Technical Plan

Purpose

Building an enterprise that provides single sign on access to a host of resources through an integrated architecture is the foundation for many of the services we will build. Web portals will enable us to provide rich and immediate access to resources and applications from a personalized view. The challenge that we face is to build an enterprise portal that takes advantage of the framework and architecture that exist at the University of Cincinnati Medical Center. This technical plan describes *how* we are going to develop the enterprise. It includes the following major topics:

1. Framework
   - Customer Access, Resources and Services
   - Applications
   - Architecture
   - Database
   - Servers
   - Directories
   - Network
   - Information Policy

2. Architectural Approach
   - Multi-tier Architecture
   - Middleware
   - Smart Digital Services

3. Integrated Database
   - Database Design
   - Personal Profile

4. Knowledge Management

5. Knowledge Integration
   - Consultant (Push)
   - Requestor (Pull)
   - Alerter (Alert)
   - Expert (Share)
6. Unified Medical Language System® (UMLS®)
   - UMLS® Relationships with the Projects

7. Definitions

8. Bibliography

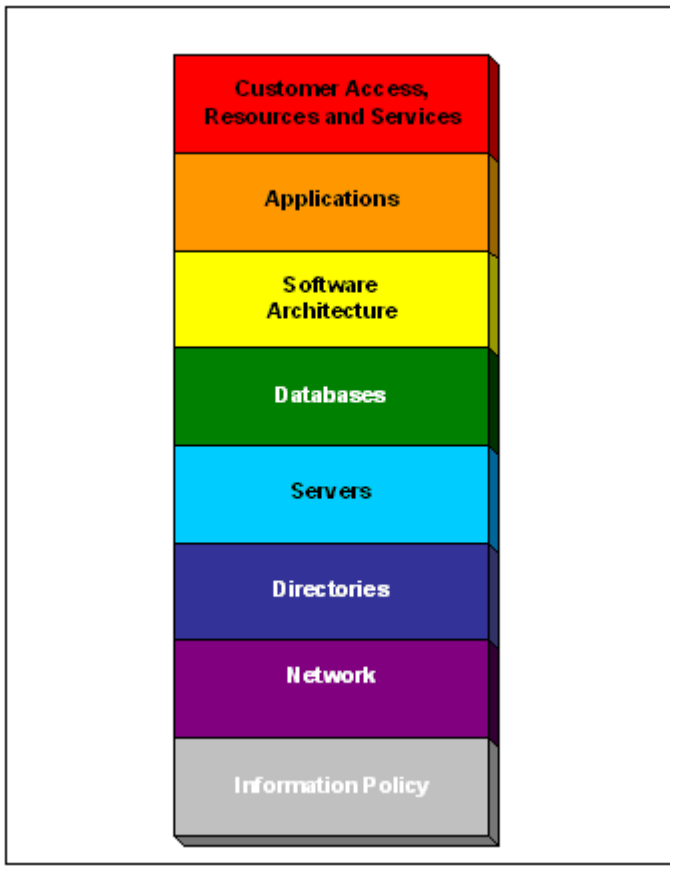
1. Framework

The University of Cincinnati has developed an eight layer information technology framework. Each layer of the information technology framework defines a major grouping of functionality.

The Medical Center relies on the University of Cincinnati Office of Information Technologies to provide the technical infrastructure for the entire University, including the Medical Center. This excellent technical infrastructure is represented by the lower layers.

The upper layers of the framework represent in the aggregate how we plan to implement our distributed computing model. All new development will continue to be based on a multi-tier architecture with web-oriented tool sets.

For a full description of this framework and the components in each layer, see Appendix 3.



