

A Knowledge Home

Personal knowledge structuring in a computer world

v. 6.03, 5 January 2008

Tommaso Toffoli
tt@bu.edu

Summary We propose developing a cultural package—let’s call it “A Knowledge Home”—aimed at giving individuals more independence and power in a world increasingly dependent on computers. Our initiative addresses an urgent social concern. Our approach is unusual in that it proposes making demands on *people* as well as on technology.

The term “cultural package” is employed here in its usual anthropological sense. For a canonical example, think of *hunting with dogs*; this “culture,” which took shape in the late paleolithic, entailed for humans to adopt, adapt, learn how best to use, and get themselves adapted to an external entity employed as an extension of their individual faculties. In our case, we want to make it possible for ordinary people to use the computer as an unobtrusive extension of their own capabilities. The computer will materially support an expanded personal information space—one’s *knowledge home*—and provide the additional processing power needed to structure and manage it. But these resources will have to be matched, on the human individual’s side, by the acquisition of substantial new skills, competences, and traditions—an extension of *ordinary literacy*.

Hardware is not really an issue; the main task will be developing the infrastructure needed to support this cultural graft. A knowledge home and the extended literacy that goes with it must fit its owner—they cannot be mass-stamped products—and yet the knowledge homes of different individuals must, like their individual literacies, retain enough commonality to permit cultural sharing—with all the power amplification that that entails.

Concretely, what will be needed most is a sustained effort at product identification, design, prototyping, evaluation, and dissemination. Beside a software substrate, this “product” will comprise tools and standards, examples and applications, a syllabus, textbooks and graded exercises, motivational material, scenarios for imitation and peer emulation, etc.—in sum, a culture.

<p>Preface i</p> <p>1 Introduction 1</p> <p> 1.1 The literacy precedent 2</p> <p> 1.2 The home metaphor 3</p> <p> 1.3 Knowledge prototyping 4</p> <p> 1.4 Computers and extended literacy 4</p> <p> 1.5 Ordinary individuals 5</p> <p> 1.6 Urgency and foresight 5</p> <p>2 Elements of a new culture 6</p> <p> 2.1 Surface expression vs deep representation . . . 7</p> <p> 2.2 A “verbing” language 10</p>	<p> 2.3 Basic tools and modular resources 11</p> <p> 2.4 Homemaking and “fire insurance” 13</p> <p>3 Designing the knowledge home 15</p> <p> 3.1 Leveraging innate mathematical capabilities . . 15</p> <p> 3.2 What shall the knowledge home know? 16</p> <p> 3.3 . . . and what know to forget? 16</p> <p>4 Conclusions 18</p> <p> 4.1 Survival value 18</p> <p> 4.2 Do we have to die 100%? 18</p> <p> 4.3 Ownership and satisfaction 19</p> <p> 4.4 A community of learning volunteers 19</p>
---	--

Foreword

By extraordinarily rapid growth sustained over several decades, computer “biomass”—not just the box sitting on my desktop, but the entire complex of hardware, software, networking, and information resources that surrounds us—has now reached a size to deeply affect the human world; in fact, it is in the process of radically transforming social ecology. Computers are increasingly making up the nervous system of small and large organizations, utility networks, machinery, and appliances; the attendant expansion of manufacturing, trading, and services has created a new economy of information plenty.

This ecology that is so favorable to the growth of new businesses and new forms of social organization does not automatically bestow a comparable empowerment on the *human individual*. Left to itself, the information industry tends to shape for the ordinary individual the role of a mere consumer of information products or operator of information appliances. Such a “colonial” relationship cuts off society from a deep reservoir of productivity; what’s more tragic, it robs the individual of a rich source of personal realization and satisfaction. Paradoxically, most of us are starving in the midst of abundance; are homeless though we may own castles; are offered magic tools but not the discipline that would make us magicians.

Besides a physical home, we all need some sort of personal “information storage and processing” homestead—a *conceptual space* which we inhabit as individuals and as individuals maintain, furbish, and continually renovate to provide us with a base from which to operate in the world. If well integrated with our mind, the computer can make possible a vast expansion of this personal space and of the resources available within it, and can provide a channel for the sharing and compounding of such personal wealth across society.

Unfortunately, for most people today the computer is an alien entity over which they have little understanding, control, or real ownership. What is on the market is so powerful and cheap (and, in a shallow sense, so easy to use) and gives access to so many resources that one can’t possibly turn down the bargain. Yet it is not clear what it is exactly that we purchase and how much ownership we can effectively exercise over it, as *cultural dumping* discourages us from developing those native skills that might allow us to make the computer our own. Thus, we travel the Information Expressway, we do business at “Information Motels”, but do not yet have an *Information Home* of our own to go home to!

This ecological explosion confronts individuals and society with an unprecedented evolutionarily challenge. Basically, three responses come to mind:

- To physically eliminate the threat (e.g., the Luddite movement, the Kaczynski case).

- To pocket the bribe of immediate gratification and ignore the long-term challenge. This risks letting ourselves as human individuals, and eventually humanity itself, slowly be pushed into irrelevance vis-a-vis the computer world.
- To strive to extend our personal capabilities by grafting computer resources onto our own individual mind, culture, and literacy—so that the power of the individual may grow hand in hand with that of computers.

By doing nothing, one defaults to the second response. We are interested in the *third* course of action, and that will require much effort. Here we discuss the premises and propose the basic ingredients of what we believe is a viable strategy.

Chapter 1

Introduction

To help others visualize one's objective—something that may not exist yet—it is expedient to make recourse to *metaphors*: objects, situations, and ideas that the audience is already familiar with and are in some way suggestive of the scenarios one wants to evoke.

The metaphors I will use for two main aspects of the Knowledge Home are basic categories of today's human life, namely, (a) the *home* and (b) *literacy*, both taken in their most ordinary sense. It will soon become evident that those are not just abstract metaphors for what I have in mind; materially, the Knowledge Home culture is meant as a *material extension*—a natural and inevitable outgrowth, as it were—of those two aspects of everyone's everyday life. All the choice that we have in this regard is help encourage that growth in a direction that will be more attuned to the needs of the human as an *individual* (which it definitely is) rather than just as a self-domesticated *eusocial animal* (which it is as well)[63].

Even as the world is noisily marching under the banner of computer triumphalism, one can make out the steady drone of computer defeatism. The subliminal message is, “*Ordinary* people will never muster the intellectual resources to truly integrate computers with their lives and minds; most likely, they wouldn't even *wish* to do so. Since only a few *specialists* can be expected to have an in-depth grasp of computers, the best they can do is turn computers into appliances that the masses can use for limited, standardized, ritualized tasks.”

Our thesis is that, on the contrary,

- C1 Ordinary people *do* have the potential to intimately incorporate the computer into their lives.
- C2 Most will find this relationship rewarding, will feel it as a natural expansion of their persona, and once they have experienced it they wouldn't choose to do without.
- C3 This empowerment of the individual will be beneficial to society. Besides leading to greater productivity, it will create more opportunities for personal satisfaction and lead to a better expression of human and material resources. Eventually, it will be deemed so essential that an individual lacking this relationship with computers will *de facto* be perceived as a substandard member of society, much as an illiterate person today.

For all this, we also maintain that

- This potential can only be realized at the cost of *substantial* personal and societal investment.

- Much research and experimentation are needed to find the most natural way for computers to become an extension of an individual's personal space.
- The free-market mechanism that so effectively drives the development of computer technology is not sufficient for the present purposes, and will have to be complemented by explicit societal goals. (How many children would “freely choose” to spend *thousands* of hours of their life in a classroom doing English and math?) Educators, scientists, and administrators will have to collaborate in finding out what is feasible, and in shaping, expressing, and carrying out society's wants in this matter.

It should be clear that we are not studying here how to “place an Internet node in every classroom” or “turn the computer into an appliance;” those initiatives address much shallower concerns than ours. What we are working at is the design—the invention, if you wish—of a novel *cultural* component that properly acknowledges and makes use of computers but that one may comfortably regard, much like one's home, as an extension of one's own persona. Our stress is on *knowledge* rather than information (cf. Doug Engelbart's early vision[23]—unfortunately soon debased by commercial and technological hubris); our concern, with the *person* rather than the gadget (cf. diSessa's long involvement[21]); with the empowerment of the *individual* before that of the *organization*.

Of course, the software industry is already involved in a related endeavor—one of enormous proportions—namely, to better integrate humans and computers as components of the “nervous system” of any large organization (cf. [66]). In fact, in the pioneering age of computers little explicit distinction was made between personal and corporate knowledge management (compare [23] with [24]). Today, however, most knowledge engineering research—having business and government organizations as its ultimate customer—primarily addresses the concerns of these collective institutions rather than those of human individuals as such. The two endeavors have fundamentally different goals.

It is true that the Knowledge Home culture may be able to leverage methods and tools developed for the sake of corporate knowledge engineering. All the while, it will have to bear with a “knowledge market” where standards, products, and incentives mostly reflect corporate rather than personal values. There is a lot out there, but how much of it is really for us?

It is also true that, as in the case of literacy (see

§1.1), a “more educated population”—to use Vinton Cerf’s words[8]—may be of benefit to organizations; but our primary objective is to benefit the human as *an individual*—organizations can and will look after themselves. Cerf’s full phrase is actually somewhat ominous: “Investments have to be made so that we have a more educated population *that can sustain a functioning Internet*” [italics mine]. To me, that sounds like “The Internet is now our God, whom we must please; if the God’s daily service require an *educated* population, well, that’s what we must provide, mustn’t we?” A wholly different vision is expressed by the title of Mike Dertouzos’s last book, *The Unfinished Revolution: Human-Centered Computers and What They Can Do for Us*[13].

To establish a viable culture is an immense task. This document is not the place for a work plan. Suffice it to say that our strategy is to initiate a *bootstrap process*—namely, to develop certain basic resources and expertise within a pilot community, which in turn will help develop more resources and expertise and a wider community. The basic approach, then, will be to bring forth a *community of learning volunteers* (cf. §4.4).

Inspiration, a precedent, and even material *prerequisites* for our initiative are found in the GNU project and the LINUX culture. Richard Stallman[57] demonstrated how effectively a dedicated, noncommercial community can deliver general-purpose tools of extraordinary utility and robustness. In turn, the fertility of the GNU project confirms Stallman’s foremost ideological statement: that a tool—never mind whether purchased or obtained for free—cannot become a true extension of oneself and part of one’s cultural heritage unless everybody has open access to its source and documentation and can potentially contribute to its development. Not incidentally, this happens to be the case for our prime cultural tool—namely, language.

Leveraging UNIX, GNU, and the ubiquity of Intel’s I32 architecture (the latter intimately tied to the unprecedented commercial success of MS Windows), Linus Torvalds demonstrated the value of a pragmatic, open-ended approach to culture building. The defining character of a culture is that it be empirically *viable*; LINUX spread and took roots because enough people and institutions liked what it had to offer. They also felt reassured that its gently-guided but basically collective development mode made it robust, cheap, constantly scrutinized, and yet open to individual creativity and custom adaptations. Finally, since everyone is in fact a *full owner* of LINUX, one is encouraged to contribute to it with the amount of dedication that only full ownership can inspire!

