CEHR Net
Large Scale Connectivity for Personal Medical Records
Technical Strategy and Open-Source Implementation

DRAFT: Version 0.3, February 4, 2003

Editors
Bryan Tripp, University Health Network (bryan.tripp@uhn.on.ca)
Jim Forbes, University Health Network (jim.forbes@uhn.on.ca)

Contributors
Neal Acharya, University Health Network
Alexei Guevara, University Health Network
Damian Horton, Courtyard Group
John Waldron, University Health Network

Acknowledgements

The clinical pilot of a prototype system has been critical in the development of this strategy. This project relied on the work of Mount Sinai Hospital, Cerner Corporation, Per Se Technologies, and Innovision Corporation.

Contents

Chapter One: Overview

Introduction

This document discusses a technical strategy for patient-controlled dissemination of clinical observations to all the health care providers that a patient sees over a lifetime.

Over time, a single person may receive professional health care services from many clinicians, at many different health care facilities. Increasingly, clinical data are stored electronically, improving availability within a single facility. There is also growing interest in making these data available to other providers that subsequently treat the same patient (e.g. to avoid gaps in medical history; to reduce duplication of previous investigations). Pockets of success have been achieved in this area, but existing solutions are far from the ideal of seamless integration on a regional or provincial scale.

We will present a new technical approach to this problem, based on a distributed network of services. This chapter discusses the key concepts and strategic decisions. Later chapters present various views of the proposed system, following the ISO RM-ODP (reference model of open distributed processing) viewpoint model.

We expect that the system described by this document will be capable of integrating clinical observations over an arbitrarily large region at low cost. Our intent is to initiate a group of open-source projects for the implementation of this system.

Background

Environment

This strategy is designed around current needs in the Canadian healthcare system. Canada has a publicly funded healthcare system. Essential medical services are paid for by various government bodies, primarily the provincial governments. The following institutions are involved in health care delivery (approximate figures):

- 25 000 family practices
- 20 000 clinics
- 1000 small and community hospitals
- 75 teaching hospitals
- X long term care centres
- X homecare agencies
- X private pharmacies

- Supporting services including privately owned labs, x-ray facilities

The vast majority of these institutions maintain some paper medical records, but the use of electronic records is growing. Many hospitals, nursing homes, homecare agencies, and pharmacies manage some types of clinical data electronically. In general, clinics and family practices make heavier use of paper.

Widely varying degrees of autonomy concerning information systems exist in different provinces. For example, some provinces mandate specific hospital information system products, while in others, products are chosen by individual hospitals, or individual hospital departments. Data standards (e.g. data model or ontology standards) are not widely used, with the following exceptions: ICD morbidity codes (http://www.who.int/whosis/icd10/) are widely used for reimbursement and retrospective statistical analysis, and HL7 v2 messaging (www.hl7.org) is commonly used within hospitals.

One Patient, One Record

A patient may receive medical care from many different institutions. For example, a single patient may have a series of family doctors as he or she moves from time to time. The same patient may be admitted to several hospitals over the years, for treatment of different problems. The same patient may use various walk-in clinics while away from home, and visit various pharmacies. The patient may receive homecare services, and later in life may spend some time in a nursing home. It is not unusual for a single Canadian resident to receive healthcare services at a dozen or more locations in the long term. Furthermore, visits to various providers may be interspersed in the short term, in the management of a single problem.

At each of these sites, the patient's medical records at other sites may have important consequences for subsequent diagnosis and treatment. For example, after a hospital stay, the patient's family physician needs to understand the test results, procedures, and medications given. Later, the same patient may receive emergency medical attention at a different hospital. The staff at this hospital should be familiar with key facts about the patient's medical history, including observations made at the previous hospital, and those made by the family physician.

Many healthcare institutions have electronic medical record systems, which allow anyone treating a patient within the institution to access the relevant records. In contrast, a paper chart can only be in one place at a time, and must be requested and retrieved manually, often with considerable delay.

In the age of the Web, we are used to accessing electronic information instantly from around the world. Those outside the medical industry may wonder why a clinician at one hospital remains unable even to access medical records at another hospital in the same city. Some of the reasons are as follows:

- The clinician may not know where else the patient is being treated, or has been treated
- The clinician will not know the patient's unique record ID at other sites, and searches based on name or other non-unique information would be time consuming and dangerously ambiguous
- Systems at other sites can not distinguish the clinician from unauthorized parties
- The clinician's application almost certainly uses different data models and terminology than systems at other sites
- The clinician's application cannot connect to the other systems, as there is no standard addressing scheme or communication protocol
- Most of the systems involved have not been secured for safe communication over the Internet

Most western countries face the same challenges. It is generally accepted that a new type of information system, often called an electronic health record (EHR), is required to meet these challenges. While there are many definitions of the term "EHR", the emerging Canadian interpretation includes the following features:

- It provides a complete longitudinal health record for an individual
- It is accessible to the patient, and to clinicians authorized by the patient
- It includes data from many sources
- The data it contains are entered or verified by health care professionals

There is also widespread agreement that an EHR should:

- Assist directly in clinical decision making, by providing data for automated decision support tools
- Provide context-sensitive links to knowledge bases for both clinicians and patients

The need for a national EHR, which provides a single comprehensive medical record for each resident, is well recognized in Canada. This need has recently been emphasized in a report commissioned by our federal government on the future of Canadian health care (Romanow, 2002).

A New Approach

This document presents a new EHR design. The fundamental concept is to link clinical systems using shared protocols and services, rather than to copy data into a large central database. In a similar fashion to the HTTP protocol (on which the Web is based), these protocols will support:

- Retrieval of content from an arbitrarily large number of source systems
- Creation of hyperlinks between parts of a patient's records in various systems

- Secure transport
- Caching
- Firewall tunnelling

To meet healthcare-specific needs, standardized services will support the following functions:

- Identifying the systems that store information about a given patient
- Unambiguous identification of a patient's records in each source system
- Collaborative authoring of problem lists, with links to supporting documentation
- Patient-driven authorization
- Referrals

The solution is based on the following technologies:

- The JXTA transport and discovery protocols (www.jxta.org)
- The HL7 Version 3 Reference Information Model and messaging standard (www.hl7.org)

Chapters 2-5 will present the processes, architecture, and services, based on these technologies, that compose this "virtual" EHR. The specifications will be neutral to language, platform, and vendor, and no restrictions will be placed on their distribution. This will allow the resulting system to expand naturally with the efforts of various interested parties.

Chapter 6 will discuss a software implementation of the required components. Competing implementations that conform to the same specifications are of course welcomed. However, as is recognized by the W3C and IETF, it is best to accompany software specifications with a working implementation. The implementation process may expose design flaws. A freely distributable implementation also facilitates adoption. It is therefore our intent to produce reference implementations of the required components using freely distributable open-source software.

An important part of the software implementation is a family of adapters that can be used to integrate the wide variety of existing clinical information systems, in such a way that the existing systems need not be modified. The adapter software will have the effect of making these systems support the standard protocols, to the extent that they are able without modification. Over time, source systems may be upgraded (e.g. to use standard terminology), which will allow them to interact more smoothly with other systems. However, experience to date leads us to believe that useful information can be retrieved from almost any installed clinical system without modification.

Related Initiatives

A few specific EHR projects should be considered in conjunction with the current project. Some readers with an interest in the current project may find one of the others more suitable to their needs. From the authors' perspective, none appears to meet the needs that drive the current strategy. However, our work is intended to complement them in various ways.

GEHR (www.gehr.org), the "good electronic [formerly 'European'] health record", is an innovative standard for representing clinical data so that it will age gracefully over time, as medical knowledge and practice change. GEHR aims to define a standard data model that can be implemented by all medical record systems. Ultimately this approach will bring more benefits than the loose coupling of existing systems that we propose. However, since GEHR requires complete replacement of clinical systems, it is unlikely to become a widespread standard within the next decade. The current strategy is complementary, in that it can provide access to a complete (if somewhat less uniform) longitudinal record in the mean time. As GEHR systems are installed, the current strategy can be used to integrate them with non-GEHR systems.

CorbaMed is a standard for integrating clinical systems using CORBA (common object request broker architecture; a distributed object technology). It defines a standard set of services for patient searches and access to clinical observations. It is competitive to HL7 in some ways, although the emphasis is different.

While the CorbaMed standard does not itself address many of the EHR challenges described previously, a solution could be built using CorbaMed instead of HL7 and JXTA, as we have proposed. The OpenEMed project (www.openemed.org) has worked in this area. It is outside the current scope to discuss the technical preferences for HL7 v3 over CorbaMed, and XML over CORBA. This is a matter of disagreement among experts, but these seem to be preferences of the industry in general. In any case, the current strategy should serve to integrate CorbaMed-compliant systems with others.

SynEx (www.gesi.it/synex) and PICNIC (http://picnic.euspirit.org) are two promising European projects with many similarities to the current project. Both favour distributed services over a central database. These projects are also committed to releasing their software under open-source licenses, making the technology very easy to re-use.

During the development of the current strategy, SynEx and PICNIC had released little in terms of software and specifications. As more of this material emerges, we hope to re-use whatever modules meet our needs. Direct collaboration may also be possible in the future.

All of these projects have the goal of rich interaction between systems. The current project is distinct in that we seek to introduce relatively loose coupling where there is little or no coupling currently. We believe this to be the only feasible option for the immediate term, nevertheless with great potential benefits. It should also be noted that this approach could be implemented alone or in conjunction with other integration methods. For example, suppose that one province had a central EHR repository while an adjacent province did not. The federated services approach could then be used to link the two provinces together, integrating the central repository of the first province with hundreds of individual systems in the second. The current strategy is designed to maximize the breadth of coverage of an EHR.

Scope

Functional Scope

This strategy is intended to support sharing of observations and interpretations among users of different clinical information systems.

Primary entry or edit of clinical observations is not addressed. This is a distinct area of concern, and a great variety of systems already exists for these purposes (e.g. the major hospital information system products). The current strategy involves leaving these systems in place, and retrieving information from them through various means (e.g. query; event-based replication).

Order entry is also not addressed for similar reasons. There is often a need for a user of one system to place an order in another system. However: 1) this activity normally occurs within a single institution, so the interaction is limited to a very few systems – large scale and plastic topology (i.e. ongoing addition/removal/relocation of participating systems) are not important concerns, and 2) effective practices are already widespread (primarily the use of HL7 order entry messages).

This strategy does address referrals. Referrals from one clinician to another often span institutional borders. Most existing clinical systems, which are designed for use within a single organization, do not address this need. Furthermore, as will be discussed later: 1) the infrastructure necessary for secure observation sharing also creates a good foundation for referrals, and 2) the referral process can be used to streamline security administration (i.e. a patient may opt to allow clinicians to authorize each other to view his or her chart).

Informational Scope

The goal of this strategy is to deliver access to the following information:

1) Textual documents (e.g. specialists' reports; discharge summaries; progress notes)
2) Allergies (primarily food and drug allergies)
3) Discrete observations
4) Drug orders
5) Problem lists

Although data entry is avoided for the most part, the system will support direct entry of the following information:

1) Problems and highlights of a patient's medical history, with links to supporting data in the chart.
2) Allergy information

Key Concepts

The system will be described in detail in chapters 2-6. Many aspects of the design reflect routine software engineering decisions. A few key aspects will be emphasized here, because of their relative importance or novelty.

