MEDINFO 2004
M. Fieschi et al. (Eds)
Amsterdam: IOS Press

A Submission Model for Use in the Indexing, Searching, and Retrieval of Distributed Pathology Case and Tissue Specimens

Ahmad H. Namini[a], David A. Berkowicz[a], Isaac S. Kohane[b], Henry Chueh[a]

[a] *Laboratory of Computer Science, Massachusetts General Hospital, Boston, MA USA*
[b] *Children's Hospital Informatics Program and Harvard-Partners Center for Genetics and Genomics, Boston, MA USA*

Abstract

This paper describes the Shared Pathology Informatics Network (SPIN) submission model for uploading de-identified XML annotations of pathology case and specimen information to a distributed peer-to-peer network architecture. SPIN use cases, architecture, and technologies, as well as pathology information design is described. With the architecture currently in use by six member institutions, SPIN appears to be a viable, secure methodology to submit pathology information for query and specimen retrieval by investigators.

Keywords:
Pathology, Distributed Systems, HIPAA

Introduction

Modern biomedical research requires access to patient materials that include tissues, diagnostic specimens, and their related clinical data. Investigators within the hospital community have recognized the value of these specimens. Tissues, in particular, have been collected for many years and used effectively to advance various areas of biomedical research. For example, tissue banks have been historically valuable in the study of human cancer. With the imminent completion of the human genome project and the revolution in molecular genetic technology, these existing tissues become increasingly important as valuable sources of information. New molecular technologies are also increasingly applicable to formalin-fixed paraffin-embedded tissues, once only the domain of scarce fresh frozen tissues. This translates into potentially large-scale research on the millions of conventionally processed tissue specimens archived in anatomic pathology departments.

Practical access to large, diverse collections of annotated pathology specimens and related clinical data across institutional boundaries is very limited, at best. Based on these needs, the United States National Cancer Institute (NCI) supports the Shared Pathology Informatics Network (SPIN)[1] initiative as a cooperative grant between two consortia, Harvard/UCLA[2] and Indiana/Pittsburgh[3]. The objectives of the Harvard/UCLA consortium are as follows:

- Define a scalable and extensible representation for tissue specimen annotation in XML that can be queried by researchers, clinicians, and the general public.
- Formulate taxonomy for confidentiality and patient consent that describes how a specimen can be used.
- Design, develop, and deploy a distributed software system so as to permit investigators to locate appropriate specimens based on desired query characteristics. The distributed system should provide functionality for indexing and searching pathology cases and retrieving specimens dispersed within the research community, while permitting those institutions participating within SPIN to maintain local control of their data. Technology requirements for SPIN users should be a standard web browser and an Internet connection.

Although SPIN's architecture, design, implementation, and monitoring represents the efforts of many researchers and staff within both consortia, this paper deals only with the Harvard/ UCLA consortium's submission model for uploading de-identified annotations. Major use cases defining SPIN functionality are first stated, along with the major components known as peer types, which provide services to realize all use cases. Then, a discussion of pathology and specimen information design is presented. Afterwards, implementation of the submission model as well as the technologies used is presented. Finally, a discussion of open issues regarding the submission model is offered.

Functionality

For years, the National Network of Libraries of Medicine[4] and its associated DOCLINE services have provided a model for centralized coordination of locally autonomous (but cooperating) resources for sharing information across medical libraries. Although, centralized coordinating of distributed services does provide an alternative, the Harvard/UCLA consortium decided that a peer-to-peer (P2P) data storage and communication model would yield the most scalable system of query transmission and results-set formation.

While traditional client-server computing environments degrade in performance with increased load, P2P architectures tend to scale better while also providing fail-over, redundant service similar to that of the Internet. Moreover, a P2P model reinforces local institutional control of pathology case and specimen data.

With a network of peers located throughout the country, each institution is required to conform to the Common Rule[5] and HIPAA[6] federal regulations that govern the research uses and electronic transfer of confidential medical records. For wide acceptance and possible legal challenges, the SPIN architecture relies on strict adherence to data confidentially, integrity, investigator authentication and authorization at every level of SPIN interaction. Moreover, SPIN must document its conformance features for local Institutional Review Board (IRB) approval, while also being flexible for possible stricter compliance for particular participating institutions.

With the use of a P2P distributed data model, the architectural design chosen is of a canonical form, whereby the codebase resident on any peer is identical throughout the entire network of peers. However, after accepting a SPIN participation policy and receiving a digital certificate, an administrator can configure his or her peer so as to perform any or all of the following top-level use cases:

- Discovery - Discover other peers within its own peer group or globally amongst all peer groups. A peer group represents a collection of peers that subscribe to a common topic and/or are geographically bound. For instance, one peer group might consist of peers interested in tumor registries throughout the nation, while another peer group might consist of all peers located within the Harvard Medical School.
- Submission - Listen for, accept and capture/record pathology case information, where the message containing the data has been de-identified and secure during communication. In addition, a codebook datastore exists in the node- and/or client-side, which map the pathology case data residing within a peer to the actual specimen. With proper scrubbing of data, the codebook is the only persistence entity that would contain identification information.
- Query and Query Aggregation - Initiate a query on behalf of an investigator or the general public, and aggregate any results from multiple peers. All results are then forwarded to the user that initiated the query. Currently two query types are permitted. The first, a statistical query computes statistical properties of the result set, e.g. mean, medium, variance, histogram. The second, a detailed query returns the result set.
- Specimen Retrieval - From the results of a query, an investigator can request any or all specimens from the specimen archive.

