

# Personal Health: From Systems to Spaces

Prepared for

Enterprise Strategy  
Veterans Health Administration  
Department of Veterans Affairs  
810 Vermont St., N.W.  
Washington, DC. 20420

Tom Munnecke  
Science Applications International Corporation  
10260 Campus Point Ct.  
San Diego, Ca. 92121  
(858) 756 4218

[munnecket@saic.com](mailto:munnecket@saic.com)

Version 1.0 July 3, 2002

Available at <http://www.munnecke.com/papers/D24.doc>

## Table of Contents

Introduction .....	2
Historical Model of DHCP .....	2
Design Pressures on the Current System.....	3
The Enterprise-Centric Approach.....	5
Future Technologies .....	5
Inverting to the Personal Health Perspective .....	6
A Future Model .....	8
The Patient-Centric Approach.....	10
Eras and the Flow of Knowledge .....	12
Lessons Learned from Y2K.....	12
Looking Forward .....	13
From “Systems” Thinking to “Space” Thinking .....	14
Comparing the Two Approaches .....	15
The Role of Open Source Technology as Foundation.....	18
Open Source .....	19
Conclusion.....	20

## Index of Figures

Figure 1. Original DHCP architecture .....	3
Figure 2 Enterprise-Centric Model.....	5
Figure 3 Comparison of Computing Power.....	6
Figure 4. Enterprises as part of Person’s health care.....	6
Figure 5 A Personal Health Space Model .....	10

## ***Introduction***

Twenty-five years ago, data processing departments developed systems to meet the needs of various users according to predefined programs. Elaborate procedures were set up to define, program, and run batch programs to meet the needs of users. With the advent of the electronic spreadsheet, however, users were able to manipulate data as they saw fit. The spreadsheet created a “space” within which they could manipulate their own information in the manner which was most appropriate for them.

This paper presents a vision of a similar transition happening in health care information systems. People will have their own information spaces within which to manage their health. And like the sometimes disruptive introduction of the spreadsheet, this space for health will have dramatic effects on our traditional way of dealing with health information.

The complexity of the information technology, health care, and organizational issues facing the VHA today are multiplying to the point where a new way of thinking about systems designs and architectures is necessary. This paper reviews the architectural foundation of the current *VistA* system as part of an era of enterprise-centric systems, and proposes a new metaphor for the future of patient-centric information spaces.

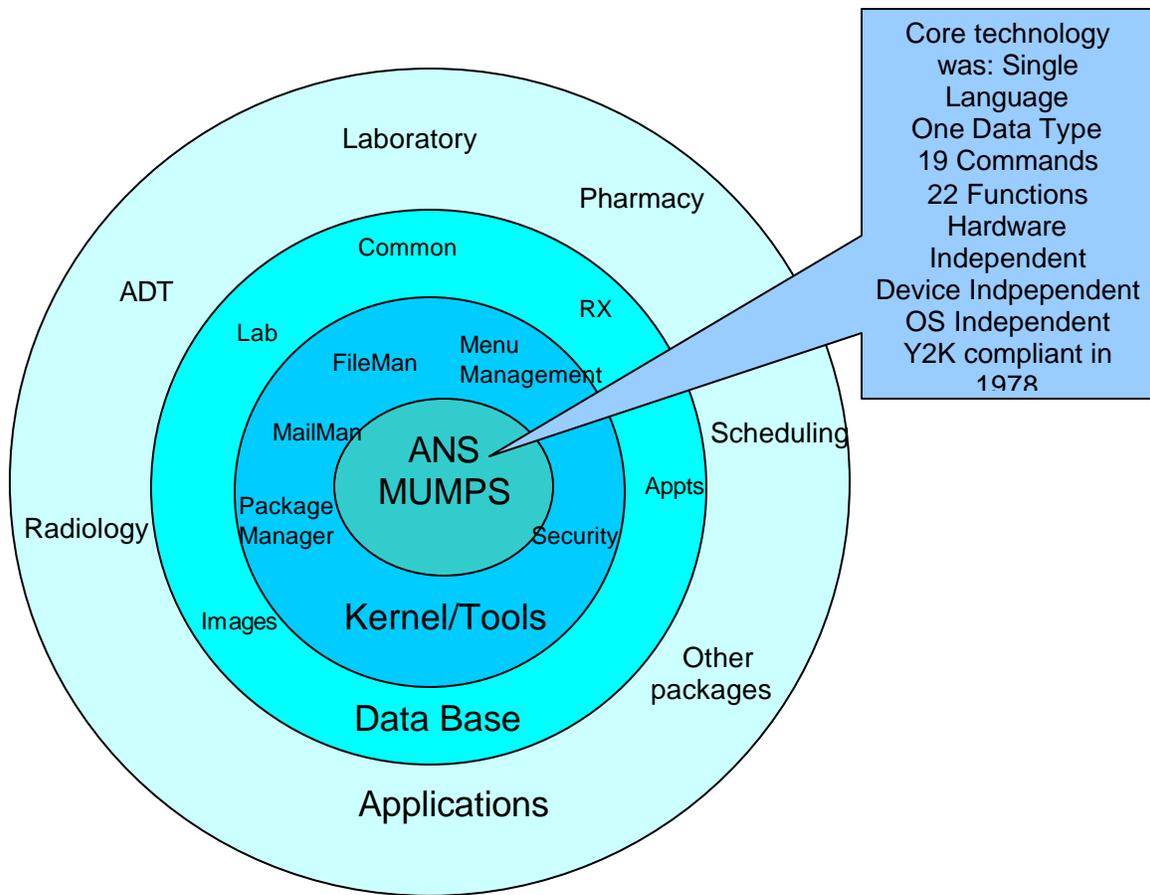
This metaphor is based on the notion of a space, rather than a system, in order to reflect the changing nature and diversity of the demands placed on it. The space metaphor implies connectivity and autonomy of elements, whereas the system metaphor implies levels of control and predefinition which may not be possible outside of the enterprise-centric model.