2. Architectural Approach

Multi-Tier Architecture

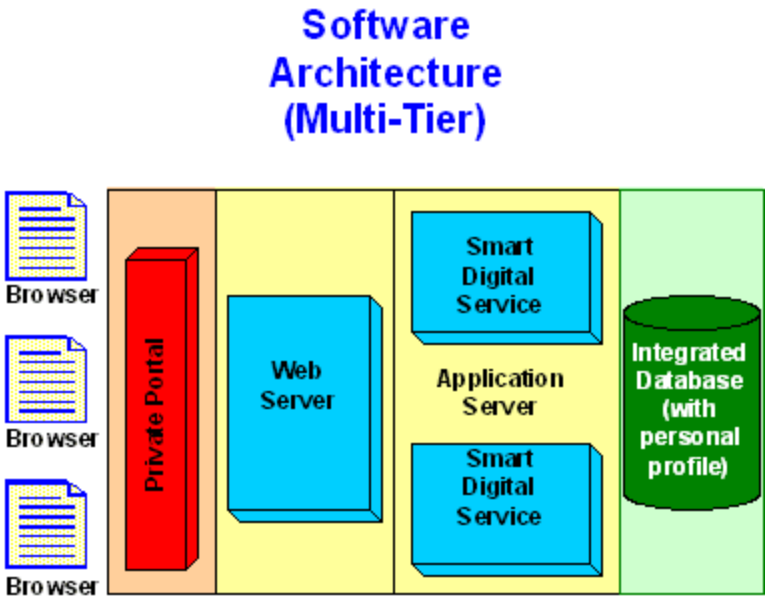
The distributed multi-tier architecture is partitioned into three or more levels:

- Client Browser and Private Portal
- Smart Digital Services (Application Components)
- Database Services

This multi-tier architecture provides a system that is easy to “scale”, to increase capacity as needed. Also, the components managed on the middle application tier can provide for a high level of sharing and reuse. Distributing more or less work to the middle tier, by using a thin or thick client, allows scaling of the system to balance the expected hardware capabilities of the clients vs. the application server. At the database tier, various techniques are available to expand the system as the volume of data and activity increases. The following diagram describes the high-level architecture.

Middleware

Middleware will be used as the implementation mechanism to connect disparate applications and media such as mainframe legacy applications and web based applications as well as to develop new mission critical applications. The Medical Center has used a series of Middleware environments since 1996. These include PERL, Web Builder (UNIX and NT environments), Cold Fusion (built on a C++ Foundation), and ASP (Microsoft's Middleware environment). Future implementations of middleware technologies will be used to develop smart digital services which will enable us to:



- Build distributed web applications on existing integrated database using modern technology from Microsoft (.NET), Macromedia, and Sun (Java 2 Enterprise Edition, J2EE, and Java 2 Micro Edition, J2ME)

- Adhere to information technology development standards in languages, meta languages (W3C: SGML, HTML and XML), database technologies and evolving standards such as web services and simple object access protocol (SOAP). Strong integration with XML will enable us to build a meta directory of all services, applications, and various types of content and media. Meta mapping will enable us to provide rich definitions at a highly detailed level using standards such as The Unified Medical Language System (UMLS).
- Integrate successfully with various types of media and architectures such as mainframe systems and multimedia based environments

Future middleware environments at the Medical Center may include the following:

- *Cold Fusion MX* – The newest version of Cold Fusion, which is currently in beta testing, provides an application framework built on top of J2EE. Built-in hooks for Flash and other multimedia applications enable rich multimedia integration with a strong database emphasis. Cold Fusion also has strong integration with open standards such as XML, WML and others.

- *Macromedia JRUN* – Certified J2EE compatible Java application server that empowers you to develop and deploy Java applications quickly with Java Servlets, JSP, JTA, JMS and EJBs.

- *.NET Framework* – It enables software integration through the use of XML Web services. .NET connected software delivers a strong toolset needed to create XML Web services and stitch them together.

