THE LISTER HILL NATIONAL CENTER
FOR BIOMEDICAL COMMUNICATIONS

A research division of the U.S National Library of Medicine

TECHNICAL REPORT
LHNCBC-TR-2004-005

ASKLEPiOS Project;
A Report to the Board of Scientific Counselors

September 2004

R. P. Channing Rodgers

U.S. National Library of Medicine, LHNCBC
8600 Rockville Pike, Building 38A
Bethesda, MD 20894

Table of Contents

1. Introduction

Mark Weiser [1952-1999], of the Computer Science Lab at Xerox PARC, is widely regarded as the founding figure of the field known as "ubiquitous computing." First enunciating his ideas in 1988, he envisaged a world with large numbers of wireless computing devices per person, with the devices spanning wide ranges of size and capability. In his vision, ubiquitous computing is inspired by "social scientists, philosophers, and anthropologists;" "its highest ideal is to make a computer so imbedded, so fitting, so natural, that we use it without even thinking about it." Although relying upon wireless communications, it "is not the same thing as mobile computing, nor a superset nor a subset." Rather, it represents a "third wave" of computing, superseding the eras of the mainframe and of the desktop PC, heralding "the age of calm technology, when technology recedes into the background of our lives. [1-3]"

The reality subsequent to Weiser's untimely demise has been anything but calm. Today's information worker is confronted with the badges, tablets, and whiteboards that Weiser worked with, along with a wide array of PDAs, pagers, mobile phones, smartcards, sensors, active and passive radio frequency identification tags (RFIDs), as well as wireless mice, keyboards and other accessories. Devices are communicating over data and telephonic wired links as well as ultrasound, Infrared Adapter (IrDA), and various forms of radio frequency protocols, including 802.11 (Wi-Fi), 802.15.4 (Zigbee), Bluetooth, Ultra Wide Band (UWB), and wireless telephony/data standards. Increasingly active research into mobile *ad hoc* networks (MANETs) anticipates a world containing mobile users, devices, and sensors, engaging in complex and shifting relationships [4].

The connection between computing and communicating is now well established; however, the multiplicity of devices and means of communication, and the fact that many technically identical systems are operated as part of managerially isolated systems, has led to clinicians with belts bristling with handheld electronic equipment. Countering this complexity are efforts to merge multiple capabilities into fewer devices — most noticeably in the effort to combine PDAs, mobile telephones, and pagers, an effort often referred to by the single word *convergence*. "Smartphone" PDA/phone hybrids of the 1990s were commercial failures, but the recently introduced Handspring/Palm Treo 600 has been a solid success, and other firms are rushing similar products to market. Distinctions between technologies are blurring: originally devised for the robust transmission of data, Internet protocols are increasingly used for voice communications as well (as with "Voice-over-IP," or VoIP). As of this writing, several firms are poised to introduce mobile telephones that invisibly switch-off between 802.11 and cell telephony networks as the user moves about.

The Internet and World Wide Web have demonstrated the advantages (and risks) of integrating information resources on fewer user interfaces and fewer devices, yet information workers, including clinicians and biomedical researchers, remain confronted with information appliances that seem needlessly numerous, non-standardized, complex, and non-interconnected. Among these disconnected islands remains the one technology that comes closest (if still remaining distant) from Weiser's view of a calm ubiquitous technology — the humble telephone. The CIA World Factbook estimates that there are at least 186 million main lines in use in the US, and over 615 million in the rest of the world. Surely the telephone is one of the most ubiquitous information appliances on the planet. It's very ubiquity, and the dichotomy we commonly make of voice versus data communication, may have blinded us to its further potential.

Project ASKLEPiOS (**A**ccess to **S**ervices and **K**nowledge, multi**L**ingually, **E**verywhere, **P**ortably, **i**n **O**pen **S**ource) is a collaborative effort with Sun Laboratories and others to explore and develop tools and frameworks for ubiquitous computing in biomedicine, focusing upon services that can be built with network servers, handheld devices, and the telephone network.

2. Project Objectives

The ASKLEPiOS project draws upon lessons in strategy learned from the World Wide Web, as invented by Berners-Lee and extended by work at NCSA [5]:

1) The use of open source and open standards encourages acceptance, unleashes the creative talents of a large community of developers, and can leverage pre-existing information systems. Use of platform- and vendor-neutral standards have played crucial roles in the evolution of the Internet and the Web; it is difficult to imagine these systems arising from a proprietary corporate culture. Two of the most important contributions of NCSA to the Web were its forms-based user interface (the first inexpensive system allowing rapid development of cross-platform graphical user interfaces) and the Common Gateway Interface (CGI, the first of many ways of hiding existing services behind a web interface), which together allowed numerous pre-existing information services to be brought quickly to wide use over the Internet.
2) Intelligent integration of existing tools and standards can have a synergistic effect upon their utility, and is an innovation in its own right. The early Web built itself upon existing concepts and standards (networked hypertext, SGML-based document markup, a multitude of multimedia standards) and invented outright with well-considered parsimony (there were numerous existing models for HTTP).
3) Simplicity is critically important in the widespread acceptance of a technology. Berners-Lee knew what compromising simplifications to make (dropping the notion of bi-directional anchors, and simplifying SGML, for example).

In the spirit of these guiding principles, the following four prototype systems are being developed:

1) ASKLEPiOS servers containing the following as HTTP services: a simple Unicode-based multilingual clinical vocabulary; multilingually-capable speech synthesis and recognition systems (used only with English in this stage of the project); and, a simple robotic chat service providing access to NLM online resources. The ASKLEPiOS servers also provide AccessGrid videoconferencing tools, and an Internet telephony server (based upon the Session Initiation Protocol, or SIP).
2) Novel gateways to an established ubiquitous communication device — the traditional telephone (a service sometimes referred to as the "Publicly Switched Telephone Network," or PSTN).
3) A prototype wireless handheld device, the SNAKE (**S**mall **N**etwork-**A**ccessible **K**nowledge **E**ngine), based upon a commercial personal digital assistant (PDA), running Linux and X windows. The SNAKE supplies the usual personal management tools and other features expected of a PDA (calendar, "TO DO" list, address book, calculator, games, audio memo, handwriting recognition, ...) as well as 802.11b wireless Internet and World Wide Web access. It also has a SIP client and AccessGrid-compatible audio and video conferencing tools, which along with access to the ASKLEPiOS server, allow real-time videoconferencing, access to speech-enabled applications, and multi-modal audio (802.11

and PSTN, SIP and non-SIP) communication.

4) A prototype information retrieval service based on the ARK (Archival Resource Key) persistent identification scheme [6]. This system, layered on top of PubMed, demonstrates one approach to the important but unsolved problem of how to deal with the degree of permanence of digital information objects.

3. Significance

The consolidation of information services on small devices will lead to fewer pieces of hardware encumbering users, more ready access to information when and where it is needed, and to an increased likelihood that existing and future services will be synergistically integrated. It will allow new forms of service to arise that were previously impractical, including easier clinical consultations between highly mobile physicians, and remote medical services being guided by ancillary care-givers or the ill themselves. Portable personalized devices with visual and speech-based interfaces will prove helpful in delivering health care to an increasingly multicultural and multilingual society. Such platforms will provide many new opportunities to disseminate NLM's information resources in more useful ways.

4. Project Components

This project relies upon the integration of open source tools capable of running under the Solaris and/or Linux operating systems, on non-mobile Sun workstations and on the ARM-based HP/COMPAQ iPAQ PDA. Numerous tools were evaluated for use and discarded in favor of the tools discussed here.

4.1 Software Applications

4.1.1 ICPC Multilingual Collaboratory (IMC)

The International Classification of Primary Care (ICPC) is a clinical classification containing 726 clinical concepts, available in over 20 languages, augmented by links to ICD-10 concepts [7, 8]. Developed by workers at the University of Amsterdam under the aegis of the World Organization of Family Doctors (Wonca), ICPC is employed in clinical information systems in several European countries. In its original incarnation, content existed in numerous different (mainly PC-specific) character encodings, which was found to adversely affect data management, quality control, and portability. The relatively small size and multilingual content of ICPC provided the perfect target for demonstrating the application of the Unicode text encoding standard [9]. The ICPC Multilingual Collaboratory (IMC) is a web-based collaboratory (based upon the use of our *webkit* World Wide Web environment, see below) with three interfaces: 1) a password-protected editorial interface which instantiates a hierarchical authority model and communication channels for review, control, and addition of translations; 2) an openly accessible read-only interface with email access to the editors (providing public access and another level of content review); and, 3) a management interface for the system administrator [10].

4.1.2 Internet-Based Teleconferencing Tools, the MBONE, and the AccessGrid

Although most users employ the Internet for one-to-one communications, its early developers realized its potential for one-to-many and many-to-many communications as well, setting aside one of IPv4's five address classes (D) for the purpose of "multicasting." In the multicast world, an IP address represents not a specific host, but rather an abstract entity known as a "session," in which multiple hosts may participate. When used properly, multicasting allows for a much more efficient use of network bandwidth (for example, a party sending a video stream to *n* other parties need only send out one copy of the stream, with copies being replicated at routers only where needed to feed downstream viewers). In the 1990s, few routers were multicast-capable, so a virtual multicasting network known as the Multicast Backbone (MBONE) was created by running multicasting software on specific hosts ("tunnels") and passing multicast traffic between them, encapsulated within conventional IP packets. Tunnels and bridges are still employed today to connect to multicast programs from hosts otherwise unable to employ multicast (many ISPs do not enable multicast services on their networks).

Two particular centers of excellence appeared during the MBONE decade, developing tools to support multimedia network conferencing: one was at the DOE/UC Lawrence Berkeley Laboratory (LBL), and the other centered on Jon Crowcroft and his Multimedia Research Group at University College London (UCL). Two of the two tools developed within these groups have proven so useful and stable that they have outlived the MBONE, have been ported to many platforms, and continue to serve as the core tools of the AccessGrid [11], the current cutting-edge network multicasting initiative. The AccessGrid project, headquartered at the DOE Argonne National Laboratory (ANL), encourages the creation of specially equipped rooms for videoconferencing, with features such as large-screen displays and robotically controlled video cameras, and has provided server and client software to manage "virtual venues," enabling the management of participants, shared software tools, and security. Hundreds of AccessGrid "nodes" have been established across the world. At its core, AccessGrid still employs two of the MBONE tools described below (*rat* and *vic*) for audio and video conferencing.