The technological, medical, and organizational pressures on the VA today dictate an approach which is adaptive to future changes as it moves to patient-centric information systems. This paper explores a conceptual model for doing so.

## ***Historical Model of DHCP***

In June of 1978, the author jotted down a sketch of a concept for a decentralized hospital information system. It was a set of concentric circles, radiating out from a core language, with progressively larger circles containing progressively more specific program and data for more specific applications such as laboratory, pharmacy, radiology, etc. The inner layer was a single language consisting of 19 commands and 22 functions. It used a single data type, a single data storage technique, and was independent of specific hardware or operating systems.

This diagram became the basis for the VA’s Decentralized Hospital Computer Program (DHCP, now *VistA*), DoD’s Composite Health Care System (CHCS), and the Indian Health Services’ Resource and Patient Management System (RPMS), and is still actively used to describe the architectures.



**Figure 1. Original DHCP architecture**

Because this design was for the Decentralized *Hospital* Computer Program, the program focused on the hospital. This architecture was created when minicomputers were just appearing. A typical computer in that era was a PDP-11 with 16 kilobytes of core memory (not megabytes) and cost about \$75,000 in 1978 dollars. A removable disk drive containing 5 megabytes was barely acceptable as carryon luggage at the airport. 300 baud communications (if any) were the norm.

### ***Design Pressures on the Current System***

There are many pressures for change in the current system. Information technology, if applied properly, can introduce new ways of adapting to these needs:

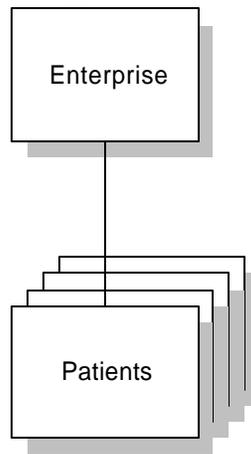
1. The Internet, health care reform, and interagency sharing of resources create the need for a common “space” for a patient’s health information.
2. The move from “brick and mortar” physical hospitals to patient centered care also moves the records for that patient. When there is no single physical location of care, there needs to be a common place for their information, independent of the organizational and institutional shifts of the individual’s providers.

3. As federal agencies share more services and information, there needs to be “home” for that shared information. There needs to be a mechanism to insure continuity of care when more than one institution is involved.
4. Telemedicine blurs space and time considerations. There needs to be a “home” for a interaction that crosses organizational and geographical boundaries.
5. As medicine moves to the concept of health care as a collaborative concept between patient and providers, there needs to be a trusted space for collaboration.
6. Patient acceptance requires a secure, trustworthy system whose security policies and activities they trust and understand.
7. Individuals may wish to allow others to access their health information. They may want to allow a pharmacist at a local drug store, their optometrist, and their dentist access to their active medications lists. They may want to maintain a private conversation and share certain portions of their records with a marriage counselor or religious advisor. These decisions must be made on a personal basis, in the context of their personal decisions relating to trust and risk, as well as regulatory issues such as HIPPA.
8. Individuals may want to customize their personal health information to their own particular needs. They may want to track their exercise at a recreation center, track their moods with a daily self-assessment, or their weight. They may or may not want to share this information with others.
9. Individuals may want to annotate or dispute their own health information.
10. People want to carry on private patient/provider communications in a secure, trusted framework.
11. The Internet is transforming how we communicate. It is creating an “associative avalanche” in which information, people, and technology connect in new and novel ways. These computing and communications capabilities may become the physician’s “stethoscope of the future.”
12. The communications revolution is providing unprecedented power. Computers can communicate around the world today as fast as the original DHCP computers could communicate within their own computer backplanes in 1980. Designing systems from a “state of connectivity” is a fundamentally different problem than designing them in isolation, as has been common in past practice.
13. The introduction of genomic knowledge into clinical practice and information systems carries with it many new issues. DNA information, for examples, relates to an entire family, not just an individual. Family members and their health care providers become trustees of a much wider range of information. A person in the private sector may find that their DNA reveals information about their brother in the military.

## ***The Enterprise-Centric Approach***

The original DHCP design was enterprise-centric. This was a necessary design perspective, and it has served the VA and other organizations durably over the decades. The enterprise is the center of attention, and its mission of providing health care includes “patient centric” care and information systems. VA is responsible to others for its operation, and is subject to oversight from other organizations, such as the GAO,<sup>1</sup> which reinforce this perspective.

This can be illustrated as follows:



**Figure 2 Enterprise-Centric Model**

The Enterprise in the above diagram could be considered a hospital service, an entire hospital, a district/region/VISN, “One VA”, or the entire health US care system. Due to the availability of computing power, communications, costs, and organizational issues, DHCP focused on the enterprise as a single hospital.