- *Websphere* – Mainframe J2EE compatible integration will be required to communicate with extant legacy systems

Smart Digital Services

The middleware will allow us to build smart digital services. *Smart digital services* are defined as shared functionality ("components") that span multiple applications and are defined not by boundaries of a given system but by the *customer's personal profile* which defines a person's knowledge needs. The objective is to develop sharable smart digital services built for change that provides specific knowledge to an infinite number of customers through personalized portals

The software development process that will be used (the unified process) is component-oriented and uses an appropriate open standard implementation language, either object-based or object-oriented. Each intelligent software component known as a smart digital service that is developed will contain a well-defined interface and encompass a set of rule-driven business functions. Each smart digital service will be developed for reuse and includes a set of one or more objects and actions that are implemented in a standard language. The objects and actions are represented by a set of software classes. Usability testing will be performed against builds (releases) early in the process so that changes can be accommodated at the lowest cost point in the project.

Systems will be developed by assembling pre-built smart digital services rather than building them from scratch. This will allow us to achieve higher degrees of flexibility and accommodate rapid change.

3. Integrated Database

The purpose of an integrated relational database is to only enter data once and then to share it broadly. The relational database allows for the managed of business rules and creation of personal profiles which are the dynamics behind the smart digital services.

Database Design

The structure of the integrated database can be visualized as a daisy: the Person Core is the center of the flower, and surrounding the center is a collection of petals, each holding the data used by a specific application. Each application manages its own data petal, but all applications get the Person data they require from the Person Core. In this way, it is possible to coordinate data in different applications linked to a single person.

Because all applications use the same Person Core, as soon as someone is added to the Person Core data about that person is available for use by all applications. A person might be added to the Person Core when he/she is first employed at the university or when an application adds that person for its own use. For example, when a non-university person registers for a Continuing Medical Education course a Person Core record is created for that person; immediately, that person is now available to all other applications linked to the Person Core, including the Research Training system.

Although each application manages its own data, applications can share data on an as-needed basis. For example, when a person becomes a researcher on a protocol managed by the Institutional Review Board (IRB) system, that person is required to take certain Research Training courses. Information about that person's new requirements is posted to the Research Training system for its use. Compliance information is available to the IRB system, when the Research Training system records the information that a particular requirement has been met.

The overall design of the database provides a framework for delivering applications across the web. A security module manages access and control issues so that different users have access to the parts of each application relevant to their needs. We are currently continuing the development of the Medical Center integrated database, a repository of personnel information for UC faculty, student, staff, and affiliates, and other people connected in various ways with the Medical Center. The database has a Person Core of key demographic information, which is available to systems at the Medical Center. Each system stores and maintains its own information in petals of the database; data entered by one system may be totally private, or parts of the data may be shared with other systems as appropriate. We are using this design to improve the sharing of information across the Medical Center. A side benefit is a reduction in the amount of data that is entered many times, for the use of different systems.

All the databases developed at the Medical Center utilize the Relational Database Model. Most of the databases are based on SQL Server, running on a Windows NT / Windows 2000 platform. Some data at the University is managed by Oracle databases.

Personal Profile

Each person will have a personal profile, listing his/her needs and levels of expertise. The common profile will be updated manually, by the person, and automatically from existing systems. The role of the personal profile is to determine the types of knowledge that are to be quality filtered for each person. For example, the personal profile data is used by the *Consultant* to push to each person at the Medical Center "context-appropriate" information – the information relevant to their specific tasks, positions, and needs, organized and presented for effective use – in a timely fashion, without overloading them with irrelevant information.

In addition to profile information determined by positions, roles and other items, the customer will be able to modify their private portal to add knowledge of interest and suppress knowledge that their profiles selected, but they do not want to see. They will also be able to specify what personal or professional knowledge they want made generally available, or made available to selected groups within the university.

For example, a Professor of Internal Medicine (job) may be conducting research overseen by the IRB (role) that involves the use of radioactive material (role), and is a member of an IRB Review Committee (position).

The integrated database contains the 'jobs' and 'roles' data; people can use a self-identification questionnaire to specify their 'positions' data. Based on this information, the database will automatically maintain a Personal Profile for each person. When people log onto the Medical Center

web site and connect to their private portal, they will see links to their context-appropriate information. (This information is pushed to them, based on their profiles.) They may also be alerted to important upcoming events, such as a Professional Training certification that will expire soon, or a seminar of interest in one of their primary research areas.