In the rest of this Introduction we illustrate the affinity between our initiative and the closest precedent that we can conceive of, namely, the development of *universal literacy*; we also explain in what sense our program goes beyond literacy, insofar as it tackles the *purposeful structuring* of knowledge for an individual’s sake—“knowledge prototyping,” as it were. Chapter 2 gives an overview of what we see as the key elements of the Knowledge Home culture and the way they are interrelated. Chapter 3 briefly touches on some foundational prerequisites for our program: What

theories and technologies will we be needing? The concluding chapter throw further light on motivations, goals, and plans.

Much of this material, taken separately, will be quite familiar to experts in various disciplines; what is new is the way we make use of it to outline a real *computational literacy* curriculum and its material support. (The common term ‘computer literacy’ normally refers to a very shallow set of human/computer interface skills. Following diSessa[21], one may speak of ‘computational literacy’ to denote the much deeper and more general competencies we have in mind.)

1.1 The literacy precedent

In ancient Egypt, as in other early cultures, writing was a *sacred art*—used by a technical priesthood chiefly for tasks of higher administration. There was substantial vested interest in equating the sacredness of the tasks with that of the art and of the persons conversant with it, and an incentive in retaining technical difficulties that would keep off the uninitiated[53]. More generally, in most literate societies through history it has been a matter of conventional wisdom (at least as expressed by the literate elite) that literacy is not, by and large, for the common people. This, for three “obvious” reasons:

- L1’ Ordinary people can’t possibly become proficient at reading and writing, because that takes a special innate talent (the “Right Stuff” syndrome[64]).
- L2’ Even if they could learn to read and write, what use would that be to them, shepherds, tillers, and stonecutters that they are?
- L3’ If, after all, they did learn the art and managed to use it effectively, wouldn’t that be a threat to social order?

Of course, the elite themselves must have known that point L1’ above was a belief of convenience (for instance, the Christian Church drew much of its rank and file from the gifted poor, and in Jewish communities it was held desirable even for housewives to be able to read the Scripture). Point L2’ is a patent *non sequitur*. Finally, point L3’ barely disguises that the perceived threat was actually to the privileged position of functionaries. Nonetheless, it took the Enlightenment to make acceptable, at least as a theoretical stance, that perhaps *everybody* could learn to read and write. It took the French Revolution to propose universal literacy in earnest. Finally it took the Industrial Revolution to provide both the effective demand and the material resources for it. Indeed, the results of a century of experience confirm that, contrary to beliefs L1’–L3’,

- L1 Everybody can learn to read and write.
- L2 Even though few read Kant or write epic poems, most individuals can write a letter, read a sign, or peruse a magazine. They find this faculty useful to themselves, and would be aggrieved if they were deprived of it.

L3 Far from being a threat, universal literacy has become one of the foundations of modern society. The illiterate is a substandard member of society, and risks being treated, if not viewed, as subhuman.

With the benefit of hindsight, one must grant that the quest for universal literacy was not a quixotic dream. But *what* was at the root of its success? If a new intellectual climate, together with the needs and resources of the industrial revolution, made universal literacy desirable, attaining that goal still wouldn't have been remotely *possible* without both a specific human predisposition and a deliberate social effort, as we shall see in a moment.

We believe in a rather close parallel¹ between points L1–L3 above and points C1–C3 of the main thesis. Thus, the following answers concerning universal literacy anticipate analogous questions concerning its incorporation of—as it were—a form of “universal computermanship.”

Universal schooling. While it is true that everybody can learn to read and write, we must acknowledge that this achievement takes *several years of mandatory, standardized, subsidized schooling*. In spite of the enormous cost, the perceived advantages for society and for the individual are so great that, in our society, to deprive a child of schooling (and thence of literacy) is a crime.

Natural predisposition. Universal literacy and the culture that goes with it are possible because they externalize, discipline, and sublimate *natural language*—and natural language happens to be something that, like walking, we humans are “wired for” from birth[9, 47]. We can train ourselves—though with some effort—to be at home with the ever more complex external constructions that society builds on language only because the foundations of these new constructions are already solidly within ourselves.

Commonality, and pragmatic standards. Literacy is not an all-or-nothing thing. It displays enormous quantitative and qualitative differences between individuals, undoubtedly owing to the interaction between inheritance, environment, and accidents of individual history. Nevertheless, literacy presumes a *commonality* of resources among individuals—lexicon, syntax, metaphors, and protocols. Even the most private diary is built with public-domain tools, and for this reason would be accessible to a wide public. Literacy is characterized by pragmatic, open-ended, public-domain standards which are adaptable to the needs of different levels of competency, different disciplines, and different registers of expression.

Personal use and competence. The social contract that makes literacy possible is not a sufficient condition for *universal* literacy. What is needed to spread, maintain, and reinforce literacy is *personal utility*—not just society's bidding. The man in the street buys a newspaper for the enjoyment or the utility he expects from it, not because his purchase will support a publisher or a political party. When he writes a personal letter, he will use a substandard expression if this is what conveys his intents, without a worry about debasing the corpus of English language. And he is *personally competent* to do his own reading and writing at

this personal level, without the intermediary of a village scribe.

Obviously, one of our goals is to verify the above premises; namely, that the achievement of a certain degree of “universal computermanship” is indeed within the reach of human nature, and, once attained, would get positive reinforcement from its rewards to the individual. We stress that what we have in mind is not merely a free ticket to an information “bread and circus,” but such open-ended expansion of one's faculties as only personal competence in the use of a resource can give. The centaur-like symbiosis between man and horse (as contrasted to flying a plane as a passenger) or the entente between hunter and dog might illustrate such a relationship between a human and an external but intimately integrated resource.

1.2 The home metaphor

Our chief objective, we recall, is to develop *cultural tools and resources* that will help the ordinary individual, supported as appropriate by the computer, to manage an *expanded personal information space*. The *home* is a good analogy, in the material domain, for the expansion of personal range that this proposal envisages in the information domain.

The home is a *physical extension* of the individual. It is at the same time a shelter, a cache of food and materials, a repository for durable goods, valuables, and memorabilia; it is a *structured* space with specialized areas, tools, and supplies for a variety of common activities; it represents an arena for intimate social interaction and a classroom for primary learning; finally, it provides a rendez-vous site and a stable mailing address. In sum, the home is an extraordinarily valuable outgrowth of our persona. Without a home we are reduced to a shadow of ourselves. Like the illiterate, the homeless risks being viewed as a *de-facto substandard* human.

Much as the home provides the individual with a base from which to operate in the physical world, we argue that it is becoming *indispensable* to develop an analogous individual home base from which to operate in the world of information. This effort, we maintain, will be the price the individual will have to be willing to pay to hold on to the title of “standard” human in a rapidly evolving information society (cf. [50]).

Three aspects of the home metaphor are especially relevant to the envisaged knowledge home:

- The home is an eminently *private* territory. We don't allow visitors or delivery people to move our stuff around or plant articles of their own without leaving a list or a map; to tamper with our plumbing and wiring; to put a camera in our refrigerator to know how much milk we have left.²
- Though devoted to personal use, the home is nonetheless a *cultural* space, in the sense that its features are

¹The contrast of this with certain arguments of [36] is only apparent. In fact, our basic positions are very close.

²Whoever has installed software in a home computer will recognize what kind of practices we are talking about[32].

compatible with those of the analogous personal spaces of most other individuals, and thus amenable to cultural transfer. Strangers will recognize my kitchen and bathroom and know how to use them—and even take ideas from them for *their* home. A similar dynamic balance between the need for privacy and the desire for cultural interchange will hold in the community of Knowledge Home owners.

- Most important for our argument: in spite of being a complex environment—which depends for its good functioning on a broad combination of technology, expertise, and custom—the home is nonetheless *managed* (and to some extent furnished and maintained) *by the very same ordinary human individuals who use it, and who, as a rule, are personally competent to do this managing themselves*. Are we to believe that the individual won't be able, with appropriate cultural support, to achieve a similar level of competence and responsibility in the management of a personal information space? Today, much commercial software aimed at the individual seems to suffer from the “white man's burden” syndrome. Though the words are “We make it easy for you,” the message is “We have to remain in control to protect you from your ignorance and irresponsibility,” and, of course, the real meaning is “We'll do our best to keep you incompetent enough to always have to depend on us!”

In sum, as Main Street lights up with the neon of ever more “information motels” and “information supermarkets,” we want to make it possible for the individual to also look forward to the warmer glow of an “information home” of their own.

1.3 Knowledge prototyping

The chief commodity to be handled by the culture we envisage is *personal knowledge*—how to organize it, extend it, exploit it, and, inasmuch as desired, share it. Much of the raw material of this knowledge will come from outside the individual, that is, from other individuals, organizations, educational institutions, the media; one's bank, doctor, lawyer; books we've read; and so forth. But, clearly, our knowledge is in a state of continual restructuring. As new information comes in we have to adapt what we already have there to the new circumstances; and even in the absence of external solicitations our mind always explores new associations, discovers better ways to order things, and continually cleans up the place and, as it were, rearranges the furniture.

Hardly ever will the acquisition of new information consist simply of *copying* some external stock (e.g., downloading a file). On the contrary, knowledge coming in and knowledge already present will have to be adapted, integrated, and to a great extent *reconstructed* at every step. While mass distribution is tooled for *verbatim duplication* (of books, compact disks, video games), the main business of our mental shop will be *knowledge prototyping*. It will be no surprise that the concepts and tools proposed here

are characteristic of a prototyping culture rather than of a mass-production culture. However much one may share materials, technique, and style with the whole tribe, the artifacts of personal knowledge are bound to be *one of a kind!*

1.4 Computers and extended literacy

The computer lineage, having landed, as it were, on an uncolonized island, is in a state of extremely rapid growth and differentiation. When we speak of “the computer” we do not refer of course to a specific object or a specific operating system in existence today. What we mean is the entire set of resources—whether lumped or distributed, hardware or software, personally or collectively owned—that will happen to be at hand for the storage, distribution, sensing, displaying, and processing of information. By drawing and integrating from this pool of raw resources, one will de facto design and reconfigure in real time a “personal computer” according to availability and circumstances. As explained in §2.4, this virtual object is what the individual will actually and truly “own.”

In ordinary literacy, a written document—a letter or a recipe—is normally a passive object, like a crystallized virus[6]; to be become active it must bodily be taken in by a human host and processed according to its nature. A book comes to life when is read by somebody who understands the language; a speech is delivered when somebody declaims it to the proper audience. Conversely, a copy of my live thoughts can be frozen onto paper by writing an essay or a legal deliberation.

In this sense, the activities of literacy, which use the human brain as a processor, are limited by the brain's processing power, memory capacity, and error rate—as well as by certain qualitative limitations—and can at best be run at the rate of a human “computer clock.” Also, most “machine cycles” of this brain are already committed to more vital and more rewarding tasks—tasks that our brain does particularly well and we wouldn't know how to offload to an external computer (could it dream, take a vacation, or write your love letters for you? play with your kids or enjoy an evening with your friends?). In sum, our brain is *overqualified* and at the same time *sorely resource limited* for many quite useful though quite unglamorous aspects of knowledge processing. Can we avail ourselves, for this purpose, of a “bureaucracy of functionaries?” For such is in essence what the computer is eminently good at providing—and literacy eminently good at handling.