In the broadest sense, DHCP was about creating a language and sharing meaning within this enterprise-centric model. The data dictionary, FileMan, MailMan, security, and access control were all focused at the hospital level. This language was embedded deep within the architecture, in the inner few layers.

The interaction with the patient was typically through medical transactions – typically, things the enterprise did to the patient, such as prescriptions, progress notes, lab results, radiological images, etc.

While retaining its nature as an enterprise, however, VHA’s effectiveness as a provider of health care will have to adapt to new perspectives on health from the perspective of the individual. Approaches such as Health eVet<sup>2</sup> and Health ePeople are steps in this direction.

## ***Future Technologies***

---

<sup>1</sup> GAO, “Veterans Affairs: Sustained Management Attention is Key to Achieving Information Technology Results.” June 2002, GAO-02-703 <http://www.gao.gov/new.items/d02703.pdf>

<sup>2</sup> <http://www.health-evet.va.gov>

Today, we can buy a computer embedded in a jewelry ring which is approximately as powerful as the computers upon which DHCP was originally designed.<sup>3</sup> Pocket PC computers costing \$599 have approximately the same capacity as a mid-sized VA or DoD hospital using a VAX computer in 1990.

Then (mid sized hospital)	Now
1978: PDP 11 minicomputer	Wearable computer embedded in Java Ring
1988: VAX 780 computer	Pocket PC with 1 GB memory card, 11 mb/sec wireless internet connection.

**Figure 3 Comparison of Computing Power**

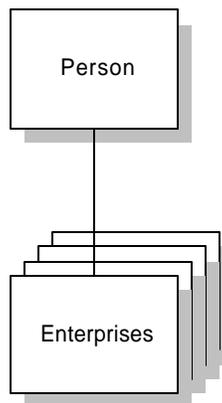
If computers and communications technologies continue to progress, we may find the equivalent of today’s supercomputers sewn into hospital gowns, and prescription bottles with computerized “labels” having as much computing power as a pocket PC today. Wireless communications protocols will allow computers to discover each other, negotiate protocols and security, and exchange information at extremely high speed. People could carry in their wallet (or store in a personalized database) whole body scans, their personal genome data, and their entire history of medical activities. These technologies create entirely new methods for people to interact with their health information.

### ***Inverting to the Personal Health Perspective***

Many great thinkers have had their greatest success by inverting their perspective: Albert Einstein imagined himself riding a beam of light, Jonas Salk imagined himself as a polio virus, and Nobel Laureate Richard Feynman discovered new ways of thinking about physics by imagining himself immersed in a messy fog of electrons.

Inverting our design perspectives from the enterprise to the person represents a powerful new way of thinking.

As some point in this technological development, we will find an inversion between an enterprise-centered view and the patient-centered view.



**Figure 4. Enterprises as part of Person’s health care**

Former VHA Undersecretary for Health Kenneth Kizer spoke of this vision in 1997:

<sup>3</sup> See the Java Ring at <http://www.mrioftx.com/html/javaring.html>

“The patient is the center of the health care universe, not the hospital. Information systems of the future have to be built around the patient - what his or her needs are, what services he or she receives, and what are the outcomes of our interventions and other efforts. We have to be able to track all these things across geography and across time. They will have to be unlinked to any specific organizational setting or geographical setting. That will require a paradigm shift in how we view our technology in the future.”<sup>4</sup>

Looking at patient-centered information systems from the viewpoint of a single organization is a fundamentally different perspective than what we find when we look from the viewpoint of the person. The enterprise sees patients flowing through it, and seeks to provide the best care at lowest cost. People, however, see a multitude of providers flowing past them. They have a much longer perspective, perhaps extending out to the next century. They see a maze of health information, sometimes reliable and sometimes sensationalized by the media or advertisers. From this perspective, as Donald Berwick states, “doctors are guests in the lives of patients.”<sup>5</sup>

This inversion has radical implications regarding information, authority and responsibility in the health care process. Speaking 5 years ago about a trend which has continued to accelerate, Dr. Kizer said,

“As a result of the availability of information on the Web, patients have ready access to research findings. Indeed, it is not unheard of today, and in fact, it is becoming increasingly common for our patients to know more about a given condition or the latest in treatment options than does the physician or other healthcare provider. Instead of being the source of information, or the fount of all wisdom, clinicians now have a new job of interpreting information and helping patients make up their mind as to what treatment options or what diagnostic modality they want to utilize. This will, again, require a different mind set as we provide these services in the future...as professionals, we have had a monopoly on the information about the diagnostic and treatment options of our patients. Now, all that has changed...largely as a result of the Internet.”<sup>6</sup>

Designing a system to deal with this messy fog of information is a formidable task. There is no single point of view, no single authority, and no single “top” from which to do a top-down design. As will be demonstrated below, our very notion of “systems engineering” is incapable of dealing with this level of complexity.

In the broadest sense, this model is creating a language and speech community around health from the personal perspective. It focuses on the health transformation process, which may or may not involve the medical transactions of the enterprise-centric model. Rather than a “system” for managing and performing these transactions, it becomes a “space” within information and health transformations may occur.

---

<sup>4</sup> Kizer, Kenneth “Forms in the Fog: Information Management in the New VA”, speech to the VA Information Technology Conference, May 19, 1997, p. 4

<sup>5</sup> “Doctor Leads Crusade to Replace Office Visits as Standard Procedure”, Wall Street Journal, May 30, 2002

<sup>6</sup> Kizer speech, p. 5

## ***A Future Model***

The enterprise-focused concentric ring model of Figure 2 has lasted for a quarter of a century. What would a new model look like, with a vision supporting information technology for the next 25 years?

