High-Performance Systems for in Silico Microscopy Imaging Studies

Fusheng Wang[1], Tahsin Kurc[1], Patrick Widener[1], Tony Pan[1], Jun Kong[1], Lee Cooper[1], David Gutman[1], Ashish Sharma[1], Sharath Cholleti[1], Vijay Kumar[2], and Joel Saltz[1]

[1] Center for Comprehensive Informatics
and Department of Biomedical Engineering
Emory University, Atlanta, Georgia, USA
[2] Dept. of Computer Science and Engineering
Ohio State University, Columbus, Ohio, USA

Abstract. High-resolution medical images from advanced instruments provide rich information about morphological and functional characteristics of biological systems. However, most of the information available in biomedical images remains underutilized in research projects. In this paper, we discuss the requirements and design of system support for composing, executing, and exploring in silico experiments involving microscopy images. This framework aims to provide building blocks for large scale, high-performance analytical image exploration systems, through rich metadata models, comprehensive query and data access capabilities, and efficient database and HPC support.

1 Introduction

Technologies for in vitro imaging of biological systems at the microscopic level have advanced significantly in the past decade. Commercial microscopy scanners are now capable of producing high-magnification, high-resolution images from whole slides and tissue microarrays within several minutes. These capabilities reduce dependency on glass slides for expert reviews to assess tissue quality and diagnose disease stage. Moreover, they enable novel *in silico* imaging studies[1] of normal and disease states of biological systems at cellular and subcellular scales. High-resolution image data offers enormous information with which to examine the spatial characteristics and relationships of subcellular structure of specimens under study. A better understanding of those characteristics can lead to better biomarkers or unveil new insights into disease mechanisms.

Software for use of digitized slides in clinical setting is typically characterized by the functionality it provides for a user to browse, view, and manually

[1] The term "in silico study" or "in silico experiment" broadly refers to a study or an experiment performed on a computer via analysis, mining, and integration of databases and/or through simulations.

P. Lambrix and G. Kemp (Eds.): DILS 2010, LNBI 6254, pp. 3–18, 2010.

annotate individual slides for tissue quality control and diagnosis. In silico experiments involving image data, on the other hand, have different characteristics and introduce a richer set of data access and processing patterns.

First of all, image data can reach very large volumes. Each image obtained from a whole tissue slide using a state-of-the-art scanner can be tens of gigabytes in size. Large studies may involve thousands of slides obtained from a large cohort of subjects. The sizes of these image datasets can range from terabytes to hundreds of terabytes – it is not too far-fetched to expect that dataset sizes will scale to petabytes, thanks to continued advances in scanning technologies. Such large scale data poses problems in storing, managing, and querying the data.

Second, image data is processed using simple and complex operations and by analysis workflows of various types in in silico experiments. Data processing operations may include filtering, correction of image acquisition artifacts, intensity normalization, registration, segmentation of structures (e.g., nuclei and blood vessels as shown in Figure 1), extraction of features, and classification of segmented structures. These operations can be combined in a variety of ways to form analysis workflows. The sizes of high-resolution images and the complexity of such operations as segmentation and classification may result in long execution times and may require large main memory and powerful computers. Clearly, large scale image analyses are good candidates for execution on parallel and distributed machines.

Third, results from image analyses, whether obtained via manual classification by an expert reviewer or through computer methods, should be expressed in a form that supports efficient synthesis of information. This is necessary to enable sharing and further exploration of results from an in silico experiment, to facilitate comparisons across multiple analyses, and to support rapid development and algorithm evaluations – a large scale study may involve hundreds of methods and analysis workflows. Rich metadata needs to be captured in order to describe analysis results (e.g., nuclear texture, blood vessel characteristics) and the context of the image analyses. With large datasets, researchers have to store, manage, and interact with large volumes of metadata about segmented anatomic structures, markups and features computed for each anatomic object, and semantic information associated with annotations (about cell types, genomic information associated with cells, etc). It is also important to model analytic procedures and pipelines used to carry out segmentation, feature generation, and classification.