Query Services

It has often been suggested that clinical systems throughout a given region should translate information into a standard format and send a copy to a central repository. A user interface for this repository could be made widely available, providing a more complete record than that which a single institution could offer. The benefits of this approach are clear, but there are disadvantages as well, including the expense of duplicate storage, limited scalability, a single point of failure (unless there are redundant copies of the database), and a likely target for denial-of-service attack. It is therefore worthwhile to investigate other approaches.

One alternate approach is for clinical systems to support a query standard instead of a standard in which they publish data to a central system. In this model, existing clinical systems remain in place and operate as usual, storing observations made within a single institution or practice. The system need only manage pointers to source systems, instead of the clinical data itself (see appendix A for contrast with other methods).

The query services model has certain advantages the central repository model:

1) Scalability is virtually unlimited. A comprehensive central repository for the Province of Ontario (for example) would have to support hundreds of thousands of transactions per day, and billions of data items. In contrast, individual systems in a query services model would not need substantially higher capacity than they currently have, regardless of the size of the integrated region.
2) Central costs for hardware, database maintenance, and interface maintenance are eliminated.
3) Organizations can retain access control over their own data. For example in the query services model, a hospital could audit access to its own data and withhold access as appropriate. Each institution could manage control independently, without the need for a common infrastructure to support this.
4) There is no single point of failure, and no target for denial-of-service attack.
5) There is a lower risk of data theft. A central repository containing health data for an entire province or country would require a team of system administrators. The administrators would be placed in a position of extreme trust.
6) Ongoing data cleanup is simplified. For example if a hospital mapped its internal procedure codes to LOINC codes, the query services model would make this

enhancement available instantly, whereas a central repository model would require an additional process for the system administrators to map codes for existing data.

One objection to this approach concerns the complexity of performing updates on non-centralized data. However, it is important to note that updates to most medical record data need not be performed on the electronic health record directly. Clinicians can continue to use local software to update local databases – it is only necessary to combine the data for review purposes.

Another common objection concerns the performance of federated systems. However, it should be noted that queries to multiple systems run in parallel. In addition, as we will discuss later, caching can be used to provide better performance than that expected in the central database model.

It is important to note that the central repository model is more viable than query services in certain situations. For example, query services cannot be used to access data in intermittently connected systems. Query services may also be inappropriate with very small systems, because the reliability of these systems may vary unacceptably.

Perhaps the best approach is to combine query services with one or more central repositories. For example, within a single province, several regional repositories might be created to store data from smaller clinical systems, or systems that lack a persistent network connection. The query services approach could be used to link these repositories with larger systems (e.g. hospital information systems) to form a comprehensive record. Some jurisdictions have existing repositories for specific functional areas (e.g. in Ontario there are existing or planned databases for transplant, cancer, and lab test data). A query services model could be used to link these repositories with hospital systems and other niche repositories as required.

Patient-Centred Care Networks

We propose that data exchange should be organized and secured using a "Care Network" for each patient. The Care Network is a dynamic, virtual network of information systems that contain observations about that patient. When a new clinician becomes involved in the care of a patient, then (with the patient's consent) he or she will join the patient's Care Network. Subsequently, when information is needed from outside the clinician's home system, queries will be distributed to other systems in the Care Network (i.e. other systems that have collected data about this patient in the past).

The purpose of a Care Network is to provide the following functions:

1) Access to a single patient's medical records.
2) A secure communication channel for exchange of data about a certain patient.

3) A limited scope for the propagation of queries and notifications. For example, a query about a patient need not be sent to every source system in the region, just to the few systems that contain information about that patient.

A care network would ideally have the following people as users:

1) A single patient
2) Guardians or agents of the patient
3) All of the clinicians involved in treatment of the patient

In order to satisfy legal requirements in many jurisdictions, and out of respect for individual privacy, creation of a care network should be conditional on the patient's consent, and user membership in the care network should be under the discretion of the patient.

Communication Using Chart References

Clinicians should have the ability to draw their colleagues' attention to parts of the distributed chart. For example, when making a referral or documenting a decision, a family physician should be able to reference relevant documents or lab results.

This capacity can be provided using a textual expression for the location of specific data on the network. These would be used much as URLs are used to communicate the locations of specific pages on the Web.

Using a URL link is an alternative to copying information, and may also serve as an alternative to abstracting information in some cases. A key advantage of this method is that the patient and originating organization retain access control over the linked information, whereas it is problematic to retain access control over a copy.

In chapter 5, we will define a Chart Item URL syntax that can be used to retrieve documents, observations, or sets thereof. Like Web page URLs, these will be text-based, so that they can be written down, communicated through email, or included in documents. The text of Chart Item URLs will not contain clinical data or information that can be used to identify the patient outside the network (i.e. by unauthorized individuals).

As an example of the intended usage, consider the following fictitious letter to a family physician on discharge from hospital:

> Dear Doctor Jones,
>
> Your patient, Mr. John Smith, was admitted to Hospital X on Dec 5 2002 and discharged on Dec 13 2002.
>
> Most responsible diagnosis: Acute Pancreatitis

Other Diagnoses: vomiting

Investigations and Results:

1) CBC (… URL would appear here …)
2) Chest X-Ray: pleural effusions (… URL would appear here …)
3) Liver Function test: normal (… URL would appear here …)

Treatment Given:

[Etc…]

As another example, consider the following fictitious entry made by a homecare agency into an on-line nursing home admission form:

> Mrs. Smith is recovering from a broken hip. She was admitted to hospital for two weeks last April. The discharge summary is available here: (… URL would appear here …). The latest physiotherapy assessment is available here: (… URL would appear here …).

In the latter example particularly, the originator assumes nothing about the patient's consent to release information. The receiver of the note cannot access the reports without the patient's authorization.

Collaborative Problem Lists

Problem lists are a key component of a longitudinal record (Dick et al, 1997; Smith & Cousins, 2002).

Whereas other clinical data are naturally distributed, problem lists are naturally shared. Observations, consultations, and orders are discrete events that are associated with a single provider. Clinicians at other sites may need to review these events, but need never edit them.

In contrast, a single problem may be managed over a long period, by various providers. Furthermore, assessment of the problem may change over time. For these reasons, everyone caring for a patient should collaboratively maintain a common problem list. This implies that problem lists should be managed by the EHR rather than queried or copied from source systems.

Other motives for managing problem lists directly include the following:

- Most clinicians do not currently have access to on-line problem lists
- Many installed systems do not support problem lists

In chapter 4, we will present an API for accessing and updating problem lists. Only one network node will contain problem data for any given patient (although different nodes may exist for different groups of patients). The API will provide a long-term, reliable repository of problems. Hospital information systems may use this as a service to drive their own problem list modules. Care network members will be able to add, update, and deactivate problems, and add comments and chart references.

Problems will be strictly coded to support features such as the following:

- Care pathways
- Decision support rules
- Customization of UI views (e.g. for diabetes patients always show latest BP, HbA1c, etc.)
- Problem-specific links to knowledge bases and best practices

Patient Resolution at Consent Time

There has been much effort invested in the development of Master Patient Indices (MPIs), tools that map the different patient IDs appearing in various systems. An MPI relies on a matching algorithm, which (for example using statistically derived weighting factors) estimates the identity of a patient in multiple systems on the basis of name, sex, address, and other demographic information. This process is complicated by misspellings, non-uniqueness, and transience of some of the data. Human intervention is generally necessary in a certain percentage of cases, and mismatches are increasingly likely with larger regions.

In the context of an EHR, the patient must consent to participate, and is therefore present during the registration process. Uncertainty can be avoided by performing the matching process with the patient's help, i.e. by asking the patient to define the search region and verify possible matches. This can be performed in such a way as to prevent patients from falsifying their identities (see chapter 2).

Performance and Reliability through Caching

The EHR software should include replication and caching mechanisms, to improve performance and reduce the risk that data are unavailable during source system downtimes.

In chapter 5, we will present the design of a cache that is suitable for this purpose.

Ad-Hoc Query

The focus of this strategy is to provide access to well-defined sets of data through well-defined queries. Ad-hoc query is a peripheral concern. However, the addition of this feature may support other applications in areas such as retrospective analysis or disease surveillance. It may also extend the system's lifetime as new requirements emerge.

These needs can be accommodated in part by supporting ad-hoc query in the cache system (also the proxy system – see chapter 5). For analysis of a block of data, the data would first be queried from source systems *en masse* and stored in a cache. Ad-hoc queries would then be performed on the cached data. This would allow (for example) sorting a small group of patients according to some clinical criteria, or follow-up analysis on surveillance-related notifications.

Note that a similar mechanism can be used to perform an EHR extract for a certain patient, for example to store a snapshot of a patient's distributed record on CD.

Dissemination Strategy

The value of this strategy will increase with the number of systems that are able to participate, which in turn depends on the ease with which existing systems can be integrated. It is therefore important to make it as easy as possible for an installed clinical information system to share the data it manages. Several means of achieving this are discussed below, including the following:

- Minimizing disruption of existing systems
- Minimizing complexity of implementation
- Releasing software under open-source licenses

Minimal Disruption

The addition of query functionality must not impede the normal operation of source systems. The primary concern in this regard is increased processing load arising from queries. However, since patients tend to be treated in short flurries, caching can be used to reduce query load dramatically. Furthermore, a proxy (chapter 5) that is populated using batched data or HL7 messages would eliminate direct queries.

Varied and flexible technical options for integrating a system with the network will make integration more cost-effective. As described in chapter 6, a variety of adapters will be created, encompassing several integration methods. A library of adapters for specific products will also be created. Since some products are very widely installed, a handful of product-specific adapters will simplify matters for hundreds of sites.

While the proxy and adapter methods will limit disruption to some extent, the system software must also be easy to deploy and administrate, to minimize financial and human resource barriers to implementation. This requirement can only be addressed by well-documented and administrator-friendly software implementations. An open specification allows competition among vendors, which may drive investment in this direction.

Minimal Complexity

In an environment with wide practice variations, a large base of legacy systems, and slow uptake of standards, the creation of a regional EHR that faithfully models the complexity and variety of healthcare does not appear to be a realistic short-term goal. Instead, this strategy aims to deliver limited functionality quickly, in a way that can improve over time.

As the next generation of systems emerges, the connectivity and security foundations provided here will also support more advanced data model and terminology standards. However, until this sophistication is widespread among source systems, the growth of this network will be made possible by minimizing the requirements for participation. Queries should assume as little as possible about the data available from the underlying source system.

Query definitions (chapter 4) will include standard terminology for use when available, and provide an elegant default for other cases. For example, some systems support LOINC procedure codes. The services will allow query by LOINC code (which would allow multiplexing of similar information form multiple systems). However, it will also be possible to retrieve a native code list from a certain system, and to then query that system using its native codes.

Open Source

Reference implementations for the project will be implemented in an open-source manner, i.e. the source code comprising the implementation will be distributed on the Web, under a license (www.opensource.org/licenses/index.php) that allows other parties to modify it and redistribute modified versions.

This method is thought to hasten evolution, dissemination, and quality control of software at the expense of most routes of commercialization. This is appropriate since the project is being initiated under the publicly funded Canadian healthcare system. It is hoped that the project will generate international interest. Some open-source projects (e.g. Linux) enjoy substantial development contributions from an international user community, although this is admittedly rare. In any case, the elimination of licensing costs, and proof against lock-in to a proprietary system, are expected to improve the growth of the network.