Note that these tools are designed for true *real-time* conferencing, in either unicast or multicast mode. Unlike the more commonly-used *streaming media* tools, in which buffering can be used to allow TCP to correct lost and damaged Internet packets, these tools employ the non-error-correcting User Datagram Protocol (UDP) to transmit real-time data, employing clever strategies to ameliorate lost data. Methods for robust transmission and bandwidth reservation are both active research topics within the Internet community.

4.1.2.1 Robust Audio Tool (*rat*)

The Robust Audio Tool (*rat*) was created at UCL for the purpose of audio communication, and complies with the ITU umbrella standard H.323. It relies upon three (IETF-standardized) protocols: Real-Time Protocol (RTP) for audio [12], Real_Time Control Protocol (RTCP) for control communications, and Message Bus (MBUS) for communication between the interface and audio components. It supports multiple unencumbered audio codecs (including ITU G7.11, the GSM telephony code, and LPC, unique to the MBONE community). It contains mechanisms for both sender-based repair (redundant transmission [13, 14]) and amelioration (packet interleaving [15]) and receiver-based repair (silence substitution, packet repetition, and pattern-matching repair). It also includes a strategy for protection against problems in the scheduling algorithms used by UNIX and Windows [16], and triple-DES encryption for secure conferencing. RTCP allows the gathering of participant lists and reception quality reports, and MBUS allows *rat* to be controlled by another process (for example, conference control tools or applications which synchronize speech to lip motion in an animation). ANL has further enhanced *rat* by adding AES encryption.

4.1.2.2 Video Conferencing Tool (*vic*)

Originally developed by the Network Research Group at LBL, in collaboration with the University of California, Berkeley, *vic* was first released as source code in November 1994 [17]. It was modified extensively at UCL to make it easier to develop alongside *rat*, to support IPv6, to repair various bugs, and to add support for additional video codecs, video capture devices, and hosting platforms. It is compliant with ITU standards H.261 and H.263, and, like *rat*, has been ported to a wide number of platforms. (Note that popular proprietary video conferencing systems such as Netmeeting and iChat are H.263-compliant, but are platform-specific, non-interoperable, and also frequently have problems with firewalls). ANL has further enhanced *vic* by adding AES encryption.

4.1.2.3 Other UCL tools

The UCL group developed other multicast/unicast conferencing tools which, although not currently used by ASKLEPiOS, may be useful in future: a whiteboard (*wbd*, originally developed by Julian Highfield of Loughborough University, and compatible with the LBL-developed *wb*) a shared-text editor (*nte*), and a tool allowing advance announcement and joining of multicast sessions ("session directory tool," or *sdr*) UCL also developed a transcoder for *rat*, which allows unicast participants to join a multicast group, or which translates between different codecs being used by participants who are using different communication bandwidths.

4.1.3 Brazil (Web Application Framework)

Brazil is a web services framework, written in the object-oriented language, Java, developed by Stephen Uhler of Sun Laboratories [18]. At its core lies a lightweight third-generation Web server that can be easily layered on top of existing services, by means of writing of small modular Java programs known as "handlers." Brazil allows clean separation of document content from its presentation, and facilitates the production of compact and readable code, through the use of HTML and Brazil Scripting Language (BSL). It is particularly well suited for this project, as a number of the other components are written in Java, its creator is our collaborator, and we are emphasizing cross-platform engineering practices.

4.1.4 Sphinx (Speech Recognition)

Sphinx arose from a multi-year DARPA-funded research program at Carnegie Mellon University. *Sphinx 1* appeared in 1988, establishing the Hidden Markov Model (HMM) approach, previously considered computationally infeasible, at the forefront of speech recognition research [19, 20]. CMU released *sphinx 2* in 2000, along with acoustic models for both English and French. Nearly three years later, *sphinx 3* was released, which was slower but more accurate than *sphinx 2* [21]. *Sphinx 4,* a joint project between Sun Research Laboratories, Mitsubishi Electric Research Laboratories (MERL), and Hewlett Packard (HP), with contributions from the University of California at Santa Cruz (UCSC) and the Massachusetts Institute of Technology (MIT), was a complete re-write of the system, in Java [22]. The architecture of *sphinx 4* was carefully considered, allowing it be be better integrated into other systems, and allowing parts of it to be used independently of the whole.

Sphinx is described as a real-time, large-vocabulary, speaker-independent speech recognition system. It relies upon four sources of information to perform its task: 1) an acoustic model; 2) a language model; 3) a pronunciation lexicon; and 4) an optional "noise word" dictionary, used to filter out breathing, coughing, laughing, and other non-language noises. *Sphinx* is supplied with well-established acoustic and language models for recognizing isolated digits and digits spoken as continuous speech (TI46, TIDIGITS) and for recognizing a 60,000 word vocabulary based on broadcast news (HUB4), as well as models of intermediate size, including ones based on 5,000 and 20,000 word vocabularies drawn from the Wall Street Journal (WSH5K, WSH20K). Pronunciation lexicons must be constructed manually, but a number of open source lexicons exist, including the CMU Dictionary.

An acoustic model is constructed by recording multiple speakers pronouncing text that is similar to that which is to be recognized later (a total recording time of 50-100 hours is generally used). It is segmented, usually at the sentence level, and supplied to a modeling system along with the corresponding text. *SphinxTrain* is an acoustic training environment supporting *sphinx* (*2*, *3*, and *4*) [23]. Incoming speech is transformed into a series of vectors ("features") representing characteristics of the sound. This allows the construction of a graph of all possible feature sequences in the entire language under consideration, which can be associated with probabilities of occurrence drawn from the language model. Audio containing speech that is to be recognized is processed in the same manner as with the training set data, and then its features are compared to the graph of the acoustic model. The Hidden Markov Model is thus a sort of giant search problem, seeking a match between the input features and the acoustic model graph with the highest associated probability. An acoustic model actually operates at the level of arbitrary units, which can be phonemes, diphones, triphones, words, ... as appropriate to the word corpus to be dealt with (for example, for the TIDIGITS acoustic model, individual phonemes are recognized, whereas for most larger models, triphones are employed).

A language model is constructed from a substantial set of written material similar to the language that is to be recognized, and enables use of either a BNF grammar or (more commonly) a *n-gram* (usually trigram) statistical approach to aid in word recognition — which also serves to disambiguate homonyms ("to," "too," and "two," for example). CMU provides an online service which accepts a sentence corpus file and returns a set of lexical and language modeling files suitable for *sphinx* [24].

4.1.5 Festival/Flite/FreeTTS (Speech Synthesis)

Festival is a multilingual (British, American, Spanish, Welsh) speech-synthesis (Text-to-Speech, or TTS) system, developed at The Centre for Speech Technology Research (CSTR), University of Edinburgh [25-27]. Written in C++, it features several APIs which allow access from the UNIX shell, Scheme, C++, Java, and Emacs. It employs diphone-based waveform synthesis. It also provides an interface to the related MBROLA project of the Polytechnical University of Mons, Belgium [28], which aims to provide text-to-speech synthesizers for as many languages as possible (and which offers voice databases and tools for adding foreign accents and emotion to synthesized speech). MBROLA is not available in source, but binaries or many platforms are available for non-commercial and non-military use. *Festival* is commonly used in conjunction with an MBROLA binary and voices, as many prefer the final stage processing ("resynthesis") and voices that MBROLA supplies. *Festival*/MBROLA have enabled others groups to create synthesizers for other languages (for example, German [29]). *Festival* can employ Sable, an XML-based speech synthesis markup language which attempts to reconcile several preceding proto-standards.

The *flite* ("festival-lite") program is written in C, and is a lighter-weight sibling to festival, suitable for use in embedded applications and on smaller devices such as the iPAQ [30].

FreeTTS is a Java derivative of *flite*, developed by the Sun Microsystems Laboratory team that helped create *sphinx 4* (see above) [31]. It attempts to combine the more efficient algorithms of *flite* with the more flexible architecture of *festival*. A technical report claims that after code optimizations, *FreeTTS* achieved execution speeds two to four times faster than the C-based *flite* [32].

4.1.6 ALICE Chatbot

ALICE (Artificial Linguistic Internet Computer Entity) is an automated chat robot, devised by Richard Wallace [33]. It was inspired by the *Eliza* program, an early experiment in natural language processing (NLP) written at MIT by Joseph Weizenbaum [34]. Although the system has twice won the Loebner Prize for artificial intelligence, contemporary workers in both AI and NLP would deny that it is representative of either discipline: it relies upon a simple query/response model using an algorithm driven by textual pattern matching, based upon laboriously created patterns written using the XML-based Artificial Intelligence Markup Language (AIML) [35]. It has spawned much ancillary activity, such as web sites that charge "botmasters" to host their personalized AIML-driven chatbots [36], and commercial services which provide lip-synchronized animated chatbot avatars [37], including one paid service which serves to tutor people in English [38]. Laying aside the ambitious claims made by its author, ALICE does provide a natural focal point for demonstrating and deploying speech services, and provides for web interfaces and the ability to interact with other software. ALICE is instantiated in at least five currently maintained independent programs, written in Java, PHP/MySQL, C++, Pascal, and Perl. We employed "Program D," written in Java.

4.1.7 *PhoneStation* (Computer Telephony System)

PhoneStation [39], developed by collaborator Stephen Uhler, comprises a hardware component (the SPARCstation-to-telephone Interface Module, or STIM) and a software component (*PhoneScript*, an extension to the *tcl* scripting language [40]). The STIM connects the workstation audio system to a single telephone line, and *PhoneScript* allows one to write sophisticated computer-driven telephony services (which can include speech synthesis), such as access to online services and information, and voice-menuing systems. Uhler is currently considering the development of a thumb-sized USB telephone interface that could be used to replace the use of the STIM; this interface would be platform-independent (as is *tcl*), thus enabling the use of *PhoneStation* on most computing platforms.

4.1.8 SIP Server and Clients

Session Initiation Protocol (SIP) is an IETF-developed signaling protocol for "Internet conferencing, telephony, presence, events notification and instant messaging [41, 42]." It does not carry audio or multimedia content (that is done using other protocols, such as those used by *vic* and *rat*), it simply enables communication. SIP is an interesting case-study in protocol design, drawing heavily from pre-existing web conventions such as HTTP and the use of URL syntax.