Imagine trying to plan a moon rocket shot knowing only arithmetic, knowing nothing about algebra or calculus. Understanding the trajectory of the rocket using only arithmetic is impossible. With algebra and calculus, however, the problem becomes solvable. One of the core issues with health care information systems is that the language which we use to deal with complexity is not powerful enough to handle the task which we face. It is as if we are trying to do a moon shot using arithmetic.

Dealing with truly person-centered health care in the coming era of genomics and proteomics, internet connectivity, and trust and privacy issues requires us to bump up a level of abstraction – or two – in the same way that we needed algebra to get to the moon.

At the core of the personal health space model is a *language*. The area surrounding it can be called its linguistic shell. This language could be considered a kind of macro language for the space, in the same way that a spreadsheet may have a macro language supporting its internal operations. What are passed to the outside, through the mediated shell, are evaluations of *expressions* in that language.

For example, at first glance, a field such as “patient sex” is a simple binary value. However, this is not a simple data element for patients who have undergone or are undergoing a sex change operation. The patient’s male or female status changes over time, and may not be the same in all contexts, even at the same time. A simple value of “Male” or “Female” does not necessarily express the full situation.

Trust relationships and privacy concerns vary with the context of the information. Genomic and proteomic information is similarly based on expressions, rather than simple bit streams. Although we don’t know how to interpret all of this today, we do know that it will be at a level of expression far beyond our current “flat field” data storage approaches. Our current level of thinking about data is equivalent to “arithmetic” level of thinking, when we need to be moving towards “algebra” and “calculus” to fully understand the issues we are dealing with.

The complexity of the internal activities of the space is all expressible in the language at the core of the space. The complexities of the underlying relationships and software within a spreadsheet are hidden from the surface view of the spreadsheet.

This language which is below the surface, and the space within which it operates has several purposes:

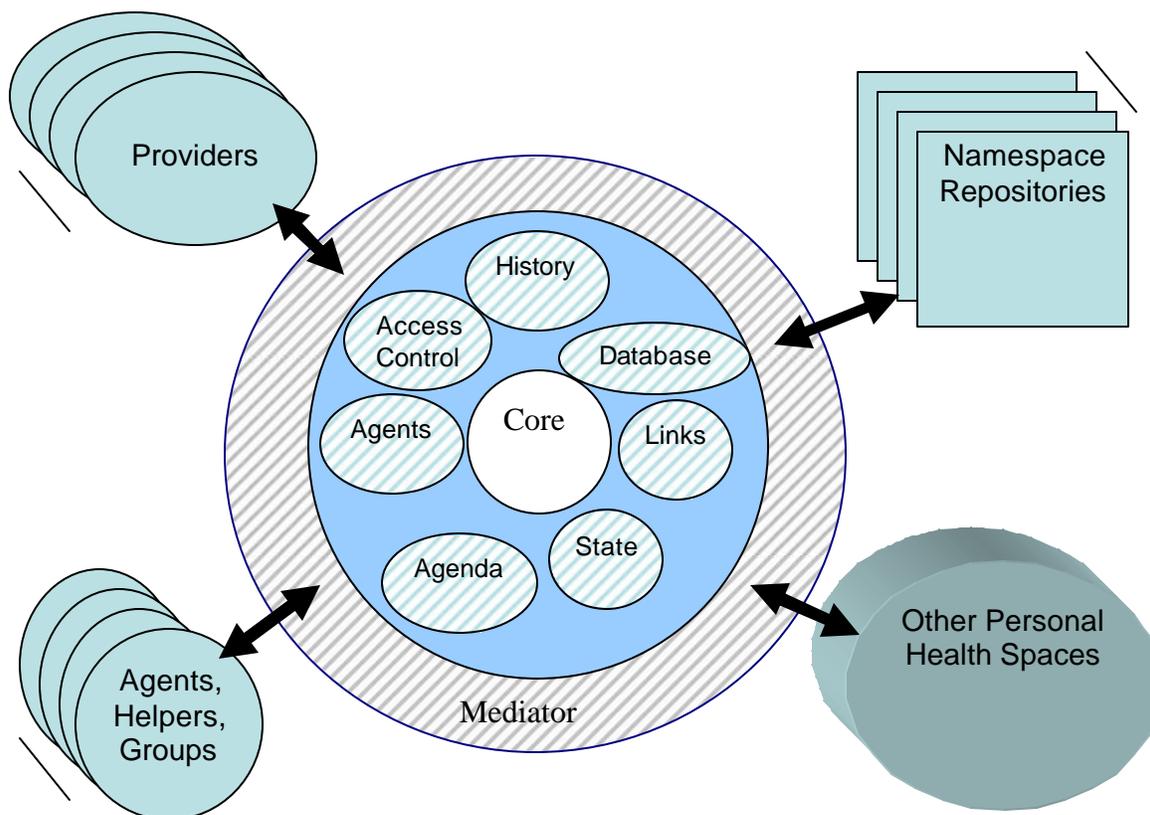
1. **Managing trust and privacy.** The issues of privacy and trust are already complicated, and soon to become much more so as personal genetic information becomes more prevalent. People will become trustees of their family’s genetic information, for example, making disclosure of personal genetic information dependent on much more than a simple personal decision.
2. **Understanding and expressing genomic and proteomic data.** How to express genetic information is itself a linguistic problem, far beyond our current relational database/tables/rows/columns type of thinking. We need a much more expressive language to be able to deal with the knowledge we will be dealing with in regard to the gene.

