CHAPTER 4

VERB FUNCTIONS AND NOUN OBJECTS

CHAPTER SUMMARY

This chapter introduces software design for knowledge machines. Based on Chapter 3, when any complex body of knowledge is accumulated around an array of “knowledge-centric objects,” the processing of knowledge entails the manipulation of such objects, subset of objects, the nature and extent of their inter-relationships. Objects also have a series of attributes that need special processing. Verb functions operate on noun objects to alter their structure and entropy. A series of such microscopic operations executed in the knowledge processor units constitute macrofunctions corresponding to subroutines and library functions in typical computer systems.

The essential software for the execution of any “knowledge program” on a given set of input objects requires a compiler to scan and search for objects and build a framework of existing Web information around such objects. Local library and Internet access provides the supporting information about related objects, their relationships and attributes. The structure of objects is thus extracted in the context that the knowledge problem is being solved. Extended artificial intelligence (AI) techniques for “bodies of knowledge” are introduced in this Chapter.

Pattern recognition (PR) is a recognized branch of AI, whereas compilation is a systematic procedure in identifying symbols and operators. In the language

---

*Computational Framework for Knowledge. By Syed V. Ahamed
Copyright © 2009 John Wiley & Sons, Inc.*
of knowledge (individual or grouped), objects take over the role of symbols and (individual or grouped) verb functions assume the role of operators. The grammar for the documentation of knowledge being more complex that of computer languages offers a large variety of operators and their syntactic variations. However, a thread of conceptual commonality exists. The recognition process (from PR) and identification of symbols (from compiler theory) overlap. Both recognition and identification are intensive disciplines calling for accurate processing and algorithmic refinement.

If parsing, lexical, syntactic, and semantic analyses are placed in an inner shell of the flowchart of a compiler for a computer environment, then statistical, syntactic, and semantic PR forms an outer shell for the flowchart of recognition procedure in a complex visual system. Now, if the two shells are collapsed into one, then a knowledge machine (KM) can be the software recipient to compile and “visualize” the body of knowledge (BOK) under process that passes through the KM.

In the knowledge society, computers and AI play a significant role in preventing nations from falling into this catastrophic cycle of boom and bust that come and go. The brave new machines [1] monitor cyclic forces dominating the such major and minor loops within the cycles. The machines also sense [2] the short-term corrections as well as the long-term trend-setting forces within and without society. Human beings and machines can indeed bring about long sustained positive growth in the wealth of wisdom in nations. Long-term corporate growth of this nature has been demonstrated in developed nations in the past. The forward movement of nations results from the growth of wealth of numerous directionalities of positive wisdom among respective nations. Such directionalities can range from the enhancement of gross national product (GNP) to the development of arts and culture within the nation.

4.1 POSITIVE AND NEGATIVE SOCIAL FORCES

Machines are gadgets and human beings are temperamental. Humans can swing in two extreme directions from positive wisdom (pursuing its absolute form) to negative wisdom (pursuing its opportunistic form). Between these two extremes, lies the materialistic wisdom that provides the means of living for the average-minded knowledge worker. On other hand, machines being driven by code can also function as the precise, incisive, and intelligent (but mindless) agents of human beings. It is humans who pursue any form of wisdom from the absolute to the opportunistic or any form in between. The directionality of wisdom may be initially encoded, but it can also be reversed later, from pristine to corrupt and vice versa.

When the human element that keeps the wisdom machine on its intended course is offered freedom, the negative slide is less strenuous. The positive movement becomes an uphill and more demanding struggle. After all, it is

---

1The demised Bell Systems and AT&T, DuPont, Corning Glass, etc.
animal instinct and human nature to maximize marginal rewards (or the ratio of rewards to effort). In a true sense, the negative slide occurs in any dynamic society naturally, unless it is blocked by the more noble efforts of socially responsible humans. In the current state of the evolving knowledge society, the well-intentioned effort of human beings or organizations (e.g., the United Nations, the Supreme Court, etc.) is greatly supplemented by positively primed wisdom bases.

The sensor and AI software in machines make them aware of the changes in the environment. They become adaptive to counter the effects of changes and yet maximize the rewards for the overall profits and stability of the systems. The design philosophy behind the functioning of intelligent systems starts to parallel the intent behind the actions of rational human beings. Social forces on human beings can be mimicked by special layers of “social” software based on the hierarchy of needs in humans and societies, and the laws of economics and ethical behavior to satisfy such needs in the hierarchy. Being culturally variable, the flavor of social software depends on the national and cultural setting. Much like human behavior, the performance of the intelligent machine will become culturally biased.

When the movements in society are tracked and monitored, the incremental change in the status of primary social “objects” (such as banking, commerce, education, health and welfare, etc.) becomes indicative of the forces responsible for the movement of these primary social objects. For example, change in the banking industry (a primary object) depends on the rate of interest, economic growth, public sentiment, etc. (a set of forces), that move the banking industry. When the cause–effect relationships in the social dimensions are extended, then a set of socioeconometric rules emerges.

It is logical to find areas of overlap between conventional (i.e., national and global) and social econometrics. The economy drives society, and society moderates it. Within this relentless cycle of inter-dependence, human beings develop concepts, wisdom, and ethics (see Figures 2.5a and b). Concept and wisdom machines, and well-intentioned human beings, can help just as much as ill-primed concept and wisdom machines, and ill-intentioned human beings can hurt.

4.1.1 Forces in Society and Operations on Objects

Society provides an environment for all objects, real and virtual, to exist in their own domains. Such domains may criss-cross many times over; however, the path through the domain remains dynamically linked to the social setting and time. In essence, the path through the domains is to resolve the social issue at that particular time. The dynamic linkages provide the intellectual pathways that provide continuity from the conception of a social issue to its resolution. Well-designed maps of the domains and pathways provide robust solutions to social issues. In many instances, the navigation that is designed by computers becomes more optimal than the weak linkages and circuitous pathways of some human beings.
Some of the objects in nature can be highly complex, independent, and volatile. The processing of such objects requires special care to simulate and predict their behavior. The processing time needs to be considerably shorter than the reaction time for such objects. In some cases, predictive processing (to compute the reaction of a predator to the movements of victim, or to determine the reaction of certain bacteria to antibiotics, etc.) may become necessary. When the forces in society are unpredictable, the reaction of the active or irrational objects can become chaotic. Unpredictable human behavior and irrational actions during disasters like volcanic eruptions or wars and famine are some examples.

Some of the other objects show (relative) immunity to environmental changes. The processing of such objects (such as numbers and constants) is relatively benign simple, and traditional computing systems are sufficient to process and predict the behavior. In social environments where noun objects are an admixture of the numerous types of objects, every object needs different sets of attributes and dependencies.

The social environment being more encompassing than the object domains provides the rules for objects to evolve, survive, and thrive. Society and objects soon form symbiotic relationships. Each modifies the other and syntactic rules for social objects to “live.” In providing an acceptable environment, the forces within society provide a framework of forces for the objects and rules for their mechanics.

Dynamic equilibrium conditions for healthy interactions between the objects are continually examined and readjusted to minimize the energy consumed when maintaining the routine functions within society. Human beings and machines learn during the process of adaptation. Movements of objects occur at the macro and micro level. The architecture, software, and firmware of machines to deal with the two levels may be as different as micro and macro economics.