The knowledge home is meant to extend the range of resources (emphatically admitting computer resources) that a host can tap through a “literate” interface, that is, resources that can (a) report to us aspects of an action or a structure, (b) allows us to specify and commission actions to be undertaken or structures to be built, or even (c) carry out *in our behalf* exchanges of this nature between two structures that are *both* external to us, such as when an Amazon script negotiates in our behalf with a used-book seller's software agent our request for an out-of-print book. Note that, though we may well be asleep when this happens,

the latter exchange can certainly be viewed as an action of our extended self just as when our hand brings a wine glass to our lips or our sheep dog nuzzles an errant sheep towards us from a distance, acting for us as a “remote robotic arm.”

There is only one catch. Let the labor of these personal servants—these external helpers, this bureaucracy of computer functionaries, whoever or whatever they may be—well come for free. We’ll still have to learn to *tell them what we want*. This is what we mean by extended literacy!

1.5 Ordinary individuals

We have carefully studied, and often cite, the report *More Than Screen Deep—toward every-citizen interfaces to the nation’s Information Infrastructure*[40], commissioned by the National Science Foundation and executed by an organ of the National Academy of Sciences. The latter used, for technical reasons, the term “every-citizen” to denote what we here more informally call “ordinary individuals.” We use this term in an inclusive way, meaning the great majority of people rather than just a technical or cultural elite; in this context, see von Rossum’s “Computer Programming by Everybody”[61]. The mismatch between the concerns of the plain citizen and some of the excesses of a technologically driven market is well discussed in [60]. Who ordered “toys for boys” and who is paying for them? Don’t even well-intentioned academics and engineers often naively envision a world populated only by their clones? And so forth.

Lest the reader discount the present approach as an idealistic one, we stress that we are aware of what anthropologists[41, 17] like to remind us of, namely, that in any country the average individual in its adult form, rather than a “noble savage,” is, not to mince words, “ignorant, greedy, gullible, and lazy.” (A more neutral term for all this would perhaps be “resistant to change beyond the age of social imprinting”—which obviously must have been a healthy trait from a survival viewpoint.) For all that, as every anthropologist will agree, the average citizen is *not* stupid, and is verbally articulate, alert, curious, interested in life, and willing to go to great lengths for the sake of self, kin, and friends. We are convinced that the ordinary individual will find value in our program.

This project has its roots in a research community. Perhaps its greatest scientific challenge will be not to confuse the needs and resources of this specialized community with those of the larger community it addresses.

1.6 Urgency and foresight

We conclude our introduction by stressing the *urgency* of this cultural project. We can’t agree more with diSessa’s plea in his book *Changing Minds*[21]: “Hardware is much less the issue than software.... The research and development of future literacies is an issue of *public trust* [italics mine] if ever there was one, but the issue does not appear on the agenda of any government agency. If the conclusions of this book are correct—or even a responsible good guess—we are making a terrible mistake by this omission.”

The computer is revolutionizing human ecology, while the individual’s raw capabilities remain essentially unchanged. If we want to maintain relevance *as individuals*—if we do not want to find ourselves strangers in our own land—we need to forge a cultural package that will provide at the same time protection, interface, and leverage between the individual and the computer world. A cultural invention—like house, clothing, the dog, literacy, and even the car!—that, even though shaped and transmitted by the collectivity, can be competently used and managed *by the individual* for the individual’s own purposes.

International scope. The appeal of the Knowledge Home culture is expected to transcend national boundaries.

Specifically, we seek substantial European participation in this project from the very beginning. Europe is grounded in historical and cultural traditions that are in many ways complementary to those of the US, and is more comfortable with letting policy be informed by the promptings of a socially responsible elite rather than merely by technological and commercial momentum.

For their relevance to the present project we may point to the pioneering role played by Europe in the development of structured programming[16] and object-oriented programming[67], Algol and Pascal, the LINUX kernel,³ and outstanding public-domain support for the T_EX environment[27] and GhostScript[38]. At a more philosophical level, also see [51], [49], and [26], among many.

Moreover, Europe is interested in the development of a more cosmopolitan computer tradition, one that would retain some independence vis-à-vis the US commercial and cultural market. For this purpose, Europe can often speak with an effective voice also for the rest of the world.

³One of the “hearts” that have been fitted in the GNU operating system “body,” yielding the GNU/Linux operating system.

Chapter 2

Elements of a new culture

1. In a world in which computers are becoming a dominant feature, what new cultural resources can help ordinary individuals maintain an adequate “personal information space” of their own?
2. How can the computer, which contributed to the challenge in the first place, be employed as part of the response?

The Knowledge Home culture is intended as an answer to these two questions. In the next four sections we’ll present the *basic elements* of the proposed culture, expanding the following outline:

- **Commuting between surface expression and deep representation.** Humans evolved in an environment that demanded continual minor inventions: every stone, every branch or piece of bone will be slightly different and will have to be adapted to the task at hand, every meal one cooks is to some extent improvised. Thus, man is “wired for” and takes pleasure in tinkering. We believe that even in our increasingly man-made environment, which abounds in matters informational, everyone should enjoy the opportunity to be an amateur *prototype designer* for their own needs. But we also believe that even one-of-a-kind artifacts are often best made by starting from a blueprint or recipe and iterating the *edit/fabricate/evaluate* loop—rather than by just “whittling down a stick” until it looks right. Today, the computer can virtually take over the “fabrication” phase (think of “painting” a picture with an inkjet printer), so that we can concentrate on the phases that more critically depend on what we want, namely editing the blueprint and evaluating the end result.
- **A “verbing” language.** To talk to this new “piece of our mind” that the Knowledge Home will provide we’ll need a conversational language somewhat analogous to a scripting language, but natively designed for a *verbal* rather than textual mode of communication. (All the same, a more formal, written variant of the verbing language will play an important role in the culture). To make possible an agile dialog between human individual and computer instrument, both parties will have to maintain throughout a conversation enough of a model of the “universe of discourse” to resolve implicit references and apply appropriate defaults. This will of course require putting some “artificial intelligence” in the language interpreter within the computer, but—we believe—only a modest amount.

- **Integration of basic tools and modular knowledge resources.** The Knowledge Home will support the individual in structuring and exploiting personal knowledge. Much of this knowledge will take the form of quite humble, user- and domain-specific “models of the world,” put together by the end user as the need arises. But how can such custom model making become effectively accessible to the less technically-minded persons? What will be needed is for the culture to directly provide as well as stimulate a market in *modular knowledge resources*. Then, rather than having to build everything from scratch, the individual will be able to act mostly as an *integrator* of off-the-shelf modular resources, using for this task a variety of generic tools and methods that will be part of the Knowledge Home’s “cultural kit.”

A good metaphor for this personal integration of given modular resources may be a city’s transportation system—subway, buses, taxicabs, pedestrian malls, etc. The works of the whole system is above anyone’s head and certainly beyond one’s control. But, with some overall knowledge of the geography of the city and some knowledge of the “user interface” (train schedules, ticket purchase and use, location of taxi stall, sign reading, etc.), each individual can synthesize out of the same raw resources a personal “virtual transportation system.” A better knowledge of structure and interface will of course allow one to build for oneself a better system.

- **Homemaking and “fire insurance”.** This section will discuss how one would go about creating one’s knowledge home in the first place. We also hint at how to make it portable, protect it from accidents, and share aspects of one’s “home design” with others.

The above cultural elements provide the foundations on which the building will rest. Architectural details are not given here and are expected to emerge only gradually: the intentional development of a culture must entail much interaction between theory and experiment, cognitive science and software engineering, individual psychology and collective sociology, wishes and reality—and thus will be a somewhat open-ended process.

Beside the comfort and satisfaction that come from feeling at home in one’s environment, the proposed culture will give a boost to an individual’s capabilities for handling information and knowledge in the pursuit of *one’s own* goals. It is understood that for many individuals an important

goal will actually be to improve their effectiveness as members of corporate enterprise or other forms of organization; though likely and welcome as a collateral benefit, this is not, however, our primary objective.

2.1 Surface expression vs deep representation

We’ve seen that an individual’s Knowledge Home will be a “mental workshop,” assisted as appropriate by the computer, for structuring personal knowledge according to one’s needs. Among the “artifacts” of this culture we may imagine a personal address book; a one-shot script for retrieving a certain episode from one’s diary; one’s accumulated and distilled “notes to oneself” on tropical fish breeding; a synthetic “group photo” containing all of one’s friends at once; a private Spanish “instructor” who’s familiar with our progress in the language and our strengths and pitfalls; a virtual “old aunt” who will patiently accompany us on the piano, over an over, as we practice a song. Like in a farm or a palaeolithic workshop, we’ll be busy *making* all sorts of concrete and abstract things.

To make a new object, a natural temptation is to directly conjure up the target object’s surface features, which are so obviously present in one’s mind, and let these features guide one’s hand: in the Genesis account, to make man God takes a piece of clay and gives it *human form*. Perhaps the most important lesson of biology and technology alike is that this approach is a dead-end! Effective, understandable, maintainable, and upgradable design requires a *blueprint*; to make any changes on the target product one edits the blueprint—not the target itself—and then lets the fabrication process that leads from blueprint to target, from “genotype” to “phenotype,” run its full course over again.

This approach is generally expensive, and is typically used only when the costs can be amortized over a large production run. In principle, however, even one-of-a-kind artifacts, such as those produced in a prototyping shop, are best made by starting from a blueprint or recipe and iterating the *edit/fabricate/evaluate* loop—rather than by just “whittling down a stick” until it looks right.

Our novel argument is that, for many processes, the computer today can take over at virtually no cost the “fabrication” stage (think of “painting” a picture with an inkjet printer, compiling a Java script, making a world-wide Web search for an incompletely described object, or having one’s home page served day and night on the Web). At this point, frequent iteration of the *edit/fabricate/evaluate* loop becomes affordable *to the individual* even for occasional, one-of-a-kind projects *of no intrinsic commercial value*; what’s more important, she can now devote all her energies to the two stages that more critically depend on what *she* wants, namely editing the blueprint and evaluating the end result.

Perhaps the most profound consequence of this approach is that at the end of this process she will much better *know* what she wanted in the first place; in a sense, she’ll have *discovered* it!

A recipe can be thought of as the “deep representation” of

a cake, from which the “surface expression”—the physical cake itself—emerges after a series of semantic expansions and transformations. If I want a less salty cake I will not wrestle with the surface expression—physically try to remove salt from the baked cake. Instead, I will edit a few words of the *recipe* and then go again through the semantic expansion—the actual kitchen work. A few examples of deep representation/surface expression pairs, together with the processes that lead from one to the other, are illustrated in Table 2.1.

Deep representation	Process	Surface expression
Cake recipe	Mix and bake	Cake
Musical score	Interpretation	Performance
Pattern	Knitting	Sweater
Genotype	Ontogenesis	Phenotype
Body of knowledge	Writing	Book
Microscopic laws	Statist. mechanics	Emergent behavior
Source code	Compilation	Executable code
Text	Text-to-speech	Speech stream
html source	Net browser	Web page
LaTeX document	TeX	Typeset page
Search pattern	Search engine	Search hits
Math formula	Evaluation	Lookup table

Table 2.1: Examples of processes that lead from a deep representation to the corresponding surface expression.