4. Knowledge Management – Enterprise Web Portal

The unifying approach of knowledge management is to design smart digital services to acquire, produce, store, distribute and integrate applications, content and business processes. The focal point will be an enterprise portal technology that provides a single point of access for each person. The enterprise web portal is an internet-based application that can be personalized to accommodate customer needs. The portal, utilizing the personal profile, is able to access organized knowledge from any device, any browser, anywhere, anytime and any place. Several vendors currently offer portal environments. Generally, these environments are proprietary and require a wide array of adjustments and modifications to preexisting frameworks and structures. We plan to build a portal environment that uses our current structure and framework and provides flexibility and the ability to change as needed. Emerging standards in portal technologies will be supported. Two standards are currently being developed.

- Java community processes' Java Specification Request 168 (JSR 168)
- Organization for the Advancement of Structured information Standards' Web Services for Remote Portals (WSRP)

The enterprise portal is an effective mechanism to counterbalance information overload by allowing for more precise, filtered knowledge (Pull, Share) and distribution (Push, Alert) of specific knowledge.

For example, a researcher who is an expert on drug therapies for AIDS can specify that he is willing to be a public speaker on AIDS drug therapies, but is not interested in having his name made available to people seeking research collaborators. Also, he can request that information about bioinformatics activities be added to his private portal page.

5. Knowledge Integration

Knowledge will be integrated by organizing it into function-based rules and self-describing structures that it can be accessed semantically (by matching profiles and other criteria) and applying sets of smart digital services. Knowledge will be organized in a location independent manner accessible by various types of internet-enabled devices using both wired and wireless technology. The knowledge-based services will be built using business rules, smart digital services and a modern integrated relational database. Four examples of knowledge-based smart digital services that distribute knowledge are the following:

- *Consultant* (Push)
- *Requestor* (Pull)
- *Alerter* (Alert)
- *Expert* (Share)

Consultant (Push)

A set of smart digital services (components) known as the *Consultant* will push or deliver specific personalized knowledge to each person based on their profile, preferences, wants and needs. The *Consultant* will assist people in sorting through the masses of knowledge automatically. Current examples of Push technology exist in several applications:

Research Training enables researchers and administrators to log in and based on their profile gain access to information and resources specific to their needs. Faculty or Staff can see a list of required courses as well as completed courses. Certification information is also available.

ePAF provides access to specific employee information based on their profile. The single login/authentication module used in ePAF, Research Training and other systems accesses profile information and builds dynamic interfaces and access to resources based on their criteria.

Other implementations of push technology will include:

- Portfolio-based credentialing: distribution of grades.
- Bioinformatics: new research results to be distributed based on a person's profile.
- Research administration: funding opportunities and appropriate progress report forms to be distributed to researchers.

Requestor (Pull)

The customer can pull or query the requestor smart digital service for ad hoc knowledge. The requestor will guide the customer to achieve the most effective appropriate knowledge based on their request.

Current implementations of pull technology exist in many applications. NetWellness' Ask an Expert is an example. People have the ability to search through over 18,000 questions and answers using a multi pronged filtering process. Viewing by area of interest and ad hoc searching either globally or by interest area enables individuals to drill down to a specific answer or set of answers that meet their criteria.

Other implementations of pull technology will include:

- Portfolio-based credentialing: students requesting remediation resources
- Bioinformatics: ad hoc search capability to locate specific bioinformatics tools that solve specific problems.
- Research administration:
    - Ad hoc search for specific funding opportunities
    - Report for compliance for departmental personnel

- Concise reports, highly processed, to allow the user to quickly draw conclusions about research data
- Conduct data mining by sifting through large volumes of biological data uncover patterns
- Robust content management systems that provide web site information in a clear and concise manner.

Alerter (Alert)

Sometimes you need to be quickly informed of critical events. Alerts are high priority knowledge-based messages. The alerter addresses the type of knowledge needs to be sent to a person immediately based on their role and the context. These event messages would typically be sent proactively to e-mail, pagers and cell phones.