Since the specifications will be open, commercial vendors may also implement and sell components of the system. Furthermore, the open-source licenses will allow resale of open-source software. This will allow companies to sell value-added versions (e.g. adding proprietary improvements or support services).

Development Strategy

Large software projects usually fail, or go overtime and over-budget (Brooks, 1995). Taken as a whole, any EHR system is complex and novel enough to involve substantial implementation risk.

For this reason, this strategy will serve only as a high-level framework for several low-risk, high-impact projects. Each project will fulfill a specific need in the larger strategy. However, each project will also be designed to accommodate major changes in the strategy, as well as re-use in other EHR projects. As each project is implemented, lessons learned will be used to refine or amend the strategy as appropriate.

This development plan allows for several long-term possibilities, including the following:

- The strategy is carried out largely as-is, as a series of small projects
- The strategy is radically changed, but individual projects remain useful in the new context
- After the first set of components is developed, the strategy is abandoned in favour of using these components to enhance another EHR system (e.g. GEHR)

The proposed development activity contributes to the final EHR in each case. Priority projects are described in the following sections.

HAPI Routing and Conformance

HAPI (http://hl7api.sourceforge.net) is an HL7 v2 framework, and is the first spin-off project arising from this strategy. It was first released under an open-source license in November 2001. It currently performs the following functions:

- Message parsing and encoding (with both XML and the traditional ER7 form)
- Enforcement of the HL7 standards for data-type content and message structure
- Message transport and acknowledgement

In addition to serving as the HL7 v2 engine for our EHR prototyping efforts to date, HAPI has been downloaded by hundreds of developers worldwide, and used in a variety of production projects.

HAPI will now be extended in the following ways:

- Rule-based message validation (including conformance profiles)
- Message routing and translation
- Secure Internet transport

With these additions, HAPI will be useable as a free interface engine. Furthermore, it will be capable of sending encrypted messages over the Internet, and validating messages according to regionally standard conformance profiles.

These features are likely to be important with any EHR topology. For example, in the central database model, HAPI could serve to extract data from a source system, translate it into a standard form (e.g. as defined by a provincial conformance profile), and send it (encrypted) to a provincial repository.

Semantic Cache

Any EHR system involves retrieval of clinical data over a wide area network. In both centralized and decentralized models, there are risks of long network delays, high query load, and network or database downtime. The semantic cache addresses these problems in either topology.

Generic Application Suite

The following services will be implemented, with corresponding update and review applications:

- Audit trail
- Consent and registration
- Problem list

These applications are key components of any EHR. They will be created according to the requirements identified in this document, as well as requirements that can be foreseen for other EHR models. For example, the following should be supported:

- Use of standalone database or integration into larger (e.g. centralized) database
- Update via API, HL7 v2 and v3, and user interface

Query Service Adapters

A family of adapters will be created for the query services that are described in chapter 4. These adapters will have limited utility if the basic strategy changed to a centralized, push-based replication (e.g. Calgary Health Region), or bulk transfer model (e.g. GEHR). However, the query services approach appears to be highly viable, with distinct advantages over other models, and it has not been sufficiently explored. Implementation of the query services adapters against selected source systems will provide data for an empirical analysis of the strengths and weaknesses of this approach.

Summary

We have outlined the main features of a new EHR strategy based on distributed services. This strategy is intended to provide an integrated view of medical record data over an arbitrarily large region, securely, with good performance and low operating cost.

The remaining chapters present technical details of this strategy, from the various viewpoints defined in the ISO Open Distributed Processing Reference Model.

Chapter Two: Enterprise Viewpoint

Users

We assume several broad categories of users, which are introduced in the following sections.

Patients

Patients will be direct users of the system. They will be able to review their own records, choose which parts of their records to expose to clinicians, authorize specific clinicians and categories of clinicians to view their records, and review an audit trail identifying who has accessed their records.

A patient's records will only be accessible on the network if the patient consents to this. Patients will be registered by agents of trusted healthcare institutions (e.g. hospitals; ministry of health; homecare agencies).

Clinicians

Clinicians will have access to the records of those patients who authorize them. They will be able to annotate the records of these patients and perform on-line referrals for them. They will also be registered by agents of trusted healthcare institutions (including medical associations).

Registrars

Patients and clinicians must be registered on the network. The registration process must ensure the correct identity of clinicians and patients, and the correct credentials and appointments of clinicians.

This will be achieved through a chain of trust, in which publicly accountable, high-profile institutions verify and assert this information. For example, if the Canadian Medical Association (CMA) registers a certain doctor, patients will then have the option of trusting the CMA's assertions about that doctor.

To put this model into practice, it is necessary for institutions to appoint registrars (accountable to the institution) who will verify user's credentials, face-to-face, during the registration process. For example, a hospital's registration clerk might also act as a patient registrar on the network. In some cases, registration may be integrated into existing procedures. For example, a homecare case manager often enters into detailed discussions with clients about information-related consent. The case manager might act as a registrar as part of this process.

Software Components

The EHR system is composed of various software components (figure 2.1) that perform different tasks. These components may be installed on the same network node, or distributed across many network nodes.

There are several “application” components. These are responsible for user interaction, e.g. displaying information to users and allowing them to navigate through it. Application components include chart review applications (different versions may be used by clinicians and patients), registration applications, and an authorization application that allows patients to control access to their records. The latter is likely to appear as part of the patient’s chart review application, although it is logically distinct.

Applications retrieve information from (and occasionally update) various information services. The information services consult an authorization service to check whether the user is allowed to access the requested information. Most information is retrieved through a “query service” (e.g. document service, medication service, etc.), which extracts information from an underlying source system through various system-specific means. The search service is used during the consent and registration process, to collect a patient’s identities in all source systems (see “Patient Registration” later in this chapter).

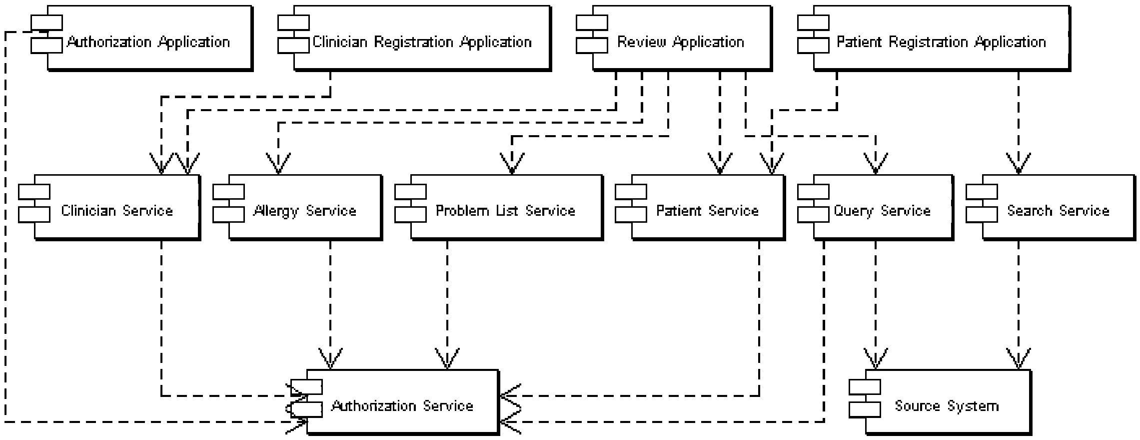

Figure 2.1: Main software components.

Security Considerations

Patient-Driven Authorization

The patient will determine membership in the care network (directly or indirectly) through the following means:

1) On an individual basis. Patients may add specific clinicians to their care networks.
2) Professional referral. The patient may opt to allow members to invite other specific clinicians to join.
3) Role-based access. The patient may opt to allow various groups of clinicians to join his or her care network, using role-based rules. For example in a hospital environment, a certain patient may opt to allow her attending physician (or all the nurses on her unit, etc.), to join the care network. In this model, the patient trusts the health care institution to correctly identify the roles of each of its clinicians, and their relationship to the patient.
4) Emergency access. A patient may opt to allow any clinician (or clinicians in certain roles) to access his or her records on an emergency basis, and to be notified of such accesses.

People, Roles, and Role Relationships

Security administration can be simplified by assigning roles to clinicians. For example, granting access to anyone in a certain role (e.g. physician) frees the patient from having to individually grant access to each person acting in that role.

It will also be useful to base access control on specific role relationships between clinician and patient. For example:

- Patient's attending physician
- Patient's family physician
- Nurse on patient's hospital unit
- Nurse in hospital where patient is admitted

Granularity

Further analysis is needed to determine the granularity of access control. Some possibilities include:

- All-or-nothing access
- By encounter
- By problem
- By observation
- By date

Fine granularity is advantageous in that it maximizes respect for privacy and access to data that patients are willing to share.

Fine granularity is disadvantageous in terms of increased complexity of administration (the burden being on the patient). Note however that access control can be performed by adapter software, so this decision does not impact source system requirements.

Although much has been written about the granularity of access control in an EHR, most discussion on the subject has a philosophical or legal underpinning. This would be complemented by empirical study. It would be helpful to analyse a random population sample to determine the amount and distribution of sensitive information. The following questions could be answered in this way:

1) How many people consider parts of their medical records to be private (i.e. they would be reluctant to share them with clinicians involved in their care for unrelated problems)?
2) How are private data distributed over institutions and time?
3) To what extent are these data intermingled with non-private data?

Defensive Authorization

Owners of source systems are trustees of the patient data therein, and may wish to perform their own authorization checks independently of the network. Therefore the identity and credentials of the querying user will be made available to source systems along with other query parameters.

Authentication and Certificate Authorities

The identity of every patient, clinician, and system on the network must be authenticated with every action.

An appropriate way of doing this is to issue a digital certificate to each user and source system. As an exception to the general non-centralized approach, it would be expedient to create a single trusted certificate authority (CA) per region (e.g. province, country). The software distributed within this region would be configured to trust certificates signed by this well-known authority.

Well-known, publicly accountable institutions (e.g. hospitals) could act as certificate authorities under the central CA. A hospital could then verify the identity (in person) of its own staff and patients and issue certificates to them under the authority of the central CA.

Some regions have existing processes for authenticating clinicians. For example, in Ontario, Smart Systems for Health has a program for assigning digital certificates to physicians. Any existing process in which digital certificates verify identity, credentials, and affiliations can be used.

Registration Processes

Patient Registration

The patient registration process is composed of the following steps:

- Identification
- Consent
- ID resolution
- Creation of record

These steps are discussed below.

Patient Identification

Patients should be identified in person using multiple identification documents. At least one of these should be appropriate for identifying the person as a unique patient in a large region. For example, in the Province of Ontario, Canada, the provincial Health Card would be the appropriate ID for most residents. In order to avoid errors associated with demographic matching, it would be reasonable for a region to insist that patients present such identification in order to register with the network.

Consent and Non-Repudiation

Following identification, the registrar must explain to the patient the purpose of the system, how it works, and what it means to agree to participate. Registration application software should print standard information sheets.

The registrar will then obtain the patient's informed consent to participate in the EHR, and the system will record this fact, including the date and the identity of the registrar.

A patient may consent the formation of a care network on his or her behalf, and subsequently deny having done so. While the care network should be disbanded at this time, the EHR should also maintain proof of the patient's original consent in order to prevent or resolve disputes.

