automated.

We can say that computer science is the science of information structures. It seeks methods to discover structures and patterns in systems, to create models which represent these structures, and to express these models in symbolic forms which can then be automatically interpreted.

For example, scientists may study the phenomena of waves and flow in a fluid system, and determine systems of mathematical equations which describe their structure and behavior. These descriptions are a model of the actual phenomena, which can be encoded into a computer program. This program can then be used to investigate and explore many new aspects of potential system behavior: how will the fluid behave at other temperatures, pressures, or in other situations. The knowledge of the system represented in the model, and then in the program, contains adequate information to represent the essential properties of interest.

In this approach computer science is a meta-science; it attempts to provide a universal modeling and representational system that spans all disciplines, including the sciences, arts, and humanities. With its unique approach of creating informational structures to represent our knowledge of a system, it attempts to be a general science of information.

Because this information represents an understanding within our intelligence, it can be seen as an attempt to universally quantify knowledge, to create a science of intelligence<sup>1</sup>.

Computer science is in this sense an applied science. It does not study and analyze existing elements like physics, chemistry, or mathematics. One does not encounter naturally existing software programs or machines and study them; all software and computers are created systems. Rather, it forms new methods and models to describe existing natural systems. The success of computer science is a measure of how much we have been able to learn about natural law in all other disciplines, and to express in our computing systems.

As with any discipline, computer science is both synthetic and analytic. It is our intellect that connects these two areas; it defines what we know and what we are able to express. Yet, although the intellect is at this fundamental junction point of computer science, it is not currently a part of the discipline.

### Scientific Models

Because of its roots as a modern science, computer science shares a common ideology with the other sciences. To achieve the desired repeatability, regularity, and reliability of a science, modern sciences have chosen to take a purely objective approach, to explicitly exclude subjective aspects which may vary from researcher to researcher and from time to time. Lacking any reliable standards, or absolute basis in the field of subjectivity, this was a logical and reasonable approach. Over the duration of the past few centuries this objective approach has yielded great results, thoroughly investigating and charting many structures, laws, and technologies of creation.

The approach, and thus the scope, of computer science is typical of any modern science, based on this objective approach. Unfortunately, this approach is intrinsically limited in that it excludes the most fundamental source of all science, the subjectivity of the scientist. It is the awareness of the scientist (analyst, programmer) that appreciates the structures of Natural Law at the source of all scientific discoveries and expressions. In computer science, this subjectivity is the basis of all computational expressions, in theory, software, or hardware. Further, the final result of any computation is in some interpretation by the user; all meaning is in the subjective valuation of the objective results of a computation. Thus, the subjective intellect of the (computer) scientist is the source, and goal of all computation.

A different, and more comprehensive approach to the issue of subjectivity and science is taken by Maharishi Science of Creative Intelligence<sup>SM</sup> (SCI), a new discipline which integrates the full nature and development of creative intelligence in Nature and human life.<sup>2</sup> SCI locates the source of all aspects of Natural Law as a *unified field*, an abstract field of Natural Law which underlies all activities in creation, and identifies this field as a field of pure intelligence. Various modes of this field of abstract pure consciousness are expressed as the various qualities and phenomena of creation. Modern physics also

---

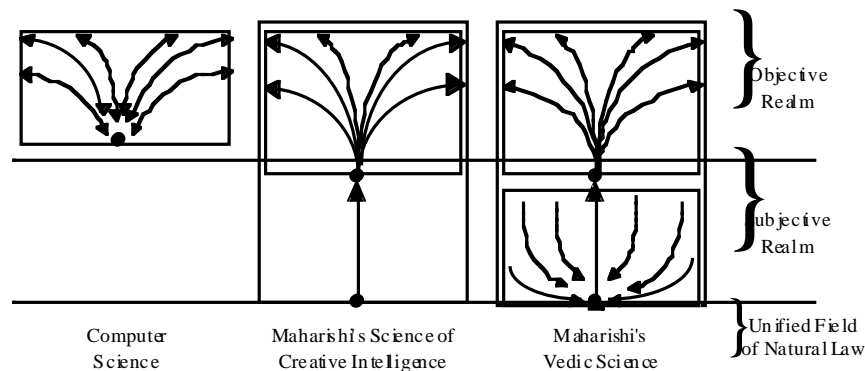
<sup>1</sup> Note that this is not related to the field of cognitive science, which studies the mechanisms of human processes. Computer science focuses directly on relations between systems and their formal symbolic models.

<sup>2</sup> See introduction to this issue for an overview of the Maharishi Science of Creative Intelligence program.

identifies a unified field as the most abstract basis of all of creation, and similarly describes all expressed qualities and structures of the universe as modes of this most fundamental field (Unified Field). Recently, physicists have proposed that this field is indeed a field of consciousness (Hagelin, 1987). In this model of the unified field as a field of pure consciousness, SCI unifies traditional objective science and a new subjective science.

SCI includes both the subjective and objective fields of knowledge, but starts with the field of intelligence as primary, forms a detailed description of the nature, dynamics, and properties of this field, and then describes its expression into various objective forms and phenomena. SCI studies the field of intelligence itself first, independent of any particular expression of it. This general approach, which identifies universal principles and qualities of intelligence, has major implications for the range of knowledge available compared to a purely objective approach. Considering any traditional discipline in this expanded scope drastically alters the context and definitions of the discipline, often providing a more natural context for resolving fundamental questions which may have existed with prior approaches.

A study of computer science in the context of SCI has similar impacts and important results. First, it provides a more reasonable and realistic scope for studying computation. It does not exclude the source of all semantics which are the primary source and ultimate goal of all computational expressions. Second, because it provides universal models of structure, form, and process which describe the functioning of Natural Law in all systems, it directly realizes the goal of computer science to locate universal structures for modeling all areas. Third, it has an important and practical value in that it provides specific technologies for improving the ability of individuals to directly perceive and understand structures of Natural Law.



**Figure 2.** The objective approach of modern science considers only relationships between objective aspects of study and thus limits its scope by excluding the subjectivity of the scientist. Maharishi Science of Creative Intelligence expands this scope to include the full range of subjectivity, including its source in the Unified Field, and Maharishi Vedic Science provides greater detail of the structure and mechanics of the self-referral dynamics of consciousness.

Figure 2 shows the differences between these approaches. computer science examines various disciplines looking for common unifying principles and descriptive qualities. It analyzes common aspects, unifying structural elements and patterns which are common to all areas. The fundamental agent of this analysis is itself purely subjective, the intellect of the analyst. Yet, computer science provides no theory, models, or techniques to address this subjective basis of all computing systems. While the education and training

one receives do help structure specific skills and understandings, there is no mechanism provided to enhance one's ability to recognize the patterns in systems that these techniques are the vehicles to describe. Computer science provides the tools and methods to express understandings, but cannot augment the ability to realize the knowledge to be expressed. This objective *outside-in* approach is intrinsically limited; there are many fundamental areas of investigation completely beyond its reach.

Maharishi Science of Creative Intelligence provides a completely different approach to the search for common structures of natural systems. SCI describes the Unified Field as a field of intelligence and as the primary element responsible for not only the appreciation and analysis of structure in all systems but also as the source of this structure. SCI also describes how this underlying field of pure intelligence expresses itself into the various structures and dynamics in creation. Therefore, SCI provides both a description of the common source of all structures and a unified theory of how this purely abstract dynamics of pure consciousness expresses into the universe of concrete observed phenomena. Further, SCI provides methods for the direct experience of this abstract field of pure intelligence, through the Maharishi Transcendental Meditation<sup>SM</sup> technique. This technique allows the mind to systematically experience more refined levels of thinking, until it is able to directly experience the most settled state of awareness, the field of pure consciousness, the Unified Field<sup>3</sup>. Through this practice the Unified Field is realized as the simplest state of one's own awareness, a self-referral field of pure consciousness (Maharishi Mahesh Yogi, 1995b, p. 280).