4.1.1.1 Movement of Macroobjects At the macroscopic level, the major national objects (such as the economy, banking, housing, public health, etc.) move precipitously due to major social forces (such as wars, recessions, famines, natural disasters, etc.). The movement of macroobjects can be differentiated (partially or totally) to determine the nature and extent of the social force. For example, the rate of change of displacement yields velocity. The social momentum for objects with estimated social inertia can thus be approximated. The instantaneous rate of change of momentum yields the social force. Given the inherent predisposition and inertia of the object (see Chapter 3), future incremental movements due to specific social changes can be estimated in any given culture or society.

4.1.1.2 Movement of Microobjects At the micro level, objects of an individualistic nature (such as homes, medicines, food supplies, safety items, etc.) move due to minor individual forces (such as job, local weather, family, etc.) within society. The nature of laws that govern the movements’ macro- and microobjects [such as the intensity of verb functions (VFs) and the inertia of noun objects (NOs)] are essentially the same, but the magnitude and impact can
be substantially different. National integration of the movement of microobjects provides a basis for the movement of macroobjects. Micro and macro social economics become inter-dependent as traditional micro- and macroeconomics.

Social processes at, among, and around noun objects become distinctly feasible in social machine tailored to emulate the behavior of objects in most (force, coerce, etc.) social settings. Major objects (i.e., the noun objects; see Section 3.2) motivate minor nouns and forces within society and provide the operators (verb functions; see Section 3.2) that can take place at, among, and around the minor noun objects. Adjectives modify the noun objects and/or their attributes, and adverbs modify the verb functions and/or intensity. Conditional clauses modify the convergence points for noun objects and/or verb functions. Such noun objects become the operands, and such verb functions become operational codes. From machine process considerations, most major verb functions are resolvable into smaller and smaller microscopic functions that the processors in the machines (knowledge, medical, educational, and computational) handle. From machine operand considerations, will most complex noun objects can be fragmented into smaller and smaller microobjects that the processors in the machines will handle. Now we have the same classic computational scenario to implement as knowledge, medical, educational, and computational machines. The combination machines are as feasible as the combined processors that handle arithmetical, logical, and graphical operations.

Forces in society propel knowledge-centric objects through the knowledge trail shown in Figures 2.5a and b. These objects do not have profound identities and firm boundaries. Human perception generally supplements enough details to anchor the ill-defined objects and move them along the left half (i.e., the D, I, and K nodes) of the knowledge trail. The path tends to become ill-defined and tedious in the right half (i.e., the K, C, W, and E nodes), and the new knowledge processors and machines supplement (1) the details from DDS or LoC classifications and their web addresses and (2) the rigor and persistence needed to bridge the right-half nodes. The need for such machines and the new algorithms becomes important because of the numerous type of objects and their behavior in response to verb functions and other objects. These intricacies of behavior of the noun objects and their classification are explored in the following sections.

4.1.2 Assistive, Neutral, and Resistive Objects

Objects have occurred in nature long before civilizations. It has become human nature to manipulate objects to satisfy needs and achieve goals. However, the efforts of humans to alter the objects are not always totally successful. A compromise (see Appendix B, Chapter 5) between the objects that exert the change and the objects that undergo the change, becomes essential.

Thus, the objects that undergo the change can be classified as assistive, neutral, or resistive (see Section 5.2.1) to the change depending on its nature (i.e., the verb function). Objects (especially other animate objects) have an attitude that is (1) willing to change a cooperate, (2) neutral toward the change, or (3) resistive to
it. Generally, assistive and resistive attitudes are displayed by intelligent objects (including living organisms), and neutral objects will follow the laws of algebra, arithmetics or physics to reach an altered state after the change is complete.

For the most part, the willingness to change is based on the unsatiated needs, forces within, and aspirations of intelligent operands. The converse is equally true when the object, that is, the kopr, objects (strongly, mildly, or weakly) resist the change. Thus, a numerical range of changes develops [an emphatic verb function on a strongly willing noun object invoking a ( + , + ) response, to an emphatic verb function on a strongly unwilling noun object invoking a ( + , − ) response]. The numerous combinations and responses are tabulated in Table 5.2 in Chapter 5.

4.2 FRAMEWORK OF KNOWLEDGE

Traditionally, human knowledge and wisdom reside in the mind. Thoughts and contemplations manipulate, manage, and mobilize knowledge and enter the domain of concepts to become axioms of wisdom to master the art of daily life. Like humans, intelligent machines having the capacity to manipulate, manage, and mobilize (MM&M) knowledge-bearing objects can also optimally perform the fine art of science in solving routine scientific and social problems. Furthermore, in emulating a human posture, knowledge machines also command, control, and coordinate (CC&C) knowledge-centric objects much like managers (see Figure 1.7) would do during the managerial activities in a corporation and institutions. However, machines need to be primed with object sets and (humanistic) tools to perform MM&M functions in conjunction with CC&C functions. The intersection space of the six functions appears to be a fertile ground for human or machine creativity. At a macroscopic level of aligning and orienting any scientific and social problem, the machine hardware, software, firmware, and operators need to be defined to suit the scientific and cultural environment for the solution. At the microscopic level of executing individual knowledge instructions, the objects and object sets become operands (k-operands) and tools become the methodology for executing the operation code (kopc) in the knowledge processor unit (KPU) of a KM.

4.2.1 Hyper-Dimensionality of Knowledge

When every classification from the DDS or LoC system is mapped to a direction in the knowledge space, the “body of knowledge” starts to become hyper-dimensional. The processing of information by the human mind has the freedom and capacity to hop freely between directions at will and also control the length of each hop. Emulation of such grace and beauty in the machine domain does not appear feasible at present. However, constrained movements may be possible.
emulated and realized. Much as an airplane cannot totally imitate the flight of every bird, the machine cannot imitate every graceful movement of objects in the human mind. Blending of truth, virtue, and grace in the manipulation of objects in a single pass of the machine appears impossible at this stage. However, the coordinated, rational, and logistic displacement of noun objects and global, goal-oriented, and optimal distribution of associated verb functions may give the machine a complementary role (almost an edge) in solving complex engineering and social problems. After all, a spacecraft flies where no bird has flown before.

The hyper-dimensional space of human thought soon becomes fragmented but an interlinked spaces in the memory banks of machines. When the forward and backward pointers are provided and preserved, recursive entries and exits in these interlinked spaces are distinctly possible for the machine to reiterate its deductive reasoning, just as human beings would weigh, consider, and reconsider their thoughts and contemplate their decisions. The responsibility of preventing circular and self-trapping loops becomes the responsibility of knowledge domain programmers. In configuring operating systems for traditional computers, software designers (aim to) prevent circular loops, deadlocks, input/output (I/O) incompatibilities, viruses, etc.

4.2.2 Intertwined Spaces of Knowledge

When the bodies of knowledge overlap, union and intersected subspaces emerge. These subspaces can enclose objects that influence other bodies of knowledge and the stability of the overall global environment that hosts numerous objects. The influence of objects and their attributes needs representation as the machine manipulates objects and their attributes during the solution of knowledge problems. In essence, the memories of machine should be designed to hold objects, attributes, and their related objects (as cross-linked objects) to whom they bear a relationship.

The three-dimensional Cartesian space has little use in the representation of knowledge. As the domain of knowledge expands to include newer objects, relationships, and dependencies, the knowledge space needs to be modified by readjusting and realigning axes and even their superposition. In a sense, the incremental knowledge may indeed modify the previous body of knowledge and displace the origin of the older coordinate system. For example, when relativistic physics was introduced, corrections to the classic knowledge and concepts in physics needed to be enhanced. As another example, when the truth about the Watergate tapes became known, the role of the Nixon White House in the coverup required revision.