Current examples of alert technology exist in several applications. eContracts, eGrants, and ePAF are all built to alert administrators, researchers, and other staff when a contract, grant, or personnel action form is ready for their stage of processing. The alert mechanism is either triggered by an event or scheduled automatically in many cases to perform needed functions. In Research Training a scheduled alerting module alerts people when it is time for them to renew a given training requirement.

Other examples of alert technology will include:

- Portfolio-based credentialing:
  - Notification to faculty of student's performance deficiency
  - Notification to faculty of student's positive performance

- Research administration:
  - Notification of over due IRB/IACUC progress report
  - Notification of over due grant progress report

- Bioinformatics:
  - Notification to researcher of new bioinformatics tool.

Expert (Share)

The *Expert* is a way of collaborating between people that have knowledge that can be shared. This requires that we pre-identify people that have similar interests and are willing to respond to specific questions. A person would have "key words" in their profile about areas of expertise. A question about a specific topic would be submitted to a collaboration tool, the *Expert*, and then dispatched to a person who is likely to have the skills to have an answer. The answer would be returned to the originator and added to the knowledgebase for use by others.

Current examples of sharing technology are most prevalent in the NetWellness Ask an Expert system. The Ask an Expert system currently supports over 150 medical experts who have expertise in over 45 health topic areas. Questions submitted into this system are automatically dispersed to experts who have been identified as having one or more areas of expertise. The system fully supports

moderated and non-moderated areas. A strenuous quality assurance process ensures the integrity of the information provided. Any answer submitted via this system is automatically added to the expert knowledgebase and available to all who access the site.

Other implementations of share technology will include:

- Portfolio-based credentialing: a student who is studying … wants to know …

- Bioinformatics: a researcher who is doing studies in … wants to ask a another researcher in the same field about what tool to use to address a specific class of problems.

- Research Administration: a researcher who is doing studies in … wants to ask another researcher in the same field about …

6. Unified Medical Language System® (UMLS®)

The purpose of this section is to describe the applicability of Unified Medical Language System® (UMLS®) in relation to the three projects. The proposal includes three projects: portfolio-based credentialing, bioinformatics, and research administration. All projects are based on a common architecture consisting of SQL Server backend, web application front end, personal profiles automatically generated and modified by individuals, and web services relying on profile data. The smart digital services built on this architecture *push* context-appropriate data to individuals, allow individuals to *pull* desired information from system, *alert* individuals to items of priority, and allow people to *share* information with others.

1. *Portfolio Credentialing:* The goal is to develop a digital portfolio documenting student/resident training in clinical skills. Students/residents will use PDAs to record data describing clinical skills sessions and supervised patient encounters. Instructors and supervising physicians will be able to assess student /resident performance in near real-time. Reports of performance will be available soon after sessions, allowing for continuous improvement in skills based on feedback. Instructors will be alerted to students with poor performance, and may push supplementary materials to them. This model can be expanded in the future to allow physicians to capture patient care data real-time, rather than dictating reports after-the-fact for processing by the medical transcriptionists.

2. *Bioinformatics*: The goal is to implement a new learning model in which researchers can specify research interests and knowledge, and pull links to appropriate training materials, research materials, and scientific resources to their portals

3. *Research Administration*: The goal is to streamline the workflows of administrative systems that facilitate research efforts. There are many related projects. The main ones relevant to UMLS are the development of an Expertise system where researchers can use a controlled vocabulary (MeSH headings) to describe their research interests and activities and several activities related to notifications about research opportunities. Based on the information in the Expertise database, researchers will be alerted to potential appropriate funding opportunities; can seek collaborators

for potential projects; can share their research interests so that others can seek them out as collaborators; and can pull links to relevant publications to their personal portals.

UMLS® Relationships with the Projects

1. *Portfolio Credentialing*: The potential in the future to use the CHMC UMLS parsing engine under development to parse clinical experience narratives into standard terminology, to automate verification of certification requirements (certain numbers of cases meeting specified criteria.) No likely application for push, pull, alert, and share concepts. If concept is extended in future for use by physicians, potential application is enhanced in value.