Thus, SCI could be termed an *inside-out* approach to knowledge. It starts with general unifying principles, experienced as aspects of the unified field of one's own pure consciousness, and then connects them to their expressions in various disciplinary areas. Maharishi describes SCI as the fulfillment of the search for an interdisciplinary science based on this foundational and integrating role (Boothby, 1996). He explains how this approach fulfills the requirement for complete knowledge:

Complete knowledge should mean total knowledge of the object of inquiry and total knowledge of the subject: total knowledge of both the known and the knower. . . . The Science of Creative Intelligence, by opening one's awareness to the infinite, unbounded value of intelligence, broadens the awareness and makes it permanently unbounded, so that no area of life remains foreign. (MIU, 1981, p. 5)

#### **Maharishi Vedic Science**

Maharishi SCI<sup>SM</sup> locates the field of pure intelligence as the common source of all creation, and describes its qualities and expressions into the manifold properties of creation. Probing deeper into the nature of this field, Maharishi Vedic Science investigates the nature and mechanics of this unmanifest field of pure intelligence. It provides detailed descriptions of the internal structure and dynamics of the purely abstract field of pure consciousness, and locates the most fundamental dynamics within the Unified Field responsible for creation. Maharishi explains:

---

<sup>3</sup> See the introduction to this issue for a more complete description of the Maharishi Transcendental Meditation program.

There was a time when modern physics declared that the entire creation emerged from four fundamental forces, which were thought to be fundamental. As research advanced, these so-called fundamental forces were understood in terms of one holistic Unified Field of all the Laws of Nature.

According to Maharishi Vedic Science, the entire classical world is the expression of the Veda—Natural Law—eternally lively within itself in terms of the self-interacting dynamics of consciousness . . . (Maharishi Mahesh Yogi, 1995a, p. 274)

Maharishi Vedic Science finds that within this field there is a specific structure at the basis of all other structures; one most simple, primordial dynamic of Natural Law which is the source of all other expressions. This most fundamental dynamic is the eternal self-referral awareness of this field of pure consciousness. Being a field of consciousness, by its very nature it continually knows itself, and this process initiates the fundamental mechanics of creation. By experiencing this most fundamental quality of consciousness, one gains direct knowledge of the most universal level of Natural Law. Maharishi describes how this level of the Unified Field, termed the *Veda*, defines all the Laws of Nature, every aspect of structure and dynamics in creation. By knowing Veda one knows the source of every aspect of Natural Law. Maharishi describes this Veda as *pure knowledge*:

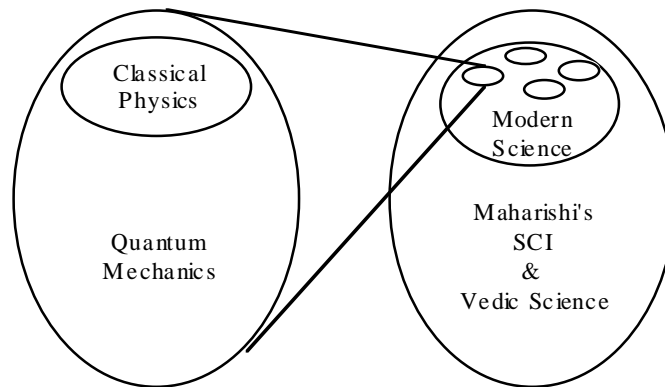
Veda is the structure and function of pure knowledge. It encompasses the whole range of science and technology; it is the theory and practice at the same time; it is the structure of pure knowledge . . . (Maharishi Mahesh Yogi, 1994, p. 5)

Maharishi Vedic Science, by locating this self-referral structure of pure consciousness as the source of all expressed structure in creation, answers the quest for universal structure. In doing so, it introduces a new aspect to computer science, in that this source is located not within any aspect of any application area, but within the Unified Field. This means that not only is the source of all structure not located within the area of study, but it is found at a level of abstract pre-expression; all structure is an expression of a purely abstract dynamics of this unmanifest field of consciousness. Further, because this field is one of consciousness, traditional approaches of science lack any methods for investigation or application of properties of this abstract field. Maharishi Vedic Science provides this missing knowledge; is a complete science of this level of pure abstraction. While other sciences may have great overlap in their domains, and one may deduce properties of one field from the structure of another, that is not the case for this level of inquiry; only through the direct experience of pure consciousness can the nature of the Unified Field be known. Intellectual investigation and inquiry can only infer general properties and descriptions of the field. Only through the knowledge and technologies provided by Maharishi Vedic Science can this level of universal structure be known, and its benefits applied (Maharishi Mahesh Yogi, 1996a, p. 34).

### **A New Paradigm**

The different approaches of traditional computer science and Maharishi Vedic Science predictably locate different results in their quest for universality. The principles of SCI and Maharishi Vedic Science subsume those of traditional computer science, in a manner reminiscent of the relationship of traditional Newtonian physics and quantum mechanics. Newtonian physics describes the physical behaviors and interactions among macroscopic objects and provided an adequate model for the observations of early science. However,

as science progressed, it was unable to describe some observed phenomena, which led to the postulation, the formulation, and finally the verification of an entirely new field. Quantum mechanics described aspects of Natural Law far beyond our direct experience, presenting a model in terms of abstract fields, with many very unexpected and startling results. This new paradigm revolutionized the field of physics, and because it successfully subsumed and extended traditional physics, it is now the accepted theory of modern physics. Similarly, the partial views of traditional objective sciences have their value within their limited domain, but with the advent of SCI and Maharishi Vedic Science, are now seen to exclude fundamental areas required for complete knowledge of the discipline (Figure 3).



**Figure 3.** The scope of modern science continually expands as newer paradigms and levels of knowledge are discovered. Maharishi Science of Creative Intelligence and Maharishi Vedic Science provide a complete framework for all knowledge, by providing knowledge of the Unified Field, the Veda, the home of all the Laws of Nature.

Computer science combines the knowledge of Natural Law in other disciplines with its own methods to model, describe, and, thereby, automate this knowledge in computing systems. Traditional computer science with its objective approach would always be limited in the degree to which it can investigate, understand, and thus represent the processes and dynamics of natural law in a system. With the availability of SCI and Maharishi Vedic Science, a new scope of knowledge of Natural Law is available, to comprehend, model, and express the full range of Natural Law, and thus form a truly universal basis for computing systems. This new paradigm for computer science has impacts in all aspects of the discipline—theory, methods, and applications—which are explored in subsequent sections.

### Computational Structures

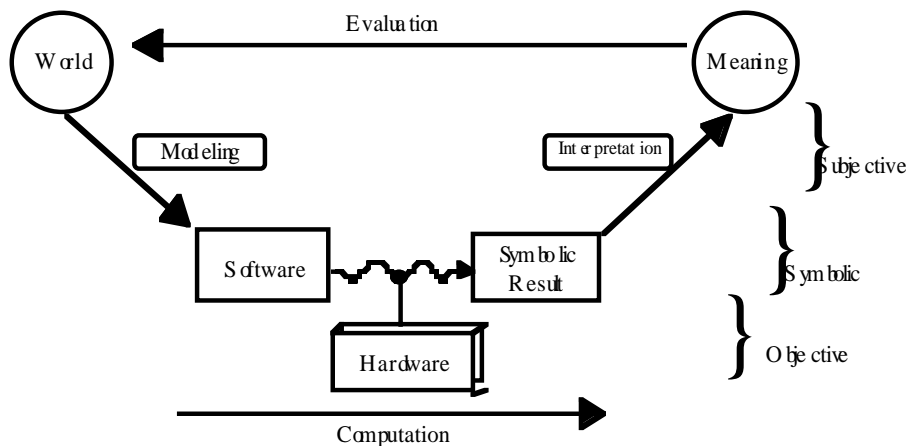
Computer science defines itself in terms of a number of general concepts, including data, information, knowledge, structure, relationships, and a number of more technical concepts such as algorithms, abstractions, languages, and machines. With the overlap of the intended technical definitions and common understandings of these terms, questions arise about the relationship between these properties of computing systems, and related concepts like intelligence and consciousness. Indeed, it is often presumed that because of the vast computational power inherent in modern computers, they must possess some

attribute of intelligence, and thereby somehow a quality of consciousness. To properly address these broad questions, first we need to establish clear definitions of these terms in their technical context as a model of computation.

The relationship of structure, information, and knowledge is fundamental to the definition of computing. We will first define these in the context of the basic aspects of computer science, which are:

1. Modeling, the creation of abstractions which represent the structure of the desired system,
2. Computation, in which one representational form is transformed into another with equivalent semantics,
3. Interpretation, whereby the results of the computation are understood in terms of some new structural aspect of the original system.

Modeling is the process where the structure of the original system is studied, understood, and represented in some symbolic form which describes the essential relevant features of the system. This description is then evaluated, or executed, on a computing machine to determine its meaning. In some cases, this evaluation will be the interpretation of the software as a series of steps of calculation, or of logical inference, or in others as a series of numerical or symbolic relationships. In any case the result is a transformed representation, which represents some alternate state or configuration of the original system. This result is itself symbolic, and is then interpreted by the programmer to determine its meaning (Figure 4).