Although other major use cases critical to SPIN functionality, such as peer registration, administration, security, and maintenance exist, these are not discussed within this paper due to length limitations.

Peer Types

At the foundation of the SPIN architecture, any peer is designated as a combination of peer types, where each type encompasses the components and services delivering SPIN functionality. Each peer type's components and services are described as follows:

- ToolsNode - A peer that listens for and accepts submissions, and then brokers the submission message to a SPIN Node. If a server-side codebook is utilized, a ToolsNode provides functionality for codebook datastore operations. Any specimen retrieval request is broadcast throughout the network, and will eventually reside within a messaging queue, where a codebook proxy can view the request message and then inform the specimen holder, i.e. the codebook owner. In addition, phonetic encoding and hashing services are provided.
- Node - A peer that provides data manipulation services for the pathology datastore. A Node will serve as a conduit for query requests from a database that in turn will be sent back to the requesting SuperNode. All Nodes can be visualized as a terminal node in the network topology.
- SuperNode - A specialization of the Node peer type. A peer type that acts as a router of messages between itself and other SuperNodes, and its children Nodes. A query request will traverse the entire SPIN topology by first initiating its request through a SuperNode.

Information Design

Peer institutions have diverse source formats of their data, ranging from text in analog paper form or as digital flat files to more sophisticated relational or object databases. In addition, source data exists with varying degrees of granularity. Some institutions having very detailed (fine-grained) pathology and specimen information, with others being less detailed (coarse-grained).

Instead of mapping directly to these in-house formats, SPIN mandates that the community of institutions adhere to an open standard developed by pathologists. This standard exists as an XML submission schema[1], in which mandatory and optional items exist. Some of the more pertinent data items within the schema are information detailing diagnosis, dimension, patient, and tissue specimen.

Before any pathology or specimen data can be submitted, the XML annotation of this data must pass schema validation. The XML annotations are de-identified and scrubbed of any "reference" corpus of cases. It is the responsibility of SPIN member institutions to have their own data de-identified and scrubbed, although SPIN is currently refining tools for scrubbing and coding.

At present, only pathology-related data exists within the schema, however, in the future it is envisioned to incorporate tumor registry definitions, sample derivatives (e.g. microarrays), consent semantics, and specimen availability with associated cost information.

Implementation

To submit data to the SPIN network, a member institution utilizes a Java application known as the SPIN Submission Tool. The Submission Tool is a standalone application which reads the XML annotations, validates the data against the previously mentioned data schema, prepares the data for secure transport to a specified ToolsNode, and then awaits response from the ToolsNode so as to update the client-side codebook. While the Submission Tool blocks, each submission is inserted into one or many Node datastores (see Figure 1).

One of the more difficult consequences of a distributed submission model is that of concurrency and datastore integrity. Concurrency design should account for how much isolation each transaction has with other transactions. In the submission of data, a transaction is defined as the aggregation of the following tasks: (1) Submission Tool reads an XML annotation and wraps the requests and sends it synchronously to a ToolsNode; (2) ToolsNode receives a submission for processing and wraps the request and forwards submissions to client-requested Nodes; (3) if the client submitter chooses to map to a node-side codebook, ToolsNode inserts entries into its codebook detailing successful Node submissions; and (4) client-side SubmissionTool receives notification of all successfully inserted cases and then adds entries into its client-side codebook.

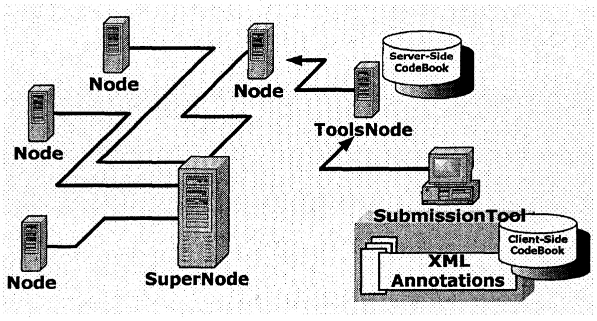

Figure 1 - SPIN Submission Process

A distributed transaction monitor has been developed whose responsibility is to maintain datastore integrity between Node databases and codebooks (ToolsNode and client-side). Integrity is maintained by ensuring that all entries within the codebook correspond to successful insertions within the Node datastore. Difficulties in achieving this are mainly due to lost network connection, overloaded resources resulting in session or datastore connection timeout, or moved or unavailable datastores.