The network should support one or more means of non-repudiation, recognizing that requirements may vary in different regions. Some possibilities include the following:

1) The patient signs for consent using an electronic tablet, and the network retains an image of the signature.
2) The patient signs a paper consent form. If necessary, the form is stored at the institution where the consent is obtained. Alternatively, a printed/bar-coded form could be used, and faxed to an independent service that retains the images.
3) The patient reads a sentence indicating consent into a microphone, and the voice is stored. This method might be particularly effective in preventing disputes by playing the voice back to the patient.

While these measures are important, the value of a signed consent form should not be overestimated. *Informed* consent is needed, and it is recognized by the courts that people sign things that they do not understand. In some jurisdictions, a registrar's witness as to the patient's understanding may have more legal value than a signature. Further research is required, to explore ways of recording indications of what the patient understands (e.g. recording a checklist of questions asked of the patient, or a corresponding voice interview).

ID Resolution

Patients are likely to be the most accurate in distinguishing their own records at various institutions from the records of other people with similar demographic information.

When a new patient is registered with the network, the patient may already have medical records at a number of institutions that should be linked. A list of potentially matching records can be obtained by sending a "find candidates" query to various systems, to search for matches on ID or demographic data items. Possible query scopes include:

1) Institutions that the patient remembers having visited
2) Municipalities in which the patient remembers having received medical care
3) The entire region
4) The entire country

Broader queries require more search parameters, to avoid large result sets.

Remote systems will return basic demographic data about the patient (e.g. name, date of birth, sex, and telephone number) and possibly information about the visit (e.g. date of visit).

The records that actually correspond to the person in question will then be determined by interview. The registrar will be prompted to ask specific questions in a way that prevents the patient from being misleading or mistaken (e.g. Were you ever admitted to River Hospital? What year? What was your phone number at the time? Have you seen a Dr. Jones on 5th street?). This process will result in a list of information systems that contain information about the patient, and the patient's ID in each of these systems. Note that primary identification is performed using photo ID; the above questions just serve to confirm records with matching demographics. Note also that this process is only performed once in a patient's life, and need not be performed by a clinician.

When a new institution joins the network, some of its patients may already be registered, so the registration process cannot be used to resolve IDs of these patients. In this case, the system will search for matches among all registered patients. In case of a possible match, the resolution process would take place the next time someone logs on to the care network. This may be months or years later, but the information will not be required before then, and this procedure allows manual verification by an interested party. A question-and-answer process similar to that described above will be automated for on-line access.

As a result of the above processes, a patient's care network will store the primary key of the patient's record in each source system, preventing ambiguity about the patient's identity when clinical queries are performed.

Creation of Record

At the end of the registration, the following information will be stored:

- A unique ID for the patient
- Basic demographic information (e.g. sex, DOB, address)
- A primary health identifier (e.g. health card number)
- Patient ID in any systems that are aware of the patient
- Non-repudiation data (e.g. voice print)

This information will be stored in an EHR node that implements the "patient service" (see chapter 4). The registration application acts as a client of a patient service node. The distribution of patient services is a deployment decision. There may be one patient service per region, one per hospital, etc., but each registration application will interact with only one of these.

Clinician Registration

The clinician registration process will be loosely standardized. The only requirements is that a registration application must issue a digital certificate to the clinician, and collect the following information and make it accessible through a clinician service:

- Name
- Professional qualifications
- Institutional affiliations
- Contact information

The trust model will be similar to that for patients. A small number of certificate authorities (ideally one per region) will correspond to well-known and trusted institutions, such as government agencies. This CA may issue certificates to other well-known institutions (e.g. medical associations, hospitals), which will be trusted to register clinicians and issue certificates to them.

A given region may already have a clinician registry that meets these requirements, in which case the creation of a new registry is not necessary.

Referrals

There are several reasons for including referral support:

- Referrals span institutional and sometimes regional borders
- From the patients' perspective, the referral process can simplify authorization, because a trusted clinician can be authorized to invite colleagues into a care network
- The ability to find a clinical entity (e.g. by name or location) is already necessary to support entity-based access control

At a high level, the referral procedure is as follows:

1. The referrer finds an appropriate target (e.g. by name, ID, specialty, or location)
2. If the referrer is so authorized, the system invites the target entity to join the patient's care group
3. The referrer initiates the referral using contact information for the target (e.g. a phone number, email address, or on-line referral form)

Email-based or Web-based referrals may include the EHR ID of the patient, or links to chart items. The target entity (having been invited to join the patient's care network) may subsequently log on to the EHR to view the patient's records.

Value-added features (e.g. maintaining a list of common targets) are deferred to the referral application, as they are independent of the basic referral protocol.

Chapter Three: Informational Viewpoint

The information model is based on the following premises:

- Due to the quality and extent of healthcare information standards, divergence from or extension of existing standards is likely to be counterproductive.
- Because of disparities between legacy systems, it is not possible for all source systems to adhere strictly to a single detailed information model.

The information model for this network will therefore consist mainly of a subset of the HL7 v3 reference information model (RIM) and vocabulary (www.hl7.org).

In many use cases, partial functionality can be attained without adherence to standard terminology. In these cases, adherence will be optional.

HL7 RIM

Service APIs, internal messaging, and repositories will map to the HL7 RIM. The following RIM classes will be used:

- Entity
- Organization
- Material
- Place
- Person
- Role
- Employee
- CertifiedEntity
- Patient
- RelationshipLink
- Participation
- Act
- ActRelationship
- Observation
- SubstanceAdministration

A D-MIM (domain message information model; a domain-specific RIM subset) that defines the common model for all of the required messages will follow in a separate document.

Vocabulary

Critical code sets include the following:

- Clinical role (e.g. family physician)
- Clinical role relationship (e.g. admitting physician)
- Location (hierarchical)
- Organization

- System
- Document category

Several other code sets, where implemented by source systems, will improve functionality. These include the following:

- Procedure
- Specialty
- Drug product
- Drug category
- Drug administration route
- Drug administration schedule

Location Codes

Certain functions (e.g. patient ID resolution and referral) are facilitated by the ability to search for people or services by location.

To support this, a set of location codes will be created at the time of installation in a given region. Specific codes are a deployment decision, but it is recommended that a hierarchical system of locations be used, with enough granularity to support scenarios such as the following:

1) At registration time, a patient cannot remember exactly what hospital she visited on vacation last year but knows the name of the town.
2) A patient has to be referred to homecare in a distant region. There is no agency in the patient's hometown, so the township or county must be searched.

Chapter Four: Computational Viewpoint

Data will be queried and updated through a set of core services, which fall into the following categories:

- Clinical query services
- Patient services
- Clinician services
- Vocabulary services

These will generally run as separate processes and separate nodes on the peer-to-peer network. Clinical Query services will present data that are collected and stored in operational source systems, while the other services will use their own persistence mechanisms.

These services will be realized (equivalently) in both 1) Java and 2) HL7 v3 (www.hl7.org) over JXTA (www.jxta.org).

The JXTA services will use HL7 version 3 query and response messages. These services will allow platform and language independent operation over the Internet.

The Java APIs will provide a convenient foundation for building user interfaces or additional services (e.g. decision support). APIs in other languages will follow as needed.

Clinical Query Services

The EHR network will include the following clinical services:

1) Document Service (for query of textual reports in various formats)
2) Problem Service (query and update of problem lists)
3) Allergy Service (for coded and free-text allergy data)
4) Observation Service (for discrete observations such as lab results)
5) Medication Service (for medication orders)

Document Service

The document service will provide read-only access to textual documents (e.g. discharge summaries, pathology reports).

The API will provide two functions: retrieval of a list of document references by category and date range, and retrieval of an actual document by reference (see figure 4.1).

Documents will normally be presented in the form in which they are stored in the source system. Ideally, source systems will provide documents in one of the following widely used, platform-independent formats:

1. CDA (HL7 Clinical Document Architecture)
2. HTML
3. JPEG (for scanned documents)
4. Plain Text
5. Adobe PDF
6. Microsoft RTF

In the case of CDA, review applications should perform an XSL transform to convert documents to a common viewable format (e.g. in HTML). RTF documents may be converted to plain text.

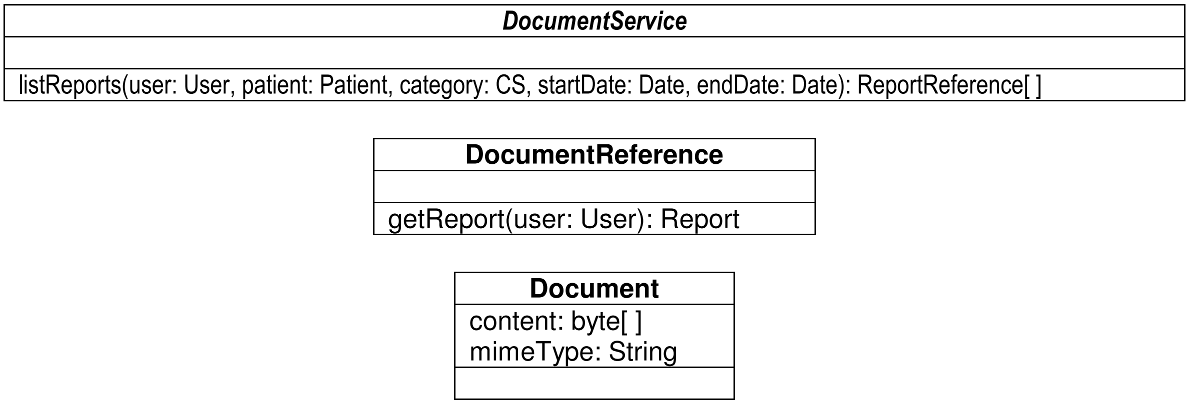

Figure 4.1: Core classes of the document service API. Patient and CS are taken from the HL7 v3 RIM. DocumentReference is a subclass of the RIM Act.

Problem List Service

The problem list is a single, coded, longitudinal record of a patient's medical problems. It is collaboratively maintained by members of the care network.

Parts of the chart may be linked to specific problems using Chart Item URLs.

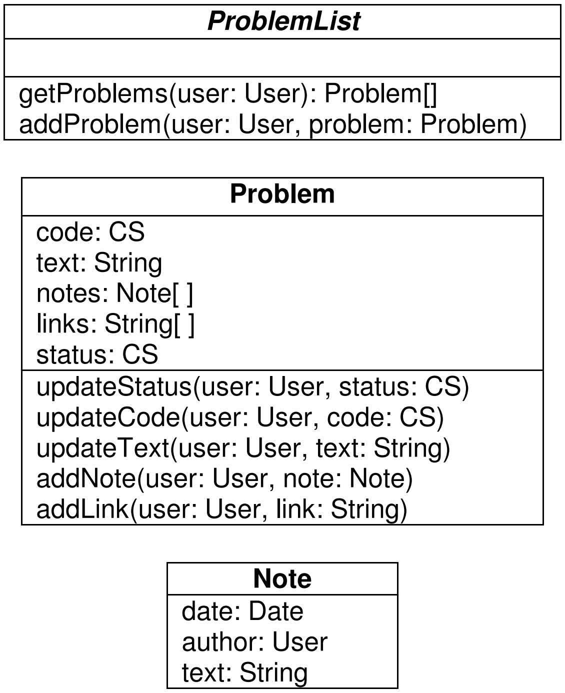

Figure 4.2: Core classes of the problem list API.