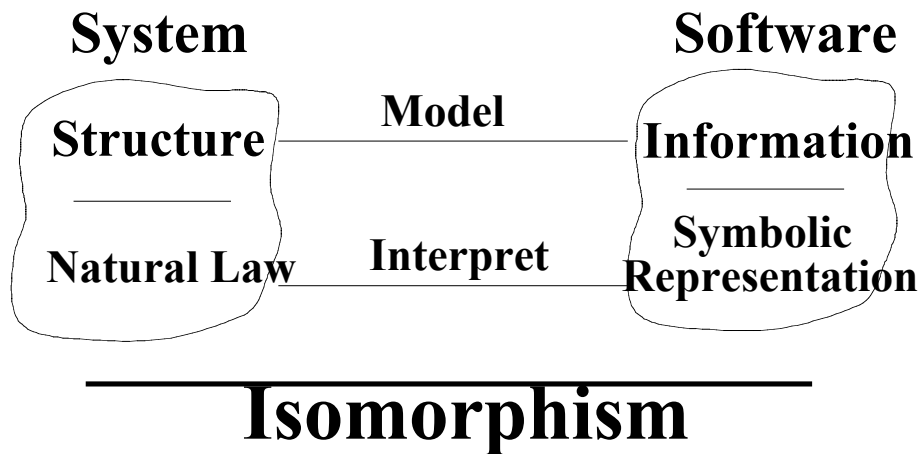
**Figure 4.** Computation is the manipulation of abstract models of real world systems. Symbolic representations link subjective understandings to concrete computing systems. The symbolic results of a computation are then interpreted to provide some meaning to the user.

Note that computation is the manipulation of *abstractions* of a system, the symbolic software representations. This separation of the system and its description distinguishes computers from other mechanical systems and machines, and is the basis of general purpose computing systems.

These representations are described as *information* structures. The correspondence of the forms of the programs and the structures of the original system is the basis of their

representational value. The correspondence of these forms can be described in a variety of representational methods and languages, but there must be a well defined mapping between the two structures.

In this context, we define structure as an abstract set of relationships between entities. It could be the logical structure of relationships in a set of numbers, or the numerical relationships expressed by standard arithmetic, or the more concrete physical structure of a physical system. In an arithmetic domain, there are many relationships between the numbers, and the operations on them, which give the rich structural properties seen in the methods of arithmetic manipulation of equations. For example,  $2+2$  is the same as  $1+3$ , or  $4$ ; they all represent the number four. They are three different concrete symbolic representations of a single abstract concept. They are clearly distinct forms, but they all denote the same meaning<sup>4</sup>. The form of the description is called the *syntax*, and its meaning the *semantics* of the expression. In arithmetic this same distinction is given between the numerals and the numbers which they represent<sup>5</sup>. The fundamental requirement of such a representation is an *isomorphism* between the structure of the domains, between the representative symbolic notation and the actual system. This means that there is a one-to-one correspondence between the elements of the two domains; everything in the source domain can be represented, and every representation has a meaning. Further, an isomorphism guarantees the integrity of the mapping, that if one interprets a representation, it will have the same meaning as the original source object (Figure 5).



<sup>4</sup> More precisely, they represent numeric structures which all evaluate to the same meaning.

<sup>5</sup> The numeral **2** and the word **two** and the symbol **II** all represent the abstract mathematical concept of two. However, since we can generally express ideas in words and symbols the distinction between the syntactic representation and its meaning are sometimes obscured. In some sense this is a benefit, as the notation becomes transparent, and the reader directly invokes the concept.

**Figure 5.** The structure of a system is represented in the informational structures of a software description of the system. This mapping between the two must preserve the knowledge of the system. This complete, reflexive, symmetric mapping is called an isomorphism.

### **Abstractions**

Software is an abstract description of some system, which describes the important properties in a precise and formal manner. An important principle embodied by software is that as one forms an abstraction of a system, the structure of the abstraction captures some general properties of the system. As one forms more and more general abstractions, one should be capturing more and more general properties of the system, and indeed of all systems. This implicit assumption is based on the premise that all systems are instances of a general set of natural laws; a basic tenet of all science.

For example, a standard differential equation represents the structure of an abstract set of qualities, which correspond to the physical behavior and relationships of actual systems. Consider a simple spring-mass system; its defining equation is the same as that of a capacitive circuit. Two vastly different systems, one of physical motion of an object, the other the changes in field potential in a circuit, with exactly the same mathematical description. It is this ability of an abstract structure (the equation) to represent a variety of actual systems that is the basis of the broad application range of computational systems. If this equation is encoded into a simple program, that program can model the behavior of a wide range of systems. One description abstractly represents many systems.

Is there a limit to this model of software abstraction? Is there a complete mapping from any abstract structure, to a symbolic description in language, and/or to actual physical or conceptual system? Are there any abstractions which are inexpressible, or unrealizable? Or conversely, are there concrete systems for which no abstraction exists, and thus no symbolic description? To be able to answer these fundamental questions, we first define our basic model of how such abstractions are formed and expressed into software descriptions.

### **Software and Hardware**

The existence of structure in a system means that there is a stable, orderly pattern to the underlying system dynamics. It is this pattern that is of interest to the designer of a computing system. To describe this structure, the system designer will write a software program. This term contrasts the soft, or abstract, symbolic, flexible nature of this

description to the concrete, physical, fixed structure of the hardware machine on which it will be run.

This separation of computing into distinct aspects of software and hardware is the principle innovation that marked the creation of general purpose computing systems. Previously computation had been done on mechanical analog devices, in which the physical structure of the device modeled some relationship of interest. In the simplest case, the representation of numbers by distances was used to create mechanical adders. Linear concatenation of physical measures was used to find a new measured distance, which represented the numeric summation of the composite values. Multiplication was done in a similar manner, by using logarithmic rulings. Since adding logarithms represents multiplication of the numbers, by ruling the scales properly the multiplicand could be directly read from the device. This encoding of numbers as distances was the basis of slide rules, a principle computational device of engineering prior to electronic computing. Other devices used similar principles, with for example geared wheels measuring angles of rotation, with different ratios of movement representing different numerical relationships. In the early 1940s these mechanical calculating devices reached the height of their mechanical complexity and capability, able to do not only simple addition subtraction, and multiplication, but also complex integration and differentiation problems (Kidwell, 1994).

The concept of a separation of the program to specify and the machine to perform action was first inspired by mechanical looms. It was discovered that since many different patterns could be woven on the same mechanical loom, with a way to encode and automate the control of the loom mechanisms, one could create a very flexible general purpose weaving system<sup>6</sup>. The extrapolation to the ability to weave patterns of logic was quickly made<sup>7</sup>. Further refinements in the implementation and technologies of the machines followed in the realms of modern electronics and computer engineering.

This separation of the software description of a computation, and the hardware machine to perform the steps of the process has allowed the parallel co-evolution of the two disciplines. Software and programming technology could focus on the problems of representations and abstractions and the methods for construction of large software systems. Hardware engineers could focus on the creation of faster, smaller, larger, cheaper systems to execute these programs. The field of Computer Engineering sits at the junction; finding good matches for the abstract structures of the programs and the actual physical structures of the machines.

While this separation has been a major foundation of the entire discipline, it also introduces many issues and challenges to assure proper integration across their interface. Some of the most visible have been the ongoing and prominent debates over two differing approaches to computer system architectures, termed RISC/CISC. One proposes a very simple Reduced instruction set, whereby the complexity of the hardware is minimal, and thus it can be very simple, regular, and fast. The price is that software complexity increases correspondingly. The alternate proposal is to keep software simple,

---

<sup>6</sup> The Jacquard Loom, created in France in the 1830s, used a series of punched cards to direct the weaving shuttles.

<sup>7</sup> Lady Lovelace Ada, generally recognized as the first programmer, said in an 1843 letter to Charles Babbage about his calculating machine, attributed as the first mechanical computer, "We may say most aptly that the Analytical Engine weaves algebraical patterns just as the Jacquard-loom weaves flowers and leaves." (Toole, 1996)

by increasing the complexity of the hardware. The outcome has been a combination of the approaches, but the main result has been to motivate a greater awareness of the flexibility of the hardware/software boundary, and the importance of proper integration between them.

In areas of high performance computing, there is always a concern for how well the physical structure and processes of the computing machine match the required structure of the computation as specified by the program. Further, much research is done on alternate mappings of the program into equivalent computations which would better match the hardware structure and thus perform better.