In addition, the magnitude of submissions can cause problems. As of last count, participating SPIN peers that have submitted XML annotations total six medical institutions totaling a set of roughly 1.4 million cases stored within their respective peers. The Submission Tool provides services for ease-of-use in preparing and loading submissions. For instance, the following features exist:

- Validate XML annotations against the schema but do not submit. This permits submissions to be "pre-processed" for validity before the long submission process.
- Graphical and automated console modes for submissions. The graphical mode is ideal for a small number of submissions processed interactively. The automated console mode is ideal for a large number of submissions and can be triggered via a timestamp so that submissions will commence in periods of lesser network activity.

Technologies

The SPIN architecture has leveraged many well-known software paradigms and technologies used in major distributed computing enterprise systems. Of these, the paradigm most useful is the n-tier architecture, whereby functionality is subdivided into individual layers. Implemented technologies associated with any tier are the following:

- The presentation-tier utilizing servlets, Java Server Pages (JSP), and browser-based applications known as applets.
- The business-tier relying on Enterprise Java Beans (EJB) and web services.
- The data-tier employing a relational database.

Any business logic, user functionality, and/or request are based on exposing components and services, which ensures a high degree of encapsulation, software reuse, and ease of maintenance and portability. All data modeling utilizes XML with transport mechanisms being either HTTP(S) or RMI-IIOP.

The Java programming language has been chosen due to its ability to run on any operating system without the need for re-compilation. All developed systems currently operate in both the Windows and Linux environments.

All chosen technologies are from the open-source software communities, which ensures virtually no software purchasing costs. These include the following:

- Apache web server for ensuring a robust, thread-safe, HTTP(S) client-server framework.
- Tomcat servlet container for responding to JSP and servlet requests.
- JBoss application server, which includes an EJB container for hosting business components (session-, entity-, and message-driven beans), along with implicit middleware addressing authentication and authorization (JAAS), messaging (JMS), and transactions (JTA).
- JNDI naming services for providing component lookups.
- AXIS that provides a SOAP-compliant web service provider.
- Java Web Services Pack for managing XML documents, consuming web services, and XML transformations.
- Struts framework's use of JSP Model 2 for developing and maintaining web applications based on the Model-View-Controller (MVC) design pattern.
- MySQL relational database for the storage of data.
- JDBC providing data services for creating, reading, updating, and deleting data from a relational database.

Open Issues

The submission model described herein has been in beta release use for sometime, with feedback from member institutions constantly driving creation and revision of feature sets. Certain open issues, which have yet to be finalized in either design or implementation stage, are discussed here.

Much attention has been paid to de-identifying patient-specific materials and protecting local and distributed codebooks. However, equally relevant issues are that of authorizing and authenticating submitters and ensuring some quality control of the submissions. As an initial design addressing authentication and authorization, the Harvard/UCLA consortium has developed a registration process, whereby an institution applies for membership, in which the institution pledges to adhere to a participation agreement. If accepted, the institution receives a public/private key pair, which permits submitters to digitally sign their submissions. The quality of the submissions is currently not explicitly being addressed, but it can be assumed that participating institutions will be of the highest quality. During the submission process, a message digest is transported to ensure the submission message's data integrity.

The submission process is quite complex. Amongst the various datastores, which host pathology case and specimen information as well as codebooks, data integrity must be assured. This is further complicated by the fact that datastores can be distributed across various peers. To ensure integrity, the design has leveraged the well-known paradigm of transaction processing, in which any one transaction must be atomic, consistent, isolated, and durable, commonly referred to as the ACID conditions. The JBoss application server provides middleware to ensure that a transaction maintains ACID conditions when all transactions steps occur on one server.

However, the complication introduced by submissions requiring distributed transactions necessitates the use of a journaling system, currently hosted on a ToolsNode. The journaling system acts as a log of pending and completed steps within a particular distributed transaction. If for any reason, a distributed transaction fails, a transaction monitor can read form the journal and cleanup broken transactions. Messaging is used to communicate with the journaling system since message queues are by definition durable.

Acknowledgements

This work was supported by grant UO1 CA91429 from the National Cancer Institute. Its contents are solely the responsibility of the authors. The authors wish to convey their gratitude to the numerous collaborators from both SPIN consortia for their comments and suggestions. Also, the authors would like to acknowledge this paper's reviewers, whom offered valuable comments and suggestions.

References

[1] Shared Pathology Informatics Network (SPIN), National Cancer Institute (NCI), http://spin.nci.nih.gov

[2] Shared Pathology Informatics Network (SPIN), Harvard Medical School, http://www.mgh.harvard.edu/path/chirps/chirps.html

[3] Shared Pathology Informatics Network (SPIN), Regenstrief Institute for Health Care, http://informatics.regenstrief.org/what/?section=spin

[4] National Network of Libraries of Medicine, http://nnlm.gov/

[5] Title 45 CFR (Code of Federal Regulations), Part 46. *Protection of Human Subjects; Common Rule, Federal Register.* Volume 56, June 18, 1991, pp. 28003-28032.

[6] Title 45 CFR (Code of Federal Regulations), Parts 160 and 164. *Standards for Privacy of Individually Identifiable Health Information; Final Rule. Federal Register,* Volume 67, August 14, 2002, pp. 53181-53273.