SIP Express Router (*ser*) [43], is an excellent free RFC3261-compliant SIP server; it is capable of acting as a proxy, registrar, or redirect server (these services are used by callers to locate one another over the network), offers many other SIP-relevant services, and is capable of interacting with a wide variety of other SIP products, providing a PBX-like capability for Internet telephony.

The SIP client *linphone* is available for the iPAQ in *ipkg* format.

4.1.9 *Asterisk* (Telephone PBX)

Asterisk is software for Linux that creates a Private Branch Exchange (PBX) for telephone communications [44]. It was released as open source by its author, Mark Spencer of Digium, a company which manufactures interface cards which allow a PC to be connected to a single phone line, or to aggregated forms of telephone service such as E1 and T1 lines. *Asterisk* supports Voice-over-IP via multiple protocols (H.323; SIP; Media Gateway Control Protocol, or MGCP), and bridges VoIP protocols and traditional telephony equipment using the Inter-Asterisk eXchange protocol (IAX). It supports 10 different audio codecs, and numerous telephony and call features. *Asterisk* running on a simple PC can provide sophisticated telephone services for hundreds of users at a small fraction of what traditional purpose-built PBXs cost. It's enormous capability comes at a price to the programmer, however: four different APIs, all complex.

4.1.10 ARK System Prototype

The ARK system prototype (http://ark.nlm.nih.gov) provides a front-end to the PubMed database (and employs PubMed unique identifiers), illustrating one vision of a credible persistent digital object identification service. Conceived at NLM, ARKs are being most actively implemented at the California Digital Library (CDL, http://ark.cdlib.org/), with additional experimentation underway at the University California, the World Intellectual Property Organization (WIPO), Rutgers University Libraries, and the Internet Archive. Open-source software that mints and maintains bindings for ARKs will be released by the CDL in September 2004.

The ARK naming scheme eases support for persistent identification of information objects. The core principle of the ARK is that persistence is purely a matter of service, and is neither inherent in an object nor conferred on it by a particular naming syntax. The most that can be expected of an identifier is to lead users to services that support persistence. The term ARK itself refers both to the scheme and to any single identifier that conforms to it.

An ARK defines a triplet of URLs that connects users to three things:

1) When typed into the location field of a web browser, the base ARK leads the user to the named *object*.
2) That same ARK, followed by a single question mark ("?"), returns a brief *metadata record* that is both human- and machine-readable.
3) Followed by dual question marks ("??"), the ARK returns a statement about the level of persistence promised by the provider that is currently providing services for the identifier. LP The ARK scheme differs from the PURL, URN, and DOI schemes [45] in that the ARK scheme recognizes that two kinds of naming authority affect persistence: the original assigner of names (the naming authority) and the current provider(s) of mapping services (such as name-to-object, name-to-metadata, and name-to-commitment) and actual content. There may be many concurrent and/or successive mapping services, as well as actual object holders. Over time, the original naming authority and its policies have less and less to do with the current object holders and mapping services, and their policies. In the context of our prototype, NLM is acting as the mapping provider, but because most of the referenced objects are held elsewhere, NLM has no commitment to the objects themselves, and so NLM's internal permanence ratings cannot be applied to the objects.

4.1.11 Ancillary Computing Infrastructure

When dealing with a large number of disparate and complex software systems, and adding locally developed software into the mix, it is crucially important to develop a reliable discipline for documenting, archiving, and retrieving work that is done. While using the tools *rcs* and *cvs*, where appropriate, for version control of individual files, we also have developed local infrastructure to standardize the UNIX environment (*Common User Environment*, or *cue*), and we have worked with the Applied Research Laboratories at the University of Texas at Austin to extend their *depot* shared software repository system, to install software as network-sharable packages [46-48]. This system facilitates standardization of the development environment by allowing the same package to be supplied to depot clients from a single depot server. Multiple versions of a given package are allowed, each package is installed in its own filesystem hierarchy for easy management and removal, and a server can offer files for multiple hardware and OS types. Every *depot* package is accompanied by a README.LOCAL file, with a standardized header describing the package, as well as detailed installation instructions. We currently have in excess of 700 individual software packages installed on our main *depot* server.

Our group web services rely upon *webkit*, documentation and a set of Bourne shell scripts which install and configure a fully-featured web server, based entirely upon open source, suitable for use as a research platform. Although the principal components of *webkit* are the Apache web server [49], the PHP server-side scripting language [50], and the MySQL relational database [51], the package entails over 90 distinct software packages, all installed independently as *depot* packages. A number of higher-level web applications have been built on top of *webkit*, including IMC, but also including *software archive*, which allows us to catalog *depot* packages which are stored offline, in a searchable manner.

4.2 Platforms & Operating Systems

4.2.1 Sun Workstations

Most of this work has been done on Sun UltraSPARC 2 and 60 computers, soon to be upgraded to SunBlade 2000 and 2500 machines.

4.2.2 Solaris & Java

The Sun workstations have been operating under Solaris 2.8 and 2.9 (soon to be 2.10), and carefully kept up-to-date with respect to security and performance patches. The systems have an up-to-date Java environment, supplemented by numerous Java packages required by this project, such as the Java Media Framework.

4.2.3 PC (i86) Hardware

We employed a Dell Dimension XPS R400 (Pentium III).

4.2.4 PC OS & Development Software

The Dell PC was set up as a dual-boot machine with Linux (7.3, later upgraded to 9) and Windows XP. This host, running under Linux, was used briefly in the early stages of installing Linux on the iPAQ PDAs (using the minicom-2.00.0-3 terminal emulator).

The *handhelds.org* site provided a pre-built *gcc*-based ARM cross-compiler for the i86 architecture, referred to as a "tool-chain." This runs under Linux on the PC, and is required when compiling from source code to create iPAQ binary executable files.

4.2.5 Wireless Handheld iPAQs

The PDA hardware configuration consisted of:

1) HP/COMPAQ Pocket PC H3870. Features: 240x320 backlit color LCD screen, 32 MB SDRAM, 16 MB flash memory, SmartData (SD) card slot, and IrDA and serial (synch/asynch) external interfaces (shipped with: Microsoft Pocket PC, v. 3.0.11171, Build 11178 (winCE)).
2) HP/COMPAQ Dual Slot PC Card Expansion Pack
3) SimpleTech Digital Media CompactFlash card, 128 MB
4) 802.11b Wireless PC Card: Lucent Orinoco Gold Card
5) Winnov VideumCam Traveller 2.0 Video Camera & PC card

The iPAQ was selected because of its unique expandability, and strong support for Linux and open source software for it. The Dual Slot PC Card Expansion Pack provides additional battery power, in addition to connections for the 802.11b and video PC cards. The CompactFlash card allows storage of the Microsoft Pocket PC operating system, prior to its removal.

4.2.6 Familiar (Linux for ARM devices)

Numerous different approaches have been taken with respect to embedding Linux in small devices [52], and it is found in various types of handheld and tablet computers, mobile and IP telephones, robots, AV and automotive equipment, and even a wristwatch. Numerous commercial PDAs employ Linux, many of them (over 25 as of this writing) documented by LinuxDevices.com [53]. The most commercially successful of these is likely the Sharp Zaurus.

Several versions of Linux exist for the iPAQ ARM-based PDA. A full version of Debian Linux, known as *intimate*, is available but consumes a minimum of 140 MB of space, which would require the use of microdrive (for which there is no spare PC slot). The older, more parsimonious Linux version known as *familiar* fits in 16 MB of PDA flash memory [54]. Among its principal features: journaling fast file system 2 (JFFS2) to access Flash memory (important, as it also attempts to distribute reads/writes across this limited-lifetime memory form); secure shell (*sshd*); anti-aliased TrueType font support; integrated Python, PyGtk, PyGDKImlib; considerable binary/library compatibility with Debian's ARM distribution; and, software package support through *ipkg* (based on the Debian package management model). It can be installed with one of two interfaces: the X Windows- and *GTK+*-based *GPE*, or the Qtopia-based *Opie*.

Although developed and maintained by a loosely-knit community of open source developers, *familiar* has received considerable support from workers at HP Laboratories, Cambridge, who also help maintain the handhelds.org web site.

5. Discussion of Current Status of Work

5.1 ICPC Multilingual Collaboratory (IMC)

The ICPC Multilingual Collaboratory (IMC) is complete and operational, and final corrections to its online documentation are underway as of this writing. The penetration of Unicode on commonly used computing platforms was poor when this work began, but progressed remarkably during the course of the work. The read-only interface for examining ICPC content uses server-side Unicode capability to produce GIF files representing text, allowing *any* GIF-capable Web client to display any of its 20+ languages, regardless of local Unicode support. The editorial interface, however, requires client-side Unicode support in order to edit ICPC content. We spent considerable effort in locating and documenting appropriate web clients and editing tools for each of the principal platforms: PC/Windows, Macintosh/MacOS, and UNIX. In the case of UNIX, we collaborated with Thomas Wolff, author of the *mined* Unicode text editor, to help add support for editing Chinese/Japanese/Korean. The IMC system is available online through the Emerging Technologies Group (ETG) web site: http://etg.nlm.nih.gov/project/icpc/.

5.2 ASKLEPiOS Server-Based Systems

The following services have been installed and tested on non-mobile Sun workstations: the *webkit* Web research platform (and other ancillary research infrastructure outlined earlier); the IMC system; the *ser* SIP server; *vic* and *rat*; the *Festival* and *FreeTTS* text-to-speech systems; the *sphinx 4* speech-to-text system; and, the Java version of the *ALICE* chatbot system.

ASKLEPiOS employs Brazil to provide web interfaces for various services, including: *FreeTTS*, *sphinx 4*, and *ALICE*. For example, one can pass text within a URL to the *FreeTTS* Brazil service, and receive an audio file in return, containing synthesized speech. We have devised AIML files for *ALICE* to instantiate *ASK*, a chatbot capable of describing our research activities.

Our collaborator Stephen Uhler is handling all of the work related to the *Asterisk* PBX system. Although it conventionally runs under Linux, he is porting it to Solaris. He is investigating the feasibility of providing a much simpler development environment for it, by layering a parsimonious *PhoneScript*-like command language on top of *Asterisk*'s complex APIs, and is attempting to extract parts of *Asterisk* for independent use.

Given the lack of a good SIP client for SPARC/Solaris machines, we have been collaborating with the current managing author of the Java-based *SIP Communicator* application (Emil Ivov), to try to get it working. If this effort is unsuccessful, we hope to use code currently being extracted from the *Asterisk* PBX system by Uhler.