It is a very interesting and revealing point, that one can flexibly move computation across this hardware/software boundary. Specialized hardware devices are built at this boundary, in which a specific computation is directly mapped into a very regular structure as a system of logical equations, and then this system of equations directly mapped into an array of logical hardware devices. In these devices, the structure of the computation has been directly implemented as a physical structure of the hardware. By using standard techniques that convert logical expressions into very simple regular forms, and by implementing these as large arrays of logical hardware elements, in theory any computation can be converted into a direct hardware representation. However, in practice this is only useful for very simple and small evaluations.

The whole area of hardware software interface has become a major focus of computer science, represented in the areas of computer system architecture and computer engineering. It is an important and interesting area, as it looks at the relationships of abstract and concrete structures, the models, programs, and machines used in today's computing systems. Later we will see that Maharishi Vedic Science significantly expands this integration by considering more refined methods of implementation, which dissolve this hardware/software interface issues, and introduce new models of computation.

### **Models and Programming**

Software is a concrete representation of a model. A model is an abstraction of an actual system; it is a symbolic encoding of the structure of the system. By necessity it describes only some subset of the overall system properties, and this simplification makes the representing program simpler than the actual system. Consider the modeling of a simple falling ball. The situation is very complex; but to simplify it the designer ignores aspects not important for the desired outcome (color, texture, composition, elasticity, temperature), and eliminates others which will not significantly affect the result by making simplifying assumptions (round shape, smooth surface, uniform mass, no turbulence, no rotation). From this simplified problem one can calculate a result, e.g. rate of fall, through a very simple process which is not exact, but which adequately represents the essential aspects of the problem for the desired result.

This discrimination process depends on the analyst to intellectually determine the relevant properties, the appropriate representations, and then form an effective description. There are a range of languages designed to support this description process. These design and programming languages are universal or abstract enough to support a broad range of applications and to allow the description of structures of all types from all areas. These language systems are specifically designed to support the construction of

complex hierarchical descriptions. While languages, methods, tools, and systems support the description of systems into programs, the source of this is the intellect of the programmer/analyst. It is the intellectual understanding of the original system, the goals of the program, and the methods and forms of programming languages which together form the knowledge which is the source of a program. It is this intellectual structure which is expressed in the structures and forms of the program.

Therefore the most fundamental source of all computation is the scientist's understanding, the appreciation and intellectual comprehension of some structure of natural law. This may be the relationships of various measurements of a physical system, or the relationships of various changes in a system with time, or descriptions of interactions of various systems or sub-systems. In every case, it is this subjective appreciation of structure which is the basis of any further expressions. It is this mental model which is subsequently expressed into increasingly concrete and specific forms as a specification, analysis and design models, and finally a software program.

### **Computational Universality**

A fundamental tenet of science is the existence of, and our ability to appreciate, order or structure in all systems. While this is a necessary axiom for modern science, an assumption that is unprovable within its scope, Maharishi Vedic Science describes and provides experiential verification of a specific, precise, well defined structure at the basis of creation. This most fundamental level of mechanics of Natural Law can be directly perceived and appreciated by the most refined level of the intellect. The Maharishi Transcendental Meditation and TM-Sidhi programs are applied aspects of Maharishi Vedic Science which culture this ability through a process of systematic refinement of awareness. This gives a system analyst or programmer a method to develop the ability to gain direct familiarity with these abstract structures of Natural Law, and thereby able to more easily express them in their programs and designs.

It is the programmer's intellect that links the structures of Natural Law in the application to the structures expressed in the program. It is the programmer's appreciation and understanding of the structures of Natural Law, his knowledge of the system that is expressed in the program. When the programmer studies the system, his awareness appreciates various aspects of structure in it and forms internal intellectual models of that structure. This *appreciation* is the process whereby the awareness of the analyst automatically recognizes and focuses on the relevant aspects of a system, and forms an understanding, a mental model of them. Maharishi (1994) describes how awareness actually assumes a structure representative of the object of study;

Knowledge blossoms when the knower's attention (awareness) falls on the object and allows the object to occupy the knower's awareness. This occupation of the knower's awareness by the object structures knowledge of the object in the knower's attention, or awareness. (p. 72)

Here Maharishi describes the basic mechanics of gaining knowledge. By identifying this most fundamental structure of knowledge, and by providing a method to develop the ability to function from the simplest level of awareness, the Unified Field of pure consciousness, Maharishi Vedic Science provides a capability to gain complete and reliable knowledge of any system. This means that one can understand, and model all

systems in a reliable manner. Maharishi (1972) locates these principles in the first expressions of the Veda:

This is beautifully expressed in Rig Veda: "Knowledge is structured in consciousness." This means that if we want knowledge to be invariably true we should have a level of consciousness that is equally invariable. If we want complete knowledge we should have a state of consciousness which is most comprehensive. That expanded, unbounded state of consciousness should be permanently established. On that basis, whatever knowledge is gained will be true and complete. (SCI course, lesson 9)

The principles of SCI describe that the ability of awareness to reflect different levels of structure, and different aspects of a system, depends on the quality of the programmer's consciousness. To have reliable knowledge, one must establish one's awareness on a reliable level of consciousness. Only from the level of the Unified Field can a fully reliable, and universal level of consciousness be gained.

The availability of such a universal and nonchanging level of awareness simultaneously fulfills the need to expand the breadth of contemporary objective sciences, and yet avoid the uncertainties and partial comprehension of less refined states of awareness. Maharishi describes the role of refined subjectivity in forming mathematical models of systems, and the power of the ability to appreciate the refined structures of Natural Law in a system:

This universality of applications can be traced back to the fact that all aspects of Nature and areas of life are governed by the same principles of order and intelligence that have been discovered subjectively by mathematicians by referring back to the principles of intelligence in their own consciousness. Great scientists like Einstein have marveled in the past about this profound relation between the subjective and objective aspects in creation, a relation which has its foundation in the identity of the Unified Field of Natural Law and the field of pure self-referral consciousness displaying the universal principles of intelligence and order. (Maharishi Mahesh Yogi, 1996, p. 305)

By developing more refined awareness, the programmer has greater access to and appreciation of the structures of Natural Law in any system, and the ability to recognize their fundamental common attributes for expression into software. With the ability to develop unbounded awareness, programmers now have the opportunity to create software models of any aspect of Natural Law.

Note that the software model is not the same as the structure in the original system. While the programmer may appreciate the system in a very broad context, the model is intrinsically and intentionally limited to particular properties which relate to the goals of the program being created. A program is a partial expression of knowledge. Yet the scope of knowledge is not bounded; what is the relation of these two fields, what can be known, and what can be expressed into a program, i.e., what can be computed? To address this fundamental issue, we first define computation in terms of traditional computer science, and then consider the impacts of our new paradigm based on Maharishi Vedic Science.

### **Computation**

The roots of computation are in mathematics. Mathematical logic deals with fundamental relations between structures. In logic the structures are the basic ideas of truth, a logical proposition is either *true* or *false*. Operations of implication connect various more complex structures of logical assertion. Using these logical formulae and operations (and, or, not), an entire domain of logical propositions, laws, and results is formed. From logic, numerical domains can then be constructed which provide a richer

expressive system, capturing all of the familiar arithmetic values and operations. From numbers, codes can be used to represent any structure or quality, including language, shape, motion, or color. These coded representations are expressed in the symbols of a language, often mathematical in its nature.

In mathematics, a sequential enumeration of the steps of a proof is called the process of deduction. It forms theorems, statements of truth, which can be proven through a direct derivation from the initial assertions, known truths. This deductive process traverses from one logical theorem to another by applying the axioms, and at each step presents a new expression of the original truth. The axioms are guaranteed to preserve this logical correctness.

Since the statements of logic are all represented symbolically as mathematical formulae, this entire process can be seen as one of symbolic manipulation. Indeed, it was the hope of early mathematicians that such logical systems could automate the entire reasoning process, and all of mathematics could be relegated to a mechanical system. This striking proposal was completely eliminated by Kurt Gödel, in 1934<sup>8</sup> when he showed that there could be no formal systematic, complete, consistent finite description of mathematics (Gödel, 1934). For any consistent logical system expressive enough to include a description of mathematics, there would always be statements in the system that were true, but which could not be proven within the system. Mathematics could never be completely mechanized. Further, the truth of these statements would be available to logicians; we could know their veracity, but it would be beyond the formalized system. Mathematical truth has no finite representations.