To discover a hypothetical space for knowledge and concepts, the notion of connectivity between hyper-dimensional, intersecting, overlapping, and curvy spaces needs a mathematical treatment. If the mathematical operations appear impossible at this stage, then the software to lead a machine through such an irregular space becomes necessary. In this knowledge space, *microcosmic* mathematical operations [such as localized partial differentiations, localized
counterintegrations, localized object dynamics (for displacements, velocities, accelerations, and forces) are feasible (see Figure 1.3).

4.2.3 Contours of Knowledge

In the physical world, civil and structural engineers have discovered (or derived the fact) that the integrated horizontal and vertical components of forces need to be balanced just as action and reaction need to be equal and opposite to maintain the equilibrium of any structure. This principle is verified by constructing the funicular polygon [3] to verify if there is any residual force that is not compensated. Such residual forces yield slow and insidious instability for the structure. The concept of evaluating the destabilizing forces on “objects” in the knowledge domain is examined in the knowledge domain.

In order to determine the stability of objects in the knowledge space, the closed contour integral of the social forces acting on an object or object group should be zero. Objects within society have bonds and structures of their own. They are subject to stability, partial stability, and even collapse [3]. Objects and object groups in society can have inertia, friction, and stick-sion effects. The response to social forces could be positive neutral or negative. Object respond accordingly with acceleration, momentum, and displacement for each of the forces if the objects are positively predisposed to forces. The opposite effect occurs for objects with negative predisposition toward such forces, causing residual tension and friction in society.

In the knowledge space when objects undergo forces in society, they tend to move swing, vascillate, or even vibrate. For example, if a force of expense is not balanced by a force of income, the object (a human, corporation, nation, etc.) moves from solvency to debt over a period of time. As an another example, if the wisdom of a nation is not balanced by a sum of internally generated concepts, innovations, and/or imported axioms (know-how skills sets, literature, etc.), then the wisdom bases in the nation are consumed and the nation drifts into bankruptcy of knowledge, concepts, and wisdom. Numerous other examples of economic entities (GNP, production, spending, etc.), manpower, education, ethics, etc., also abide by this rather mundane and mechanistic rule for the dynamics and stability of objects.

4.2.4 Funicular Polygon of Concepts

In the knowledge space, if the funicular polygon of concepts does not close, the contribution of that particular body of knowledge (paper, report, book, movie, documentary, etc.) should equal the residual vector to close the polygon of concepts. This vector is generally hyper-dimensional since the coordinates at the beginning of knowledge (before processing the body of knowledge) and end

---

3This concept is borrowed from the theory of structures based on the closed/open contour of the funicular polygon, emphasizing that a residual horizontal and/or vertical force remains active on any structure if the polygon of forces is not closed.
of knowledge (after processing the body of knowledge) can occupy different knowledge (and subject) coordinates. This residual force contributes to the movement (progress) of the objects and is a result of the change in entropy of that particular body of knowledge. The magnitude of the movement depends on the bias (friction, stick-sion, predisposition) in society. Forces in society inducing change (i.e., verb function) and the mechanics of (noun) objects become computable.

Bodies of knowledge that have the most impact on society deliver or contribute a linear (highly directional) vector from the known into the unknown and establish a localized contour of knowledge at the end point of the vector. Such a contribution is a breakthrough, and it can be precisely spelled out. Even though such contributions are rare, classical examples (such as relativistic corrections, Maxwell’s equations, notion of pi, Heisenberg’s principle, unified gas laws, anti-slavery, etc.) have occurred. Quantum jumps in the knowledge domain do not generally follow an articulated game plan.

A totally closed funicular polygon of concepts is an exercise for the mind rather than a motivation. The beginning and end are the same in the hyper-dimensional space of knowledge. The motivation for the next thought is the open end of the polygon of concepts that a body of knowledge does not close till the polygon is closed. Concepts are thus born in the minds of those who dare to jump in and attempt to close an open end of the funicular polygon in the knowledge domain.

4.3 COMPILATION OF KNOWLEDGE

For a moment, consider a “body of knowledge” (BOK; e.g., a paper, book, document, etc.) as a “program” to convey a message. Next, consider the basic processes (lexical, syntactic, and semantic analysis) embedded in traditional compiling (Sections 4.2.1.1–4.2.1.4) of programs by computer systems to extract symbols and operators in programs. Now, superpose the principles of compilation on the body of knowledge by invoking the routine machine executable steps involved, but to extract the “knowledge” (noun objects, verb function, and structure) encoded in that body of knowledge.

Largely, the program structure of a well-written software module is reflected in a well-organized document accompanying the module. Both have their own parallel structures. One basic difference is that most documents written for human use are poorly structured, but the human mind is forgiving and adept to extract the message from the document. Such a process is also akin to the human filtering of poorly structured audio signals to recover the message from poor-quality audio signal waves. The practice of message filtering by human senses is evolutionary, even though the principles are recently documented. The process of knowledge filtering by machines is not sufficiently evolved to extract the genesis of knowledge from poorly written documents by and for human beings.

Whereas the science of compilation (of computer programs) is well documented, the science of compilation (of knowledge from message-carrying documents) is still to follow. For initiating a platform to compile knowledge, we
propose that BOKs be subjected to the same rigorous steps in the compilation of programs written in any standard computer language.

### 4.3.1 Lexical Analysis for Knowledge

As the first function in compilation, lexical analysis permits the machine to scan and overview the program for permissible key elements and the authentic operations on and among such key elements. Parsing is a required function in lexical analysis. It permits the computer system to identify and classify the key elements (noun objects) embedded in the program (BOK) and authenticates that the elements in the program are indeed valid and recognized by the system (from a dictionary). Traditional operators (+, −, ×, ÷, etc.) are recognized and tabulated for syntactic analysis to verify that such operations requested in the program are indeed permitted in the language of the program.

Computer programs are well-structured BOKs written in any standardized language that can be compiled. In processing, generic BOKs are also programs, but the syntax and semantics are far more flexible. They can be seriously error-prone. However, if the parsing can be relaxed to tolerate the loose structure of generic language and filter out its flexibility, it becomes a multilevel process (presented next) to identify noun objects and verb functions first, and then interpret the relationships that identify the structure of knowledge as a graph.

### 4.3.2 Parsing Bodies of Knowledge

Parsing for symbols and tokens is a standardized procedure in compilation. However, in dealing with complex objects and BOKs, parsing for (noun) objects, relationships, their attributes, and operators tied to the objects becomes essential. Operators are equivalent to verb functions that operate on noun objects to alter the entropy of the object or object group. The function is equivalent to the execution of a operation code on an operand. When the operation is an executable operation on the operands [i.e., not a no-op (NOP) or pseudo-operation], the result is computed and stored at a preassigned location.

#### 4.3.2.1 Parsing for Noun Objects

Noun objects (NOs) should be considered as dimensioned symbols in typical computer systems. The numerous dimensions should accommodate a unique or group address for the object, attributes, qualities, linkages, pointers, precedent, antecedent and following object addresses, location and identification of the object location in any knowledge tree that holds the noun objects, etc. It is indeed a complete digital image of the object.\(^4\)

\(^4\)This type of digital image is used by security agents to hold and store the images of ordinary police suspects, members of organized crime, terrorism suspects, etc.
1. Parsing for information-bearing noun objects (Section 3.2.1)
2. Parsing for qualities of noun objects:
   (a) Nature being positive, neutral, or negative to move toward A, M, or O types of wisdom
   (b) Reactivity being intelligent, linear, or nonlinear
   (c) Intelligence to verb functions by being assistive, neutral, or resistive to verb functions (see Sections 3.3.1–3.3.3; Chapter 5 further discusses the nature of noun objects)
3. Parsing for relationships (and linkages) between noun objects (Section 1.3; see Figure 1.1)
4. Parsing for attributes of each noun object (Section 1.3.1)
5. Parsing for relationships between attributes of each noun object (Section 1.3.1)
6. Searching for related objects, attributes, and their relationships, of associated (derived and/or dependent) local and Internet bases

The first pass of parsing for noun objects is very passive. In any body of knowledge, the function is merely to identify the noun-objects and distinguish them from nonobjects. The location of the objects in each sentence and its role are determined in the context of the structure of the sentence. The location and role of each sentence are determined in the context of the structure of the paragraph, etc., until reaching the highest node to evaluate the context of the body of knowledge in the context of the universal and accepted domain and dimensions of knowledge. An entire array of unidentifiable noun objects in any body of knowledge (in any language) deems such a body of knowledge gibberish.