5.3 SNAKE Handheld Wireless Devices

We installed *familiar* with the X Windows system and *GPE* user interface, using a complex multi-stage bootstrapping process, which we documented thoroughly on our group web site [55]. We are the first group to document an installation method relying entirely upon Linux and open source (other sites have documented use of a Windows PC to send bootstrap files to the iPAQ). Starting with version 0.5.3, we have since upgraded to 0.6pre, 0.7 pre-release, and finally 0.7.2. Using *ipkg*, we supplemented the base installation with various other applications such as the *vi* editor, *vic*, *rat*, and the *linphone* SIP client.

The *rat* audio tool required modification, to: 1) make its graphical user interface fit on the iPAQ screen; and, 2) make its audio component work properly with the ALSA audio driver (this work is in progress as of this writing).

We are currently experimenting with installation and use of a Java Virtual Machine on the iPAQ, which would open up opportunities to use Java audio capture tools, *FreeTTS*, and parts of *sphinx 4*.

6. Evaluation & Discussion

In the instance of IMC, the working system was cloned to a Linux server in Amsterdam, and presented to an internationally representative panel of ICPC editors at an ICPC workshop there in June 2004. Workshop attendees agreed that the system would likely produce a striking improvement in the efficiency of their work in maintaining and extending ICPC. It is intended that our Amsterdam-based collaborators will take on support of the system as it enters production mode; we have discussed plans as to how they could evaluate the impact of IMC in formal publishable terms, through documenting the rates at which new translations are contributed and completed, the rate at which errors are detected and repaired, and through assessments of changes in the awareness and use of ICPC.

ARK research and development is most active at the CDL, and evaluation of the technology will be left to that group.

With the exception of IMC, the work described here is still at the proof-of-concept prototyping stage. In this stage, the questions to be addressed for each component are, in order:

1) Does this system work acceptably at a technical level?
2) Is this system something that I would like to use routinely?
3) Is this system something that others would like to use routinely (and, if so, in what sort of applications)?

As components of ASKLEPiOS progress beyond the prototype stage, they will be applied to real-world situations, where the means of evaluation will vary as appropriate.

Initial technical assessment of the iPAQ yielded mixed results. The device itself, suitable as a prototype, is still too physically unwieldy to deploy for routine use. The video frame rate remains low — approximately 6 fps, with detectable latency which could interfere with some applications where audio and video must be tightly synchronized. Video images are of rather poor quality, and highly sensitive to lighting conditions. Audio quality has varied widely depending upon which wireless network was in use; using a Cisco Aeronet 350 802.11b base station, we observed latencies from 500 to 1500 msec, which is unacceptable for telephonic communications. However, using an early model Apple Airport base station, we observed clear audio with latencies well under the usual cutoff for acceptability of 250 msec. We were not able to determine the source of latency in the Cisco setup, a task which may be much more difficult now due to recent changes in management of local wireless networks (see below). Work on SIP and speech applications is still at too early a stage to make helpful assessments.

We have identified and contacted several other groups that have experimented with the *familiar*/iPAQ platform. USC/ISI conducted research into military surveillance applications of the platform under DARPA funding in 2000-2001, with an emphasis on power analysis, making important contributions to the software for this device [56]. The "Baby GRASS" project [57], has used it to run a Geographical Information System (GIS/GRASS). Haining Liu of the Video over IP Group in the School of Information and Computer Science at University of California, Irvine, got *vic* and *rat* operating as a pilot project [58]. Motorola Laboratories, as part of their Internet 2 initiative, attempted to integrate iPAQs running *vic* and *rat* with AccessGrid nodes [59]. Neither ISI nor Liu nor Motorola addressed the issue of the *rat* user interface not fitting properly on the iPAQ display, and *rat* audio was made to work by installing an alternative audio driver, potentially breaking other audio applications.

Although there are excellent open source packages for dealing with facsimile (fax) transmissions, we have chosen not to include fax services within the ASKLEPiOS framework, as we consider fax technology as being rapidly eclipsed by more efficient network equivalents (it could be added in future if this evaluation changes).

Earlier sections of this report may lead some readers to conclude that we believe that telephony has a strong future. In fact, although we believe that telephonic *functionality* (voice communication) has a strong future, we foresee the disappearance of the telephone in its current form, as the merger of computing and communications continues; oral communications will become integrated with other modalities of information exchange on a wide variety of physical devices. As with most such technological shifts, the transition period will be marked by the newer technology (the computer) imitating the older (telephony), as do most current "SIP telephone" sets; over time, the full flexibility of computer telephony will be revealed to users by software applications that deviate more from the established conventions of traditional telephony, and integrate themselves into the other multimedia capabilities of the underlying computing platform. Given the huge installed base of traditional telephone sets around the world, this transition may take many years to complete, occurring first in wealthier areas of the world. *PhoneStation* and *Asterisk* provide a pathway for the large existing telephone infrastructure to be integrated into this transition.

With consumer- and industrial-grade commercial speech recognition systems available, why use *sphinx*? There are compelling reasons, most of them related to the fact that it is open source. First, it is multi-platform, and not tied to specific computing hardware. Work built upon it will be more widely usable. Second, proprietary systems present you with their own fixed language models. Most consumer-grade systems allow these to be supplemented by training for an individual speaker, but you can not create an underlying speaker-independent model of your own design, and the ones the vendor offers may not be satisfactory for a given application. Third, work built using *sphinx* is unencumbered and freely sharable, making it more suitable for collaborative research, and ensuring an stable path forward in time (which is more than a hypothetical concern, as demonstrated by the collapse of industry leader Lernout and Hauspie). Finally, commercial systems are not modular — you can not pick off components for embedding within other applications.

7. Potential Applications

Potential applications exist at the levels of the computer operating system, application software, and human activity.

Mobile wireless Internet-connected devices with speech interfaces would be a boon to the disabled, and to physicians who work with hands occupied (as for pathologists and surgeons). They would have myriad applications for emergency response personnel. There is much need for the development of good unencumbered cross-platform server-side and client-side (perhaps Java applet-mediated) web content readers.

A single handheld wireless device that can act as a telephone, pager, Internet access device, and videoconferencing platform could replace the current collection of clinical belt-line accessories. It would allow integration of information resources and means of action in a manner now difficult: for example, it would be much easier to tie together a specific patient with that patient’s electronic medical record and a list of care-givers, along with the various ways in which those care-givers can be contacted (email, SIP or conventional telephony, etc.). NLM databases could be accessed via typed input or speech, and output obtained as text or spoken language. Full telephony services could be provided over 802.11 networks, with 802.11-to-PSTN bridging provided via *PhoneStation* or *Asterisk*. (bypassing the recurring service fee charged by pending commercial 802.11/PSTN mobile phone services for use of their internal proprietary bridge).

Data underlying the PDA-like services (calendars, address books, "TO DO" lists, ...) of the hand-held Internet-connected SNAKE could be maintained on a single, secure, well managed (backed-up) central server on the network, and synchronized dynamically with copies on the SNAKE, with no action required on the part of the user.

The multimedia capabilities of a properly designed SNAKE would improve the quality and coherence of clinical training, as bedside rounds could now integrate comments and images from radiologists, pathologists, and others who are presently seen at other times and places, making their findings more clear and compelling through their immediacy. Part of a resident's morning rounds duties could be to prepare lucid summaries of literature findings that could be integrated on-the-spot with a particular patient's clinical findings and laboratory results, in a manner that could be shared by everyone present in the palm of their own hand. Radiologists, pathologists, and others are already archiving images, sometimes within their formal reports, and a multimedia platform could retrieve these images on demand in any location needed, to be integrated more naturally into teaching rounds, family conferences, or collegial discussions. By freeing up physicians from the tyranny of proximity, clinical consultations could be more easily arranged, and follow-up discussions more frequent.

We employ the iPAQ as a SNAKE prototype because its expandability makes it a convenient research platform. The framework of tools envisaged here will be of value regardless of the physical format of the final device. Although some of the applications may run suboptimally on today's equipment (speech recognition may be slow, images poor), evolution in processor, display, and camera technology will rapidly catch up.

8. Plans for Further Work

Our primary outside collaborator, Stephen Uhler, is concentrating on evaluating *Asterisk* and developing tools for handling audio using the SIP and Inter-Asterisk eXchange (IAX) protocols on Solaris and Linux. Inventor of the Brazil framework and *PhoneStation*, he provided the striking SLIM demonstration (discussed below) and has been helping us to develop a Brazil handler-based web interface for *sphinx*. We have concentrated on developing the iPAQ SNAKE prototype, dealing with Internet-based videoconferencing, working with *sphinx*, and developing biomedical applications that utilize NLM resources.

8.1 Clinical Evaluations of SNAKE Prototype & Computer Telephony Applications

The SNAKE prototype currently in use is impractical, in part due to its unwieldy backpack with external 802.11 card video camera, and video card, but also due to its sluggish performance for conferencing, and its limited battery capacity. In early August 2004, HP announced a new iPAQ model, clearly intended to compete with the Treo 600 phone/PDA hybrid. The h6315 employs the XScale ARM chip, which is double the speed of the processor in the model we currently use; it contains twice the memory (64 MB SDRAM, 64 MB Flash ROM), contains 802.11b support, and supports Bluetooth. It also serves as a mobile phone, supporting the Global System for Mobile Communications (GSM) protocol and the associated General Packet Radio Service (GPRS, a protocol for data transfer using mobile telephones, with a *theoretical* maximum speed of 171.2 kbps). The initial service provider will be T-Mobile. Most importantly for this project, it has a built-in camera, which is being advertised as a still camera, but which (we are assured by colleagues at HP Laboratories) will be capable of video. One limitation is that the camera does not appear to be aimable independently of the iPAQ case, so that the device may still not be optimal for videoconferencing. We would like to replace our current iPAQ platform with this model (thus dispensing with the external PC card backpack we currently require to support 802.11 and video services). Although still far from the optimal device for ASKLEPiOS, such a device would be deployable in realistic clinical settings. (In future, we can anticipate the release of both increasingly capable handheld devices, and also of better wireless services, such as "Enhanced Data rates for GSM Evolution," or EDGE, a component of forthcoming "third generation" GSM services, or 3GSM, which offers three times the data rates of GPRS).