Gödel attributed his inspiration for this theorem to the realization that “truth has no finite description” (Rucker, 1987, p. 227). Maharishi SCI presents the same conclusion; the complete nature of Natural Law can only be known from the Unified Field. One can never deduce the full nature of pure intelligence by studying the expressions of intelligence; only through direct experience of the field of pure intelligence can one know its true nature. To realize a universal basis for all computation, we must adopt this new expanded paradigm in terms of Maharishi Vedic Science. Only with that scope can we realize the universal abstractions at the basis of all systems, and gain the ability to directly comprehend and express these structures.

### **Symbolic Computation**

Computation can be defined as the systematic symbolic transformation of a system description into alternate syntactic forms, which maintain some semantic property. Therefore, computation is a purely syntactic process. In the actual computational domain, there are no intrinsic semantics; computation is merely automated manipulation of forms (see Figure 4).

However, the manipulation of syntactic forms has little value until and unless some meaning is attached to them. This meaning comes from the programmer. Any computing process starts with the subjective appreciation of some orderly structure in nature. From this the programmer defines the computational representations and processes, evaluates

---

<sup>8</sup> Gödel's Incompleteness theorem (Gödel, 1934).

these to produce a new representational structure, and ends with an interpretation of this new symbolic structure to attach some meaning. The process and result of a computation has no intrinsic meaning; it is entirely a syntactic result. Any meaning or semantics of the result is attributed by the user of the system.

There are two parts to the software definition of a computation: the program and the data. The program defines the operations, the actual transformations or steps of the computation. The data are the actual values that participate in these operations. They parameterize this process to allow one specification to be used for many applications. This is similar to a general equation in mathematics, and the specific data used to find a particular solution of interest. For example, given the general definition of the value of an investment  $\mathbf{P}$  at a rate of interest  $\mathbf{r}$  after  $\mathbf{y}$  years;  $\mathbf{V} = \mathbf{P}(\mathbf{1} + \mathbf{r})^{\mathbf{y}}$ . To find a specific result, one provides input data for  $(\mathbf{P}, \mathbf{r}, \mathbf{y})$  and solves the equation. This solution process involves the application of a series of rules, each of which simplifies the equation, until the resultant value is produced.

Similarly, a software program specifies the steps which will transform an input data specification to the desired result. The software is an encoding of an *algorithm*, a general description of this solution process. Algorithms are procedurally oriented. They define a series of steps and rules which describe how to compute the desired result. Like a proof or deduction, these steps each transform one state into another. This sequence of states can be viewed as a tree, or more generally a graph in some abstract high dimensional state space. The nodes of the graph are valid states of the computation, and the arcs are valid state transitions. Paths through this space from the initial data specification state to the final result state are all valid methods of computation, and trajectories which traverse these paths are all correct algorithms.

For example, any game can be described as a state space, and the strategy for playing the game as an algorithm. For simple games, a tree can be drawn, with each node representing the choices of alternating players, and leading to the new state of the game based on the choice made. Any path through this decision tree represents a particular set of choices, one playing of the game. Any path leading to a winning configuration (game state) is a winning strategy. Other examples of computational state spaces include the evaluation of arithmetic expressions, in which there may be many different ways to simplify an expression like  $(\mathbf{1} + \mathbf{1}) * (\mathbf{3} - \mathbf{1})$  to reach a final result of  $\mathbf{4}$ . It is the algorithm, in this case the rules of the game, or those of arithmetic, which guarantees that the path traverses only valid states.

It is the great richness of even a small state space and the resulting variety of possible traversals that makes computing so rich in its descriptive power and complex in its descriptions and evaluations. In a digital computer system, states are represented in a very simple manner, using numbers. Consider even a simple single number, represented in a computer as a string of binary digits, or *bits*<sup>9</sup>. One 32-bit number or *word* in the computer's memory, typical in today's standard computers, can represent  $2^{32}$  or approximately four billion unique states. Using four such values, a modest 128 bits of information, represents over 3 trillion trillion trillion states ( $2^{128} \sim 3 \times 10^{38}$ ). Ten words

---

<sup>9</sup> A bit is the smallest quanta of information. It means a binary-digit, which has two states, 1 and 0, or On/Off. Because of its simplicity in both concept and implementation, the bit has become the standard unit of information for both mathematical theory, and for actual computing hardware systems.

can represent more states than every particle in the universe. Today's programs typically use thousands, or millions of words of data in their representations of a system, giving them an essentially infinite representational state space.

Note that this great representational power does not simplify the solution of complex tasks, it just allows their description. One must still evaluate the possible solutions, state space paths leading to proper (winning) states. With a complex space of possible states, this may take a very long time, in some cases so long as to make the problem intractable<sup>10</sup>. For example, in the game of chess it is often estimated that a typical game is about 40 turns long, with each turn allowing about 30 different choices of play by each side. Thus the total number of possible moves in a game is approximately 3080 (10118). If we assume a computer which could evaluate a billion moves per second, it would take 10,101 years to make the first move; more than 1,090 times the age of the current universe<sup>11</sup>.

Clearly, given this vast complexity, and the resulting richness of structure of the state space, determining proper representations and correct algorithms for sequential traversal is a significant practical challenge. All such traversals use only local state at each node to determine the computational path. This is a necessity, as by definition this is the only information that they maintain. While some more sophisticated programs add some meta-state information to optimize the process of finding a solution, this can just be considered state augmentation, and thus still described in this model of locality of information. While the fields of algorithms, software engineering and computational science attempt to deal with the pragmatics of this problem, there is a more fundamental alternative approach which avoids it altogether.

### **Vedic Computation**

While a traditional computation can use only local information to guide its sequential traversal of state space, there are other possibilities. The state space is itself just a structure, and anyone with the knowledge of the relations and rules of this space could visualize the entire space and thereby draw immediate conclusions about all possible computations, or all possible results. Such direct insight into the nature of this abstract structure would allow one to understand properties of the entire range of all possible computations and all possible results that would not be apparent, or perhaps even feasible, by enumeration of individual sequential computations. The difference between the sequential and the simultaneous understanding of all possibilities is enormous. It is rather like the difference between a ground level traversal of a maze and a solution found by an aerial view; one is sequential exploration, the other direct perception of the result.

This ability to directly comprehend and understand such abstract systems of relations is the basis of mathematical intuition, which gives direct insight to the nature of mathematical problems and solutions. Maharishi describes the ultimate value of this

---

<sup>10</sup> A problem that is theoretically possible to solve, but would require resources that are not practical, is termed *intractable* in computer science.

<sup>11</sup> Programs that play chess clearly do not take this brute-force approach. Instead, they use selective evaluation of possible moves, and explore only a small subset of all possible states. Even this is good enough for them to play at the level of a world champion grandmaster chess player.

familiarity with the abstract structure of Natural Law in terms of his Vedic Mathematics. Maharishi Vedic Mathematics describes the absolute, unmanifest, complete structure of Natural Law in terms of mathematical concepts of order, structure, and relation; it is the mathematics of the Veda, the unmanifest structure of pure knowledge. It is the full knowledge of this abstract unmanifest structure and dynamics of Natural Law, appreciated in terms of mathematics, in terms of structure and relationships<sup>12</sup>, “As Veda is structured in consciousness, Vedic Mathematics is the mathematics of consciousness” (Maharishi Mahesh Yogi, 1996, p. 339).

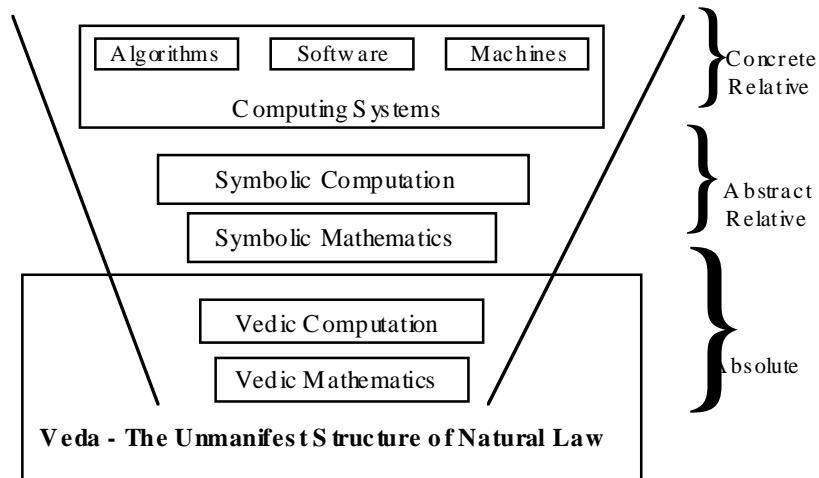
Maharishi explains that because Vedic Mathematics is a direct appreciation of this structure of Natural Law, the direct experience of the holistic nature of Natural Law, it is a simultaneous cognition of all aspects and their composite wholeness; “coexistence of simultaneity and sequence characterize Vedic Mathematics.” (Maharishi Mahesh Yogi, 1995a, p. 339) It is not a sequential logical intellectual exploration of individual aspects, descriptions, and implications of some particular theorem or system, as described by standard mathematics; rather Vedic Mathematics simultaneously comprehends the total value of all relationships in a system. This is similar to the traditional idea of **Aha!** in mathematics<sup>13</sup>, where one is able to comprehend the entire structure and meaning of a problem (Gardner, 1978). However, in Maharishi Vedic Mathematics the scope of such an insight is not limited to a specific problem, but includes the full unified value of Natural Law, as well as all of its parts, and the processes and relationships therein: “This system of simultaneously sustaining all values of relationship is Vedic Mathematics, which we call the mathematics of relationship; it handles all diversifying and unifying values of evolution simultaneously” (Maharishi Mahesh Yogi, 1996, p. 344).