4.3.2.2 Parsing for Verb Functions The parsing for verb functions (VFs) can also be equally detailed. Two fold equivalency is evident: verb in the sentence structure, and operator in the CPU. Verb functions being equivalent to operators (arithmetic, logical, matrix, complex, etc.) need scanning and the context in which the operations take place. For example, if the machine is to search for a rare ailment in the human species, then the verb function (köpc) is the “search,” the noun object (koperand) the “human species,” and the verb modifier (specialized köpc from the set of search köpcs) a “rare ailment.” All three parameters become instrumental in obtaining the correct answer from the search engine I/O device of the knowledge machine. Such a search process needs to be identified and tagged to the knowledge instructions. Processes of this nature are common in the computation of data (within the CPU) when operators and operands have to be consistent with the capacity of the machines (i.e., integer operations on integer variables, floating point operation on floating point variables, matrix operations on arrays, etc.) to execute an instruction. Typically, the parsing for the verb functions has at least six variations:
1. Parsing for operative verb functions (Section 3.2.2)
2. Parsing for the nature and intensity of verb functions
   (a) Nature being positive, neutral, or negative (+, 0, −) to move toward
       absolute, materialistic, or opportunistic types of concept and/or wisdom
   (b) Intensity being strong, mild, or weak (Section 3.2.2),
3. Parsing for tense (past, present, or future) of each verb function
4. Parsing for conditional, dependent, and linked verb function
5. Parsing for relationships between the nature and tense of each verb function
6. Searching for the related verb functions of associated (derived and/or
   dependent) local and Internet bases, and their nature and inter-relationships

The first pass of parsing for verb functions is very passive. For any given
body of knowledge, BOK, the function is merely to identify the verbs and actions
and distinguish them from nonaction-oriented verbs. The location of the verbs
in each sentence and its role are determined in the context of the structure of
the sentence and of the noun objects identified in the previous parse. Such a
parsing of operators including their relationship with symbols is standard proce-
dure in the compilation of a computer program. An entire array of unidentifiable
verb functions in any body of knowledge (in any language) deems such a body
of knowledge to be as challenging as a sentence without a verb. Furthermore,
unidentifiable verb functions in conjunction with unidentifiable noun objects deem
the body of knowledge to be challenging gibberish or like listening to an alien
corneration.

4.3.2.3 Parsing for Relationships of Noun Objects Relationships be-
 tween noun objects and their attributes give structure to knowledge. The contour
of any body of knowledge is composed of the peaks and valleys of relation-
ships. Lateral and hierarchical relations offer horizontal and vertical dimensions
to offer a snapshot of the profile and the expanse of its contents. However, the
hyper-dimensionality of knowledge forces the contour, profile, and expanse to
become a multidimensional and complex object. Such objects are conceived by
the mind much faster than by a machine. Much like computer imaging software
that can present pictures by filtering or enhancing the light of certain wave-
lengths, the knowledge-processing software can present images in any Library of
Congress (LoC) or Dewey Decimal System (DDS) subject classification, or any
other classification such as the artistic, literary, or even emotional content. Such
knowledge-processing software is as probable now as image-processing software
was during the days of von Neumann; it is just four decades too late.

Lateral and hierarchical relations between objects (and their attributes) form
the basis for a graph embedded in the body of knowledge. Largely, the graph is
the graphical or symbolic representation of the structure and stability of concepts
embedded in knowledge. Perfect knowledge will have eternal and indestructible
concepts (and even ethical values) within it. In reality, the machine (and human
beings) will learn to processes incremental concepts from imperfect knowledge
for a limited timeframe. Primary concepts are relationships between primary noun objects related by primary verb functions giving rise to the main graph of knowledge. Branches and twigs in the graph result from secondary objects and attributes with weak relationships.

Parsing of relationships in the machine helps to construct a graph of concepts. Syntactic and semantic analysis will then be able to verify the validity of the graph in comparison with any identifiable concept structure from universal knowledge-concept bases around the Web addresses.

4.3.2.4 Parsing for Nature of Verb Functions  As much as noun objects are related, the verb functions are also intertwined. Verbs indicate action. Different segments of action may be inter-related just as noun object may be intertwined. For example, if the action is to pass an examination (noun object), the examination may be in numerous subjects (related noun objects), and passing implies the practice of different approaches to pass the tests in different subjects. Hence, the execution of a task on a noun object can lead to numerous responses and results (see Section 3.3.1).

4.3.2.5 Extended Artificial Intelligence Strategies for Bodies of Knowledge  Pattern recognition (PR) is a recognized branch of AI, and compilation is a systematic procedure in identifying symbols and operators. In the language of knowledge, objects take over the role of symbols and verb functions (individual or grouped) take over the role of operators. The grammar for the documentation of knowledge, being more complex than that of computer languages, offers a large variety of operators and their syntactic variations. However, a thread of conceptual commonality exists. The recognition process (from PR) and identification of symbols (from compiler theory) overlap. Both recognition and identification are intensive disciplines calling for accurate processing and algorithmic refinement.

If parsing, lexical, syntactic, and semantic analyses are placed in an inner shell of the flow-graph of a compiler for a computer environment, then statistical, syntactic, and semantic PR forms an outer shell for the flowchart of recognition procedure in a complex visual system. Now, if the two shells are collapsed into one, then a knowledge machine (KM) can be the software recipient to compile and “visualize” the BOK under process that passes through the KM.

The software integration of the two shells needs to satisfy the goals of the KM per se. Between the I node and K node in Figure 3.6, the KM software serves as the compiler–visualizer of (information-bearing) objects buried in information and the relationships between such objects. Between the K node and C node in Figure 3.6, the KM and CM (concept machine) serve as a compiler–visualizer of (knowledge-bearing) notions buried in the knowledge and the relationships between such notions, etc. The complexity of the compiler–visualizer becomes more and more severe as its functionality is customized for functions at the seventh and eighth nodes of Figure 3.13.
If the goal of the knowledge trail is movement toward Aristotle’s (type I) ideals, that is, truth, virtue, and beauty (TVB), then the goals themselves become the rewards. Conversely, if the goal of the knowledge trail is movement toward Aristotle’s (type II) objectives, that is, deception, aggression, and hate (DAH), then the goals become a self-consuming cannibalistic trap. Both destinations are manifest in society. The KM becomes a means of the mind to an end for the soul. The writings of Carl Jung point to similar notions about modern man in search of a soul.

4.3.3 Syntactic Analysis of Knowledge

Syntactic analysis in data-processing systems is essential to preventing the CPU from trying to execute impossible or ill-structured instructions. In the same vein, knowledge operations (kopcs) in the knowledge domain should be executable in the KPU in the context of the knowledge operands (koprs) that will undergo operation. As for the context of the kopc requesting any particular knowledge instruction, the instruction should be a legal operation that the KPU can execute, thus making syntactic analysis necessary.