Pending correction of problems in our ARM cross-compiler environment, we will complete our modifications to the *rat* audio tool, encapsulate the results into an *ipkg* package, and share this the handhelds.org community. We will complete testing of the SIP client, *linphone*. In the intermediate term, we will install and test a Asterisk-compatible IAX audio client being produced by our collaborator Uhler. The SNAKE platform will be the target of several experiments in speech interface design (discussed below).

In the longer term (12-18 months, depending upon how quickly *familiar* and the rest of our iPAQ software environment can be transferred to the new HP 6315) we will attempt to evaluate the SNAKE in a clinical environment. We are engaged in discussions with members of the Department of Internal Medicine at the University of California, San Francisco, with respect to executing clinical assessment studies of the device. The currently envisaged study design (subject to further elaboration) is as follows:

Hypothesis:

The use of hand-held video conferencing can contribute to improved remote supervision of residents/nurses by the "physician of record" at moments when he/she can not be physically present, leading to improved health care.

Methods:

Eight teams of residents will be divided into two groups. The control group will conduct themselves as usual, relying mainly on telephone conversations with "physicians of record" when such physicians can not be present for direct patient evaluation (as at night, or when the physician is at another site). The test group will employ SNAKES for such conferences, using encrypted 802.11 wireless video-teleconferencing over UCSF's clinical virtual private network (VPN). The physician will employ a desktop computer with *rat* and *vic*. The study will continue for a period of between 6 and 12 months.

Evaluation Criteria:

This use of the SNAKE will be evaluated according to the following criteria: 1) frequency of use; 2) user satisfaction (both resident's and physician's); 3) number of occasions in which the physician considers that use of the SNAKE is likely to have changed diagnoses or management plans, with estimates of the likely effect on outcome of care. Length of stay and patient throughput will be compared between the control and test groups.

Colleagues in the Department of Laboratory Medicine, University of California, San Francisco (Drs. Hamill and Terrazas), are eager to explore the application of computerized telephony to pressing problems in their department. Their clinical laboratory has recently been moved across the city from the main clinical center (additionally, the laboratory already had several satellite facilities). In a pilot project at UCSF several years ago, R.P.C.R. demonstrated the use of MBONE technology to transmit clinically usable microscopic images of bone marrow aspirates, for the purposes of being evaluated by a pathologist at a remote site. In the first stage of the current shared endeavor, we have installed an AccessGrid node within the Department of Laboratory Medicine, which will be used initially to perform clinical teaching rounds between a physician/microbiologist working in the laboratory (a facility removed from the main campus), and an infectious disease specialist and his Fellows, working in the hospital on the main campus. At least two simultaneous video streams will be in use, supplying facial images from the infectious disease participants, and live images of microbiological culture plates and microscopic images from the microbiologist. A single audio stream will carry full-duplex voice communications between the two sites. This system in anticipated to be fully operational by the end of 2004. It will be evaluated as follows:

Hypothesis:

The use of video teleconferencing can substitute for a face-to-face conference in the context of microbiology/infectious disease rounds, and can improve efficiency of teaching when participants are widely separated and would otherwise have to travel to meet.

Methods:

Instantiate the system described above. Have the participants at both ends complete surveys for the periods before and after implementation, rating the two regimes according to: frequency of rounds; amount of time taken for the meetings (including transportation time in the case of the initial regime), clarity of the didactic materials, and overall satisfaction (accepting specific comments as well).

Evaluation Criteria:

Compare the ratings obtained before and after use of teleconferencing.

In the next stage of collaborative work, during 2005, two computer-telephony systems will be prototyped. We will be using *Asterisk* for this application, as we wish to build the system using readily available hardware which may be easily replaced and repaired. Integration of these systems with existing UCSF clinical information systems will be conducted by UCSF personnel.

The telephone tasks we automate here are currently expensive problems for the UCSF laboratory, which estimates that it consumes the equivalent of between 2 and 3 FTEs during 16 hours each day (calls fall off during the "graveyard" shift). Given the disciplined time management practices in the laboratory, evaluating the impact of these systems in pure dollar terms will be straightforward.

1) A system enabling physicians to call the lab by phone, enter their security code, and retrieve recent laboratory results for their patients.

2) A system which would assist with calling up responsible parties when emergency laboratory results are observed. For example, when an ambulatory patient on anticoagulants is found to be at risk of hemorrhage due to his medications, a laboratory technician must dial telephone numbers from a supplied list of care-givers until one is located who can quickly take charge to ensure the patient’s safety. This automated system will take over dialing out for responsible parties until one is found, then pass the call to the technician to complete the pass-off of information.

Both systems will be evaluated in the following manner:

Hypothesis:

The computerized telephony service will reduce the FTEs required, with no decrease in technician or care-giver satisfaction, or detrimental effects on patient care.

Methods:

Gather data for the period of 6-12 months prior to deployment of the system, to compute the FTEs dedicated to the task. Then deploy the computerized telephony system, and gather data about FTEs required after the system is declared fully operational, for the same period of time used prior to deployment.

Evaluation Criteria:

Compute the changes in FTEs before and after deployment; perform surveys of technicians and care-givers to assess overall satisfaction before and after the new system is deployed. Write case studies of any instances where either the manual or semi-automated systems cause or encounter special problems (for example, inordinate delays or miscommunicated results).

Systems already exist capable of providing the above services, but they are expensive, proprietary, and limited to telephone communications. The ASKLEPiOS open source framework would dramatically lower the cost barrier for developing such services, and provide multimodal communications capability, which we would like to explore beyond 2005: the direction that work takes will depend upon the degrees of success in the other areas of work outlined here.

8.2 Speech Interface Research & Development

Given NLM's considerable expertise and interest in natural language processing (NLP), the appearance of *sphinx 4*, the world's first serious open source speech recognition engine, is of particular import (note again the comments at the end of section 6). Evaluation of such systems is based upon measuring accuracy rates: what proportion of words within a target document are recognized? However, training and then evaluating *sphinx 4* is anything but trivial. The HMM algorithm is essentially a huge brute-force pattern-matching exercise. The number of parameters at play in determining the performance of *sphinx 4* is large, and the parameters have complex interactions.

Workers in speech recognition rely upon simple empirically-derived rules-of-thumb in training their engines, as there are no rigorously determined statistical guidelines. In creating new acoustic models, between 10 and 30 speakers are generally employed. On the order of five hours of spoken material are gathered from each speaker. It is important to use realistic speakers of the sort to be recognized, rather than trained voice actors (there are of course many variables at play in selecting speakers, whose speech patterns may differ widely based on age, sex, level of education, dialect, etc.). It is also important to use audio sampling rates and ambient noise levels that accurately reflect the targeted final working conditions. There are of course trade-offs: using more speakers for training improves the speaker independence of the final system (overall accuracy for randomly selected speakers), but at the cost of reduced accuracy for any one speaker (note: *sphinx* can be trained to a specific speaker, with improvement in recognition rates for that speaker; about 20 minutes of speech from the individual is required, which can be folded into a pre-existing acoustic model using maximum-likelihood methods). A larger training vocabulary means that the final engine will recognize more words, but again at a cost of reduced overall accuracy. Finally, great care must be taken in defining the problem: just what kind of speech is to be recognized? That influences the choice of a "unit of recognition" to be employed in the language and acoustic models: for example, when recognizing numbers, the vocabulary is small (good for increased recognition rates) but the words themselves are short and sometimes resemble one another (bad for recognition rates). The HMM algorithm is often found to work better in such circumstances if the unit of recognition is taken to be an individual phoneme.

Taking into account then that devising speech recognition models remains more art than science, and that ambitious goals may require substantial time and resources, we outline a number of research and development targets, listed in increasing order of difficulty. Some of these projects may need to draw upon the linguistic expertise of NLP-specific collaborators within LHNCBC:

Short Term (Year 1):

Hypothesis:

Performance of *sphinx* is comparable to commercially available speech recognition engines.

Methods:

Using a minimum of 10 acoustically diverse speakers in quiet ambient conditions, measure recognition rates for the digit-recognition and Wall Street Journal models supplied with *sphinx*, using an adequate sampling of digits and excerpts from the WSJ as the test materials. Repeat study using a commercial speech recognition engine (IBM’s ViaVoice) instead of *sphinx*.

Evaluation Criteria:

Compare recognition rates between *sphinx* and ViaVoice.

Hypothesis:

Individual speaker training improves recognition rates for that speaker, to a similar extent in *sphinx* and ViaVoice.

Methods:

Using the speakers from the above study, train the engine for each individual speaker and repeat the measurement of recognition rate.

Evaluation Criteria:

Compare the changes in recognition rates before and after training, for both *sphinx* and ViaVoice, and compare these figurers between the two systems.

Hypothesis:

Use of a language model such as WSJ is inadequate for purposes of searching the biomedical literature.

Methods:

Using the speakers from the previous studies, measure recognition rates for the existing WSJ model, but applied to the reading of a test set of biomedical abstracts, obtained by using queries from one of NLM’s query test sets (oriented toward Internal Medicine and Family Practice) to search PubMed.

Evaluation Criteria:

Compare the recognition rates obtained for biomedical abstracts to those obtained in the initial study for reading WSJ materials.

Intermediate Term (Years 2-3):

Hypothesis:

Creation and use of language and acoustic models appropriate to the Graeco-Latin language used in biomedicine will yield improved recognition rates for biomedical literature, as compared to use of the WSJ model.

Methods:

Use between 10 and 30 representative speakers, working in low ambient noise conditions (suitable for literature searches in quiet settings, but not recognition from ER telephone calls). Draw training materials from the literature for Internal Medicine and Family Practice using elements from NLM query test sets appropriate to those disciplines, such that about 5 hours of speaking time per speaker is obtained. Employ *SphinxTrain* to create a new acoustic model.

Evaluation Criteria:

Evaluate recognition rates using other speakers (minimum 10 in number) and published materials not used for the training process. Train the engine for each individual speaker, and repeat the measurement of recognition rates.

Hypothesis:

Sphinx provides a usable source of software components

Methods:

This is really an engineering evaluation: look at the code design to see if *sphinx 4*'s modular design allows, in principle, the use of components for ancillary applications; could one, for example, create a phoneme recognizer, and use it behind a web site to create an open source biomedical language pronunciation dictionary?

Evaluation Criteria:

See if a phoneme recognizer can be successfully extracted from the code and run independently.

Hypothesis:

Speech recognition via a SNAKE can be made more efficient by placing parts of the *sphinx* recognition engine on the SNAKE itself.