2. *Bioinformatics:* UMLS may provide enhanced ability to provide scientific resources, by resolving descriptions of research interests to UMLS terms prior to searching.

3. *Research Administration:* The project uses controlled vocabulary. The potential to use UMLS to enhance features, by letting researchers to seek out potential collaborators, share information with potential collaborators, and pull notifications of relevant publications without restriction to a controlled vocabulary. Also, use of UMLS may expand range of publications searched beyond those with MeSH indexing.

7. Definitions

Application components: component objects, smart digital services, deployed on an application server, which handle primarily application logic and business rules, but also are used for distributed computing services. This “middle” tier is deployed on an application (transaction) server.

Architecture-the fundamental organization of the system as a whole, just as the architecture of a building is the organization of the spatial and design elements of the building to meet the client’s functional needs.

Browser (Client) Front End: A graphical user interface, usually handled by a browser. This tier is deployed on an independent client. . This tier can be implemented as either a “thin” or “thick” client. For systems using thin clients, processing that may occur on the client side includes validation, rich multimedia support, interactive user interfaces and other functions to enhance or enrich the user’s experience; for system with thick clients, much of the application processing is distributed to the clients and less happens in the middle tier.

Database Services: The database underlying the entire system, which accesses information for processing and updates information as required, is based on relational and/or object-oriented database products, and is deployed on a database server.

Distributed computing model: processing functions and data can reside anywhere on the network or the commercial Internet. There is less reliance on proprietary software and more emphasis on open standards. With the distributed model, applications are scalable by adding more servers or upgrading existing servers and cost substantially less expensive then mainframes.

Middleware: the software libraries and tools that address the implementation needs of the distributed computing model. Middleware enables the creation, integration, and management of large-scale, distributed applications in both homogeneous and heterogeneous environments.

Multi-tier architecture: an architecture that is based on the distributed computing model. Multi-tier applications share the processing, formatting, presentation, and storage functions across clients and servers.

Smart digital services: Smart digital services are defined as shared functionality ("components") that span multiple applications and are defined not by boundaries of a given system but by the individual's personal profile which defines a person's knowledge needs. Also, see application components.

8. Bibliography

Len Bass, Paul Clements and Rick Kazman. Software Architecture in Practice. Addison-Wesley, 1998.

Grady Booch. Object Solutions: Managing the Object-Oriented Project. Addison-Wesley, 1996.

Grady Booch, James Rumbaugh and Ivar Jacobson. The Unified Modeling Language User Guide. Addison-Wesley, 1998.

R.G.R. Cattell. Object Data Management. Addison-Wesley, 1991.

Jim Conallen. Building Web Applications with UML. Addison-Wesley, 2000.

Margaret A. Ellis and Bjarne Stroustrup. The Annotated C++ Reference Manual. Addison-Wesley, 1992.

Martin Fowler. Analysis Patterns: Reusable Object Models. Addison-Wesley, 1996.

Martin Fowler. UML Distilled: Applying the Standard Object Modeling Language. Addison-Wesley, 1997.

Martin Fowler. UML Distilled. Addison-Wesley, 1999.

Erich Gamma. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.

Stanley B. Lippman. C++ Primer, Second Edition. Addison-Wesley, 1991.

Stanley B. Lippman, Josee Lajoie. C++ Primer, Third Edition. Addison-Wesley, 1998.

Scott Meyers. Effective C++. Addison-Wesley, 1992.

Kendal Scott. UML Explained. Addison-Wesley, 2001.

Bjarne Stroustrup. The C++ Programming Language, Second Edition. Addison-Wesley, 1991.

Clemens Szyperski. Component Software. Addison-Wesley, 1998.

David A. Taylor. Object Technology: A Manager's Guide. Addison-Wesley, 1997.

Peter van der Linden. Just Java. Sun Microsystems Press, Prentice Hall, Fourth Edition, 1999.

Ann L. Winblad, Samuel D. Edwards, David R. King. Object-Oriented Software. Addison-Wesley, 1990.