For example, if a KM is to construct a graph of objects and relations from a body of knowledge, then the type and format of the body of knowledge need declaration and must be acquired in memory locations from an input source. As a precursor to actual execution, the machine may instruct the extraction of noun objects and their attributes together with the relationships between such objects, their attributes, and the relationships between attributes of attributes. Elements of the graph, such as branches, twigs, and terminal objects, also need identification and labeling, so that a complete graph may be constructed.

Inconsistent and incorrect modules are tagged for the semantic analysis to verify the validity and appropriateness of the module of knowledge. In a sense, every module of information and knowledge ($I \leftrightarrow K$) is thoroughly validated syntactically and filtered for consistency with the structure of knowledge from the local and Internet knowledge bases.

4.3.4 Semantic Analysis for Knowledge

The precise meaning of the noun object can vary from one occurrence to the next. The appropriate meaning will be identified in the local and global context of the information that is being communicated during this phase of compilation. The true identification of the meaning in the context of its usage also becomes essential. The human mind is well adept at this type of filtering that blocks the unsuitable meaning of any noun object to clutter the overall message in the body of knowledge. In a sense, the KM finds the embedded relationships to other noun objects and their attributes as it evaluates which meaning bears the most correlation. The check can be completed if certain combinations are already declared acceptable, questionable, or denied in accordance with the rules of the particular context. Knowledge compilers can indeed be written for science-, fiction-, business-, and economics-oriented bodies of knowledge.
Such compilers are frequent in traditional computer systems. Fortran, Algol, C*, Prolog, and graphics compilers exist. When the application is extensive, specialized application programs are developed. Graphical user interfaces (GUIs) further facilitate the flexibility and usage of such programs. The uniqueness of the knowledge-level compilers is that they unify the older object-oriented compilers to address social and human problems with passive and active objects that may have linear and nonlinear responses. When such compilers generate executable code for KMs with KPUs capable of handling numerous objects, related objects, their attributes, and their relationships, then the KMs can perform comparably to any traditional mainframe computer system.

A sense of reality and energy balance between noun object and verb function is implied in the semantic relationships between objects and verbs. Reality and energy balance have a numerical relationship. For example, an ant moving a mountain forces a fictional nature or highly improbable reality, whereas a machine moving a mound of snow becomes readily acceptable as a reality. In the realistic directions of sciences, the semantic analysis of knowledge programs “weighs and considers” [4] the energy balance of operations. The compiler thus has the potential ability to tag fictitious (or highly improbable) operations in a realistic scientific program and vice versa. Other semantic violations are also tagged by the compiler. When Web knowledge bases are available, the compiler has the option of looking up the particular knowledge base to assign a confidence level accepting or denying the validity of any proposed knowledge function. In a sense, the compiler acts as a reference to the proposed knowledge function within a program. Similar lookup procedures are used in Mycin [5] and NeoMycin [6] systems in the medical field.

This implication invokes a sense of realism (or possibility) or fiction (or impossibility) in the syntactic and semantic relations between noun objects and verb functions. For example, a module of knowledge (i.e., a sentence or paragraph) may represent reality or a virtuality. When reality is implied in scientific subjects (i.e., DDS classifications 500 and 600), the laws of sciences5 (such as laws of dynamics, thermodynamics, and stability of structures, also derived from the same DDS classifications 500 and 600, etc.) are applicable, and verifying

5Subjects in the Dewey Decimal System (DDS) classification, are 000 Generalities, 001 Knowledge, 002 The book, 003 Systems, etc., 100 Philosophy & psychology, 101 Theory of philosophy, 102 Miscellany of philosophy, 103 Dictionaries of philosophy, etc., 200 Religion, 201 Philosophy of Christianity, 202 Miscellany of Christianity, 203 Dictionaries of Christianity, etc., 300 Social sciences, 301 Sociology & anthropology, 302 Social interaction, 303 Social processes, etc., 400 Language, 401 Philosophy & theory, 402 Miscellany, 403 Dictionaries & encyclopedias, etc., 500 Natural sciences & mathematics, 501 Philosophy & theory, 502 Miscellany, 503 Dictionaries & encyclopedias, 504 Not assigned or no longer used, etc., 600 Technology (Applied sciences), 601 Philosophy & theory, 602 Miscellany, 603 Dictionaries & encyclopedias, 604 Special topics, etc., 700 The arts, 701 Philosophy & theory, 702 Miscellany, 703 Dictionaries & encyclopedias, 704 Special topics, 705 Serial publications, 706 Organizations & management, etc., 800 Literature & rhetoric, 801 Philosophy & theory, 802 Miscellany, 803 Dictionaries & encyclopedias, 804 Not assigned or no longer used, etc., 900 Geography & history, 901 Philosophy & theory, 902 Miscellany, 903 Dictionaries & encyclopedias, 904 Collected accounts of events, etc.
syntactic rules make the module of knowledge syntactically appropriate with the other modules that constitute the rest of the module. Virtual operators (such as magic, witchcraft, demon-worship, etc.) on virtual objects (such as fairies, witches, demons, etc.) may be executable in the KPU environment to create a fictitious atmosphere of computer games and cock and bull entertainment.

4.3.5 Knowledge Machine Code

The actual machine code depends on the hardware capability of the KPU, the macro and subroutine libraries in the compiler. An extensive set of macros and subroutines is not written for the overall processing of knowledge programs. An efficient knowledge compiler will make the processing of human and social computer programs as dependable as the processing of scientific programs when the optimizing Fortran V compiler was introduced in the 1960s.

4.4 DERIVATION OF KNOWLEDGE

New knowledge can be derived from existing knowledge at three levels. At the first level, the execution of a simple or complex verb function (kopc) on one or more noun objects \([kopr(s)]\) causes an incremental change in the entropy by altering the status of noun objects. The convolution of the VF’s on NO’s may also be altered in the KPU. These functionalities occur in the KPU hardware and, typically, these changes may be too small to offer new knowledge. However, when the functionalities are compounded and cascaded by sophisticated knowledge programs, the change in entropy of knowledge can be significant. The KPU has access to similar objects in the object, and an element of embedded intelligence can be implanted in the KPU itself to predict a massive change in the entropy if the objects are appropriately adjusted. However, a series of such assembled programs from the compilation of a knowledge program can produce new information from a BOK from which the noun objects are parsed.

At the second level, the execution of an entirely consistent (application) program on one coherent and self-contained BOK creates an increased change of entropy and transforms BOK to BOK’. This second-level functionality occurs in the machine hardware. All or some of object relationships, their attributes, their relationships, and attribute of attribute relationships would have undergone modifications, thus causing greater changes in entropy. The KM or WM has access to similar objects and BOKs, and artificially intelligent agents of embedded intelligence can be implanted in the machines to predict a massive change in entropy if the BOKs or objects are appropriately adjusted. As a next step, a series of such assembled application programs from the compilation of massive social and political actions can produce new and more beneficial social and political actions for a society from which numerous BOKs are drawn.

At the third or universal level, when an entire array of programs (dependent, quasi-independent, or independent) operate on a global collection of BOKs (e.g.,
the real world, a universe, or a society or nation), then the change of entropy in the knowledge of that universe can become unpredictable and even chaotic. Examples of such conditions occur in seismic eruptions (see Section 1.6.4), stock markets, or even nations undergoing severe dynamic conditions. Examples of situations in which the changes are controlled but nonchaotically include the migration of herds, evolution of species, natural erosion of land masses, etc.