Methods:

Parts of *sphinx* will be placed on the iPAQ: the acoustic capture mechanism and feature-recognition engine. The remaining tasks will be done on the ASKLEPiOS server, working in conjunction with the SNAKE via the network connection. A simple speech recognition task (recognition of numbers and a small vocabulary) will be performed by 10 separate speakers, under two regimes: doing feature recognition on the SNAKE and the remainder of processing on the ASKLEPiOS server, versus capturing audio on the SNAKE but doing recognition processing entirely upon the ASKLEPiOS server. Recognition rate and speed will both be measured.

Evaluation Criteria:

Recognition rates and speed will be compared between the two test regimes.

The preceding projects dealt with speech recognition; the following projects explore speech synthesis, which is a much simpler and more mature technology.

Short Term (Year 1):

Speaking web site

There are commercial systems which provide "speaking web sites."

Hypothesis:

A Java applet used with *sphinx* can provide equivalent performance to a comparable commercial system while providing superior engineering features.

Methods:

Devise a Java applet to harvest content highlighted by a Web client user (or, alternatively, the content of an entire frame) and return synthetic speech from the ASKLEPiOS FreeTTS server. Compare the performance of this system to a public demonstration project using a commercial solution, in place at NIH (http://nihseniorhealth.gov). Have at least 20 subjects use both systems (randomizing the order of presentation) and rate them both according to: speed, ease of use, clarity of speech, and overall impression.

Evaluation Criteria:

Compare the ratings found for the two systems.

Speech on the SNAKE

Hypothesis:

Adequate biomedical speech synthesis can be performed on a device like the SNAKE.

Methods:

Install a Java Virtual Machine will be installed on the iPAQ, allowing the installation of *FreeTTS*. Obtain a small set of biomedical abstracts using several queries from a NLM query test set. Have at least 20 subjects evaluate the quality of each of the synthesized versions of the abstracts, according to: speed of synthesis (running alone and with videoconferencing running concurrently). Have the same material presented via the audio on the SNAKE, but with synthesis being done on the ASKLEPiOS FreeTTS server (randomizing the order of presentation of the two versions of each abstract).

Evaluation Criteria:

Compare the ratings to determine the quality of SNAKE-synthesized speech with server-synthesized speech.

Although not strictly speech interfaces, the following two experiments are closely related to demonstrating and providing speech services.

Short Term (Year 1):

Unicode on small devices

Hypothesis:

Full Unicode support is possible on a handheld SNAKE.

Methods:

Attempt to install native Unicode support on the iPAQ by installing the Cyberbit glyph set (employing a microdisk or SD card for the required storage).

Evaluation Criteria:

Test the installation by accessing Unicode-containing web sites such as the IMC server.

ASK: an Internet search bot

As simple as it is (or, perhaps *because* it is so simple), *ALICE* offers intriguing possibilities. A demonstration exists in the form of SLIM (Sun Labs Instant Messenger), developed by collaborator Uhler, which looks at first like a conventional text-based Internet chat room. However, it also allows the participation of telephone-based participants (who hear the comments of others via speech synthesis, and whose comments are heard by the text-based participants through hypertext links which lead to the playing of audio files containing his/her spoken remarks). ALICE is another participant in this chat room, and can spawn searches of Google and of the Encyclopedia Brittanica based on queries posed to it during chat. We have modified SLIM, by developing AIML files for ALICE, to create a chatbot we call "ASK," which can describe our research; we will continue to develop AIML files which guide users in accessing NLM resources, searching PubMed and other NLM online systems based on chat interactions with users. ASK is intended to provide high-level guidance only.

Hypothesis:

ASK is a useful adjunct to NLM's web site.

Methods:

ASK will be implemented as described above. Two test groups of at least 20 members each will be composed, one representative of members of the general public, and the other of health care providers. The amount of prior exposure to NLM's web resources will be determined for each participant. Each subject will asked to devise ten medically relevant questions they wish to answer, and then will engage in a chat with ASK, and with the NLM web site, in an effort to find answers. One half of each group, randomly assigned, will use ASK first; the other half will use NLM's web site first. We will further evaluate this system by creating a test set of several hundred questions received by NLM from database users, posing them to ASK. Advice on where to find answers to these queries will also be obtained from a NLM librarian.

Evaluation Criteria:

Users will score their level of satisfaction for each query for both ASK and the web site, with respect to speed, ease of use, perceived usefulness of replies/content, and overall satisfaction. ASK will be compared to the "gold standard" librarian for correctness of advice.

The following studies combine speech recognition and synthesis work; we can anticipate better success with the (simpler, more mature) synthesis systems.

Intermediate Term (Years 2-3)

Verbal shell

At the border between OS and application levels of UNIX, one can envisage a "verbal shell," which can be used to interact with the computer via speech. The open source "Bourne-again shell," or *bash*, will be modified by adding "speak" and "listen" commands to its repertory, which will in turn draw upon the HTTP-based speech capabilities of the ASKLEPiOS server (care being taken to allow confirmation/correction of spoken input). A verbal shell will be useful both to issue commands to the computer in real-time, and to write standalone speech-based applications, using the shell as an interpreted computing language. The *Festival* and *flite* TTS systems have been used in this spirit in conjunction with Speakup [60], a

Linux OS TTS interface, and Emacspeak [61]. Such an approach requires the definition of a command vocabulary for controlling the shell's behavior (what to call ">", for example). Speech recognition systems are in use by, for example, software engineers with severe carpal tunnel syndrome. However, existing systems have been awkward to set up due to their being platform specific (for example, UNIX workstation programmers have had to have parallel PCs set up just for speech recognition, with additional software handling communication between the two systems).

Hypothesis:

A verbal shell will be slower for use than a keyboard-driven shell, but acceptably usable.

Methods:

The verbal shell will be implemented as described. Two test groups of non-visually-impaired subjects will be composed: one containing software engineers who are already accustomed to using speech technology for computer tasks other than dictation, the other with software engineers who have not used it or used it only for dictation. Both groups will be assigned several tasks, including the entry and execution of small Bourne shell programs.

Evaluation Criteria:

Users will rate speech synthesis for clarity, speed, and overall user satisfaction. Speech recognition will be rated by speed, recognition rate, and overall user satisfaction.

Add a speech interface to the ASK search bot

We will add ASKLEPiOS speech recognition/synthesis capabilities to the ASK search agent. The study design will be identical to that used for the initial ASK evaluation, but in addition, the recognition rate will be measured for speech recognition, the speed of query completions measured and compared to those of the earlier study when using typed input/output, and scores obtained from users for satisfaction with both recognition and synthesis.

Looking further ahead, exciting but more speculative projects (beyond the scope of the current proposal) would depend upon success with the projects described here. Both *sphinx 4* and *festival*/*FreeTTS* are capable of dealing with any human language. NLM could find and/or develop recognition and synthesis models for other languages (for example, Spanish) which might help serve its mission. One can envisage the multilingual content of NLM's Metathesaurus (of which ICPC is but a small part), serving at the core of an automated biomedical translation tool. Speech recognition could be used to index the content of audiovisual materials, along with timestamps for synchronization to the source (which would facilitate automated translation of the language as well).

9. Summary

The ASKLEPiOS project is exploring novel means of integrating non-mobile computer servers, telephones, and portable wireless hand-held devices (SNAKES), through a framework of open source tools facilitating PSTN-based and Internet SIP-based telephony, videoconferencing, and wireless data services, facilitated by a PSTN-to-Internet gateway, speech recognition/synthesis services, and a simple pattern-recognition-driven robotic chat service. The work is guided by the example of the World Wide Web in striving for simplicity, and seeking to achieve synergy between existing technologies by facilitating their closer integration.

Asklepios: the Myth, the Acronym (A PostScript)

The Myth:

As with most myths, the story of Asklepios appears in many variants, which agree on main points. His mother was mortal, his father the god Apollo. His mother was killed by Apollo's sister Artemis (sent to the task by Apollo) for falling in love with a mortal man, and Asklepios was snatched from her womb (by Apollo himself, in one telling) prior to her body being burned on a pyre. The infant Asklepios was taken by Apollo to a wise centaur, Chiron, who raised him and taught him the skills of healing. Married, he had as many as three sons (two of whom appear as physicians in Homer's *Iliad*), and as many as five daughters, the best known being Hygeia. Famed for his wisdom and skill, he was assisted by the goddess Athena through a gift of blood taken from the Gorgon, Medusa. Blood arising from a Gorgon's left side was deadly, but that taken from her right side could raise the dead. Asklepios met his end when Zeus struck him dead with a thunderbolt for raising a man from the dead. In some tellings, this was due to Zeus's distress in seeing Asklepios tampering with the natural order of the world; in Pindar's variant, Asklepios is killed because he performed this act in response to "... a lordly bribe, gold flashing in the hand ... [62]" (the social significance of the latter version has not been lost on contemporary medical professionals [63]). As with many acts of violence, the death of Asklepios produced a reprisal, when Apollo slew the Cyclops that had forged Zeus's thunderbolt. After an interval in Hades, Asklepios ended up as both a constellation (Ophiuchus, the "serpent-bearer") and a god. His name is still invoked today in the Hippocratic oath, which begins: "I swear by Apollo the physician, and Asklepios, and Hygeia, and Panakeia, and all the gods and goddesses, that, according to my ability and judgment, I will keep this Oath and this stipulation ..."

Asklepios's principal symbol, the *karykeion*, a single serpent wound around a wooden staff (and the origin of another of our acronyms, SNAKE) is widely used as a symbol within the modern medical profession (as, for example, by the AMA), but has often been confused with the staff of the god Hermes, the *caduceus*, which is winged and has *two* serpents coiled around it [64, 65].). This confusion is more common in commercial and military settings, and in the United States as compared to Europe.

The Acronym:

Given the symbolic importance of Asklepios to western medicine, it would be surprising not to find his name in wide use within the profession (his name has been given to medical clinics, pharmacies, museums, ...). After adapting the name as an acronym for the project described here, it was discovered to have also been used by an international collaboration in telemedicine and problem-based learning, involving personnel at the University of Missouri, Howard University, Yale University School of Medicine, the University of Regensburg, Aristotle University (Thessaloniki, Greece) and the University of La Laguna (Santa Cruz de Tenerife La Laguna, Spain) [38]. However, this group does not appear to have used the name as an acronym, was inconsistent in the use the Greek-derived (Asklepios) versus Latin-derived (Asclepius) spellings, and does not appear to have been active since 2001.