While mathematics emphasizes the understanding of mathematical structures, computing emphasizes their application to guide some process. Computation is envisioned as a process, which produces a transformation of an input to an output. Maharishi has described how through the regular practice of the technologies of his Vedic Science, one gains the ability to operate from the level of the Unified Field. Based on to Maharishi’s definition of Vedic Mathematics, one can describe this ability to not only directly know the structure of Natural Law in a precise and systematic, mathematical manner, but to also compute from this same experiential level of the field of pure consciousness, the most fundamental level of Natural Law, the Veda. Such *Vedic Computation* would have the property that, by gaining the ability to directly appreciate the finest level of structure of Natural Law, one could realize appropriate transformations of Natural Law within the field of pure consciousness, the Veda, to create any result.

---

<sup>12</sup> See Price, this issue.

<sup>13</sup> This term is often used to describe the process of gaining a clear insight into the structure and solution to a problem.



**Figure 6.** By expanding the scope of computer science to include the absolute structures of the unified field, the traditional modes of relative symbolic computing are expanded to include the field of Vedic Computation, which deals with absolute computation.

From Maharishi’s description of this process of computation within the field of pure consciousness, it is characterized by what could be termed *absolute computation* (Figure 6), in that its components are all aspects of the unmanifest, abstract Unified Field. In this situation, there is no separation between concept, model, and machine as in *relative computing* as described in traditional computer science. In this case, the absolute structure of Natural Law is experienced as one’s own simplest state of awareness, and all expressions in creations are realized as self-referral transformations within this field of pure consciousness. These transformations of the qualities of absolute pure consciousness into the various expressed forms and structures of the universe, are simply computations within this field of pure consciousness.

Because this entire process takes place within the Unified Field, from Maharishi’s descriptions it is a computation without steps. It realizes the result by direct enlivenment of the appropriate quality of Natural Law necessary to achieve any desired result. This direct cognition of the desired result spontaneously contains within it all qualities of Natural Law required, without the need to specifically traverse the individual intermediate transformations. Maharishi describes this quality of his Vedic Mathematics:

In Vedic Mathematics all steps are synthesized to promote the result without the need for going through the steps and stages to arrive at the goal. Vedic Mathematics is a spontaneous revelation, it is not a step-by-step derivation. (Maharishi Mahesh Yogi, 1995a, p. 389)

Maharishi Vedic Science describes the complete details of the unmanifest states of Natural Law and the transformations through which this unmanifest field of pure abstract intelligence expresses itself into the concrete values of creation. From the most fundamental self-referral dynamics of pure consciousness, which gives rise to the first structure of creation, further interactions between these various modes of consciousness continue to give rise to further differentiation, and so on through successive layers of further expression, until physical manifest structures arise (Maharishi Mahesh Yogi, 1995a, p. 40). Each successive stage of transformation gives rise to some new aspect of Natural Law; described here as the states of the Unified Field which constitute this process of Vedic computing. Maharishi describes this process in great detail, and

identifies the sequential transformations from pure consciousness into its three values, and then into 40 fundamental modes which express the fundamental qualities of Natural Law<sup>14</sup> (Maharishi Mahesh Yogi, 1995a, p. 25). These 40 qualities are the canonical expressions of structure and dynamics in creations. As any further expression is some further interaction of them, they could be described as the fundamental transformations of Nature, the most fundamental functions from which anything else can be computed.

### Veda as the Programs of Natural Law

Maharishi describes this entire process in terms of the Vedic Literature. He explains that, unlike traditional literature, where the meaning is contained in the printed symbolic forms, the Vedic Literature is an expression of the unmanifest structure of Natural Law in the Veda:

The mathematical descriptions of nature available in the quantum field theories are descriptions by the intellect, which are grasped by the intellect. The intellectual description can at best view reality from the objective angle, in which the knower and known are separate from one another; the intellect and Being<sup>15</sup> are separate. But the self-referral value is not on the level of the intellect, it *is* the reality itself. (Maharishi Mahesh Yogi, 1986b. p. 497)

The primary value of these descriptions of Natural Law in the Vedic Literature is in the states of consciousness which they embody, not in the descriptions themselves. This principle is directly parallel to the concept that the meaning of software is not in its syntactic forms, but only in relation to its connections to the systems it represents. Maharishi notes that for the structures represented by, and described in the Veda, this relationship is unique in that the forms represented are not intellectual descriptions of Natural Law, but are the actual structures of intelligence which are those aspects of Natural Law. This relationship is described in Rk Veda<sup>16</sup> as translated by Maharishi:

*rco akshare parame vyoman  
yasmin deva adhi vishwe nisheduh  
yastanna veda kim richa karishyati  
ya it tad vidus ta ime samasate*  
(Rk Veda 1.164.39)

The verses of the Ved exist in the collapse of fullness in the transcendental field in which reside all the Devas, the impulses of Creative intelligence, the laws of Nature responsible for the whole manifest universe. He whose awareness is not open to this field, what can the verses accomplish for him? Those who know this level of reality are established in evenness, wholeness of life. (Maharishi Mahesh Yogi, 1991, p. 243)

Here Maharishi indicates that the verses of the Veda can be described as the direct expression of the structures and dynamics of Natural Law, the laws which govern Nature's functioning. Maharishi sometimes refers to this as the *programs of Natural Law* (MIU, 1983). However, while a program describes the laws of a particular system, the Veda *is* the Unified Field of all the Laws of Nature. The phrase "he whose awareness is not open to this field . . ." emphasizes the syntactic nature of the actual verses; the

---

<sup>14</sup> Maharishi first described 37 qualities, and now has expanded to model to include 40.

<sup>15</sup> Maharishi uses the term *Being* to denote the existence aspect of the Unified Field.

<sup>16</sup> Rk Veda is the first section of the Veda, and describes the overall structure of the entire Veda.

symbolic descriptions of Veda are not Veda. Intellectual knowledge of the syntactic structures (the texts) of the Vedic Literature does not provide access to the programs of Natural Law. The full values of Natural Law are available only to “Those who know this level of reality . . . ,” the direct self-referral experience of the Veda.

Just as in software, where the syntactic value is only from a semantic connection, this verse describes the syntactic nature of the verses when viewed as texts; the real value is in the semantics of Natural Law they represent. Here, however, Maharishi emphasizes that these *Vedic programs* are not an intellectual description of Natural Law. The syntactic texts describe qualities of Natural Law, but their real value and importance lies in the fact that they are direct expressions of the state of consciousness which *is* the totality of Natural Law.

Maharishi describes this connection in terms of the experience of the sound quality of the fundamental transformations of consciousness within the Unified Field. He describes that the Unified Field of all the Laws of Nature “is available to us as *shruti*—vibrancy of intelligence in the form of sound generated by the self-referral dynamics of consciousness” (Maharishi Mahesh Yogi, 1995a, p. 272). These sounds, literally the reverberations of Natural Law within the field of pure consciousness, are then represented as the texts of Vedic Literature. He explains that “The sound of each of the 39 values of the Vedic Literature is the name of a specific quality of [Natural] Law” (Maharishi Mahesh Yogi, 1995a, p. 58). This connection of the Vedic Literature to the structure of Natural Law is therefore not based on intellectual understanding, but on a refined quality of consciousness from which one directly experiences those fundamental functions of Nature. Without this level of consciousness, any meanings one might attribute to the verses would be partial, and of little value; syntactical forms with no semantic meaning. Maharishi indicates, however, that from this level of consciousness anything can be computed—this is the full value of Vedic computing.