In the following subsections, the representation of machine architecture to handle these three levels of change is discussed. The KPU configuration, and knowledge and wisdom machine configurations are explored next.

4.4.1 Microscopic or Basic Assembly-Level Knowledge Instructions

In basic assembly-level knowledge (BALK) instructions, a typical verb function (VF) with numerous knowledge operation codes (kopcs) can operate on a complex object BOK with numerous noun objects (NOs), operands or koprs. A typical KPU layout with data base support as shown in Figure 4.1. To verify the validity

![Figure 4.1: Logical configuration of the knowledge-processing unit to execute a standardized VF on a complex authenticated object. All the knowledge operation codes (kopcs) are microscopic verb functions, and all the knowledge operands koprs are microscopic noun objects. Only permissible verb functions can be executed on legitimate noun objects. The numerous control circuit chips now becomes the equivalent of many instruction registers and their control memories, in a MIMD or pipeline central processing unit architecture. CCC = control circuit chip.](image-url)
of the operation, a database of permissible *kopcs* is provided on the left side of the figure. Verification of the match between *kopcs* and *koprs* becomes crucial in the KPU because of the dynamic nature of the objects. Operation code-operand mismatch can generate unpredictable results in the KPU and cause instability in the program’s execution.

The databases of related objects, their attributes, and attributes of attributes also permit the changes that the *kopc* will bring about on the entire BOK. When the noun objects are classified by type of objects, the respective VFs can also be arranged to match. In Figure 4.2, the layout of a VLSI chip for the KPU is shown. The operand, relational, and attribute databases, and logic and control circuit space for the KPU, are shown on the right-hand side of the figure. If the databases are not large, the cache memories can be used instead of traditional data bases and the entire KPU can be accommodated on a single chip or wafer. There are two aspects to the decoding and execution of the *kopcs* shown on the left side. At the database level (extreme left of the illustration), a new and updated list of *kopcs* is generated from the local and Internet bases. The control memories to decode the *kopc* are shown in the center.

### 4.4.2 Application-Level Knowledge Programs

Application-level knowledge programs (ALKPs) operate on complex objects and have the necessary *kopcs*, macros, subroutines, and library functions to modify the entire structure and characteristics of objects, their attributes, etc., of the components of knowledge. The extent of modification or enhancement depends on the nature of the application programs.

An example of such application programs is available in the executive and management information systems (MISs) of a corporation. The simplest of such MIS programs is the report generation MIS that scans the complex object (i.e., the corporation) and reports the vital signs (such as the price/earning, liabilities/assets, balance sheets, etc.) of that object or entity. More complex MIS functions (such as corporate realignment, production scheduling, maximization of profits, etc.) require dynamic profiling of all the essential objects (noun objects) and entities (complex noun objects with relationships, attributes, etc.) in the corporation and the effects of altering executive policies (verb functions).

Other examples of complex entities occur in medicine, hospital management, and national and econometric modeling of numerous segments of the economy (such as consumption, savings and taxes, or industrial production, investment, and government spending, etc.). An architectural configuration of the corresponding knowledge machine is shown in Figure 4.3.

The applications of the knowledge machine are generic in almost all disciplines of scientific study where the dynamic equilibrium can be modeled by mathematical equations or statistical behavioral patterns. The knowledge machine has the final edge over traditional computing systems because of Web access, AI techniques, and predictive behavioral pattern synthesis that offer knowledge machines significant advantages over normal computers especially in social problems.
Figure 4.2  Spatial layout of the knowledge-processing unit to execute a standardized kopc on an authenticated object. Note that the objects are knowledge-centric noun objects that occur in any body of knowledge. Specialized kopc for different groups of noun objects are executed in the logic and control space of the unit. CCC = Control memories and control circuits. DB = Data bases, Atr = Attributes.
Figure 4.3  Extension of Figure 4.1 to application-level programs to change the structure and entropy ($E$) of a body of knowledge (BOK) to the derived entropy ($E'$) of a newly computed body of knowledge (BOK'). The symbol * indicates a machine-based knowledge convolutions.
The synergy due to statistical, AI, and predictive pattern recognition techniques over and above the computational techniques (i.e., arithmetic, logical, matrix, and scientific) in the knowledge machine forces the machines to behave in a humanistic mode with humanistic characteristics in dealing with complex objects. When the rational side of the attributes of the objects in the complex BOK is supplemented by inclinations and charismatic choices, the humanistic functions and decision of any KM can be slanted to become almost human.

4.4.3 Universal-Level Knowledge Programs

These programs extend the operations of a knowledge machine further such that the machine is able to operate its knowledge functions on numerous complex objects in any universal setting and alter the entropy of the universe from where the complex objects are drawn. When the complex objects reach the boundaries of the universe of knowledge that encompasses them, the functions become unpredictable because of the truncated scientific, statistical, AI, or predictive capabilities of the machine. The machine can only come to a halt because the basic kope in any KM is based on some prior knowledge about the objects, its related objects, its attributes, etc.

A typical configuration of such a machine is shown in Figure 4.4. If a machine could possibly track the transition between the two universes depicted in Figure 4.4, then the machine would have all the reasons for the evolution of the universe. A more localized and realistic version of a machine for discovering unknown diseases is shown in Figure 4.5. The noun objects (NOs), that is, human diseases in this case, are systematically mutated to find the existence of another hidden universe of an unknown disease. The symptoms of the patient are matched with the derived symptoms resulting from the mutated diseases, their relationships, the local objects, their attributes (ATs.), and the ATs of the ATs. In order to discover the order of the hidden disease, the symptoms and conditions of a suspected patient (with the unknown disease) and ailments and attributes of known diseases are processed by AI routines [5, 6] in the application programs in the KM or WM. The process is exhaustive and all the clues are self-tuned to maximize the probability of positive identification. The cure may also be synthesized by the KM or WM by tracking and mutating the generic medicines for the known diseases.

However, when the knowledge is minimal or inconclusive, the machine does have the ability to propose a hypothesis (see the section on WM) and test its validity. This type of “educated guessing” is practiced by scientists based on intuition or hunch [7]. Like a human being, the machine can systematically override the rational or logical blocks just to explore the unknown universe or objects in it that do not abide by our rationalities.

These functions that are unique to the WM may become initial machine-generated attempts to second guess knowledge where none exists. The human element of an WM is absent here since human interference can impede the semi-logical or exhaustive search that the machine is capable of performing.
Figure 4.4 Extension of Figure 4.3 to global-level programs to change the structure and entropy ($E$) of a complex unbalanced universe $U$ consisting of numerous knowledge-centric objects with an entropy level $E$ to the derived level of entropy ($E'$) of a newly balanced and computed universe for derived knowledge-centric objects in a different state of transition, stability or equilibrium. The symbol * indicates a machine-based knowledge convolution. S, M, E and V stand for sun, moon, earth and venus.
Figure 4.5 Extension of Figures 4.3 and 4.4 to comprehensively isolate (i.e., analyze, analogize, conceptualize, generalize, rationalize, synthesize, systematize, etc.; see Section 3.1.2) an unknown disease $D_u$ from the knowledge bases of the known diseases $D_1...D_n$. The path to $D_u$ may involve intuition and creativity of the human mind. Human control of the functions is an essential part of knowledge and wisdom machine operation. Ats. = attributes of any disease $D_i$. 

An unpredictable path of possible convergence to an unknown disease object $D_u$ derived (predicted, extrapolated, synthesized, combined, etc.) by the APs. The alteration of all the possible objects $D_1$ through $D_n$, their localized objects, their relationships, attributes (Ats.), and Ats. of Ats. is done by knowledge in KBs. 