3. **Managing software agents.** We can expect software agents to operate in these spaces.<sup>7</sup> These agents will assist in decision making, and track long term, possibly lifelong, trends and information. These represent “verbs” in a language of health, unlike the “noun” based language which is prevalent in today’s system. For example, the Unified Medical Language System (UMLS) has few if any verbs in it.
4. **Managing the person’s agenda and schedules.** This includes the patient’s personal agenda of appointments, goals, exams, protocols, triggers, and other time-oriented activities timed around the patient. This may involve short range schedules (a cholesterol screen next week), or long (flexible sigmoidoscopy tests ever 5 years after age 50), to intergenerational issues relating to genetics within families.
5. **Managing meaning over time.** The meaning of information can change over time. For example, the significance of the release of genetic information today can take on an entirely new significance in the future. The language must be able to deal with this change of meaning over time.
6. **Providing infrastructure managing information.** The information contained in the personal health space will be varied and complex. It will not necessarily be recordable as “data” in a relational database and “program” stored in precompiled programs. Data may have complex interrelationships with other data, programs, time sequences, and knowledge bases. For this reason, the core of the space is based on a language, not simply data tables.

---

<sup>7</sup> See, for example, the Guardian Angel project at <http://www.ga.org/ga>

## The Patient-Centric Approach



**Figure 5 A Personal Health Space Model**

The above diagram illustrates some of the concepts of the personal health space. At the center is an individual's shell, containing the information relating to that person's health. It connects with the outside through a linguistic shell, through which all information comes and goes. This can be likened to a web server, which holds information inside of it in a variety of forms, yet presents it to the outside world in specific formats.

The elements which are represented by a "stack" with an arrow by them represent information sources which change over time. For example, medical nomenclatures from 10 years ago were relevant to information stored 10 years ago, but new nomenclatures may have extended or made obsolete over time.

The components of this model are:

- ? **Core:** These are the core services of the space. It provides the core language service for the space, as well as the central controller for the other activities within the space.
- ? **Links:** these links associate the core space with other elements outside of the space. This is not necessarily a separate database, but could also represent

- embedded links from objects throughout the space. Web-based URLs are one approach to this, but other, more permanent approaches may be required.
- ? **Database:** a storage area for structured and unstructured data. This may include a variety of object-oriented, relational, and free text-oriented data.
  - ? **History:** This is an historical log of activity, transcripts from agents and executable scripts. A history of previously disclosed information will allow future analysis of releases to determine if old information has a new context.
  - ? **Access Control:** This logic deals with the trust, privacy, and regulatory compliance issues which mediates the exchange of information in and out of the space.
  - ? **Agents:** These can be considered to be long-running programs which track the long-term flow of information within the space. Agents may run for the life of the person, perhaps longer. They need to be able to “sleep” for years, then wake up at a specific time or event. Given that they may wake up in a new era, they must be able to deal with different eras of information, namespaces, interfaces, and the like.
  - ? **Agenda:** This represents a kind of To Do list for the space, holding information about tasks and information which is yet to be done. A prescription for a drug requiring close monitoring of liver function, for example, could post an agenda item to monitor this appropriately.
  - ? **State:** This represents the storage of state information for the space. The basic web protocols, for example, are “stateless,” which gives rise to the need for storing state between sessions. Cookies are one form of state information. The space has a repository for its own state.
  - ? **Agents, Helpers, Groups** describes formal or informal relations with the persons space, perhaps online support groups for specific diseases, community resources, or software agents which can be purchased or rented.
  - ? **Providers** are the enterprises which provide medical care for the individual. These “flow” past the personal health space, as the relationship between providers and patients can be assumed to be constantly changing.
  - ? **Other Personal Health Spaces** People may interact between their spaces, sharing and collecting information as is relevant to them. Family members who share genetic information become part of a web of trust relating to the patient.
  - ? **Namespace Repositories:** There are a large number of files, dictionaries, tables, and knowledge bases which are involved in the health care system. Individual hospitals may have specific files and tables, for example, which room identifiers

are for the psychiatric wards. Over time, these will change, but the information linked directly or indirectly to them still needs this.

## ***Eras and the Flow of Knowledge***

Our information environment is rapidly changing. In the last 30 years, for example, information technology has gone through several eras of technology:

- ? Mainframe computers with batch oriented interfaces
- ? Minicomputers with terminal based timesharing interfaces
- ? Microcomputers with dedicated personal operating systems
- ? Web-based computing with universal access to global information

Medicine similarly has gone through eras:

- ? Sanitation/public health
- ? Biological/disease model
- ? Chemistry
- ? Genomic and proteomic

Any design created today must be resilient enough to deal with future changes and new eras. It must be durable yet adaptable.

We can predict with near certainty that new eras will emerge, but not the specific technology which drive them. Despite these changes in our information technology and our knowledge of health, we need to provide a continuous means of dealing with our health information needs.

The space must be able to deal with a succession of eras, rolling forward to new eras on a predictable timetable, even though the destination is unpredictable.