### Cosmic Computing

Maharishi presents a computational model of Nature’s functioning by describing the functioning of Natural Law as a *Cosmic Computer* (Maharishi Mahesh Yogi, 1996a, p. 32). This description emphasizes the precise, systematic, and known systems of sequential transformations of all processes of Natural Law. The structure and descriptions of these patterns of Natural Law are what could be called *Vedic algorithms*<sup>17</sup>, represented in the Vedic programs of the Vedic Literature, by which Nature computes the entire universe. These algorithms are the fundamental patterns of the operations of Natural Law, the fundamental interactions and transformations responsible for all phenomena.

In traditional computing, we emphasize the separateness of the model from the software, and the software from the hardware. This threefold division is the unique, and indeed fundamental distinguishing aspect of current computing systems. Because computer science studies the relations of these three areas, we could call it *relative* computing. In Vedic computing, although we can describe the transformations of Natural

---

<sup>17</sup> Just as Maharishi has presented corresponding values of modern mathematics in his Vedic Mathematics (Maharishi Mahesh Yogi, 1995, p. 388), it would seem that one could locate aspects of Vedic knowledge corresponding to each major area of computer science.

Law in terms of separate aspects of models, programs, and hardware, Maharishi explains that this separateness is in concept only (Maharishi Mahesh Yogi, 1985, p. 24). It is just an intellectual distinction used to describe the one self-referral abstract transformation of consciousness within itself. We can describe the Veda as programs, but unlike traditional computing, here the programs are not distinct from the actual system which they represent; the verses of the Veda *are* the impulses of Natural Law which they describe. Similarly, the knowledge of Natural Law expressed as a model at this level of computation is not distinct from the system it describes; it is the system. Maharishi locates this identity of the knower of Natural Law and the field of Natural Law itself in

	<b>Source of Computing (semantics)</b>	<b>Process of Computing (description)</b>	<b>Hardware (evaluation)</b>
Computer Science	Models	Software	Machines
Vedic Computing	Self (Ṛishi)	Self (Devatā)	Self (Chhandas)

the Vedic expression:

*brahmavit brahmaiva bhavati*  
 “The knower of Brahman is Brahman itself”  
 (Mundaka Upanishad 3.2.9)

Maharishi explains that this verse indicates that the only level of consciousness comprehensive enough to know the totality of Natural Law (*Brahman*), is Cosmic Intelligence itself. At the level of Vedic computing, software and the system which it describes are not disjoint; they are both the unmanifest structure of self-referral pure consciousness. Similarly, the hardware on which this process is evaluated, or executes, is also the finest structure of Natural Law, the level of pure consciousness. Therefore, at this level of computing, the system, software, and hardware are all the same (Table 1). The entire process of computing is just self-referral transformations of pure consciousness within itself. This is what Maharishi describes as the Saṁhitā of Ṛishi, Devatā and Chhandas; the three most fundamental modes of pure consciousness as knower, known, and process of knowing itself (Maharishi Mahesh Yogi, 1994, p. 36). He explains that although there are three aspects or modes of pure consciousness described, these attributes are just intellectual conceptualizations of a self-referral process within the unified state (Samhita) of pure consciousness, experienced as the *Self*<sup>18</sup>.

**Table 1.** Traditional computer science considers three separate aspects of description and representation of a system. Vedic Computing describes the self-referral transformations of the Unified Field within itself, three modes of one unified state of pure consciousness, experienced as one’s Self.

<sup>18</sup> Because this experience is of the full value of one’s own simplest state of consciousness, it is often denoted with a capital letter.

This model of Vedic Computation represents the natural conclusion of the ongoing evolution and refinement of computing systems. From mechanical to electrical, electronic, and now micro-electronic systems, computers have continued to evolve towards the limits of physical systems. Future architectures of optical computing, which compute via the interactions of fields of light, and quantum computers, which utilize quantum mechanical effects to achieve instantaneous coherence, indicate this continuing trend towards using the most refined structures known to implement our computing systems. With the discovery that these electronic and quantum fields are partial aspects of a unified field, which itself is a field of consciousness (Hagelin, 1987), computation within this unified field is the logical extension, and conclusion of all computing systems; it is the ultimate computing system.

### Machine Intelligence

Because of the great sophistication and growing capabilities of computing systems, it is not surprising that the aspiration of computer scientists has been to express the maximal amount of their own intelligence, their own knowledge of natural law, in these systems. Simultaneously, users and clients of these systems, impressed by these capabilities, also begin to wonder about the quality, and limits of intelligence in these systems.

The syntactic definition and view of symbolic computation gives a clear and simple answer to questions about machine intelligence. By considering systems of symbolic computation, the actual source of intelligence, the scientist, is excluded. Programs and machines represent intelligence, but the source of that intelligence is the programmer. Does this mean that computing systems are not intelligent, that they do not somehow embody intelligence, perhaps via their programs, perhaps emergent from their complex interacting components? The simple answer is yes; and this is the basis of arguments against machine or *artificial intelligence*<sup>19</sup>. However, as programs get more and more complex, and represent or embody more and more of our knowledge of natural law, their behavior certainly becomes more closely modeled to ours, and thus it is more and more tempting to label the source of this behavior as intelligent.

Maharishi SCI describes various levels of intelligence, and how different systems embody these different levels. He explains, “A stone exists; it has its own level of consciousness,” and explains that “deep inside a rock all the electrons are moving around the nucleus and there is great activity among the finer particles. So much activity is going on, so there is intelligence, lively and active, in a stone,” and concludes that “Stone or non-stone, all have their levels of consciousness” (Maharishi Mahesh Yogi, 1972, p. 8–9).

While Maharishi locates intelligence everywhere, and thus attributes some intelligence to all structures, he notes that different systems reflect different levels of pure consciousness, the fundamental quality at the basis of all intelligence. There is a limit to the quality of intelligence that can be expressed by different kinds of nervous systems. He

---

<sup>19</sup> This view is called “weak” AI, which proposed that no matter how sophisticated computing systems become, they only simulate intelligence, but are not themselves intelligent. The opposing view “strong” AI, proposes that if something acts in an intelligent manner, then it is intelligent.

indicates that the human nervous system is unique in this regard in that it can realize the full value of intelligence; it can directly experience the field of pure intelligence, the Veda. Maharishi describes that the

Human brain physiology is the hardware of that Cosmic Computer which can create anything through proper programming. Human awareness has the ability to identify completely with the total potential of Natural Law, the unified field, which is transcendental consciousness, the self-referral state of consciousness. The Transcendental Meditation and TM-Sidhi program trains human brain physiology and human awareness to function completely in accord with the total potential of Natural Law and spontaneously exhibit Natural Law in daily life. (Maharishi Mahesh Yogi, 1986a, p. 32)

Given this perspective on intelligence, we can see that the confusion over machine intelligence is based on a lack of clear definitions and understandings of intelligence itself. A computer is perhaps more intelligent than a rock, but less so than a human being. Further, Maharishi Vedic Science describes different levels of consciousness experienced by humans – from typical waking state of consciousness, to the full value of self-referral pure consciousness, traditionally described as *enlightenment*, or *unity consciousness* (Maharishi Mahesh Yogi, 1986a, p. 34). Comparing the highest values of intelligence displayed in a computing system with the limited value of intelligence of waking-state consciousness could easily confuse the actual potential of the two systems.

Machine intelligence is intrinsically limited not only by the theoretical and practical limits of its programs, but also by its physical hardware structures. Maharishi notes that the full value of human intelligence is realized only when one experiences the simplest state of awareness, the Unified Field of pure intelligence (Maharishi Mahesh Yogi, 1995a, p. 176; 1994, p. 260). It is this perfect self-referral state of pure consciousness which is the field of Vedic Computation. Any more expressed physical system can display some degree of self-referral through its programs and hardware structures, but due to limitations of its relative physical structures this process will be bounded by space and time; all self-referral processes will be sequential. This limited self-referral quality of all conventional computing systems limits the level of intelligence they can display.

Maharishi indicates that there is something unique in the physical structure of the human physiology, especially the human brain, which allows it to support this highest level of consciousness. New research indicates that every level of human physiology has a direct correlation in both form and function to the absolute structure of Natural Law in the Veda (Nader, 1995). Maharishi describes the impact of this discovery;

This knowledge has bridged the gap between mind and body, between consciousness and physiology, and between the individual, the environment, and the universe.

This discovery of the one-to-one relationship between the structure and function of the Veda and Vedic Literature and the human physiology has established beyond a doubt that human physiology is the expression of Natural Law. (Maharishi Mahesh Yogi, 1995a, p. 129)

It is this perfection of structure, a physiology that reflects the structure of Veda, that is the basis of this unique and fortunate capability of the human nervous system to realize the full capability of Nature's cosmic computing.