Allergy Service

The allergy service allows query and update of food, drug, and environmental allergies, etc. There is a preference for coded data. However, recognizing that coded data are unavailable from many source systems, and that free text is better than a lack of information, codes are not required.

Allergies can be presented in two ways:

1) As a list of observations, each of which is recorded by one clinician
2) As a list of problems, any of which may be noted by multiple clinicians

The former is more suited to a distributed environment and does not require clinicians to do any additional work (assuming allergies have been entered into some source system). The latter is more useful for review and decision support.

In order to provide both benefits, the allergy services will be structured with distributed observations and (optionally) a single coded problem list per patient that references these observations.

The single service (per patient) that manages the problem-oriented list will also manage an observation list, so that users may record allergy observations whether or not their local systems support this.

Applications that require a complete list of allergies should combine the problem list with any observations that are not linked to a problem entry. If all observations are linked, a non-redundant, coded list will be available.

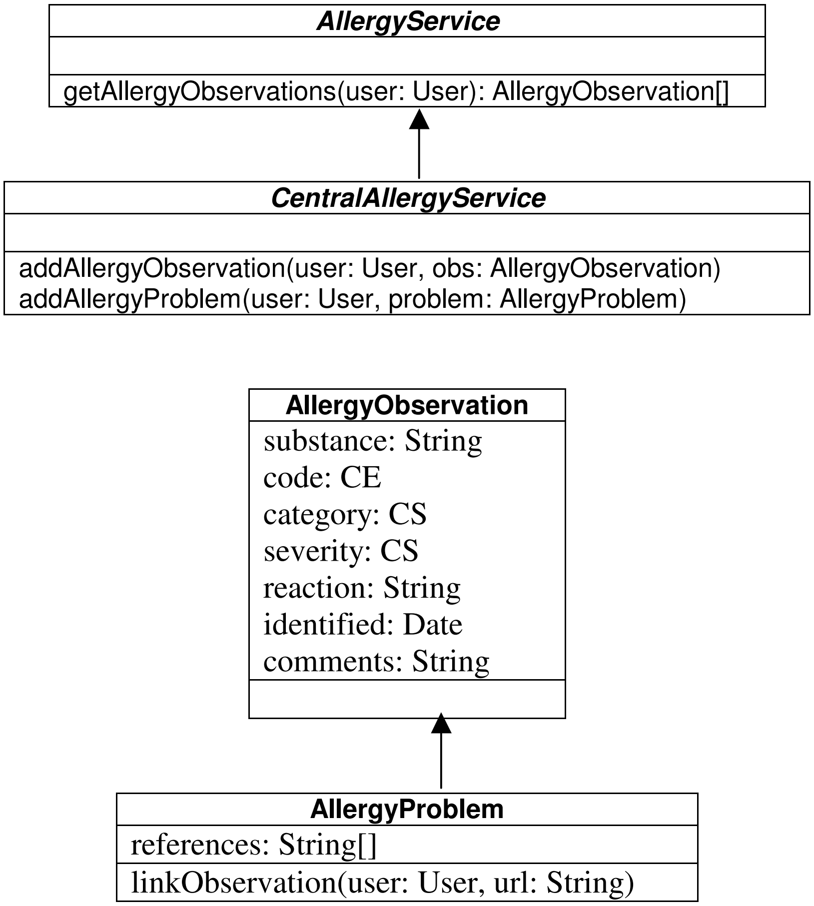

Figure 4.3: Core classes of the allergy service API. CS and CE are taken from the HL7 v3 RIM.

Observation Service

The observation service will provide read-only access to discrete clinical observations (e.g. lab results).

It is preferred that source systems provide standard procedure codes. Provision of these codes will facilitate the following:

- Queries for results of one procedure type (e.g. trending a lab result) across multiple sites
- Use of data by clinical decision support systems

However, it is recognized that many of the systems currently in production to not support these codes. Where standard codes are unavailable, clients will first query a particular source system for a list of its native codes. The native codes will then be used in subsequent transactions (e.g. they could be incorporated into a user interface so that the user could select procedures from this list). This will provide, at a minimum, access to human-interpretable observations from one system at a time.

ObservationService
listObservations(patient: Patient, type: CE, startDate: Date, endDate: Date): Act[] getBatteryCodes(): CE[] getDiscreteObservationCodes(): CE[]

Figure 4.4: Observation service API. The Patient, CE, and Act classes are taken from the HL7 v3 RIM.

Medication Service

The medication service will provide read-only access to medication orders. Medication administration records are not included because of dramatically greater data volume and lesser value in a longitudinal record.

As with the observation service, the minimum requirement on source systems will be to provide un-coded data, but coded data are strongly preferred.

This API will provide a single query method, with which medication orders are queried by date range (figure 4.5).

MedicationService
listMedicationOrders(patient: Patient, startDate: Date, endDate: Date): MedicationOrder

MedicationOrder
name: String code: CE form: CE

effectiveTime: GTS route: CE dose: PQ doseAdminPeriod: PQ maxDose: RTO instructions: String

Figure 4.5: Medication service API. The MedicationService class represents a single source of medication order information (e.g. a pharmacy information system). The MedicationService is used to provide a list of medication orders.

Clinician Service

The clinician service provides the following functions:

- Search by name, location and/or institution, and qualifications
- Query for contact information
- Query for relationship with specific patients (optional)
- Update patient relationships

The maintenance of patient relationship information is an optional feature that will simplify authorization where available. For example, a certain patient may specify that any nurse working on a hospital unit in which he is staying be allowed to view his chart. These role relationship data will typically be updated through HL7 v2 messaging.

ClinicianService
find(givenName: String, familyName: String, region: SC, facility: SC, qualification: CS): Clinician[]

Clinician
getEmailAddress(): String getRelationshipToPatient(user: User, patient: Patient): CS[] updateRelationshipToPatient(user: User, patient: Patient, relationship: CS)

Figure 4.6: Clinician Service API. Patient, SC, and CS are taken from the HL7 RIM.

Patient Services

Patient Service

The patient search service will typically be implemented in conjunction with one or more registration applications. A registration application will populate a patient data store, which will act as the data source for a patient search service. Note that a given region may have one or many patient services, each with a corresponding data store (e.g. one per province, one in every hospital). This is a deployment decision. Each registration application interacts with a single patient service, but it does not matter which one.

A patient search service will provide the following functions:

- Search for patient using demographics and identifiers
- Return demographic and care network information by UID
- Possible match resolution

The latter function requires some explanation. When an existing system joins the EHR, some of its patients may already have care networks (see chapter 2). The system will search for each patient (using whatever identifying information it has), and will notify each care network that might represent a matching patient. Later, members of the care network can confirm whether the match is accurate. If so, the care network will store the new ID and the new system will become a care network member.

PatientService
findCandidates(user: User, patient: Patient) : Patient[] getPatientDetail(uid: String): Patient registerPossibleMatch(known: Patient, new: Patient) getPossibleMatches(user: User, known: Patient) : Patient[] confirmPossibleMatch(user: User, known: Patient, new: Patient, match: boolean)

Authorization Service

Users will perform queries and updates within the context of a patient’s care network. Only authorized users will be allowed to join the care network, therefore only trusted clinicians will be able to view any part of a patient’s records.

Patients may also mask any parts of their records that they do not wish even trusted clinicians to view.

It should be noted that further analysis is required to determine the following:

1) Appropriate levels of granularity for masking data
2) Whether to allow masking of different parts of the chart from different users (this would increase complexity dramatically)

This is discussed in more detail in chapter 2. In the meantime, we will make the following assumptions:

1) Data may be masked by institution, date range, or specific data item.
2) Some facility for masking different items from different users is needed, because masked items should not be hidden from the patient. However, no further distinction between users will be made until the necessity is demonstrated.

Systems within masked institutions will still join the care network, so that data therein can be viewed by the patient.

There are several different ways of integrating query services with access control services. These include the following:

- The query service asks the authorization service for permission to reply to each query
- The query service sends query results to the authorization service, which filters them and returns the viewable results
- All queries are made through the authorization service, which retrieves information from the query service and filters it as needed
- The query service sends the query definition to the authorization service, which responds with a list of exclusions relevant to that query
- The query service retrieves a complete list of exclusions from the authorization service from time to time, and uses it to filter results locally

We will use the latter model. This will allow partial results to be viewed when a query spans masked data. It is also expected to provide the best performance, as authorization data can be cached by the query service for use in multiple queries.

This decision introduces a slight risk, in that when a patient changes a mask, the change might not be reflected in queries for some time afterwards (e.g. an hour). However, if a clear time limit is defined and understood by the patient this is not likely to be perceived as a breach of privacy. Furthermore, in most cases it is unlikely that a patient would update his or her masks while receiving medical care. The obvious exception is that a patient may wish to immediately grant access to a sensitive item on the request of a clinician, but in this case, the patient could show the item to the clinician directly.

AuthorizationService
getMask(user: User): Any[]

Terminology Service

In contrast with the other services, which are distributed, the terminology service will normally be centralized. Each node in the network will be configured with the network addresses of a small number of identical and redundant instances. The purpose of this service is to provide an authoritative coded vocabulary for an entire region.

The terminology service is not global, in that individual regions may elect to use different sets of terms. For example, an independent network could be set up within a hospital, with a service that uses hospital-specific codes. HL7 v3 vocabulary will be provided with the default installation.

Applications through which data entry is performed should obtain valid code lists from this source. For example, an application that registers a clinician with the EHR should obtain its list of valid role codes from the regional terminology service.

Some codes may be used for informational reference. For example, the administrators of a pharmacy information system may wish to consult the terminology server to obtain regional codes for administration route. These codes would not be used for data entry or as search criteria, so they would not be embedded within a user interface. However, a textual reference might be used to map codes within a pharmacy system to the regional standard.

TerminologyService
getVocabularyDomains(): CS[] getEntries(domain: CS): CE[]

Chapter Five: Engineering Viewpoint

This chapter describes the ways in which the various system components can be connected.

At a high level, the EHR consists of two kinds of executable components: applications and services. Users interact with applications, which in turn use services to retrieve or update data. Services may also use other services. Various independent parties will deploy instances of these components.

For an application to use a service, it must conform to the necessary message model, supply the necessary security credentials, and know the network address of the service.

Message Model

HL7 v3 will be used for internal messaging. Since the service APIs are mapped to the HL7 RIM, mapping queries and responses to v3 messages is expected to be straightforward. Specifications for the required messages will follow in a separate document.

Connectivity

Both client-server and peer-to-peer connectivity models will be supported.

The peer-to-peer model will allow the natural growth of an arbitrarily large network, without manual connectivity administration. In this model, services advertise their addresses, and applications discover them automatically as needed.

The client-server model is more appropriate for smaller-scale implementations (e.g. to integrate applications within a single hospital or a small town). In the client-server model, application will access services at pre-configured locations.

While an EHR will include many instances of most applications and services, a vocabulary service should be relatively more centralized in order to provide a single dictionary of codes. For this reason, the vocabulary service might be deployed at a known address even in a peer-to-peer network.