Once we accept that “making” an object really means concentrating most of our effort on designing the genotype (left column of the above table) rather than building the phenotype (right column), we will realize that our primary “tool” will be an *editor* rather than, say, a milling machine. In this context, one can’t overestimate the advantages of a true general-purpose text editor like EMACS[56] over a WYSIWYG “word processor;” the latter may be quite effective as a glorified business typewriter, but becomes a liability—pedagogical and practical—when one’s job is not merely writing but design and “knowledge structuring.”

The discovery of deep representations of natural structures, and, conversely, the design of “machine tools” for fabricating structures corresponding to a given deep representation, have been the main business of science and technology in the last few centuries. But, until recently, this design philosophy used to be accessible only to a technological “priesthood.” As we shall see, the real novelty of the computer is that it can give the man in the street effective access to this mode of operation and to the conceptual vistas which that opens up.

The primary teaching of the Knowledge Home culture is that the individual shall grow alert, almost as a reflex, to the fact that most information structures encountered in ordinary life can usefully be viewed as the surface expression of deeper, more abstract and pregnant representations; and shall become conversant with using that powerful servomechanism, the *edit/fabricate/evaluate* loop, to evolve a desired target object through manipulation of its deep structure. This attitude encourages seeking the “essence” of the information structures at hand so as to make them

more *useful*—legible, compact, easy to search, share, upgrade.

One may wonder why the edit/fabricate/evaluate discipline, which is so vital to the professional, isn't already widely practiced by the general public. Two reasons come to mind. First, the individual may just not be able to afford the enormous investments, running costs, and delays inherent in the *fabricate* stage of the above loop. Second, the layman may be ill at ease with editing at the “genotype” level—which is, after all, a form of *programming*; may lack the attention span to encompass the edit/fabricate/evaluate loop as a whole; or may be unable to manage the practical mechanics of keeping this feedback loop going. We shall respond to these two classes of objections right below.

Historically, it used to be that most of the labor and technical expertise were demanded by the *fabricate* stage. It's materially easy to copy and make some changes to a recipe (*edit*), and I can judge for myself whether the cake tastes good (*evaluate*); but am I really about to knead dough, whip cream, make caramel, etc., and finally clean up the mess? Isn't it simpler to buy a store-made cake?

The good news is that today, when it comes to matters informational, *fabricate* may in effect have come to be the *least* demanding of the three stages. The computer—this universal machine tool[39]—can carry out most “fabrication steps” in an automated way, inexpensively and almost instantaneously. Think of a C++ compiler that turns a page of source code into a living program; of an MP3 player that turns the song layout we just put together into a “demand performance” for our half-hour of jogging; of an XML application that turns a page of XML text into an interactive on-line catalog. When I write a scientific paper using TeX, it often happens that, after altering a single formula or adding a reference, in order to see the overall effect I will have the entire paper typeset afresh. Years ago, this step would have demanded hours of work of a proof editor and days of a typesetter; today, TeX does the job professionally, in a fraction of a second, on my home computer.

To be able to do all this, the computer—it's true—has to be *programmed*. That is, in the first place somebody must design the software engines—say, an industrial-grade C++ compiler or the TeX/MF suite—that pull the above feats. These engines may be extraordinarily complex; they were not programmed by the individual who so casually uses them, and certainly were not “thrown together” just for that specific use. Their development required deep knowledge and all sorts of technical expertise (as well as much teamwork), and easily used up a number of man-hours and an amount of infrastructure far exceeding the lifetime resources of any individual. One of the miracles of human culture is that today anybody can reap the fruits of such a collective effort essentially for free—as any Linux distribution will attest.

The question remains whether such tools for “genotype-to-phenotype conversion,” granted that they may be effective and affordable, will ever be simple and practical enough for the ordinary person to use: once we get such beasts, will we be able to ride them? But this has to do less with

the complexity of a tool or of its functions, and more with proper design, interfacing, documentation, cultural transmission, and, at bottom, utility and motivation (cars are getting more and more complex, and yet easier to drive). We'll have more of this in the present section and in §2.3.

A good analogy for a complex tool with open and open-ended access to the layman is the *piano*. The design, fabrication, and operation of a piano embody centuries of technological development. No knowledge of that is needed for even small children to begin using the instrument and develop musicianship at their own pace and according to their talents. While the instrument is complex and powerful, the calling parameters—that is, to use Bach's words, just “pressing the right key at the right time”—are as simple and transparent as can be; nevertheless, these are not “buttons” that recall canned tunes, but “handles” to abstractions for *notes* (cf. footnote 10). In the musical hierarchy, piano keys are, admittedly, still very close to the level of actual sounds: most of the higher-level processing is still done in the mind of the player. Thus, the piano is a conceptually modest extension of the individual, and yet very empowering one. The conversion from deep representation (score, remembered melody) to surface expression (actual music), carried out no matter how imperfectly by a human with the help of a piano, empowers one—if growth of personal musicianship is what one has in mind—better than listening to hundreds of perfect, “canned” CD performances!

We'll now turn to the second difficulty, that is, to what extent the average individual will be ready to embrace—or even just grasp—the “genotype-to-phenotype” approach to getting things done.

The selling point of much of today's computer technology is that it managed to replace a clumsy command-line interface by an intuitive, graphical, “What-You-See-Is-What-You-Get” interface. Though such an interface is undoubtedly natural for many tasks, exclusive reliance on the WYSIWYG philosophy is, in our view, pedagogically harmful and a cultural dead-end. According to Tognazzini[60],

In the early days of the personal computer we were trying to sell an unproven technology to a skeptical world. We could not depend on people investing weeks or months of self-education in a system they did not yet know would improve their lives. We had to make things easy. However, sometimes, what was easy in that first 20 minutes was not necessarily the right solution for maximum efficiency over the long haul.¹

We need to establish the relevance today of principles that drove the design of the original graphical user interfaces, as embodied in such lists as the Principles of Macintosh Design.² Which of [these principles] represented “training wheels,” and which represented needs, wants, and limitations of ordinary citizens?

Tognazzini's questions are just as relevant today as they were years ago. We see a steady tendency from most quar-

¹Also cf. *In the Beginning . . . Was the Command Line*[58].

²Reference [3] in our bibliography.

ters to patronizingly underplay the potential of the ordinary individual. The claim of a “friendly interface,” the boast that a product “doesn’t even need a manual,” the assurance that the application is 100% menu-driven and “you can do everything with point-and-click”—all these promises are, to a greater or lesser extent, *fraudulent*. There is more honesty in “You can play the piano in five easy lessons!”, which at least acknowledges you’ll have to learn something yourself.

The philosophy of the Knowledge Home is to encourage people to think in terms of adaptable tools rather than of canned applications, and to put more emphasis on the *transformations* than on the *data* being transformed. (Incidentally, the term “verbing” in the verbing language is a pun not only vis-à-vis *scripting* languages, but also as it hints at a more *verb*-oriented than *object*-oriented pedagogy.) Specifically, the envisaged curriculum will cultivate the achievement of regular and confident trade between different *levels of representation*. At any given time, attention will typically move between two levels, and these are what we call, in a relative sense, “surface expression” and “deep representation.”

Through graded tasks and tools we ultimately aim to leave the individual in possession of one fundamental asset, namely awareness of the distinction between the surface expression of a structure—an activity, a document, a body of knowledge, etc.—and its deep representation; between source and target of an assembly, translation, or transformation process; in brief, between “*what-you-see*” and “*where it came from*.”

Besides mere awareness of the distinction between levels, we also want to develop an appreciation for the advantages (already mentioned above) that accrue from doing one’s “building” and “remodeling” work at the deep-representation level rather than directly on the surface expression. Once the culture has made these advantages clear and effectively accessible, there is no reason not to enjoy them even when the object in mind is a prototype or a small production run: once mastered, the edit/fabricate/evaluate loop is no less appropriate to the home context than to industrial-scale production.

A qualitative idea of what all of this will be like can be had by looking at the embodiments of another, older human/computer culture, namely UNIX, with its programs, utilities, languages, applications, conventions, and mystique. It is not our goal, of course, to redo UNIX (or, for that matter, Windows): we want something with much more *internal state*—so that it can grow into an individual’s own companion—and much more feedback and “proprioceptive sense” (too much in UNIX is based on *blind reckoning*). Yet there is much in the UNIX spirit that can be used as inspiration for the knowledge home.

We must note that both the UNIX “composable module” philosophy and the “accessibility of genotype” theme have long found advocacy and solid embodiment in the GNU project[57], indefatigably propelled by Richard Stallman. In addition to a comprehensive suite of software tools and operating system components, such as *emacs*, *gcc*, *bash*, and *make* (which the Linux kernel had the fortune to find ready-made and which it accreted around itself so as to

become the GNU/Linux operating system), GNU has endowed the community with a rationale for software openness and with cultural templates like the General Public License (GPL or “copyleft”) and the *info* documentation discipline.

Our approach, however, should not be confused with an attempt to impose “universal technicianship.” What we want to encourage, and specifically help set standards for, is reasonable universal *accessibility* to deeper levels of representation. Even levels that are not intended to be accessed by the casual user will be documented and open to visual inspection and manual modification (either directly or through an editor that accompanies the documentation). Equally important, a *rationale* for the existence and the format of these lower-level structures will be provided. Likewise, the processes that deal with the different levels of structure and convert from one to the other will be in the form of accessible modules (see §2.3), that one can call separately and run with the data of one’s choice. To this end, a process’s function will be described and motivated, and its invocation procedures clearly and concisely documented, especially by using graded examples; more extensive documentation will not intrude, but will remain available for technical and reference purposes.

It must be noted that high-level functions can still be conveniently pre-packaged for the sake of the first-time user or for frequent, self-contained tasks, and made available through point-and-click at a graphical menu. But, in addition to that, the component modules will be individually accessible to the more mature or inquisitive user, who can put them together in different combinations and call them through a script[43] (see §2.2), perhaps to be carried out at a future time or a remote place without requiring the user’s physical presence.

In the construction and reconstruction process that will be used to give life to the Knowledge Home culture we shall try as far as possible to practice what we preach, openly abiding by *culture regeneration* (or *recompilation*). That is, we make an all-out commitment to (a) document the *rationale* of the choices that are being made and (b) keep track of the web of dependencies between these rationales (cf. the *make* discipline of UNIX). When a major choice change is made, we will make a sweep through the chain of dependencies, retracing the entire culture generation process (most of this, it is hoped, will be done automatically, but some will perforce have to be done by hand)—recompiling programs, regenerating manuals, etc., so as to leave as little trace as possible of the former state (except for historical purposes). We want to avoid burdening the new culture with capricious legacies even before it is born.

It will be important to ascertain how fast and how well the deep-representation vs surface-expression paradigm can be absorbed and productively employed by the layman and used as a basis for further cultural build-up. To obtain some data for this purpose, we are developing a number of *split-screen* applications in which a “genotype” is manipulated in the left half of the screen while the right half immediately displays the corresponding “phenotypic expression.”

A canonical example is a pure text editor on the left and the typeset copy on the right, with a \LaTeX compiler, for example, connecting the two halves in nearly real time. The user edits the text, inserting \LaTeX commands and macros on the left of the screen, and monitors the effects on the right, thus learning and acquiring confidence (by the help of the immediate feedback) with operating directly at the deep-representation level. In an analogous fashion one can edit budget data and see the corresponding changes on a tax return, or edit a musical score (it doesn't have to be in pentagram form) on the left, perhaps adding repeats, da capos, and transpositions, and hear the corresponding developments in the played music. Though separate packages with similar capabilities of course already exist, our aim is to develop a way to do things of this kind

- more casually and extemporaneously, on the one-of-a-kind, lightweight, still unformatted information structures that crop up in one's day; and
- from the vantage point of a unified "mental platform" residing in the individual, rather than using a collection of sundry gadgets purchased from unrelated "mall shops."