Iteratively derived universe for a collection of diseases, their local objects, (Ats.), and Ats. of Ats. for the unknown disease $D_u$. 

Altered object sets ($D_1$ through $D_n$)
The role of intelligent agents in exploring or deriving universal knowledge has yet to be investigated. When the entropy of knowledge is low, the machine’s ability to be both instigative and intuitive or “institutive” needs a numerical estimation. Predicting a probability the machine will uncover new knowledge based on no knowledge, is an exhaustive search. Every logical block in the direction of the search needs to be overridden. In a sense, the reason to block a search should become a reason to search. Interpolating the directionality of the search is based on the machine’s sensitivity to break down the DDS or LoC system into finer steps. When the machine reaches its own numerical level of estimation of direction and probability, the search for new knowledge can be terminated. In a human environment, such initial limits are found only to be broken. The creativity of human beings still wins and a human intervention would be in order. Under these circumstances, the machine for universal-level knowledge becomes a super-sensitive wisdom machine with the capacity to override its own knowledge programs.

4.5 KNOWLEDGE MACHINE SOFTWARE HIERARCHY

The knowledge machine needs the traditional attire of software to clothe the bare machine. Being supported by traditional computer systems, the knowledge machine primitive software can be uploaded by the lower-level “slave” systems. At the lowest level of knowledge functions, the machine code for loading the knowledge control memories (KCMs, if any), kopcs, koprs, fetch, store, linkage loaders, etc., needs to be primed into the KPU.

The initial structure of the software is suggested in Figure 4.6. Instructions at the most primitive level provide the capacity for the machine to perform the elementary kopcs (e.g., OROR object register-object register, single-object fetch, single-object store, etc.) that are cascaded to perform complex kopcs. The utilities and libraries fall at levels 2 and 3, and different sets of libraries will become necessary based on the application. Knowledge machines like medical computers or educational systems will have applications in the social domain and serve as intelligent robots tailored to the individual needs of the user.

Individual human needs are well articulated by Maslow [8] and extended by Ahamed [9], and specialized libraries to meet such needs ranging from the most primitive (physiological and safety needs) to the most sophisticated (search and unification needs) can be evolved. In traditional computing environments, we have routines ranging from the trivial (arithmetic and logic routines) to the sophisticated (fast Fourier transform and complex matrix inversion routines).

A platform of traditional computer systems become desirable to build a full-fledged knowledge machine. The integrated hardware and software integration of a traditional computer is shown in Figure 4.7a. The layers of software can vary substantially between machines customized for different applications. Central or distributed processing environments are feasible. The input/output devices and processors can vary radically. The CPU can handle
numerous types of (single data, SD; multiple data, MD; or pipelined) data streams and execute single or multiple (SI or MI) instructions, etc.

4.6 KNOWLEDGE HARDWARE AND SOFTWARE SYSTEMS

In traditional computers, the building blocks [CPU(s), memory(ies), I/O subsystem(s), bus structure(s), and switch(es)] prevail in almost all general-purpose machines.

Specialized computers may have additional elements such as sensors, robotic control devices, digital signal processors, etc. In general, specialized architectures are more expensive and demand additional interfacing software. In order to execute higher-level knowledge-level programs, as shown in Figure 4.6, the

![Software hierarchy for a knowledge machine. This hierarchy is similar to the software hierarchy for traditional computer systems. It permits program writers to use the knowledge machines to resolve social, human, local, and global issues. BALK = basic assembly-level instructions for KPU-based systems, KOPC = knowledge operation code for the KPU, KBS = knowledge-based systems, ALKP = application-level knowledge programs, ULKP = universal-level knowledge programs (see Section 4.4), $C_3I =$ command, control, and coordinate with intelligence, UN = United Nations, IMF = International Monetary Fund.](image-url)

**Figure 4.6** Software hierarchy for a knowledge machine. This hierarchy is similar to the software hierarchy for traditional computer systems. It permits program writers to use the knowledge machines to resolve social, human, local, and global issues. BALK = basic assembly-level instructions for KPU-based systems, KOPC = knowledge operation code for the KPU, KBS = knowledge-based systems, ALKP = application-level knowledge programs, ULKP = universal-level knowledge programs (see Section 4.4), $C_3I =$ command, control, and coordinate with intelligence, UN = United Nations, IMF = International Monetary Fund.
inner layers of the knowledge ware (KW) need some enhancement to make the knowledge machine consistent with the capacities of a knowledge processor unit. Knowledge machines can also be built to abide by the multiplicity of KPU architectures. Some of the themes that can be readily enumerated are single-process single-object (SPSO), single-process multiple-object (SPMO), multiple-process single-object (MPSO), and multiple-process multiple-object (MPMO) designs. These variations are presented in [6].

The very inexpensive chip sets that constitute the traditional computer justify this platform. The hardware in the immediate vicinity of the KPU that is the bare KM hardware assumes a configuration as shown in Figure 8.17. Bus structures, object handlers, and object caches become complex but addressable and manageable. In Figure 4.7b, four computers in the platform act as slave machines and addressable I/O subsystems. The outer layers of the KM software function to act as compilers for the HLL knowledge programs, and as loaders and linkers for the executable knowledge code.

A four traditional computers make up the platform of a KM in Figure 4.7b. The functions of these four computers are depicted in Figure 4.8, and they, in turn, perform the following four essential functions in any complex knowledge based computational facility:

(a) The Dewey Decimal System (DDS) and/or Library of Congress (LoC) subject and content identification and address management system identify objects in the context of the knowledge program (compile program, step 1) and with reference to the knowledge bases in the local and global knowledge bases and libraries.

(b) The management of executable programs and libraries is more elaborate than that in traditional computers because numerous subprograms can be invoked as a result of the prior executable steps. Both compiler and interpreter features may be necessary to execute the more complex knowledge programs with human entities as objects embedded in the program.

(c) In terms of disk and database management, a series of secondary storage systems may be necessary to store (1) discipline based (DDS or LoC) objects and their attributes, (2) cross-linkages to other related objects and their attributes, and (3) precedent and descendant objects.

(d) Dynamic access management exists for objects, attributes, object linkages, and primary and secondary linkages in the problem solution and to further its verification and optimality.

Knowledge machines can act as object machines by removing some of their special knowledge-processing features and content-based addressing and processing capabilities. The differences between object processor units (OPUs) and KPUs are presented in Chapter 8. Knowledge machines can also act as wisdom machines by incorporating the traditional search for Aristotelian TVB, that is, truth (universality), virtue (social benevolence), and beauty (innate elegance and inherent balance) as the machine executes knowledge programs.
Figure 4.7 Design for the innermost software layers of a traditional computer and a knowledge machine. PI = privileged instruction, IO = input/output, DMA = direct memory access routines, Boot = bootstrap software, BAL = basic assembly-level assembler and SW, HLL = higher-level language compilers and SW, App = application-level programs, KL = knowledge level, OOP = object-oriented programs, K prefix/suffix designates knowledge-level functions.
Figure 4.8 Partitioning of basic computational functions within a knowledge machine. Four essential steps are executed by the conventional computers shown in Figure 4.7. DB = database, KB = knowledge base, DDS = Dewey Decimal System, LoC = Library of Congress, OOP = object-oriented program, RDBMS = relational database management system.

For example, if the information filters presented in Chapter 6 seek out a perfect combination of TVB and also attempt to filter out every hidden deception, arrogance and hate (DAH) in any BOK, then the KM-based machines will struggle to make sense from every piece of information. Preprocessed input information will facilitate the final outcome from the KM. However, the convergence toward wisdom becomes faster if the information is processed at the K and C nodes before the W nodes along the knowledge trail depicted in Figure 2.5.