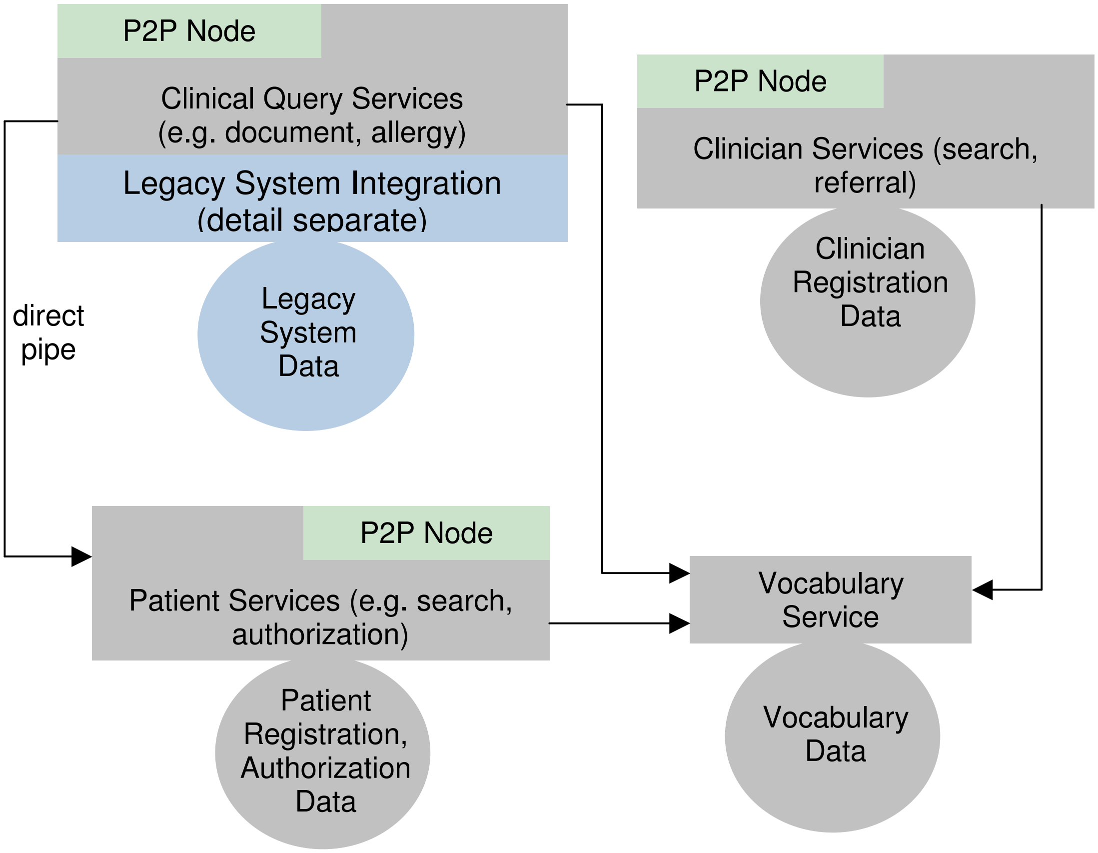

Figure 5.1: High-level service architecture. Services may be deployed at widely known network addresses for direct client-server access. Most services will also have P2P adapters so that they can operate in this mode as well. However even in a P2P network, a single, well-known vocabulary service is advised.

Client-Server Operation

Applications will be configured so that they can access a limited, pre-configured set of services at known addresses. In this mode, query messages will be posted over HTTP.

The client-server configuration is intended for small, well-defined deployments. This configuration simply connects clients to services directly. Care networks, firewall tunnelling, and message encryption are not supported.

This configuration is appropriate for use within a single hospital, or within a small community through VPN (virtual private network).

Peer-to-Peer Operation

Client-server deployment works well with a limited number of nodes. However, over a large region there are two problems with this approach:

1) Services will be added or removed very often in a large network, so an automatic mechanism is required for keeping track of services and their locations.
2) The total volume of queries on the network will grow very large. If all systems received all queries, the load on each source system would be too great. Therefore, it is essential to have a means of sending queries only to systems that know the patient.

These problems will be addressed using a hybrid peer-to-peer networking model. A peer-to-peer network is a dynamic network in which applications and services (all of which are "peers") find other peers automatically.

A distributed peer-to-peer topology will be used in conjunction with client-server, to take advantage of the strengths of each model (Minar, 2002). For example, multiple registration applications will communicate with a single registration server. The server, rather than the applications, will operate as a peer.

JXTA Protocol

The system will employ the JXTA peer-to-peer protocols (www.jxta.org). JXTA has the following desirable features:

- Platform-independence
- Language-independence
- Open-source Java language bindings
- Good scalability (improved by "rendezvous peers")
- Secure communication channels
- Firewall tunnelling
- Peer groups

Care Networks via Peer Groups

The "care network" concept is discussed in chapter 1. A patient's care network is a virtual network within which all EHR activity concerning a patient takes place. It includes all the source systems that store data about the patient, and all applications being used by members of the network at any given time.

Care networks serve the following purposes:

- Provide a secure environment for information exchange among authorized parties
- Limit the set of source systems to which queries are sent regarding a specific patient

Care networks will be implemented using JXTA "peer groups". A JXTA network is divided into concentric peer groups, each of which provides a context for advertising the presence of peers and services. Peer groups also provide security, in that a peer group may require peers to present credentials before being allowed to join. All JXTA peers belong to the global "network peer group". Within this, various industries create peer groups for various purposes (commonly file sharing, chat).

The current project will define an EHR peer group, which will be the basic point of access. It will contain peer groups for administrative services, and a peer group for every patient on the network.

Each patient peer group will have a JXTA "membership manager" that is hosted by the peer in which the patient has been registered. When clinicians try to join a patient's peer group, the membership manager will grant or deny access based on the identity and role of the clinician, using authorization information available through the patient service.

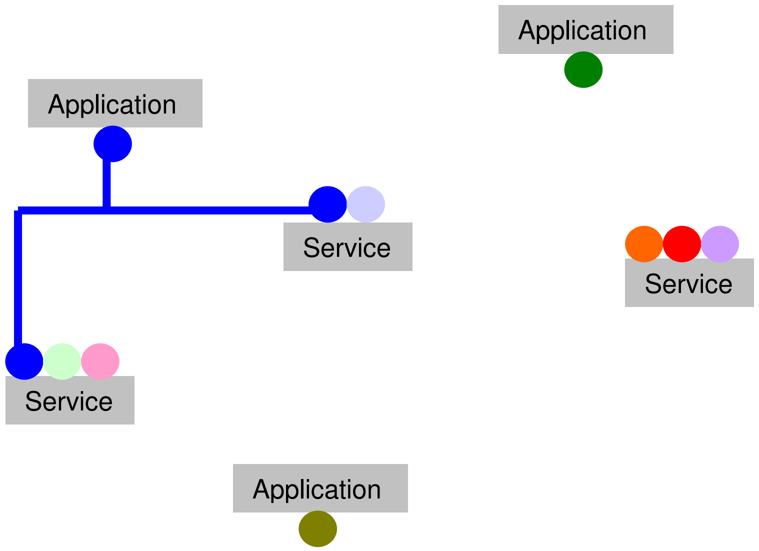

Figure 5.2: Ad-hoc Care Networks. The complete EHR network consists of applications and services running in various locations throughout a region. In this figure, different care networks (each corresponding to a different patient) are represented with different colours. A service will typically provide data about multiple patients. When an application is used to access data about a particular patient, the user signs on to the patient's Care Network (shown as a connection between the circles of one colour).

Scaling to Millions of Peer Groups

The normal way in which peer groups are used in a JXTA network is that an XML "advertisement" file is created and propagated through the network. The advertisement identifies the existence of the peer group and contains sufficient information (excluding security credentials) for a peer to join.

With a peer group for every patient, this method could result in millions of XML files on some nodes, causing performance and storage problems.

Instead, each registration service will store patients' peer group definitions in the patient repository. This will reduce the number of advertisements managed by any single service to a million or less in most cases, and these will be rapidly accessible through an LDAP directory instead of the file system.

A "patient search" peer group, consisting of registration services, will be used to scope patient queries. Applications will query these services to retrieve the data necessary to create the advertisement document locally. The reconstructed advertisement can then be used to join the peer group, without it ever having been published.

This method should scale very well. It also does not detract from the dynamic nature of the network, since broadcast methods can be used for queries within the "patient search" peer group.

Security

From the engineering perspective, the primary security consideration is secure transport via encryption. This is handled differently in the client-server and peer-to-peer deployments.

In the client-server deployment, message encryption will be deferred to a later phase. At present, this model is intended for use within a firewall or VPN.

In the peer-to-peer model, message encryption is an integral part of the JXTA protocols. Messages will be transmitted over "secure pipes", which are not visible outside the immediate peer group.

Chart Item URL Syntax

Ultimately, a clinical query will be expressed as a HL7 v3 message, directed at a known network address. However, we will also define a concise textual syntax for these queries in the form of a URL (uniform resource locator).

The URL representation will allow clinical documents to include links to related items in the patient's distributed chart. The links will also allow references to chart data within notifications and written communication between clinicians.

The BNF description of the chart item URL, following the conventions of RFC 1738 (Berners-Lee et al., 1994), is as follows:

```
url =  scheme “:” schemepart
scheme = “x-ehr"
schemepart = service “:” patient “:” request
service = *alphadigit
alphadigit = alpha | digit
patient = *alphadigit
request = query-request | item-request
item-request = “item=” item-id
item-id = *alphadigit
query-request = “query=” type “;” category “;” date-range
type =  “DOC” | “MED” | “AL” | “OBS” | “PROB”
category = *alphadigit
date-range = date “&” date
date = year month day
```

The following example would retrieve specific data item #4426 from system #100567, for patient 9989834:

```
x-ehr:100567:9989834:item=4426
```

The following example would query discrete observations under local code 112345 in system #100567, for patient 9989834:

```
x-ehr:100567:9989834:query=OBS;112345;20021024&20021105
```

Cache and Proxy

While not strictly necessary to operation of the network, a caching system will have great benefits in terms of performance and reliability. The intent is to provide a general caching mechanism that can be inserted at various points in the system. For example:

1) An institution could deploy a cache at the point of access for external parties. The cache would store frequently accessed results, reducing the query load on operational systems.
2) An institution could deploy a cache in series with outbound requests. Recently accessed results would then be stored within the institution’s LAN, improving performance and removing dependence on uptime of remote systems.

The cache will accept queries associated with the major services, search its own repository for matching results, obtain missing results from remote services, and return the combined results to the calling system.

The requirements are different from file caching (with the exception of documents), in that a given query may encompass both cached and un-cached data. The scenario is similar to database caching, except that knowledge of clinical processes can be used to optimize the aging of cached data.

Aging Algorithms

When original data can be changed, there is a danger of cached data becoming out of date. Thus caching systems typically discard or update data after a certain time.

There are two useful features of clinical data that can be used to optimize the aging algorithm for the PRO cache system:

1) When a patient is present at one location (e.g. admitted to a hospital), new observations about that patient are unlikely to appear at other locations.
2) Clinical observations and documents are less likely to be edited as more time passes after they are originally authored.

Metadata

The caching system will store metadata that defines its contents in RAM, so that a very fast determination can be made as to whether or not the required data are available there.

Set Operations

A query defines a set of data to be returned. Set operations are required on queries and cache metadata, to determine which data are not contained in the cache and must be queried remotely.