2.2 A "verbing" language

In the partnership we envisage between human individual and computer, the latter is better viewed as a pragmatic and unpretentious extension of the capabilities of the former, much like a piano to the pianist or a dog to the hunter—rather than an external "artificial intelligence" genie. For the dialogue that will take place between a human individual and his/her knowledge-home "extension" we plan to develop a **verbing** language. Functionally, this will be somewhat analogous to a *scripting* language: interactive, general-purpose, flexible and robust rather than efficient at any cost, suitable to act as a conversational shell for more specialized packages. However, the verbing language will be designed from the start as a *spoken* medium, capable of maintaining and referring to a substantial implicit "world of discourse;" thus, in terms of internal structure, it will resemble more a remorselessly stripped-down version of *natural* language.

A more formal, text-oriented dialect of the verbing language will be used for batch-mode directives and other non-interactive tasks. Though the oral mode of the verbing language will have conceptual primacy in the Knowledge Home, the written mode will be cultivated (as it happens with English) as the instrument of choice for more deliberate information-structuring activities, as we shall see below.

Until recently, the job that in popular culture would most naturally be associated with the computer was "programmer." Programming tends to be viewed as a necessary but technically demanding and not particularly pleasant task—something one gladly delegates to others. At the other extreme of the spectrum there is the "user," for whom the computer is basically a virtual appliance: by running different *applications* the computer may be turned at will into a video game, a word processor, a cash register. Modern

graphic-interface wizardry relieves the user of even the last semblance of programming.

In the knowledge home, the computer will be viewed not as much as an external appliance as an extension of *our mind*. We know how to talk to ourselves; how will we talk to this new "piece of our mind"? Will we be able to think in point-and-click language? Shall we all learn to speak Java? The design of a verbing language addresses these concerns: the language will be used by ordinary individuals for verbal communication with the computer at the level required for the construction, maintenance, and enjoyment of their knowledge home; in other words, for talking in a simple fashion with their extended selves.

Actually, before debating what language one should speak one would have to first ask *who* it is that one is speaking with. What kind of "animal" will our information home be and what will be its view of the world? This immediately raises the issue of knowledge representation as from within a knowledge home and, in short, of *machine intelligence*. We shall deal with this issue in more detail in §3. For the moment, we may note that, except for the shallowest applications, natural language understanding by computer requires access to a *semantic model* of the world of discourse[12].³ Constructing and maintaining an adequate model of an external system (say, of human physiology for medical diagnostics) may be an immense task—even after drastic simplification and much arbitrary filtering. In the case of the verbing language, however, we don't have to resign ourselves to an expensive and unfaithful model of the outside world; our primary world of discourse is the very contents of the knowledge home, which is *by construction* totally and directly accessible to the computer. Given these unique circumstances, a verbing language for the knowledge home can start supporting a perhaps limited, but deep and sincere, dialogue between human and computer without requiring an enormous up-front investment in artificial-intelligence machinery.

Scripting. The term 'verbing' (short for 'verbalizing') suggests an analogy with scripting languages, but reveals a different emphasis and scope. As we said, the verbing language is conceived from the beginning as a *conversational language*, and will be used by the computer as well to talk back to the human. In this interactive mode, the language will heavily rely on semantics. Within the same language, however, one will shift into a more formal "scripting dialect" for one-way, written, batch-mode communication, which has substantially different requirements.

Though the verbal, interactive component of the verbing language retains methodological priority, the pedagogical value of the scripting mode cannot be overemphasized (cf. [42]). A script captures the results of a mental effort so that we don't have to repeat that effort every time we want a complex action done; at the same time, unlike a keyboard macro or a recording, a script is open for inspection, recall, and modification. Unlike a sequence of point-and-click incantations, a script has a syntactical and semantic struc-

³This is not to deny the overwhelming role that statistical models may play in the early stages of natural language processing[15], and the inextricable contribution of syntax to semantics.

ture similar to natural language. (Scoping, data-typing, and dependencies are appropriate for an interpretive language: temporary structures are kept simple, are built as the need arises, and discarded after use.) The moment a sequence of commands becomes routine, it is encapsulated, given a name, and becomes just one new command; use of modifiers (do this *in an hour*) or iterators (do this *every weekday*) greatly enlarges the scope of a base command form.

Scripts are a vehicle eminently suited for *cultural transmission*, as they provide a structured way not only to specify an action but also to *document* it, i.e., reveal the aspects of *intention* and *design* behind the action. Scripts permit a short design/test/modify cycle, and are much more tolerant of concatenation, deletion, mutation and recombination than conventional programs.⁴ Moreover, a scripting language lends itself well to a gradual transition from keyboard/mouse/screen input/output to *verbal dialog*, which we expect will become more and more common in the future in dealing with computers.

In practice, an interim version of the scripting dialect would need to be developed first, as a communication medium for immediate use and to provide support for the other components of the cultural program. The development of a full verbing language will continue for many years: it is an ambitious undertaking and will require combining a wide range of expertise with substantial new research.

Besides programming languages more suited for the needs of large critical project and large organizations, there are a number of languages that attempt to provide a more immediate and individual relationship between human and computer and were motivated by innovative pedagogical approaches; we may mention LOGO[1], FORTH[7], and SCHEME[55]. These all demand a higher level of personal technical commitment than we envisage. Attempts to dumb down the act of programming may of course backfire; according to Dijkstra, “it is practically impossible to teach good programming style to students that have had prior exposure to BASIC: as potential programmers they are mentally mutilated beyond hope of regeneration.”[20]

A scripting language like PERL may be very handy for computer professionals steeped in the C-shell culture, but is totally mystifying for the non-adept. Applescript[4] has aspects worth imitating, such as the discipline it enforces on application program interface, but has too narrow a scope for our purposes. A philosophy more germane to ours can be found in Tcl or SmallTalk. Perhaps what comes closest to our motivations is Boxer[22]; though in that project the stress is more on visual than verbal aptitude, much of the underlying philosophy is directly applicable to our concerns. Pragmatically, the most relevant vehicle for the first stage of our journey may turn out to be the all-round language *Python*, whose developer shares many of our objectives[61],

⁴In the our verbing/scripting/programming distinction, a *program*, in the usual sense of the term, is the software counterpart of a piece of complex, high-precision machinery whose demands for compactness, efficiency, and reliability justify a large technical investment and long design latency. In the same machine-shop metaphor, a *script* is more like a prototype, and a *verbing exchange* is more like a use-once jig.

and which might provide the starting point for the scripting dialect (cf. above).

But, in essence, all of the above are *computer* languages; what we are after is a graded range of computer-oriented specializations of *human* language. This will require of the ordinary human a substantial cultural investment in appreciating the resources and limitations of the computer ecology, but not the acquisition of the language habits of a different “species.”

2.3 Basic tools and modular resources

Knowledge can be defined as any kind of successful adaptation (from the knowledge owner’s viewpoint[49]) to certain aspects of the world. Ultimately, the Knowledge Home will succeed only if it will allow one to develop more useful *models* of what one cares about in the world. We envisage quite humble, user- and domain-specific “models of the world,” put together by the end user as the need arises. For this, besides a number of basic tools and skills in possession of the individual, we anticipate the availability of suitable prefabricated components.

Modular knowledge resources. Making a new “primary” model (a street map of Boston, an adventure movie, a scientific theory) requires competence, dedication, and resources that are beyond the reach of the average individual. But one can buy a map, learn a theory, and watch a movie made by others. In the milieu of culture, the individual will mostly use models of general availability, adapting them to personal circumstances and combining them in a custom fashion into more complex models.

Thence, from our viewpoint, the ordinary individual will mostly play the role of *model integrator* rather than of *primary-model maker*.⁵ In Engelbart’s words, “one would more often be engaged in structuring a composite process than in structuring a computer process.”[23, p. 97] Rather than having to build everything from scratch, the individual will be able to act mostly as an integrator of off-the-shelf modular resources. It is in this context that much useful work can be performed from the verbing language vantage point, without requiring the more specialized craft of a heavy-weight programming language.

As far as the development of the Knowledge Home culture is concerned, we stress that the only stock models that are effectively usable by the lay model integrator are those that can *readily* and *reliably* be combined with other models. Design issues include proper task segmentation,⁶ interfacing, documentation, validity range, correctness of algorithms, reliability of the source data, etc. Much as in LEGO pieces, both the capability to be systematically composed and the potential for variety and usefulness of the composite structures depend on thoughtful design of the “combinatorics” of the pieces. Much as LEGO parts do not

⁵Of course, a particularly skilled integration of a number of primary models may become to all purposes a new primary model; think, for example, of the *MATHEMATICA* package[65].

⁶E.g., will a report be produced as a formatted page only fit to be printed, or in some intermediate, structure-tagged form suitable for input to other programs?

combine with erector-set parts, or CMOS gates with ECL gates, stock models for use in the knowledge home will have to be conceived as parts of “logic families” obeying a definite discipline. It is in this sense that we speak of *modular resources*.

Modularity, “object orientedness,” and standardization are, of course, a primary concern of modern software design, but are mostly implemented so as to meet corporate needs and resources rather than match the goals and capabilities of the individual. Note, however, that, in the commercial world, distribution channels and sources of revenue do evolve. Google Earth would be just an expensive “toy for boys” if all one could do were scroll and zoom on a fancy globe through a computer-game like dashboard. But Google betted on people wanting to do who knows what kind of smart things once they were given access to Earth’s database, and so provided a full API (Application Programmers Interface) to it. Through this modular “software connector,” now a computer can be put “behind the wheel,” so to speak, and scripted to rummage through the Google Earth treasure house in pursuit of much more complex knowledge quests.

Initially, an effort will have to be made to retrofit pre-existing resources (e.g., a GPS interactive atlas) with a knowledge-home interface (basically, an interface to the verbing language). As the culture we envisage takes hold, it will become advantageous for third-party developers of primary models to provide versions of their products in which the integration with the knowledge home is built-in. Of course, some effort will have to be devoted to creating original models and modeling tools specifically targeted at the knowledge-home environment. Obviously, in all this one should keep in mind leveraging relevant existing standards.⁷

Basic tools. While, from the standpoint of the individual, the above modular models will typically come from outside, the means used to integrate these resources into custom constructions of a certain permanence will come from *within*, i.e., will be part of one’s acquired (well-ingrained or at least readily refreshable) skills in handling the implements of the received cultural toolkit. (Here we shall not make a sharp distinction between basic skills as internal and tools as external, since many tools are mental or conceptual—such as, for instance, the ideas of *recursion* or of *feedback loop*.)

Beside deliberate projects, basic tools will also be used daily to casually produce, as the need arises, less ambitious, disposable “knowledge jigs”—for instance, a substitution pattern to be used in a specific search-and-replace renaming spree. In brief, cultural implements will be used on one hand to materially make small contrivances in-house, and on the other to assemble larger projects by providing the connective tissue between “organs”—while the organs themselves may come from external “organ banks.” Products like those of the GNU culture will play an obvious—and vital—role in this context.