Acknowledgements

Ziying Sherwin provided important programming support for all phases of the project; summer student Yi Zhao helped install software on the iPAQ and in testing and installing *PhoneStation*; summer student Pavan Rudravarum updated software on the iPAQ and revised the *rat* GUI and audio to work better on this device. This work would not have been possible without the valuable advice and experience shared by many network colleagues, including: Jonathan Abbey (Applied Research Laboratory, University of Texas, Austin), Nelson Beebe (University of Utah), Michael Finke (Multimodal Technologies), Jamey Hicks (HP Laboratories, Cambridge), Orion Hodson (formerly of UCL, now Microsoft Laboratories), Emil Ivov (France), Doe-Wan Kim (ISI East), Mark Leisher (New Mexico State University), Kevin Lenzo (CMU, Cepstral), Bob Olson (ANL), Brian Schott (ISI East), Terrance Sullivan (Canada), Kabe Vanderlaan (Motorola Labs), William Walker (Sun Microsystems Laboratories), Thomas Wolff (Germany), Carl Worth (ISI East), and anonymous members of the handhelds.org *irc* chat room.

References

1. Weiser, M. Ubiquitous Computing. 17 March 1996. http://www.ubiq.com/hypertext/weiser/UbiHome.html.
2. Weiser, M. Some computer science problems in ubiquitous computing. Communications of the ACM. July 1993.
3. Weiser, M. The computer for the twenty-first century. Scientific American. September 1991: 94-10.
4. Miller, LE. Wireless ad hoc networks bibliography. NIST, 3 August 2004. http://w3.antd.nist.gov/wctg/manet/manet_bibliog.html.
5. Schatz, BR, Hardin, JB. NCSA Mosaic and the World Wide Web: Global hypermedia protocols for the Inter net. *Science* 1994; 265: 895-901.
6. Kunze, J, Rodgers, RPC. The ARK Persistent Identifier Scheme. July 2004. (Internet Draft, submitted to Internet Engineering Task Force).
7. Lamberts, H, Wood, M. ICPC: International Classification of Primary Care.. Oxford: Oxford University Press, 1987.
8. Okkes, IM, Jamoulle, M, Lamberts, H, Bentzen, N. ICPC-2-E: the electronic version of ICPC-2. Differences from the printed version and the consequences. *Family Practice* 2000; 17: 101-107. http://www.fampra.oupjournals.org/.
9. Unicode Consortium. The Unicode Standard, Version 4.0. Reading, MA: Addison-Wesley, 2003.
10. Rodgers, RPC, Sherwin, Z, Lamberts, H, Okkes, I. ICPC Multilingual Collaboratory (IMC): A web- and Unicode-based system for distributed editing/translating/viewing of the multilingual International Classification of Primary Care. Proceedings, MEDINFO 2004. San Francisco, California: 7-11 September 2004: (in press).
11. The AccessGrid Project. http://www.accessgrid.org/.
12. Schulzrinne, H, Casner, S, Frederick, R, Jacobson, V. RTP: a transport protocol for real-time applications. Internet Engineering Task Force, January 1996. Request for Comments (Proposed Standard), 1889.
13. Kouvelas, I, Hodson, O, Hardman, V, Crowcroft, J. Redundancy Control in Real-Time Internet Audio Conferencing. Proceedings of AVSPN 97. Aberdeen, Scotland, UK: September 1997.
14. Perkins, C, Kouvelas, I, Hodson, O, Hardman, V, Handley, M, Bolot, JC, Vega-Garcia, A, Fosse-Parisis, S. RTP Payload for Redundant Audio Data. IETF Audio/Video Transport Working Group, September 1997. RFC2198.
15. Perkins, C, Crowcroft, J. Effects of Interleaving on RTP Header Compression. Proceedings of IEEE Infocom 2000. Tel Aviv: March 2000.
16. Kouvelas, I, Hardman, V. Overcoming Workstation Scheduling Problems in a Real-Time Audio Tool. Proceedings of Usenix Annual Technical Conference. Anaheim, California: January 1997.

17. McCanne, S, Jacobson, V. vic: A Flexible Framework for Packet Video. ACM Multimedia. San Francisco CA: November 1995.

18. The Brazil Framework. http://research.sun.com/brazil/.

19. Singh, R, sphinx speech recognition systems: historical backdrop, T. 10 October 2003. http://www-2.cs.cmu.edu/˜rsingh/homepage/sphinx_history.html.

20. Lee, KF, Hon, HW, Reddy, R. An overview of the SPHINX speech recognition system. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 1989; 38: 35-46.

21. Placeway, P, Chen, S, Eskanazi, M, Jain, U, Parikh, V, Raj, B, Ravishankar, M, Rosenfeld, R, Seymore, K, Siegler, M, et al. The 1996 hub-4 sphinx-3 system. Proceedings, DARPA Speech Recognition Workshop. 1997.

22. Lamere, P, Kwok, P, Walker, W, Gouvea, E, Singh, R, Raj, B, Wolf, P. Design of the CMU sphinx-4 decoder. 8th European Conference on Speech Communication and Technology (EUROSPEECH 2003). Geneva, Switzerland: 1-4 September 2003.

23. SphinxTrain: acoustic modeller. http://www.speech.cs.cmu.edu/SphinxTrain/.

24. Sphinx Knowledge Base Tool. http://www.speech.cs.cmu.edu/tools/lmtool.html.

25. Black, AW, Taylor, PA. The Festival Speech Synthesis System: System documentation. Technical Report HCRC/Technical Report-83. Edinburgh, Scotland, UK: Human Communciation Research Centre, University of Edinburgh, 1997. http://www.cstr.ed.ac.uk/projects/festival.html.

26. Taylor, PA, Black, A, Caley, R. The architecture of the festival speech synthesis system. Third ESCA Workshop in Speech Synthesis. Jenolan Caves, Australia: 1998: 147-151.

27. Clark, RA, Richmond, K, King, S. Festival 2 - build your own general purpose unit selection speech synthesiser. Procedings, 5th ISCA workshop on speech synthesis. 2004.

28. The MBROLA Project. http://tcts.fpms.ac.be/synthesis/.

29. IMS German Festival. Institut fu:r Maschinelle Sprachverarbeitung, Universita:t Stuttgart. http://www.ims.uni-stuttgart.de/phonetik/synthesis/festival_opensource.html.

30. Black, AW, Lenzo, KA. Flite: a small fast run-time synthesis engine. Fourth ISCA Workskop on Speech Synthesis. Perthshire, Scotland, UK: 2001.

31. FreeTTS 1.2beta2 - A speech synthesizer written entirely in the Java programming language. http://freetts.sourceforge.net/docs/index.php.

32. Walker, W, Lamere, P, Kwok, P. FreeTTS - a performance case study. SMLI Technical Report-2002-114. August 2002.

33. A.L.I.C.E. Artificial Intelligence Foundation. http://www.alicebot.org/.

34. Weizenbaum, J. Computer Power and Human Reason. San Francisco, CA: W. H. Freeman, 1976.

35. Wallace, RS. The Elements of AIML Style. ALICE A.I. Foundation, Inc.. http://www.alicebot.org/.

36. Wallace, RS. Be Your Own Botmaster. ALICE A.I. Foundation, Inc.. http://www.alicebot.org/be.html.

37. Oddcast’s VHost SitePal. http://www.oddcast.com/sitepal/.

38. Asclepios Home PageDave E.S.L. Bot. http://lomiweb.med.auth.gr/asclepios/http://www.alicebot.org/dave.html.
39. Uhler, SA. PhoneStation, moving the telephone onto the virtual desktop. Proceedings, USENIX Winter Conference. San Diego, California: USENIX Association, 25-29 January 1993: 137-140.
40. Ousterhout, J. TCL: an embedded command language. Proceedings, USENIX Technical Conference. Washington, DC: USENIX Association, January 1990: 133-146.
41. Session Initiation Protocol (SIP). http://www.cs.columbia.edu/sip/.
42. Janak, J. SIP Introduction. http://www.iptel.org/ser/doc/sip_intro/sip_introduction.html.
43. iptel.org SIP Express Router. http://www.iptel.org/ser/.
44. Asterisk: the open source Linux PBX. http://www.asterisk.org/.
45. Persistent identifiers. Preserving Access to Digital Information (PADI). National Library of Australia, August 2002. http://www.nla.gov.au/padi/topics/36.html.
46. Manheimer, K, Warsaw, B, Clark, SN, Rowe, W. The *Depot*: A Framework for Sharing Software Installation Across Organiz ational and UNIX Platform Boundaries. LISA IV. 17-19 October 1990. (Available at: `http://www.forwiss.uni-passau.de/archive/marchiv/systemverw altung.html`).
47. opt_depot. ARL, University of Texas at Austin. (Available at: `http://www.arlut.utexas.edu/csd/opt_depot/opt_depot.html`).
48. Rodgers, RPC, Sherwin, Z. A Management System for Network-Sharable Locally Installed Software: Merging RPM and the Depot Scheme under Solaris. San Diego, CA: 2-7 December 2001: 267-272. Proceedings, 2001 LISA XV.
49. Laurie, B, Laurie, P. Apache: The Definitive Guide, Second Edition. Sebastopol: O'Reilly & Associates, 1999.
50. Choi, W, Kent, A, Lea, C, Prasad, G, Ullman, C. Beginning PHP4. Birmingham: Wrox Press Ltd., 2000. (775 pages).
51. DuBois, P. MySQL. Indianapolis: New Riders, 2000. (756 pages).
52. Linux and PDAs (white paper). http://www.clavib.com/linux_pda_white_paper.asp.
53. LinuxDevices.com. http://www.linuxdevices.com/.
54. handhelds.org. http://www.handhelds.org/.
55. Emerging Technologies Group (ETG). http://etg.nlm.nih.gov/.
56. Distributed Systems (PADS) iPAQ Research. http://pads.east.isi.edu/.
57. Stankovic, J, Neteler, M, Flor, R. Experimental mobile wireless GRASS based GIS for handheld computers running GNU/Linux. Proceedings, Open Source GIS GRASS Users Conference. Trento, Italy: 11-13 September 2002. http://grass.itc.it/grasshandheld.html.
58. Liu, H. Wi-Fi Based Wireless Mobile Audio and Video Conferencing. 26 January 2004. http://www.ics.uci.edu/˜haining/ipaq/wifi-av.html (haining AT ics.uci.edu).
59. Labs, M. Linux iPAQ: Mobile Multimedia Platform. http://internet2.motlabs.com/ipaq/index.htm.
60. Speakup. http://www.linux-speakup.org/speakup.html.