To query source systems for exactly the un-cached data may require multiple queries, which may be less efficient than a larger query that overlaps the cache. For example, suppose the cache contains (for a particular patient and a particular source system) the radiology reports from a single day. If a new query requires radiology and pathology reports for that entire month, it may be more efficient to use a single query for this information than to use non-overlapping queries (i.e. pathology reports for month + radiology reports for part of month up to day + radiology reports for part of month following day). For this reason, the cache system should always perform (for each source system) the smallest single query that will return the desired results when combined with cache data.

Persistence

Cache data will be persisted in RIM form, using the JDO API. In Java, Enterprise EJBs are often used for persistence. While these have the advantage of being distributed, this is not an important consideration here. JDO has the much more important advantages of better performance and inheritance support.

Proxy

It is anticipated that many source systems will not support direct query, but will make their data available through declarative HL7 version 2 messages.

Even where direct query is possible, it may be undesirable. For example, some source systems may have frequent (e.g. nightly) downtimes, heavy user loads, or poor performance.

For these cases, the persistence component of the cache will be re-used to create a proxy of the source system. Data from a source system will be copied into the proxy system (using batch transfers or HL7 messaging) and queries will run against the proxy. The aging mechanism of the cache will not be used in proxy repositories.

HL7 v2 conformance profiles will be created for a set of interfaces into the proxy, and the HAPI HL7 engine will be used to parse inbound messages an populate the repository.

Chapter Six: Technology Viewpoint

As discussed in chapter 1, this strategy will be implemented in a series of small, focused projects. Resulting software will be released under open-source licenses, allowing free re-use and redistribution by other parties.

Much of the initial development work will be done with Java™ technology, because it offers platform independence, strong typing, good error handling, and other safety features, as well as a large base of existing open-source software. However, the JXTA protocols and HL7 v3 messages are not language-dependent, so the system can be expanded as needed with applications and services written in other languages.

Re-use of existing open-source products is favoured over new development. Existing modules from open-source systems in healthcare (e.g. OpenEMed, PICNIC) will be used where possible. The following products are also expected to be useful:

- JXTA Java Reference Implementation (for JXTA networking)
- HL7 Java SIG implementation of HL7 v3 (for internal HL7 v3 messaging and persistence of RIM objects)
- Jakarta Tomcat (for web UI based on Servlets and JSPs)
- JBoss (EJBs in support of web UI)
- OpenLDAP (LDAP implementation)
- PostgreSQL or MySQL (relational database used in cache)
- Jakarta ObjectRelationalBridge (OR mapper and JDO provider)
- [Certificate Generator TBD] (used to create X.509 digital certificates)
- jTerm (terminology server)

The remainder of this chapter introduces (at a high level) the following initial group of projects:

- HAPI Routing and Conformance
- Semantic Cache
- Audit Trail
- Problem List
- Consent and Registration
- Query Adapters
- Default Review Applications

Detailed documentation regarding these projects is ongoing, and will be managed separately from this document.

HAPI Routing and Conformance

HAPI (http://hl7api.sourceforge.net) is an HL7 v2 engine. HAPI was originally released in November 2001, and has been downloaded by several hundred developers worldwide. It currently provides the following features:

- Message parsing and encoding (in both XML and ER7)
- Enforcement of the HL7 standards for data-type content and message structure
- Message transport and acknowledgement

The following additional features are required:

- Message Routing
- Validation and Conformance Testing
- Secure transport

Message Routing

HAPI currently has mechanisms for routing parsed HL7 messages to various consumer classes within a single executable program. Additional functionality is required to allow routing of both parsed and unparsed messages, at high volume, between producers and consumers running in different processes. Messages must be routable according to the values of any field in the message header segment. Routing should be configurable using graphical methods, and custom routing code should be supported for advanced cases.

The basis of the implementation will be integration of HAPI with the JMS (Java message service) API (http://java.sun.com/products/jms/) for message handling. JMS manages messages using “destinations”. Destinations include queues (1 consumer per message) and topics (0 to many consumers per message). HAPI will interact with JMS destinations in a way that automates HL7-defined application-level processing rules.

Routing and transformation of messages will be accomplished using “Converter” objects. A Converter will consume a message from one destination, perform a conversion, and then post the converted message to another destination. Thus, to obtain the form of the message that a particular system requires, the interface administrator will simply link the client to the appropriate Destination. Format Converters will map between message formats (e.g. Java object; XML-encoded; ER7-encoded). Content Converters will alter data fields according to frequently used rules or custom Java code.

Validation and Conformance Testing

HL7 messages must be validated against two sets of rules: 1) the basic rules defined in the standard, including rules for message structure, encoding, and data type content, and

2) additional constraints defined for a particular implementation (e.g. this EHR). The latter rules are expressed in terms of XML "message profiles" (see HL7 v2.5 section 2.12).

HAPI currently enforces HL7-defined rules for all messages. This feature should be improved by allowing administrators to turn individual rules on and off as needed. This will bring several benefits: 1) increased speed due to non-execution of irrelevant rules, 2) ability to define additional rules (e.g. additional constraints on telephone number format) that may not apply in all situations, and 3) disablement of rules that are not appropriate for a given deployment. Various types of rules will enforce data type content, encoding constraints (e.g. conformance to an XML Schema), and validation of messages against conformance profiles. A facility will also be created to enforce profile constrains in application code.

Internet Transport

HAPI should also be enhanced so that it can send encrypted messages through firewalls, over the Internet. To support firewall tunnelling, an intermediate HAPI node must be exposed on the Internet, so careful security analysis is required.

Semantic Cache

[Add high-level implementation description …]

Audit Trail

[Add high-level implementation description …]

Problem List

The Problem List project has yet to be initiated. This project will begin with review of existing problem list systems, and development of a requirements document.

Particular attention must be focused on the granularity of problem codes. Institutional problem list applications typically 10 000 or more codes. This presents a significant usability challenge, as it takes time for clinicians to navigate this code set. Two questions must be investigated early: 1) are the coding requirements the same for a longitudinal problem list as for an institutional problem list, and 2) what are the criteria for using existing problem databases?

To illustrate the motivation for the latter question, Canada maintains a large database of ICD-coded morbidities, for billing and analysis purposes. It might be possible to re-use this database, reducing data entry by clinicians, if 1) the data are available quickly enough after problems are observed, and 2) ICD is expressive enough to be useful in this context.

Patient Consent and Registration

Agents of well-known healthcare institutions will register patients in the EHR, as described in chapter 2. The main functions of the registration application will be as follows:

- Login
- Collection of patient consent and non-repudiation data (e.g. signature graphic; voice recording)
- Search for candidate records (records in various source systems that may correspond to this patient)
- Confirmation of candidate records with patient
- Storage of patient demographics and IDs within various source systems
- Creation of patient's unique ID on the network
- Creation of digital certificate
- Creation of CD with certificate (and review application, in a later phase)

Query Adapters

The purpose of the clinical query services is to provide access to data in source systems (i.e. operational clinical information systems).

A wide variety of clinical systems exists, and no single method of retrieving information from them is suitable in all cases. The task of integrating source systems can best be facilitated by creating a family of adapters, each designed to accommodate a different type of system (see figure 6.2).

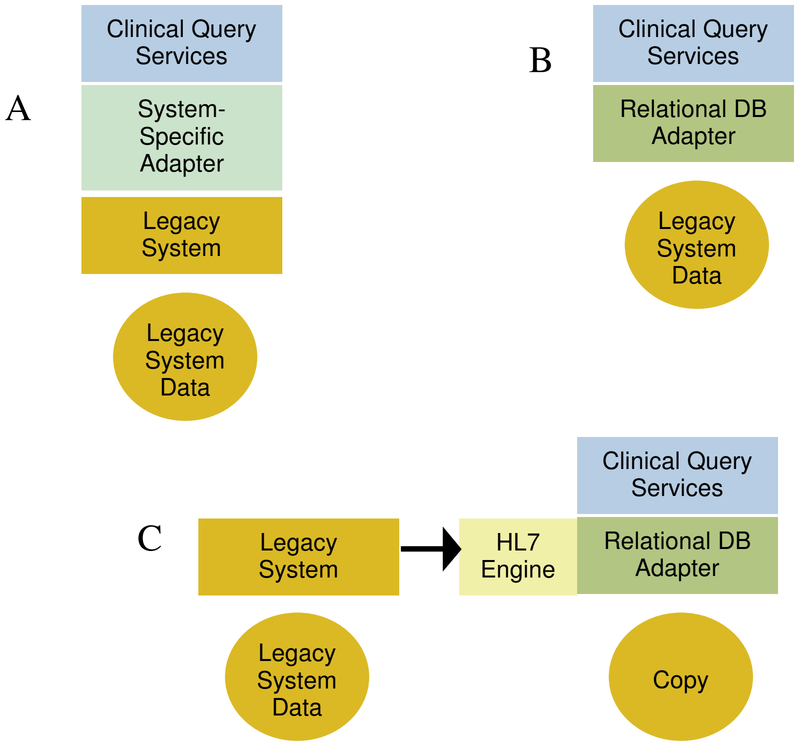

Figure 6.2: Examples of legacy system integration methods. A) An adapter for a particular product, B) A relational DB adapter that can be configured by a hospital's DBA, C) a proxy adapter (in which data are copied from an HL7 stream into an accessible database)

Technology-Specific Adapters

The following technologies are commonly supported by clinical information systems:

- ODBC and JDBC
- HL7 version 2

Generic adapters (not tailored to any specific product) will be written to retrieve data from source systems using these technologies. Deployment against specific source systems will require substantial configuration and/or customization on a case-by-case basis.

Relational Database Adapter

A variety of object-relational (OR) mapping tools exists. The main purpose of these tools is to simplify object-oriented programming by mapping objects to relational database tables, so that (once this is done) programmers need only deal with objects. There has been great attention and competition in the area over the last few years, and currently there are several open-source products available. We will deploy such a tool, partially configured to work with the core query services.

We will also provide a code-mapping tool for translating the limited set of codes that are used as query parameters (e.g. “document category”) to native codes.

HL7 v2 Query Adapter

Some systems support the HL7 v2 query-response mechanism. An advantage of this mechanism over ODBC is that HL7 defines the meanings of data elements, which simplifies analysis and mapping.

The clinical query services will be implemented in the form of HL7 queries, using the HAPI HL7 engine. The query and response message formats will be defined using conformance profiles (see HL7 v2.5 section 2.12).

HL7 v2 Proxy

Many clinical systems declare internal data changes using HL7 v2 unsolicited messages. This mode is more commonly supported than HL7 queries.

Such systems can be integrated by creating a proxy (see chapter 5). A set of conformance profiles will be created to define the various messages required to populate this proxy. HAPI will listen for messages and populate the proxy repository.

In general, some message transformation will be necessary to manage differences between the proxy’s message definitions and those of the source system. Most commercial interface engine products are suitable for this task. This functionality will also be added to HAPI (see HAPI section earlier this chapter).

Product-Specific Adapters

There are hundreds of clinical information systems on the market, but a limited number of these are widely installed. Over time, as the EHR expands to include new source systems, a library of system-specific adapters will emerge. It is expected that even a modest number of adapters will greatly facilitate the growth of the network.

In keeping with the open-source model, those who deploy the software against source systems for which adapters do not yet exist will be encouraged to develop them and contribute them to the project. Interested vendors may also offer adapters to their systems as commercial products.