In the Knowledge Home, the acquisition of basic tools

⁷Such as the Universal Modeling Language[25] and a variety of Structured Markup Languages[54]—chiefly the XML lineage.

is where most direct engagement will be required of the individual—you cannot learn to ride a bicycle “in theory” or by correspondence! Correspondingly, a major component of the Knowledge Home project will be the development of a *curriculum* and all the instruments needed to implement it: what concepts should be mastered and at what level, what practical, skills and automatic responses should be acquired; syllabuses, primers, teaching notes and aids, graded sets of problems, tutoring strategies, proficiency evaluation criteria, etc.

Useful as all this apparatus may be, one has to remember that, anthropologically speaking, the natural way for humans to learn everyday skills is through *imitation*—through seeing other people “do it.” The learning process may be enhanced by deliberate teaching, but even this is often best done simply *by example*.

Personal transmission of skills will be encouraged in the Knowledge Home. It will be important to build a tradition of progressive examples, useful analogies, and conceptual simplifications which knowledge-home owners (cf. §2.4), in their role as tutors, will more easily employ by having themselves learned through these devices.

It is impossible to stress enough the importance of proper documentation at all stages of creation, dissemination, and operation of a culture. Cultural transmission is much more effective if raw directives are accompanied by their *rational*.⁸ I may say, “To call my cousin in Rome you just dial 0113906818283,” but I will make the task easier to comprehend and remember (and to debug as well) if I explain, and at the same time document, “011 gets you out of the United States, 39 lands you in Italy, 06 is for Rome, and 818-283 is my cousin’s home.”

By having their attention called to them—especially through the practice of scripts—the user will naturally be initiated to a number of concepts and tools that belong to a primary layer. For example, much power comes from promoting from an intuitive level to a somewhat more intentional status certain *name/usage* (or “meta”) mechanisms—namely, what we technically call *quoting*, *escaping*, *redirection*, and *piping*; here one may also mention *recursion*, *scoping*, *encapsulation* or “information hiding,” and *data typing*. Mechanisms of this nature are already lavishly used by natural language—which we all competently use without conscious effort. And don’t we all “push the stack” when we answer a phone call while we are cooking a meal, and know how to return to the suspended task? We may easily get lost, it’s true, if the recursion goes deeper than three or four levels, but that’s a general limitation of our short-term memory⁹—not an indication that our native grasp of recursion is poor. In matters like that, computer help that is modest but well-integrated with ourselves can enormously extend our reach.

At a slightly higher, more cultivated level will be the acquisition of basic tools that somewhat stretch our native intuition and thus involve a modicum of discipline to be

⁸True or alleged—the difference is not so important for mnemonic purposes.

⁹Remember, from Psychology 101, the famous “seven plus or minus two items”?

mastered. Such may be, for example, the use of *regular expressions*, invaluable for searches, enumerations, productions, and substitutions. (The poor man’s version of this, namely, wildcard characters, can be picked up with no effort; as for full-fledged regular expressions, a plainer syntax than the ones tolerated by computer hacks will of course make them much more palatable.) The verbing language itself will provide uniform conventions for arguments, modifiers, comments, etc. by regular linguistic means, much as ordinary language does for direct and indirect objects, adverbs, conjunctions, and parenthetical expressions.

The UNIX environment must be credited with introducing important prototypes of basic tools, and also has a tradition of readily available, if not always friendly, documentation (cf. [2]). However, UNIX tools evolved in a more primitive computer world (where, for instance, conciseness was prized over readability), and to be used well require the skills and the mentality of a “guild craftsman.” A systematic, critical re-spin of reference-level tools and conventions will be necessary to create something that the ordinary user can thoroughly assimilate and use as a solid anchor. We will proceed concurrently in a top-down mode and a bottom-up mode. The top-down component begins with the high-level design of structures, standards, and concept-oriented documentation, while the bottom-up component involves the construction, adaptation, and evaluation of various software tools.

The linguistic/computational environment we envisage will provide a number of default *algorithms and data structures* (such as dictionaries, stacks, and lists; hashing mechanisms; link processing; customizable sources of randomness) for implementing those objects and actions that will be part of the most basic knowledge-home ménage and corresponding lexicon. In this process of language construction we shall seek feedback from experimental subjects: what ordinary individuals typically wish to do, can do, and can be taught to do with basic tools in matters of managing basic items of personal knowledge—professional or amateur interests, personal endeavors, one’s social web, etc.

In addition to producing original *documentation*, we plan on restructuring much existing documentation to better fit the concept-oriented nature of the computer culture we are developing. In the spirit of the genotype/phenotype complementarity mentioned in §2.1 and the idea of scripting as remote extension of oneself, documentation will be developed that is not only “machine readable” in a trivial sense, but suitable for machine access at a reasonably deep semantic level. From this level, surface-level documentation and other maintenance-oriented material appropriate to the needs of the moment will be generated on demand. Experiments such as GNU Texinfo and Knuth’s “literate programming” [34] will provide inspiration, but clearly we’ll have to go much further than that.

When all is said, the Knowledge Home’s basic tools will essentially be those of general *computational literacy* (cf. right before §1.1) grafted on existing linguistic competence and all that is behind it. Note that this is no repackaging of an old concept under a new name. General computational

literacy as a transmissible cultural package *does not exist yet!* The task that we set ourselves is to design one, and a *viable* one to boot.

2.4 Homemaking and “fire insurance”

Here we address some issues concerning ownership of a knowledge home. Let ‘owner’ denote the individual who enjoys the use of and manages this expanded personal information space; by convention, the owner will be a *she*; for definiteness, she’ll be a college student.

A snapshot of the envisaged knowledge home would show just an ordinary-looking computer. One may visualize this as processor, disk space, and an Internet connection, but may actually be some other computational support, as explained in §1.4. The difference is not in the material, but in its use.

If we looked at the knowledge home from the viewpoint of an information systems manager, we’d find the usual hierarchical structure with directories, data files and program files. But try to ask its owner to describe its contents from her vantage point: She will turn out to be *familiar* with its functional organization, in the same sense as she’s familiar with the contents of her house even though she won’t know or care about the internals of the microwave oven. She’ll be able to tell us what the different things she sees are *for*. She will know the *names* of most things she employs,¹⁰ some names having come with an object when she acquired it or with a tool when she learned to use it, and other names having been given by her to the things she made herself.

“You must have given much time and love to making your knowledge home a space fit for you, and you certainly have many precious things in it. Aren’t you afraid that a fire or a disk crash will take it all down?”

“That would be a nuisance,” our user will answer, “but no tragedy. You see, I have a *script* that describes what my knowledge home should be like, what it should contain and where. A routine program keeps track of minor changes or additions, but whenever I entertain major renovations I go over the script myself (I got my first home script from a friend) and plan the changes by *editing the script*;¹¹ then I try out the changes (just running the scripts builds me a brand-new knowledge home in a few minutes) until I’m comfortable that the new home is what I want. If I don’t like it, I can always go back to the old one. My physical home may burn down, but my knowledge home will be up and running again in a few minutes—maybe in my friend’s computer until I have a new one of my own.”

“But your script is only a few pages long, and you have a hundred terabytes in your hard disk,” we observe. “Where does all that information come from, when you have to rebuild your knowledge home from scratch? And what kind of wizard do you have to be to insure that everything will come together as it should?”

¹⁰ “Naming is perhaps the most powerful abstracting notion we have, in any [natural or computer] language.” [55]

¹¹ Again, editing the genotype in order to modify the phenotype; cf. §2.1.

“As far as I’m concerned, there are basically only two kinds of objects—the ones I made myself and those I got from outside. For the latter, which constitute most of the bulk, my script has pointers to the sources and can download materials and automated assembly instructions from there; I don’t have to know the details.

The objects that I made myself are mostly texts, drawings, scripts, and shop models which came directly from my hand and are literally the sweat of my brow. As in the case of the primary script that describes my knowledge home’s contents, this is literally *small change*—how many megabytes do you think I can personally generate in my entire life?¹²—of which I keep copies in the usual safe places. There are, in addition, some irreplaceable personal data of greater bulk, like family photos, home movies, hard-to-find books;¹³ I find it more convenient to keep a backup copy of these with a professional data-vault provider. In sum, even if the information I care for is much too large for me to know in detail (as if I ever wanted!), I know what my possessions are and I can give directions on how to restore them if lost.”

The above dialog throws light on our use of the terms ‘personal’ and ‘structuring’ as related to knowledge. One can rely on the professional expertise of movers, plumbers, and decorators, but the basic plan and management of the knowledge home are the natural sphere of interest of the owner.

¹²Even a computaholic like Steven Wolfram claims only 100 million keystrokes[10].

¹³A “personal digital library” may of course be a much more structured collection than an “attic trunk”[37]. But it will still consist of digital *objects* that are materially owned by the homemaker, of *links* to relevant external material (one may imagine one’s personalized catalog to the city’s public library), and *descriptive material* that provides a “guided tour” of portions of the library. In any case, the irreplaceable components of such a structure boil down to low-volume scripts and texts and some amount of bulky personal data.

Chapter 3

Designing the knowledge home

Of the many forms of false culture, a premature converse with abstractions is perhaps the most likely to prove fatal to the growth of a masculine vigor of the intellect. [George Boole]

All I Really Need to Know I Learned in Kindergarten.
[Robert Fulghum]

[This chapter will discuss some design aspects of the proposed culture—what can be expected and/or furthered both in human nature and in terms of technological prerequisites. Most of the chapter still has to be written.]

3.1 Leveraging innate mathematical capabilities

As we’ve noted, literacy builds upon, and expands in ways that it would have been hard to foresee at its beginning, our ordinary, innate linguistic capabilities, our mnemonic and pattern-recognizing talents, and other general knowledge-integrating capabilities.

Note that it is clerks of ordinary bureaucracies—not great scientists or writers—who comfortably produce and use contract specifications with, say, seven levels of itemization. Literacy is no longer just a matter of putting ordinary speech onto paper and vice versa (cf. Jong’s classic monograph[42])—some sort of verbatim “tape recorder”! A contract can be drafted, examined and revised at leisure without the real-time pressure of oral delivery, compared with other contracts, circulated, reviewed by collaborators, etc. The resources of the cultural medium make it possible for quite ordinary individuals to leverage the evolutionary “old” mechanisms of language, together with innate capabilities for problem solving, design, and culturally assisted tool development, into novel kinds of production.

Besides language, all humans display on their day-to-day routine a wealth of talents of a logic, mathematical, geometric, or dynamical nature. They may be convinced that “linear algebra”, “homomorphisms”, or “Kirchhoff’s theorem” (if they ever heard of them) belong to the nebulous jargon of science, yet they competently use mental reflexes that are the prototypes of those very scientific concepts: namely, they expect *proportional response* to a stimulus, they sooner catch on to an idea via a *metaphor* than through a long literal explanation, and they readily conjecture appropriate *conservation laws*: even a child of three is surprised when a ball that rolls behind a box does

not soon *reappear* on the other side. It’s hard to overestimate the amount of tacit mathematical and computational capabilities we are all *perforce* endowed with (cf. [19]). Systematic observations made by Piaget and his school[45] suggests that these rather general organizing capabilities of the human mind have a functional role somewhat analogous to that of *category theory* in mathematics.