61. Emacspeak. http://emacspeak.sourceforge.net/.
62. Greek Mythology: ASKLEPIOS God of Medicine. http://www.theoi.com/Kronos/Asklepios.html.
63. Murphy, JP. Asklepios, God of MedicineAsklepios: Hero of Medical Caring 03-27-96. 18 April 1996. http://www.willamette.edu/cla/exsci/asklepios.htm ‘http://w3.trib.com/˜murphy/04-18-96.htm.
64. Blayney, KT. The Caduceus vs. the Staff of Asclepius. http://drblayney.com/Asclepius.html.
65. Stolinsky, DC. Asklepios. *Ann. Int. Med.* 1 August 1996; 125: 253-254. letter: http://www.annals.org/cgi/content/full/125/3/253-c.

Curriculum Vitae

R. P. Channing Rodgers

Clinical Specialty Consultant

Education and Training:

American Board of Pathology	Diplomate in Clinical Pathology	1984
University of Utah College of Medicine	MD	1976
Harvard College	BA *cum laude* (Biochemical Sciences)	1972

Research and Professional Experience:

Clinical Specialty Consultant, Lister Hill National Center for Biomedical Communications, National Library of Medicine, National Institutes of Health, Bethesda, MD, 1992-present

Assistant Research Chemist, Department of Pharmaceutical Chemistry, University of California, San Francisco, CA, 1987-1992

Assistant Professor, Department of Laboratory Medicine, University of California, San Francisco, CA, 1984-1992

Resident in Laboratory Medicine, Department of Laboratory Medicine, University of California, San Francisco, CA, 1982-1984

ICP Fellow, Unité de Médecine Expérimentale, International Institute of Cellular and Molecular Pathology, Université Catholique de Louvain, Bruxelles, Belgium, 1980-1982

Research Associate, Laboratory of Theoretical Biology, National Cancer Institute, National Institutes of Health, Bethesda, MD, 1978-1980

Research Fellow, Department of Nuclear Medicine, Middlesex Hospital Medical School, University of London, London, UK, 1977-1978

Resident in Internal Medicine, St. Mary's Hospital & Medical Center, San Francisco, CA, 1976-1977

Selected Pertinent Activities:

Member, Standards Development Committee, National Information Standards Organization (NISO), 2004-present

Participant in Internet Engineering Task Force (IETF) standardization activities, 1992-present

Founding Member, International World Wide Web Conference Committee (and chair of live audio-video multicasts, conferences II - VII), 1994-1998

Founding Chair, NSF/NCSA World Wide Web Federal Consortium, 1994-1995

Created and maintained HyperDOC, a suite of content and applications comprising NLM's first public World Wide Web service, 1993-1995

Organizer of special Interest Group for Networked Information Discovery and Retrieval (SIGNIDR III; one of the first scientific/technical conferences to be multicasted live over the Internet, and the first from NIH), 1993

UCSF-developed software packages licensed through Office of Technology Licensing, University of California, Berkekey (Enhanced troff/TranScript document preparation system, bibIX bibliographic preprocessor for ETT system, academIX office tools, System Manager's Toolkit), 1992

Selected Publications:

1. Rodgers, RPC, Sherwin, Z, Lamberts, H, Okkes, I. ICPC Multilingual Collaboratory (IMC): A web- and Unicode-based system for distributed editing/translating/viewing of the multilingual International Classification of Primary Care. Proceedings, MEDINFO 2004. San Francisco, California: 7-11 September 2004: (in press).
2. Kunze, J, Rodgers, RPC. The ARK Persistent Identifier Scheme. July 2004. (Internet Draft, submitted to Internet Engineering Task Force).
3. Rodgers, RPC, Sherwin, Z. A Management System for Network-Sharable Locally Installed Software: Merging RPM and the Depot Scheme under Solaris. San Diego, CA: 2-7 December 2001: 267-272. Proceedings, 2001 LISA XV.
4. Rodgers, RPC. Searching for Biomedical Information on the World Wide Web. *J Med Pract Man* 2000; 15: 306-313.
5. Rodgers, RPC. Java and Its Future in Biomedical Computing. *JAMIA* 1996; 3: 303-307.
6. Stites, DP, Rodgers, RPC. Clinical laboratory methods for detection of antigens and antibodies. In: Stites, DP, Terr, AI, eds., Basic and Clinical Immunology. Norwalk, Connecticut: Appleton and Lange, 1996. Ninth edition.
7. Rodgers, RPC, Srinivasan, S. On-Line Images from the History of Medicine (OLI): Creating a Large Searchable Image Database for Distribution via World-Wide Web. Geneva: 25-27 May 1994: 423-431. Proceedings, The First International World-Wide Web Conference.
8. Rodgers, RPC, Baddeley, A. *Journal of Applied Probability* 1991; 28: 539-552.
9. Rodgers, RPC, Levin, J. A critical reappraisal of the bleeding time. *Semin Thromb Hemostas* 1990; 16: 1-20.
10. Beebe, NHF, Rodgers, RPC. <PLOT79>: A comprehensive portable Fortran scientific line graphics system, as applied to biomedical research. *Comput Biol Med* 1989; 19: 385-402.
11. Rodgers, RPC. How much quality control is enough? A cost-effectiveness model for clinical laboratory quality control procedures (illustrated by its application to a ligand-assay-based screening program). *Med Decis Making* 1987; 7: 156-167.
12. Rodgers, RPC. Data analysis and quality control of assays: a practical primer. In: Butt, WR, ed., Practical Immunoassay: The State of the Art. New York: Dekker, 1983: 253-308.
13. Rodgers, RPC, Tannen, R. Rapid and accurate determination of total lung capacity (TLC) from routine chest radiograms using a programmable hand-held calculator. *Comput Biol Med* 1983; 13: 125-140.
14. Rodgers, RC, Hill, GE. Equations for vapour pressure versus temperature: derivation and use of the Antoine equation on a hand-held programmable calculator. *Br J Anaes* 1978; 50: 415-424.

Ziying Sherwin
Curriculum Vitae

Education

University of Delaware, Newark, Delaware, 1997-1999
Master of Science: Computer and Information Sciences
Advisor: Dr. Keith Decker
Zhejiang University, Hangzhou, Zhejiang, China, 1992-1996
Advisor: Dr. Miaoliang Zhu
Bachelor of Science: Computer Science and Engineering

Experience

Senior System Engineer, Contractor of Lister Hill National Center for Biomedical Communications, National Library of Medicine, National Institute of Health, Bethesda, MD, 2000-present
Software Engineer, Bell Atlantic, Arlington, VA, 1999-2000
Research Assistant, Natural Language Processing and Artificial Intelligence Laboratory, University of Delaware, Newark, DE, 1997-1999
Teaching and Research Assistant, Center for Artificial Intelligence, Zhejiang University, Hangzhou, China, 1996-1997
Computer Programmer, Fire and Rescue Dispatch and Operations Center, Hangzhou Fire Department, Hangzhou, China, 1995-1996

Publications

"ICPC Multilingual Collaboratory: a Web-and Unicode-Based System for Distributed editing/Translating/Viewing of the Multilingual International Classification of Primary Care", RPC Rodgers and Ziying Sherwin, MEDINFO 2004, in process
" A Management System for Network-Sharable Locally Installed Software: Merging RPM and the Depot Scheme under Solaris",RPC Rodgers and Ziying Sherwin, LISA XV fifteenth system administration conference, 2001
"A Chinese Character Filter for Map", Ziying Shao and Miaoliang Zhu, Journal of Computer Aided Design and Comput
er Graphics, China, Vol. 10, No.2, 1998.

Biographical Sketches of Collaborators

Timothy Hamill, MD (University of California, San Francisco)

Dr. Hamill is Clinical Professor in the Department of Laboratory Medicine, University of California at San Francisco (UCSF), and Director of the UCSF Clinical Laboratory. He has had longstanding interests in computer applications to his field, and collaborated with Nikon in the development of a robotic microscope for clinical pathological examinations.

John Kunze

John Kunze is a senior development architect in the California Digital Library's preservation program. His current focus is the creation of long-term durable digital references (ARKs) to information objects, backed up by standards for kernel metadata and permanence ratings, and by search systems that make them discoverable. Recently he has worked at the University of California, San Francisco, and at the U.S. National Library of Medicine. John's background is in computer science and information systems. He designed, implemented, and ran UC Berkeley's first campus-wide information system, which was an early rival and client of the World Wide Web. The system also demonstrated the first interoperation of the Z39.50 library search and retrieval protocol. John made major contributions to the standardization of URLs, Dublin Core metadata, and Z39.50. In an earlier life he was a BSD Unix hacker whose work lives on in today's Linux and Apple operating systems.

Enrique Terrazas, MD, MS (University of California, San Francisco)

Dr. Terrazas is Clinical Assistant Professor within the Department of Laboratory Medicine, University of California at San Francisco (UCSF), Director of the Mount Zion Hospital Clinical Laboratory (a UCSF-affiliated institution), and Director of the Clinical Pathology Residency Program at UCSF. He is very active in developing and supporting information systems within the Department of Laboratory Medicine.

Stephen Uhler (Sun Microsystems Laboratories)

Stephen Uhler worked on the technical staff at Bell Laboratories (Whippany, NJ) on user interface management systems, then joined Bell Communications Research (Bellcore) when it was founded in 1984. There he devised MGR, a network-distributed windowing system which predated both X and NeWS, and *PhoneStation*. After a brief stint with a Bay Area startup, he joined Sun Microsystems Laboratories n 1993 as a Senior Staff Engineer and Principal Investigator. Leader of the Web Applications Technologies group, he has been responsible for exploring the convergence of web application servers, proxy technologies, and wireless PDAs to provide architectures for the next generation of web services. He has written parts of the TCL programming language, invented the Brazil framework for Java-based network applications, and devised numerous demonstrations of novel computing applications, ranging from the use of power-line networks for home automation to the use smartcards for authentication tasks. Mr. Uhler is recipient of the Sun Presidential Award. (See full resume at: http://pages.sbcglobal.net/sau/resume.pdf).