To begin, model implementations will be constructed against the Java APIs of two major HIS products: Cerner Corporation’s PowerChart and Per Se Technologies’ Vista. These products are chosen because of their availability to those currently involved in the project.

Appendix A: Data Sharing and Federation Techniques

Methods of Sharing Clinical Data

There are several common methods of sharing clinical information between clinicians in a given region. These include the following:

1) Require all clinicians to use the same software. The software vendor may then provide a proprietary means of passing data from one installation to another. This approach has been adopted in many regions. However, it is not practical in regions that have existing software investments with various vendors, nor does it permit the sharing of data between regions (for example when a patient moves or travels). There are also inherent drawbacks, such as the difficulty of finding a system that meets the needs of different clinical groups, and (with proprietary systems) vendor lock-in. While the advantages and disadvantages can be argued, it is unlikely that this approach will be adopted universally.

2) Visual Integration. The user interfaces of multiple systems are synchronized, so that on a given desktop, all open applications display information in the same context (i.e. they display information about the same patient and possibly the same procedure). A good example of this technique is the CCOW (clinical context object workgroup) standard. However, this method is inappropriate as a regional strategy. Within a single city dozens of different software products may be used – it is not practical to license or install all of them for every clinician, or to expect clinicians to become simultaneously proficient with each user interface.

3) Mutual Replication. Information entered into one system is sent to one or more other systems (e.g. via HL7), so that each system can present all required information in its own user interface. This approach works well with a small number of systems, for example within a single institution. However it becomes less feasible the more systems are involved, because of semantic differences between systems and sheer volume of data.

4) Federation. Clinicians enter information into the systems that they use in their daily practice, and this information is combined into a longitudinal record separately (through various means). Federation has the following advantages: 1) it is more scalable than the other approaches (i.e. it is practical to integrate a wider region in this manner), 2) clinician and institutions have maximum freedom to equip themselves with the information systems that best meet their particular needs.

Federation Approaches

There are several ways in which information from multiple clinical systems can be federated.

1) Central Repository. In this model, data from source systems are replicated into a large regional database (figure A.1). Clinicians are given access to a user interface through which information in this database can be viewed. For regional integration, this approach improves on the "mutual replication" method (see previous section) in that there is only one large database, instead of many large databases. However, as the size of the region increases, maintenance costs increase, and performance of the database is threatened. The database is a potential single point of failure, so it should be replicated, further increasing the cost. Another concern is that organizations forfeit control of their own records. However, there are also clear advantages to this approach: 1) it allows collection of data from intermittently connected systems, 2) since all the data are in one store, ad-hoc query can be supported using the native query methods of that store.

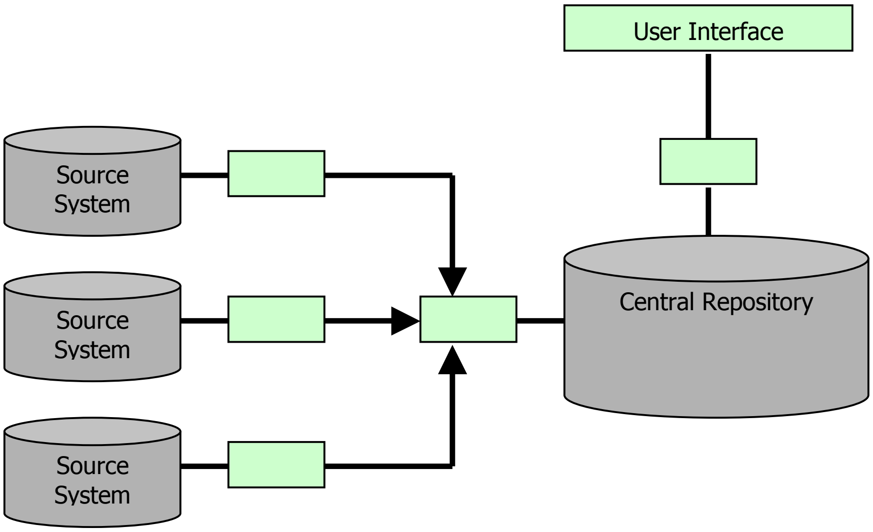

Figure A.1: Central repository model.

2) Query Service. An alternative to the central database model is to provide instead a relatively lightweight central service, which queries source systems on demand (figure A.2). The advantages of this approach are reduction in cost and increase in scalability, compared to the central database approach. However, it is still necessary to maintain the service software centrally. In some regions it may not be obvious who should pay for this or how the costs should be shared. This method is also inappropriate for obtaining data from intermittently connected systems, and extra measures must be taken to support ad-hoc queries.

This method also has the apparent disadvantage of increasing load on the source systems, because they are queried on demand instead of sending all data to a central location.

However, since patients are often treated in short flurries of activity (e.g. a hospital visit; a series of appointments with the family physician), augmenting the central service with a relatively small short-term cache could dramatically reduce the volume of queries that actually reach the source systems.

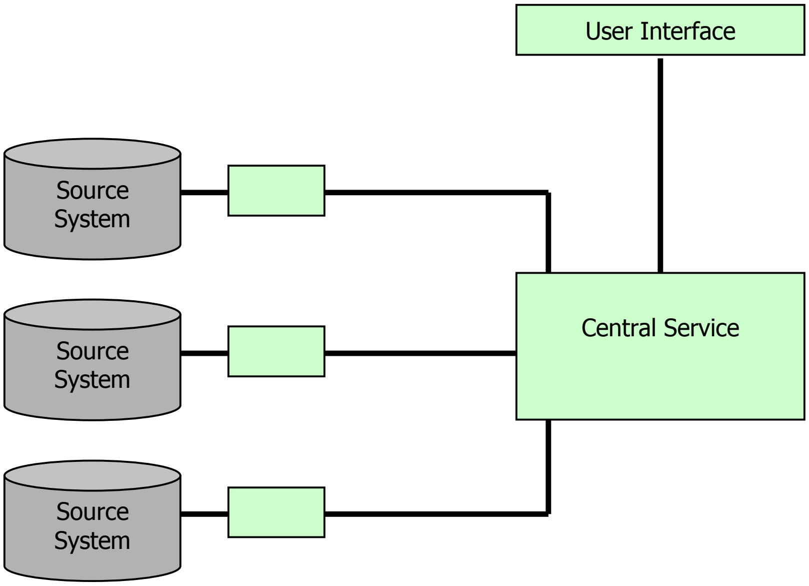

Figure A.2: Central service model.

3) Peer-to-Peer. Like the Query Service method, source systems are queried on demand, and the results may be cached. However, in this model there is no central service through which queries are routed. The central service is replaced by a protocol by which applications discover source systems and query them directly (figure A.3).

The peer-to-peer model eliminates the need for a central service of any kind. It enables the growth of ad-hoc regional networks, which can merge into larger networks with little effort. It also eliminates the danger of a single point of failure or a single point of vulnerability to a denial-of-service attack. However, this approach has the disadvantage (compared to the central service model) of increased complexity.

4) Hybrid model. The preceding models can be combined in various ways. For example, within a province, large institutions with persistent network connections could support query services, while smaller or intermittently connected systems could copy data to a central database. Alternatively, groups of closely affiliated institutions (e.g. providers within a certain city) could use a common central repository, and query services could be used for larger scale integration of the whole province. Rather than prescribing one topology for an entire region, various approaches can be used where they are most appropriate.

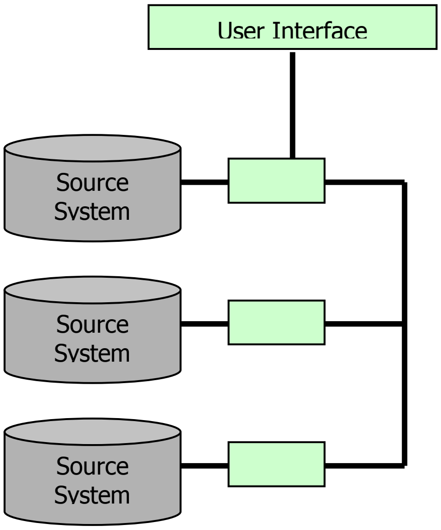

Figure A.3: Peer-to-peer service model

Glossary

Care Network: A virtual network that provides a secure boundary and limited scope for data exchange about a single patient

Community Hospital: Primary care hospital.

Document: A textual record made by a clinician concerning a patient. Examples include discharge summaries and progress notes.

EHR (electronic health record): A comprehensive source of information about a patient's medical history, ideally spanning the patient's whole life. EHR is a common term in medical informatics, and many slightly varying definitions have been published.

HL7: A prominent standard that defines the content and structure of messages sent between clinical information systems (www.hl7.org).

Granularity: Refers to the spectrum of fine-grained to coarse-grained data. For example, a set of discrete haemoglobin measurements has finer granularity than the same information stored as a single block of text. Generally, finer granularity allows software to do more with data (e.g. a set of discrete haemoglobin results can be trended or graphed, while a block of text can only be displayed as-is).

JXTA: A peer-to-peer protocol (www.jxta.org)

Observation: The record of a clinician's assertion of a discrete fact about a patient. Examples include lab test results, blood pressure measurement. Observations are distinguished from orders and documents. Also excludes medical images and their interpretation (the latter are treated as documents).

Peer-to-peer (P2P): A family of methods for connecting information systems so that they communicate directly with each other, rather than (or as well as) with a central server.

Problem List: A list of clinical problems experienced by a single patient. Problems are things that require medical care (e.g. stroke, broken leg).

Registrar: A person responsible for maintaining records about people in the EHR network, including patients and clinicians. A registrar is responsible for registering people in the system.

RIM (reference information model): The HL7 (www.hl7.org) standard information model, on which messages from version 3 of the HL7 standard are based. The RIM is a concise, abstract model of a wide variety of healthcare processes and data.

Service: A piece of software that performs some (generally well-defined) function for other software. A service is distinguished from client software, which is a consumer of one or more services. A service often operates at a distance from the client.

Source System: A clinical information system that supplies data to the federated EHR network.

References

Berners-Lee T, Masinter L, McCahill M, 1994, Uniform Resource Locators, IETF, RFC 1738

Brooks, FP, 1995, *The Mythical Man-Month: Essays on Software Engineering*.

Canada Health Infoway, 2002, Pan-Canadian EHR Survey, Phase 1, Results and Analysis (http://www.canadahealthinfoway.ca)

Dick RS, Steen EB, & Detmer DE, 1997, *The Computer-Based Patient Record: An Essential Technology for Health Care, Revised Edition*, Institute of Medicine

Krishnamurthy S, 2002, Cave or community? A critical examination of 100 mature open source projects, http://opensource.mit.edu/papers/krishnamurthy.pdf

Minar N, 2002, Distributed Systems Topologies: Part 2, http://www.openp2p.com/pub/a/p2p/2002/01/08/p2p_topologies_pt2.html

Romanow, RJ, 2002, *Building on Values: The Future of Health Care in Canada*, Commission on the Future of Healthcare in Canada.

Smith J & Cousins A, 2002, The cross-continuum regional electronic health record: consensus and charting a path forward, Proc eHealth 2002

Smith C, 2002, Open source software and the NHS: white paper, NHS Information Authority, http://www.nhsia.nhs.uk/def/pages/features/i_250202.asp