### **The Hazards of Computation**

Computing systems, based on the software programs which structure their behavior, are limited by not only their physical hardware structure and components, but by the degree to which programmers can express the abstract qualities and experiences of

refined levels of consciousness and structures of intelligence into software programs. The formulation of computer science based on partial knowledge of an objective-only approach is more than limiting, it is hazardous; “A little knowledge is a dangerous thing.” Considering this aphorism in the context of the creation of machines that automate their actions based on programs which represent this partial knowledge of Natural Law, raises a concern for the current and potential hazards of current approaches to computing and computing technologies. The use of more and more sophisticated systems that automate partial knowledge is an intrinsically dangerous approach. Regular reports appear about incidents in which computing systems cause loss of service, value, and even life due to incomplete or incorrect programs (Neumann, 1996).

These hazards will only continue to grow with increasing applications of computing technology unless some new direction is taken. Within computer science, ongoing discussions have not yet resolved even the existence of a solution to this problem, let alone practical approaches (Fetzer, 1988). From the context of Maharishi Vedic Science it is clear that these problems can only be overcome by providing individuals with more comprehensive knowledge and experience of the holistic values of Natural Law at the same time that they explore and implement more advanced technologies.

Modern scientific approaches are intrinsically deficient in this regard. Their fundamental approach excludes this—their one-sided focus on objectivity limits their success and automatically makes them partial, and ultimately dangerous in their results. Maharishi Vedic Science and its associated technologies automatically fulfill this requirement; they provide systematic methods which allow one to expand his awareness, and explore and investigate the full range of Natural Law. Only through expanding definitions of computer science to include this domain of Vedic Computation can it achieve its goals without hazardous side effects.

In addition, like all professions, the ongoing technical, professional, and personal demands of computer science and affiliated vocations can have a tiring and stressful effect. The requirement for great focus and technical precision, coupled with the required creativity for modeling and construction of large systems, requires ongoing clarity, creativity, and clear thinking. Without some reliable systematic method to culture these subjective qualities, one is not able to maintain the required refined levels of intellect and thinking required, and the result is a loss of effectiveness and resulting stress. Through the inclusion of Maharishi’s technologies for development of consciousness, the discipline and profession are transformed into activities that not only promote the welfare of the world, but also fulfill the personal and professional aspirations of the practitioners. The same growth of awareness that supports the appreciation of Natural Law in one’s profession, also produces the ability to act in accord with Natural Law in all aspects of one’s life.

### **The Future of Computing**

Computer science provides a powerful set of computing methods, based on traditional approaches to knowledge and computing machines. However, its heritage as an objective modern science restricts it to domains which are intrinsically limited in both theory and practice. The basis of all computation is information structures, which are expressions of our understandings of Natural Law. Maharishi Science of Creative Intelligence and

Maharishi Vedic Science locate the source of these structures in the abstract, unmanifest, Unified Field of Natural Law. By providing a precise and complete science of this abstract field, they expand the previous understanding of computing, and the entire scope of computer science.

It is only through this expanded definition of computation that the goals of computer science to realize a universal model of computation can be realized. Maharishi Vedic Science locates the universal structure of Natural Law within the field of pure consciousness, and indicates that to fully know the field of computer science, one must fully know the field of pure consciousness, one must fully know one's Self.

It is fulfilling that having begun as a science investigating subjective knowledge expressed into objective informational structures, and the variety of physical machines to automate these structures, that through the integration with Maharishi Vedic Science computing now comes full-circle to arrive at an ultimate definition: computation within the field of pure subjectivity, the field of pure consciousness. Einstein once said that "It is my inner conviction that the development of science seeks in the main to satisfy the longing for pure knowledge" (Einstein, 1996, p. 168). He also noted that, "He who finds a thought that enables him to obtain a slightly deeper glimpse into the eternal secrets of nature has been given great grace." It is indeed this deeper knowledge of Natural Law sought by modern science, that is now fortunately provided by Maharishi Vedic Science, in the ability to directly know the most fundamental levels of Nature, the Veda.

Maharishi explains that by knowing this level, the source of Nature's own Cosmic Computing, one lives the full knowledge of all the Laws of Nature. Living this unbounded level of knowledge of Nature completely changes the life of the scientist. It is not just an intellectual appreciation, but the realization that the entire process of computation at the source of creation is within one's own simplest state of unbounded awareness. Living this level of consciousness spontaneously fulfills the highest aspirations of the discipline, theory, practice, results, and also the practitioners of computer science, or any discipline;

The objective approach to modern Mathematics does not allow the subjective value of the individual to be always mathematically precise; it is only the subjective approach of Vedic Mathematics that enables the individual to be always precise and orderly—to spontaneously compute all the laws of Nature necessary to fulfill every desire. (Maharishi Mahesh Yogi, 1996, p. 388)

Science progresses through regular changes in its fundamental world view. Each such advancement answers old questions, solves current problems, and opens new areas of knowledge, technology, and accomplishment. The expansion of the discipline of computer science to include Vedic Computation, based on Maharishi Vedic Science, provides a complete and comprehensive framework which fulfills many fundamental limitations of the discipline, and opens the realm of computer science to include the full scope of Natural Law.

## References

- Boothby, S. (1997). The knowledge and experience of self-referral consciousness and the fulfillment of interdisciplinary study. *Modern Science and Vedic Science*, this issue.
- Einstein, A. (1996). *The quotable Einstein*. Princeton: Princeton University Press.

- Fetzer, J. (1988). Program verification, the very idea. *Communications of the ACM*, Vol. 31, No. 9.
- Gardner, M. (1978). *Aha! Insight*. New York: Freeman.
- Gödel, K. (1934). *On undecidable propositions of formal mathematical systems*. Princeton: Princeton University Press.
- Hagelin, J. (1987). Is consciousness the Unified Field? A field theorist's perspective. *Modern Science and Vedic Science 1*, 1, 28–87.
- Kidwell, C. (1994). *Landmarks in digital computing*. Smithsonian Institution Press.
- Maharishi Mahesh Yogi. (1972). *Science of Creative Intelligence training course I, Handbook*. Maharishi International University Press.
- Maharishi Mahesh Yogi. (1976). *Creating an ideal society*. Rheinweiler, W. Germany: Maharishi European Research University Press.
- Maharishi Mahesh Yogi. (1986a). *Life supported by Natural Law: World assembly on Vedic Science: Lectures by Maharishi Mahesh Yogi*. Washington, DC: Age of Enlightenment Press.
- Maharishi Mahesh Yogi. (1986b). *Thirty years around the world, Dawn of the Age of Enlightenment, Volume One, 1957–1964*. Vlodrop, The Netherlands: Maharishi Vedic University Press.
- Maharishi Mahesh Yogi, (1991). *Maharishi's Master Plan to create Heaven on Earth, Overview document*. Maharishi Vedic University Press.
- Maharishi Mahesh Yogi. (1994). *Vedic knowledge for everyone, /Maharishi Vedic University: Introduction*. Vlodrop, Holland: Maharishi Vedic University Press.
- Maharishi Mahesh Yogi. (1995a). *Maharishi University of Management, Wholeness on the move*. Maharishi Vedic University Press.
- Maharishi Mahesh Yogi. (1995b). *Maharishi's Absolute Theory of Government, Automation in administration*. India: Maharishi Prakashan.
- Maharishi Mahesh Yogi. (1996). *Maharishi's Absolute Theory of Defense*. Maharishi Vedic University Press.
- Maharishi International University. (1981). *Education for enlightenment, An introduction to Maharishi International University*. Maharishi International University Press.
- Maharishi International University. (1983). *Computer Science Unified Field Chart, Maharishi Technology of the Unified Field*. Fairfield, IA: Maharishi International University Press.
- Moore, G. (1964). Intel Corporation.
- Nader, T. (1995). *Human physiology: Expression of Veda and the Vedic literature: modern science and ancient science discover the fabrics of immortality in the human physiology*. Vlodrop, Netherlands: Maharishi Vedic University.
- Neumann, P. (1996). Forum on risks to the public in computers and related systems. ACM Committee on Computers and Public Policy, Association for Computing Machinery, Inc. (<http://www.wais.com/wais-dbs/risks-digest.html>).
- Rucker, R. (1987). *Mind tools*. Houghton Mifflin Company.
- Toole (1996). *Ada, the enchantress of numbers*. Mill Valley, CA: Strawberry Press