## ***Lessons Learned from Y2K***

Computer languages today can be classified as early binding or late binding. Early binding languages fix the characteristics of data, for example, at the time a computer program is created with a compiler. Late binding languages, however, fix these characteristics at the time the program is executed. For example, an early binding language will declare YEAR to be a 2 digit number each time this field of information is used in a computation. A late binding language will discover that YEAR is a 2 digit language at the time it encounters the field. If the field is 4 digits, (the solution to the Y2K problem), the program will still be able to handle the data.

The benefits of early binding are that the program will run faster because it does not have to adapt to varying data definitions when it executes. The drawbacks are that it makes the program more “brittle” and subject to failure when unexpected changes occur. The Y2K problem was a manifestation of this form of early binding. Early binding thus saves CPU and memory costs at the expense of flexibility.

The benefits of late binding are greater adaptability. The drawbacks are that the program spends more time checking the characteristics of the data it is handling, and may

not be able to optimize the storage format of the data. The costs of early binding are measured in terms of decreased CPU and memory efficiency.

## **Looking Forward**

As we look forward, we may ask if there are any issues today which are like the activities in the late 1900's which lead to the Y2K issue. Information we enter today will be bound by characteristics in the future unknown to us today. This has many implications:

- ? Data encrypted today will need to be encrypted by other technologies tomorrow in order to keep them secure. Increases in computing power are sure to make today's encryption techniques obsolete sometime in the future.
- ? DNA information released today may in the future prove to reveal different knowledge than is expected.
- ? Software agents may be launched to track a person's long term health process, but knowledge about that process may change during the life of a patient. For example, a child may be tracked by a vaccination agent which examines the child's vaccination state every 6 months and makes recommendations. It is likely that the recommended vaccination schedule will change at some time in the child's life. The agent must be able to adapt to future protocols while handling current ones.
- ? Nomenclatures and dictionaries are constantly changing. For example, the UMLS system is growing at about 5% per year, and about 5% of the terms in the system change every year. Knowledge which is based on these terms must deal with a continuous flow of terms and nomenclatures.
- ? Information storage, communications, and processing can all be assumed to change rapidly in the future. For example, wireless communication may become so ubiquitous that the concepts of "local" and "remote" databases dissolve into one.<sup>8</sup> The processing and storage of the information in the space may be distributed across many nodes for security or performance reasons.

Components in the space will need to "roll over" to new eras. Each would be active for a given era, and the continue as a reference level to allow their successor model retain compatibility.

## **Binding to the Future**

What is the "glue" that is capable of tying all this together? How can we move forward with an information architecture today which can adapt to these fuzzy, but certain changes in the future?

The spreadsheet program is an example of this kind of "glue." Prior to the spreadsheet, analysts and programmers had to laboriously study, document, and program every specific interaction between users and computers. Committees would decide whether a program was to produce a report sorted by one field or another. Analysts

---

<sup>8</sup> This is similar to the way that email has eliminated the notion of distance. An addressee can be down the hall or halfway around the world, with no apparent difference to the sender.

would carefully lay out reports so that columns were a specific width, formatted in a specific way.

The arrival of the spreadsheet changed this dramatically. Users could simply click on a column to reformat the data if they wished. They could draw pie charts or scatter plots with just a few clicks, unthinkable complex and expensive operations just a few years earlier. They could learn how to use a spreadsheet once, for say a budget report, and apply these new skills to a variety of applications, such as an expense reports or departmental phone lists.

The spreadsheet program provided a set of tools one level of abstraction below the users' problem level. It served as an interpreter between many common office computational problems.

### ***From “Systems” Thinking to “Space” Thinking***

Enterprise architectures tend to be focused on the notions of systems. A variety of systems are implemented to achieve specific requirements and goals of the organization. Metrics are used to calculate cost/benefit ratios of alternatives. These systems frequently parallel the splits inherent in the organization charts, and subsequently must be “integrated” through a series of interfaces into enterprise-wide systems. This tends to limit the boundaries of their systems thinking to that which is under their administrative control.

The broader we consider the topic of health, the more acute the problem becomes. Over larger spans of time and providers, the levels of interaction will increase:

- ? The more we understand genomic based medicine, the more important family and ancestral information becomes. Information can be created and shared in one context, which at a future date can have unexpected new significance.
- ? The more tumultuous our health care system, the more likely there will be disruptions in the continuity of care.
- ? The more we focus on preventative aspects of health, the more our attention is drawn from the purely biological model of disease.
- ? The more distrust in the current system, the more complementary or “quack” approaches will emerge.
- ? The more legislation for patients rights there is, the greater the risk of malpractice, which increases the prevalence of defensive medicine.
- ? The more we automate our medical records, the simpler it will be for malpractice lawyers to “datamine” for malpractice suits, making the medical record even more of a defensive document.

Given these forces, despite claims of “patient centric,” hospital and medical information systems tend to be driven to ever deeper into “stovepipe” orientations. Their understanding of their patients will focus on medical transactions, things that their enterprise *does* to a specific patient.

The model for truly patient-centric information is radically different than that of a single enterprise's information architecture. This model becomes much more web-like, rather than a single hierarchy of information. Each person is different, interacting with a

continually changing array of enterprises. In the enterprise model, the enterprise is fixed and patients flow by. In the personal model, the enterprises flow by.

Enterprises with well defined lines of authority to a single manager have the potential for taking a systems approach to their operations. However, in the somewhat chaotic and constantly changing realm of personal health information, there is no such single authority. There are many autonomous entities interacting, in highly complex and contextually specific ways.