One feature of our approach to crafting the cultural suite discussed here is the direct leveraging of universal proto-scientific capabilities into pragmatic resources to be used in building and maintaining one’s knowledge home, without going through the entire process of *first abstracting all the way up and then concretizing all the way down*—which is what is typically done (or attempted) in much conventional scientific training (see diSessa’s extensive material[21] in this regard).

This does not mean that people of the Knowledge Home culture will be expected, by some sort of magic, to spontaneously fire up whatever innate talents they have in, say, handling recursion, and from there invent all by themselves a toy version of the LISP programming language! What we mean (see, e.g., Resnick[52]) is that

- It is part of our experimental agenda to determine at what level innate computational faculties (that one might a posteriori label “intuitive mathematics,” “physics,” “game-theory,” etc.) can be supported and boosted, for every-day modeling purposes, by a well matched “mental shop.”
- Tutorials, examples, tools, and documentation will be designed so as to elicit and reinforce this competence, and exploit it for the building and maintenance of one’s knowledge home.

Note that the issue of what proto-scientific talents humans do or do not possess must be approached, for our purposes, as an empirical matter. For example, it appears that, in contrast with superb language talents and visual/spatial processing, our mind is poorly equipped (or perhaps just too impatient) to handle probability questions and complex inference chains[46, 31, 44, 5]. As a consequence, inference tasks in the management of the knowledge home will have to be supported by a different strategy than, say, syntactical expansion.

3.2 What shall the knowledge home know?

Conversely, what basic innate capabilities shall be built into the knowledge home?

Earlier we mentioned the verbing language (cf. §2.2), which will be the most visible expression of the knowledge home. The surface features of this language may be an interesting exercise in computational linguistics, but the real design must begin by sketching an idea of what sort of entity shall dwell *beneath* the language itself. What will this entity be like, that we'll be "extending our mind with and talking to"? What kind and what level of "animal intelligence" shall flicker inside the knowledge home?

Questions related to this have been asked since the beginning of artificial intelligence studies and even earlier—by Thomas Aquinas, for one. I favor the approach of Hofstadter (e.g., [29]) and Dennett (e.g., [18]), which is more concerned with genuine, deep-seated capabilities than with here-and-now deliverables.

The semantic machinery underlying the verbing language must itself display some innate competence comparable in kind if not in degree with that of humans. For instance, a command to print a file will not be interpreted merely as a terse description of an action to be taken ("send byte stream to printer port"), but also, and primarily, as the *intention* (and corresponding *expectation*) to have a certain goal achieved. If no ink lands on the paper after the specified action is taken, the command will not be deemed fulfilled and further measures (such as retries, remedial or alternative actions, warnings) will be attempted by the system in order to meet the *expectations*.

Competencies of this kind are of course being engineered in any well-designed adaptive system—be it a VCR or a fighter plane. The point we are making here is that they should not have to be reinvented for each new application, but should already appear as generic semantic properties of the verbing language. In that language, "Do x !" should by default signify "Do whatever is in your power to get x achieved!"

How hard will it be to put this kind of generalized competence in the knowledge home?

In designing the human mind, evolution must have proceeded in an empirical, incremental, *exadaptive* fashion[28], building gradually out of whatever came in handy at the moment. We are more in a hurry than that! Fortunately, our design target—the knowledge home—is by far a much more modest affair than the human mind; besides, cultural design has the advantage over natural evolution that it proceeds in an *intentional* fashion—literally, by "intelligent design"—and thus may attain its goal over a straighter and shorter course. In particular, the bottom-up, epigenetic approach can be complemented by a top-down approach, where abstractions are designed in from the outset.

To this purpose, we think it will be advantageous to incorporate the basic elements of *category theory*—mathematics' most abstract "shell"—directly into the very fabric of the knowledge home.

In brief, while a specific branch of mathematics may in-

quire as to "Who did what to whom" (e.g., "Which elements of an action group map certain points into themselves?"), category theory is content with more general knowledge such as "Someone did something to somebody," that is, that a certain type of operator (no matter what the specific operator is) has two parameters, the first dealing with an action (perhaps of some specified kind) and, likewise, the second dealing with an actee. If the operator is called with a different number or kind of parameters, this high-level error must be addressed *before* worrying whether, for instance, the named action is indeed known or available to the system.

Seen from this perspective, Piaget's lesson[45] is that it must have been evolutionarily beneficial for the human mind to grow comfortable with some basic categorical abstractions. Who doesn't readily appreciate the meaning—and the *usefulness*—of a phrase such as "That was a *who-did-what-to-whom* question"? Likewise, we think that the knowledge home (here meant not as the culture, but as the hardware/software embodiment of one's personal knowledge-structuring support) should be granted similar benefits by virtue of design.

We have seen that ordinary literacy, by providing material and cultural props, allows one to extend the power of natural language. In a similar way, computational literacy such as that supported by the knowledge home will allow one to extend the power of natural computational aptitudes. If the knowledge home comes "wired" with somewhat *more* (rather than less) and more dependable category-theory structure[62] than we are innately competent with, with its help we'll be able to amplify our level of competence in this respect. Note that, like Moliere's uncultivated character who "spoke prose without knowing it," most inhabitants of a knowledge home will not have to become competent in the *theory* of category theory—or even be aware of their progress in it. All that matters is that, with the help of the provided instruments and examples, they will playing "by ear" some better "music" for their own enjoyment, much as MS-DOS users make effective use of regular-expression abstractions when they type wild cards (the stars in a command like `del budget199*.*`), without even having to *know* that they are making a regular-expression search.

3.3 ...and what know to forget?

A knowledge home will always be at risk of becoming a "knowledge attic," full of memories that used to be somebody's fond ones but now are no longer claimed by anyone. The same culture that will help us build a knowledge home (see §2.4) will also provide traditions and tools that will encourage us to keep it clean.

To begin with, most documents and scripts, unless actively recalled, will by default be made to gradually "fade" (they may acquire a yellowish tinge when displayed, they may be given a lower precedence in searches, etc.) and eventually disappear from sight—though not necessarily from the system. But that will not be enough. Sifting, sorting, consolidating, throwing away, though not glamorous endeavors, are important *creative* efforts that require dis-

cipline, knowledge, personal choice, motivation—all things that the so-called “personal computing” market wouldn’t touch with a ten-foot pole, and which instead the Knowledge Home culture will not be afraid to abet.

Equally important, how can we design a knowledge home to be culturally *adaptable* and yet *faithful* to its owner? A puppy may equally well be grown into a policeman’s dog as a burglar’s one, but after that neither owner would want his dog to switch sides. Similarly, I will want my knowledge home to be “smart enough” to know how to adapt to me, and at the same time “dumb enough” to remain attached to *me!* I want it to be usefully interconnected to the Internet, but not to be so strongly influenced by it as to slowly morph into a house I can no longer recognize. We may want to put in, by design, what ethologists call an *imprinting* period, after which our knowledge home will shed much of its (by then no longer needed) adaptability.

Chapter 4

Conclusions

[Also this final chapter is under construction and may undergo major changes.]

4.1 Survival value

The farmer prays to Zeus for rain, the potter for dry days.

Here we address the personal survival value of the knowledge home. *Relevance* of a piece of information, *usefulness* of a model, etc. are subject-relative. My knowledge is tied to *my* successful adaptation to *my* environment. Similarly, my supermarket’s database serves the supermarket in *its* environment. As a rule, we as individuals cannot expect that organizations, social groups, and, in general, any structures that have to “make a living” in order to persist[17], will primarily care about our own individual well-being—though they may happen to do so when their success is correlated with ours.

Our point is that, at bottom, *the ultimate custodian and judge of an individual’s well-being can only be the individual itself*. If good models of the world are important to the individual’s adaptive response to the world, it can only be up to the individual to make sure that the right models get put together. The cultural tools—the “discipline”—we plan to develop to help individuals gather, integrate, and deploy knowledge for personal use start from the assumption that the individual will *want* to acquire *some* competence (cf. [59]) in the use of these tools. To ask to what extent this want may come from inner desire, from societal inducement, or from economic need (why do people “want” to go to college?) is not our primary business here. Our business is to make sure that the means are consistent with the objective. If the goals are enhanced individual adaptiveness and personal satisfaction, the solution cannot be merely *technological*—a package that one buys at the computer store. The “package” that we propose does envisage computer tools and standards (which will mostly be public-domain) and computer applications (many of which may well be commercial), but includes as an essential part the development of individual competence in the “feed and care” of one’s models of the world. The latter is an *educational* endeavor.

4.2 Do we have to die 100%?

Non omnis moriar, multaque pars mei vitabit Libitinam. (I shall not entirely die; in fact, a large part of me will avoid the grave.) [Horace, Odes III-xxx, 6]

Here (the section title, borrowed from [11], is clearly a paraphrase of Horace parting statement) we address the issue of “sharable ontologies:” sharing and bestowing one’s models of the world at a deeper level than the surface representation provided, say, by a book.

We were touched by Donald Knuth’s home page (www-cs-faculty.stanford.edu/~knuth). Here is a man who has an enormous wealth of organized knowledge to share—and most deliberately started doing this sharing thirty years ago with the first volume of *The Art of Computer Programming*[33]. On his way, he realized he needed a deeper-level interface between his manuscript pages and the printer, and he devoted several years to creating T_EX, METAFONT, and WEB—three companion computing environments that “know” a lot about the printer’s craft, the mathematician’s notational magic, the art of typeface making, and the need for a program to communicate with other humans as well as with computers[35]. *Note that this kind of knowledge cannot be captured by a passive book; it requires an active structure that, though created by humans, lives inside a computer.*

Knuth then went back to writing on “a hermit’s schedule”. “I retired early,” he says, “because I realized that I would need about 20 years of full-time work to complete my main life’s work, *The Art of Computer Programming*.” In a tongue-in-cheek wager with the devil, Knuth estimates that his fifth volume will be ready in 2009; “and after Volumes 1–5 are done, God willing, I plan to publish Volume 6 (the theory of context-free languages) and Volume 7 (compiler techniques).”

Knuth knows what “personal knowledge structuring” means, and he knows it with a vengeance. He would like to transmit to us as deep a representation as possible of his internal journey—not just a travelogue—so that we can re-live and extend his explorations. Unfortunately, adequate cultural tools for externalizing, transmitting, and then re-internalizing in another person one’s knowledge are not developed yet. An ideal person to contribute to the design of these tools would of course be Donald Knuth himself, but he has already “given” beyond the line of duty; with the rest of his allotted time he cannot both give us his knowledge (as best one can do this with today’s tools) *and* develop the right tools for knowledge transmission.

Literacy allows us to reach farther in space and time. An accidental spin-off of this historically primary purpose is that an off-line “text file” supports deeper structure and richer interaction than on-line speech. Thus, born out of the need to better communicate with others, literacy allows us to better communicate with *ourselves*.

A knowledge home would allow us to co-opt external resources “better to think with” in our daily undertakings. The artifacts of this culture—scripts and other “thought files” one may have developed on one’s “platform”—will have enough commonality of format to at least “run” (if not necessarily to make much sense) on the platform of anyone who shares the same cultural conventions.

In this way, a culture primarily devised for thought support will also provide a vehicle for “thought transmission” that goes beyond the traditional literacy means.