When the movement along the knowledge trail is not pursued vigorously and systematically, the purely philosophic discussions about information, knowledge, concepts, and wisdom nodes, that is, I, K, C, and W nodes, create a tangled web of thought. In the next section, we illustrate the movement from the B (binary) node of Figure 2.5 to the TVB node after the E node in the same figure based on the dictionary definitions of the seven nodes given in that same illustration.

4.7 CLASSICAL MIGRATION PATH OF KNOWLEDGE

The definitions for the terms “binary,” “data,” “information,” “knowledge,” “concept,” “wisdom,” and “ethics” are taken from the American Heritage Dictionary and appear in the boxes under each node. Figure 4.9 depicts the meanings of the words “binary” through “ethics” under each of the seven nodes. When the words are read casually, the mental path through all seven nodes is irregular and often circular.
The multiplicity of the meanings of each node is also shown. From a purely philosophic consideration of the meanings, the general directionality for the movement is clear because the meanings in the right-hand side boxes (mostly) refer to the terms in the left-hand side boxes, confirming the notion that the progression of thought processes in the knowledge trail (Figure 2.5 and 3.4) occurs from left to right.

As indicated in Figure 4.9, the lack of a theoretical framework for knowledge that is proposed in Chapter 3, can force the human thought process to become trapped in circular loops of philosophy and mental road blocks. Knowledge machines can traverse the knowledge trail based on the computation that an object derived from a prior node can satisfy the criterion for the next node. The methodology is derived from achieving a high confidence value (see Figures 3.8 and 3.9) that the new entity will pass the tests of prior scientific investigations of that particular object in that particular DDS or LoC direction. The methodology is highly specific.

Tracing the semantic lineage as arrows leads to Figure 4.10. Numerous circular loops are also evident between the B and D nodes, around the K node, and between the E and TVB nodes. The arrows that lead to the K, C, and W nodes are external inputs in the knowledge trail arising from human intuition and creativity. The arrows going out of the I, K, and E nodes indicate the effect that information, knowledge, and ethics and values nodes have on society.

### 4.7.1 Knowledge Traps

In tracing the left to right movement of knowledge in Figure 4.10, four major traps (circular paths) are evident between the B and D nodes, at the K node, between the I and K nodes, and between the E and TVB nodes. Numerous jump-in points (K-4, C-2, 3, and W-3b) and jump-out points (I-6, K-5, and E-4,5,6) are also apparent. Long jumps (C-1, D-1, D-3, and W-1,3,4) between nodes also appear to result from the imprecise definitions of words such as “binary,” “data,” “data structures,” “knowledge,” “concepts,” “wisdom,” and “ethics.” Historically, human language has carried greater ambiguity than computer programs written in a structured language.

In an effort to reach the goals of wisdom and ethics, ancient thinkers successfully avoided the knowledge traps, unsubstantiated shortcuts, jump-ins, and jump-outs and formulated classical theorems in mathematics, oracles of wisdom, and scriptures of truth, virtue, and beauty. In order to be scientific and instill rationality, we propose that the migration path between nodes be precise and programmable (see Chapter 3, Figures 3.5, 3.6, 3.8, and 3.9).

### 4.7.2 Transition at Generic Nodes

For the sake of completeness, we represent the transition between any $(i - 1)$ to node $i$ in Figure 4.11. The major findings in any particular discipline are denoted in the minor node between $(i - 1)$ and $i$. The knowledge remains trapped in
Figure 4.9 The meaning of the words “binary,” “data,” “information,” “knowledge,” “concept,” “wisdom” and “ethics” are shown in the boxes under each node (and taken from the American Heritage Dictionary). When the common words are shown as lines with forward and backward, activity then a messy diagram as shown in Figure 4.10 is generated.
Figure 4.10 When the commonality of the words is replaced by arrows in the knowledge trail, it becomes unduly complicated and at least three circular loops near the B,D; I,K; and E,TVB, etc., nodes are formed. These loops depict the jagged path of society to reach its mature and stable state, in spite of the closed loops that block freedom of thought in many instances.
CONCLUSIONS

Figure 4.11 The discussions about any “subject matter” are replaced by tests for the D, I, K, or C material to be satisfactorily supported by all the current research in that subject matter. The ultimate condition for transition lies in the proof of the D, I, K, or C material and can be confirmed by the current research on that topic in the transition zone.

this node until a scientific methodology is established to move from \((i - 1)\) to node \(i\). The condition occurs if and only when the cumulative findings of all scientific contributions satisfactorily confirm that “information” can be classified as “knowledge,” that “knowledge” can be classified as “concept,” that “concept” can be classified as “wisdom,” or that “wisdom” can be classified as “ethics.” All the rigors of mathematics and methodologies of sciences are applicable to make the transition.

When the procedure of Figure 4.11 is streamlined and the iterations for the research are complete, then the diagram for the transitions for all the nodes converges to a figure like the one shown in Figure 3.8. Ideally, knowledge machines should be able to perform all the functions of conventional object computers and information-processing systems. Application-oriented designs will yield efficient performance and optimal solutions.

4.8 CONCLUSIONS

In this chapter, the basic concept of the operator (a verb function) operating on a generic entity (a noun object) is introduced as an incremental convolution. The result is the effect of a single component of a verb when it interacts with a single element of a noun object. Numerous such results are generated and they ac
cumulate. The effect is an alteration of the state of the object, its attributes, and its linkages for future use. A machine-based knowledge convolution is the cumulative effect of such incremental convolutions on the entire object. The integrated incremental convolution results are systematically merged with the original object, its attributes, and its linkages to create the altered object.

Conceptually, the effect is that of slightly altering the characteristic of an object as the machine executes a knowledge operation code (kopc). When a series of basic knowledge-level kopc s are executed, then the whole picture and perspective change and the processed object becomes an altered object. Numerous such incidents occur in nature or the real world: when a flower blossoms, when a student goes through college, when the rain falls, etc. When the knowledge programs accurately portray the state of change that the objects undergo in the environment or society, then the transitions can be tracked accurately and the direction of change can be adjusted to generate objects and entities that are desirable in any social setting.

Some changes are more profound (e.g., a tornado, an earthquake, a new member in a family, etc.), and the corresponding knowledge programs are more detailed and intricate. Knowledge machines like traditional computer systems can be tailored to suit the object types, the nature of change, and the environment.

The nature and architecture of the knowledge processor units (KPUs) differ considerably from those of control processing units traditional (CPUs). These differences are examined in detail in Chapter 8. It is possible to design a platform for the KPU that is built on numerous CPUs. Such CPUs perform mundane tasks such as numerical computations, matrix and array processing, signal processing, statistical functions, and hypothesis testing.

The nature and complexity of the software (including the operating systems) also differ considerably from those of traditional computers. The basic software operations are more closely aligned to object and attribute processing rather than numeric and algebraic processing. Such an extreme difference occurs between traditional computer systems and the electronic switching systems (ESSs) used in typical circuit-switched environments. However, the commonalities between two systems persist, and both mainframes and highly specialized ESSs have prevailed for the last six decades.

If knowledge machines are designed to serve human and social needs, then the major contributions in both the computer and communications fields will propel a new generation of integrated social domain computers and networks dedicated to serving society. The object banks of such machines can also be personalized to suit individual users, thus making the machines well suited to be the Personal Companions (PCs) rather than personal computers (pcs) etc. of users.

REFERENCES