The World Wide Web is an example of this kind of organization. Although it appears to be a chaotic mess of web pages, search engines today are able to locate specific information with great speed. Many different types of communities and communications are thriving, allowing people to connect with each other in ways never before possible.

There is no central control to the web. There is no predefined structure to the way web sites are designed. There is no central manager to eliminate redundancy. Yet the web allows hundreds of millions of autonomous people and organizations to interconnect in new ways.

The web was designed to be a “space” for information, according to its developer, Tim Berners-Lee, “The web was not a physical “thing” that existed in a certain “place.” It was a “space” in which information could exist.”<sup>9</sup> While others were building “things” (systems) for information, he was building a “space” which grew to become the web as we know it today.

### ***Comparing the Two Approaches***

<b>Systems Thinking</b>	<b>Space Thinking</b>
<b>The system is defined by its structure</b>	<b>The space is defined by that which is connectable and discoverable</b>
There is a well defined boundary describing the limits of the system	There are basic constraints on the operation of the space, but within these constraints the elements are fairly autonomous
There are a limited number interactions (interfaces) outside the system	By definition, there is no “outside” to the space, because anything that is connectable is within the space.
These interactions are definable in advance	Interaction may be definable, but new connections are discoverable.
These interactions are well defined	Interactions may be ad hoc and highly specific to certain contexts
<b>The system is rational</b>	<b>There may not be any “one correct way” to use the space</b>
There is a set of rules governing the behavior of system	Rules define underlying constraints.
Participants know the rules affecting	Many of the constraints would be hidden in

<sup>9</sup> Berners-Lee, Tim, **Weaving the Web, The Original Design and Ultimate Destiny of the World Wide Web**, Harper San Francisco, 1999, p. 36

them	the infrastructure.
Each user and component behaves according to the rules	There is greater autonomy of the elements within the space
The process is repeatable	Each interaction may be unique and irreversible.
The integrated behavior of the system does not create any conflict or paradoxes between components	Fewer rules leads to fewer conflicts
The system does not modify or refer to itself	The space is built on the concepts of self-reference.
<b>There is a set of requirements</b>	<b>It may not be possible to describe the operation of the space.</b>
Requirements can be unambiguously expressed	There may not even be a language within which to express requirements.
There are those who know or can find out the requirements	Everyone may use the space in a different way.
The requirements can be expressed independently from the operation of the system	The space is its own definition. For example, one cannot define the web other than pointing to it and saying, "It is this."
The organization can come to agreement on a single set of requirements	There is no one organization which controls the space.
Users know what they want without seeing it	Users discover their needs based on what is available.
The organization has the resources to develop the requirements	Less organization, less requirements implies less need for resources.
<b>System can be decomposed and integrated</b>	<b>Space is generated from simple initial conditions, evolves over time.</b>
The whole system is equal to the sum of its components	The whole of the space is greater than the sum of the parts
Complexity can be reduced by breaking big problems into smaller independent components	Complexity emerges as an evolutionary outgrowth of the space.
The effects of integration will not change the components	Elements within the space are continuously adapting to whatever else is happening.
Components developed separately can be integrated into a whole	Each agent will evolve within the space, automatically connected to its environment.
<b>There is an authoritative point of view from which requirements can be stated</b>	<b>There is no central authority to the space.</b>
An authority controls each component	Elements control themselves within the constraints of the space
The scope of control of these authorities can be integrated as their components are integrated	
The authority accurately reflects the	Self-controlling

needs of users of the components	
Other points of view can be subordinated to the authoritative point of view	
Users whose needs are not met will not develop their own systems independent of the authoritative one	This is a fundamental aspect of the evolutionary process. Some elements will try different things. Those which are “fit” will reproduce.
The functions of the system map closely with the functions of an organization chart.	Functions will develop as necessary in their environment. Elements expend resources on unnecessary functions will lose out to elements which are more fit.
<b>The environment is fixed</b>	<b>Environment is constantly changing and adapting to itself</b>
The environment perceived at requirements time will stay the same during development and deployment.	The elements change their own environment.
Deploying the system does not change its environment	Environment and elements co-evolve.
The environment has the capacity to support the system as it scales up	If space is “autocatalytic” it will grow larger as it accumulates more elements. i.e. Amazon.com did not “use up” the web, but rather expanded it.
<b>The behavior of the system can be understood by aggregating transactions</b>	<b>Some interaction can be characterized by transactions, but other is transformation.</b>
The basic unit of interaction is the transaction	Basic unit is the health transformation
Transactions are linear	Transformations are non-linear
“Bottom line” is achieved by “rolling up” transactions	The may be no “bottom line”; transformations cannot necessarily be added
The organization shares a common categorization of transactions	Transformations may be highly contextual and personal to the individual.
“Flow” is assessed by periodic “snapshots” and aggregations of transactions	The basic model of the space is a continuous flow of information.
<b>The system can be optimized</b>	<b>The space places a premium on adaptability which may require redundancy.</b>
There is a single value by which organization can be measured	There is no single metric with which to determine “optimal”
There are predictable results from specific interventions	Individual transformations may be unique to the individual.
The system will scale and adapt	Space is designed to be scalable, adaptable from the outset.
The system will cope with growth of	The space <i>is</i> its own evolution.

the organization	
<b>There are resources to accomplish the project</b>	<b>Open source development dynamics do not follow traditional models of scarce resource allocation</b>
<b>Trust</b>	<b>Trust focuses on webs of trust, based on individual elements and chains of trust.</b>
The organization is developing the right system	Successful systems are those which are the most trustworthy.
Correctness is defined according to meeting requirements	“Adaptability” replaces the notion of “correctness”
People trust the deployed system	Untrustworthy elements or activities dissipate.
People trust the system to operate safely	Trustworthy elements are selected as part of the fitness function of the space.
People trust the system security	This is one of the constraints of the system
People trust the systems engineering paradigm	The space evolves according to general constraints
That there is organizational acceptance to the idea	Elements which are not accepted do not thrive.