4.3 Ownership and satisfaction

We are incessantly solicited to “easily expand our powers” by becoming ourselves part and extension of some other organism’s body—a company, a party, a nation, a church, a frequent fliers’ club, an Internet charter group like AOL or MSN. This, of course, has been going on in nature since time immemorial: there are great advantages to being a cell of multicelled organism rather than a freelance unicellular microorganism, or being a dependable and well-rewarded pollinator in symbiosis with a matching plant species. What we are proposing here is a way to expand our powers by extending *our own* body, as it were, thus retaining a major ownership share of the resulting system. This approach may require a little more effort, but it gives—at least to me—much more *satisfaction*.

4.4 A community of learning volunteers

The development of the Knowledge Home culture will require vision and common sense. It will need sound judgment in social, economic, and technological matters. It will have to be solidly grounded in human psychology, computer science, and evolutionary sociology—and will have to live with the realities of the market.

Most of all, especially at the early stages, it will require a dedicated community of learning volunteers who’ll use themselves as experimental subjects; who’ll inoculate themselves with test strains to determine their potency, side effects, and synergy with other strains; who will design, construct, evaluate—and tear down—countless buildings; who will have the patience to imagine living in a completed knowledge home when only few of its rooms and facilities are yet in place.

We are reminded of the story of Eliezer Ben-Yehuda, who (be he judged a saint or a tyrant) achieved the “linguistically impossible” feat of reviving a dead language—Hebrew. Over many years, while the arch was being built, he, his family, and his close associates supported the arch stones with their bodies until the keystone was in place and the arch could stand by itself.

Bibliography

- [1] ABELSON, Harold, *Turtle Geometry: The Computer As a Medium for Exploring Mathematics*, MIT 1981.
- [2] ABRAHAMSON, Paul, and Bruce LARSON, *UNIX for the Impatient*, 2nd edition, Addison–Wesley 1996.
- [3] APPLE COMPUTER, Inc., *The Apple Human Interface Guidelines*, Addison–Wesley 1986.
- [4] APPLE COMPUTER, Inc., *Applescript Language Guide*, Addison–Wesley 1994.
- [5] BENNETT, Deborah, *Randomness*, Harvard 1998.
- [6] BLACKMORE, Susan, and Richard DAWKINS, *The Meme Machine*, Oxford 2000.
- [7] BRODIE, Leo, *Starting FORTH*, Prentice–Hall 1987.
- [8] CERF, Vinton, “An interview with Vinton G Cerf,” *IEEE Spectrum* **37**:1 (Jan. 2000), 43–44.
- [9] CHOMSKY, Noam, *Language and Mind*, Harcourt Brace Jovanovich 1972.
- [10] CHOWN, Marcus, “Principia Mathematica III,” *New Scientist* **171**:2305 (25 August 2001), 44–47.
- [11] COCHRANE, Peter, “Do we have to die 100%?” *Electronic Design* (14 October 1996), 64.
- [12] COLE, Ronald, et al., *Survey of the State of the Art in Human Language Technology*, Cambridge 1996, 279–283.
- [13] DERTOUZOS, Michael, *The Unfinished Revolution: Human-centered computers and what they can do for us*, Harperbusiness 2001.
- [14] DIBONA, Chris, Mark STONE, and Sam OCKMAN (editors), and Chris DIBONA, Brian BEHLENDORF, Scott BRADNER, Jim HAMERLY, Kirk MCKUSICK, Tim O’REILLY, Tom PAQUIN, Bruce PERENS, Eric RAYMOND, Richard STALLMAN, Michael TIEMANN, Linus TORVALDS, Paul VIXIE, Larry WALL, and Bob YOUNG, *Open Sources: Voices from the Open Source Revolution*, O’Reilly 1999.
- [15] COMERFORD, Richard, John MAKHOUL, and Rick SCHWARTZ, “The voice of the computer is heard in the land (and listens too),” *IEEE Spectrum* **34**:12 (Dec. 1997), 42–47.
- [16] DAHL, O–J, EW DIJKSTRA, and CAR HOARE, *Structured Programming*, Academic 1972.
- [17] DAWKINS, Richard, *The Extended Phenotype: The long reach of the gene*, Oxford 1990.
- [18] DENNETT, Daniel, *Brainchildren—Essays on designing minds*, MIT 1998.
- [19] DEVLIN, Keith, *The Math Gene*, BasicBooks 2000.
- [20] DIJKSTRA, Edsger, *SIGPLAN Notices* **17**:5.
- [21] DISESSA, Andrea, *Changing Minds: Computers, learning, and literacy*, MIT 2000.
- [22] DISESSA, Andrea, “Twenty reasons why you should use Boxer (instead of LOGO),” *Learning and exploring with LOGO*, Proc. Sixth Europ. Logo Conf., Budapest, Hungary 1997, 7–27.
- [23] ENGELBART, Douglas, “Augmenting human intellect: A conceptual framework,” *SRI Project No. 3578* (October 1962).
- [24] ENGELBART, Douglas, Richard WATSON, and James NORTON, “The augmented knowledge workshop,” *AFIPS Conf. Proc.* **42**, NCC (June 1973).
- [25] FOWLER, Martin, with Kendall SCOTT, *UML Distilled: Applying the Standard Object Modeling Language*, Addison–Wesley 1997.
- [26] GLASTONBURY, Bryan, and Walter LAMENDOLA, *The Integrity of Intelligence: A bill of rights for the information age*, St. Martin’s 1992.
- [27] GOOSSENS, Michel, Frank MITTELBACH, and Alexander SAMARIN, *The L^AT_EX Companion*, Addison–Wesley 1994; GOOSSENS, Michel, Sebastian RAHTZ, and Frank MITTELBACH, *The L^AT_EX Graphics Companion*, Addison–Wesley 1994; and European T_EX support from the DANTE project.
- [28] GOULD, Stephen, and Elisabeth VRBA, “Exadaptation—a missing term in the science of form,” *Paleobiology* **8** (1982), 4–15.
- [29] HOFSTADTER, Douglas, *Fluid Concepts and Creative Analogies: Computer models of the fundamental mechanisms of thought*, BasicBooks 1995.
- [30] JACKENDOFF, Ray, *Patterns in the Mind: Language and human nature*, BasicBooks 1994.
- [31] JAYNES, Edwin, *Probability: The Logic of Science*, (edited by Larry BRETTHORST), Cambridge 2003.
- [32] KATZ, Laurent, “Le PC sera-t-il encore longtemps un espace privatif?” *INFOPC* No. 143 (Jan 1998), 19.
- [33] KNUTH, Donald, *Fundamental Algorithms*, Vol. 1 of *The Art of Computer Programming*, 3rd ed., Addison–Wesley 1997.
- [34] KNUTH, Donald, “Literate programming,” *The Computer Journal* **27** (1984), 97–111.
- [35] KNUTH, Donald, and Silvio LEVY, *The CWEB System of Structured Documentation*, Addison–Wesley 1993.
- [36] KOBLITZ, Neal, “The case against computers in K-13 math education (kindergarten through calculus),” *The Mathematical Intelligencer* **18**:1 (Jan. 1996), 9–16.
- [37] LESK, Michael, *Practical Digital Libraries: Books, Bytes, and Bucks*, Morgan Kaufmann 1997.

- [38] MERZ, Thomas, *PostScript and Acrobat/PDF: Applications, troubleshooting, and cross-platform publishing*, Springer-Verlag 1997, including a separately available GhostScript User Manual.
- [39] MCCURDOCH, Pamela, *The Universal Machine—Confessions of a technological optimist*, McGraw-Hill 1985.
- [40] NATIONAL RESEARCH COUNCIL (the full authorship is: Toward an Every-Citizen Interface to the Nation’s Infrastructure Steering Committee, Computer Science and Telecommunications Board, Commission on Physical Sciences, Mathematics, and Applications, National Research Council), *More Than Screen Deep: Toward every-citizen interfaces to the nation’s information infrastructure*, National Academy Press 1997.
- [41] NEEL, James, *Physician to the gene pool: Genetic lessons and other stories*, Wiley 1994.
- [42] ONG, Walter, *Orality and Literacy—The technologizing of the world*, Routledge 1982 (reprinted Methuen 1988).
- [43] OUSTERHOUT, John, “Scripting: Higher-level programming for the 21st century,” *IEEE Computer* **31**:3 (March 1998).
- [44] PEARL, Judea, *Causality: Models, Reasoning, and Inference*, Cambridge 2000.
- [45] PIAGET, Jean, et. al. (translated from the French and edited by Terrance BROWN, with preface by Seymour PAPERT), *Morphisms and Categories—Comparing and transforming*, Lawrence Erlbaum 1992.
- [46] PIATTELLI-PALMARINI, Massimo, *Inevitable Illusions: How mistakes of reason rule our minds*, Wiley 1996.
- [47] PINKER, Steven, *The Language Instinct: How the mind creates language*, Harper 2000.
- [48] PINKER, Steven, *How the Mind Works*, Norton 1997.
- [49] PLOTKIN, Henry, *Darwin Machines and the Nature of Knowledge*, Harvard 1993.
- [50] RIDLEY, Matt, *The Red Queen: Sex and the evolution of human nature*, Penguin USA 1995.
- [51] POLANYI, Michael, *Personal Knowledge*, Harper 1962.
- [52] RESNICK, Mitchel, “Rethinking learning in the digital age,” in *The Global Information Technology Report: Readiness for the networked world* (G KIRKMAN, ed.), Oxford 2002. Also, “Sowing seed of a more creative society—New technologies help students navigate the creative thinking spiral,” *Learning and Leading with Technology*, Dec/Jan 2007/08, 18–22.
- [53] ROBINSON, Andrew, *The Story of Writing*, Thames and Hudson 1995.
- [54] OASIS (Organization for Structured Information Standards), ISO/IEC 8879:1986.
- [55] SPRINGER, George, and Daniel FRIEDMAN, *Scheme and the Art of Programming*, McGraw-Hill 1990.
- [56] STALLMAN, Richard, *GNU Emacs Manual*, 6th ed., Free Software Foundation 1987.
- [57] STALLMAN, Richard, “The GNU Project,” in [14]; also see <http://www.gnu.org>.
- [58] STEPHENSON, NEAL, *In the Beginning ... Was the Command Line*, Avon 1999.
- [59] THAYER, Lee, “The functions of incompetence,” *Vistas in Physical Reality*, papers in honor of H Margenau (LASZLO and SELLOM ed.), Plenum Press 1976, 171–187.
- [60] TOGNAZZINI, Bruce, “Ordinary citizens and the national information infrastructure,” in [40], 271–278.
- [61] VON ROSSUM, Guido, “Computer Programming for Everybody,” proposal to DARPA, Corporation for National Research Initiatives (July 1999), www.python.org/doc/essays/cp4e.html.
- [62] WALTERS, RFC, *Categories and Computer Science*, Cambridge 1992.
- [63] WILSON, Peter, *The Domestication of the Human Species*, Yale 1991.
- [64] WOLFE, Tom, *The Right Stuff*, Bantam 1983.
- [65] WOLFRAM, Stephen, *The MATHEMATICA Book*, 3rd edition, Cambridge 1996.
- [66] WRIGHT, Robert, *Nonzero: The Logic of Human Destiny*, Vintage 2001.
- [67] ZAMIR, Saba, “It’s a big world: Interview with Bjarne Stroustrup,” *Object Magazine* (April 1998), 20–24.