### ***The Role of Open Source Technology as Foundation***

The original DHCP software architecture was written during the transition from centralized mainframe computers to decentralized minicomputers in the late 1970’s and early 1980’s. It was based on a single language, ANS MUMPS. It used a single data type (string), single database technology (globals based on multi-way Btree storage algorithm), 19 commands and 22 functions.

This architecture was remarkably durable, as it continues to be the core for nearly all hospital information systems in the VHA, DoD, and Indian Health Service even after 23 years. The computing platform for the technology has migrated from PDP-11, VAX, Alpha, and Intel hardware. The operating systems on which this software has operated have been standalone MUMPS, VMS, Unix, DOS, and Windows. Although MUMPS (later named M) technology support has shrunk to a single commercial vendor, there is some activity to create an open source version of the software.

The term “open source” was not around at the time of the MUMPS development. In particular, the Internet was in its infancy, and VA was not connected to it. There were few examples of large-scale collaboration such as are evident in today’s open source community. However, there were many similarities:

1. All development was based on a single ANS language standard.
2. Software was designed and written to be hardware independent. If there were operations which needed to be specific to a particular computer, operating system, or terminal, they were abstracted to a separate part of the system. If the system had to move to another environment, only these limited facilities needed to be changed.

- The goal from the beginning was to create a long-term, open community of developers. At the initial meeting describing the system, for example, representatives from VA, DoD, and Indian Health Service were at the meeting. The software was adopted by a group in Finland, who sent programs and ideas to the VA.

## Open Source

The goals of the original DHCP architecture were quite similar to the open source software movement today.

“The basic idea behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing.

We in the open source community have learned that this rapid evolutionary process produces better software than the traditional closed model, in which only a very few programmers can see the source and everybody else must blindly use an opaque block of bits.”<sup>10</sup>

Some comparisons of open source and commercial technologies are:

Class of Software	Commercial	Open Source
Server	Windows 2000	Linux
Desktop	Windows XP, 98, 95	Windows
Database	Oracle, SQL Server	MySQL, Berkeley DB
Mail Client	Outlook Mail	Eudora
Mail Server	Exchange	SMTP, POP, IMAP based servers
Programming Language	C#, Visual Basic	Java, PHP, Python, Perl, C++, Javascript
Web Server	IIS	Apache
Productivity Suite	Microsoft Office	Sun's Star Office

Open Source software has several potential advantages:

- ? VA would not have to pay software license fees.
- ? The software would be available to a larger body of users.
- ? Other developers would be motivated to improve the software and contribute them via open source procedures
- ? Other organizations which adopted the software would find that they had a greater degree of shared architecture. This could lead to higher levels of

<sup>10</sup> Open Source Initiative, <http://www.opensource.org>

standardization and expressiveness of interaction. (See Language vs. Data Interaction, below)

- ? VA could lower its costs of software development as it discovered an increasing body of software, modules, and information available in open source form.
- ? VA could lower its maintenance costs as it discovered that open source methodology can be more adaptive.

There are potential disadvantages to the open source approach as well:

- ? The quality of the software may not be up to commercial standards. Just because a project is declared “open source” does not mean that it will be “high quality.” While Linux has proven itself to be a highly reliable operating system, there are other open source projects which are of poor quality.
- ? Support for open source software is typically through a community of users via email. Professional support tends to be via independent consultants, who may or may not have the needed skills. Commercial sources typically have trained customer support engineers available to hire.
- ? The entire approach to open source software is relatively new, and there is no guarantee that it is a long term, viable form of software development. The economic model for developing and maintaining open source software may not be sustainable in the long term. The success of the approach is based on a sufficient number of players to create a critical mass of interaction.
- ? Open source is an extremely popular subject at the moment, and may be subject to the boom/bust cycle that has

## **Conclusion**

Twenty-four years ago, it was a challenge to get acceptance of the original enterprise-centric model of figure 2. “Why should we go to the expense of using a general subroutine when we can do it more cheaply with hard code?” was a common question. “We can’t wait for the next release of the kernel software to implement this function, so we have to do it ourselves at the application level” was also very common.

The inner layers of the model were critical to the success and evolution of the architecture, yet the visibility was greatest at the outer levels. Over the years, pressures for expedited solutions tended to inflate the outer layers of the model with software which should have been embedded deeper into the core. Investment in the inner layers of the software gradually waned, to the detriment of the entire stack of software.

The vision of a space presented in this paper is an attempt to build on the successful aspects of the past model, as well as technological trends we see already and can anticipate in the future. These trends may be disruptive to current thinking and procedures, and may appear today to be weak or insufficient to their task. However, such is the nature of emerging technologies. It is necessary to develop a strong foundation from which to base our future information technologies. This will require a vision, investment, and collaboration in a common infrastructure far more powerful than today’s.