Furthermore, comprehensive query support is needed. Researchers would like to query anatomic structures and objects, semantic annotations on objects, and spatio-temporal relationships in order to mine, explore, and correlate the characteristics of specimens under study and integrate with other types of data such as omics and clinical data. A researcher may, for example, want to search for blood vessels by not only shape features like length or thickness but also by their types. In an algorithm evaluation scenario, queries may look for the amount of overlap between objects detected by different algorithms or differences in classification results from an algorithm and a human. A whole slide image may contain millions

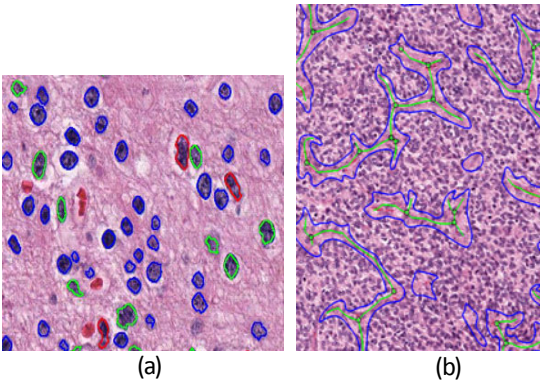

Fig. 1. Examples of image markups: (a) Nuclei; (b) Blood vessels

of anatomic structures, which may have complex shape and texture characteristics, hence there may be millions of annotations associated with the image. A repository of analysis results should be able to support queries on terabytes of image data and hundreds of millions (even billions) of anatomic structures, features, and semantic annotations.

We have presented and discussed solutions to the first and second challenges elsewhere [13,14,12]. In this paper, we propose and discuss a data warehouse framework to support storage, management, and querying of results from in silico image studies. We present an implementation of the core repository infrastructure of the proposed framework. This implementation uses an object-oriented model, called Pathology Analytical Imaging Standard (PAIS) [21], for representation of image analyses. It employs a relational database management system, IBM DB2, for data storage and management.

2 An Example Application Scenario

There are a variety of studies that make use of microscopy imaging, including characterization of the tumor microenvironment and comparative analysis of tissue microarrays. Here, we present a multi-scale integrative research project in cancer research as an example application scenario. We will use this example to illustrate the requirements and design choices to be presented in the following sections.

The In Silico Brain Tumor Research Center (ISBTRC) is a research project funded by the National Cancer Institute as one of the six In Silico Research Centers of Excellence (ISRCE, https://wiki.nci.nih.gov/display/ISCRE). The overarching goal of the ISRCE program is to carry out novel scientific research by analyzing, mining, and integrating publicly available biomedical datasets. The ISBTRC conducts hypothesis-driven translational research on brain tumors. Initially the research will focus on mechanisms for better classification of diffuse gliomas and on the biology of disease progression. This research makes use of

complementary genomic, pathology, and radiology brain tumor data from the Cancer Genome Atlas (TCGA), Rembrandt, and Vasari studies.

The ISBTRC is undertaking multiple approaches to study gliomas. One of them involves systematically executing and evaluating in silico experiments to look for relationships between 1) nuclear shape and texture in microscopy images and gene expression profiles defined by molecular clustering analyses and 2) the characteristics of angiogenesis (as detected in microscopy images), gene expression profiles, and neuroimaging features. In the in silico experiments designed for the pathology data, high-resolution images from whole slides are reviewed by expert pathologists as well as analyzed by computer algorithms. The pathologists mark up histological entities of interest on a selected subset of slides, annotate the structures (i.e., assign a classification value), and grade each selected slide. Computer algorithms segment anatomic structures, compute a set of features (ranging from the area and elongation of a nucleus to the bifurcations of blood vessels), and annotate each segmented structure with a semantic classification value (e.g., astrocytoma or oligodendroglioma). The pathologist reviews are used in validation of image analysis methods and to improve the algorithms' segmentation and classification results. Markups and annotations from multiple algorithms also are compared to assess the relative performance of the algorithms.

3 A Framework for in Silico Experimentation with Pathology Images

As we have alluded to in the introduction section, in silico experiments add new data access and processing requirements on top of the basic requirements of viewing and manually annotating individual slides. In a typical study, volumes of image data will be analyzed and mined by computer algorithms to look for morphological patterns that can assist in developing new hypotheses or proving/disproving a hypothesis. Since it may not be feasible to manually examine and classify each slide in a large study, multiple computational methods and workflows may be employed. By comparing and evaluating results from different analyses, a researcher can assign a confidence level to the experiment outcome. The researcher may also design an experiment to rapidly evaluate algorithms in their early stages of development to assess algorithm accuracy and speed. This type of experiment would involve running the algorithms possibly many times against one or more datasets as well as querying, retrieving, and comparing results from other algorithms and previous runs.

We describe at a high-level a software framework to address these types of data access and processing requirements. An illustration of this framework is provided in Figure 2. The framework consists of four main components. We now briefly describe these components.

Analytical Workflow Component. This component implements support for execution of analysis methods and workflows. For large datasets, it should take advantage of parallel and distributed machines and enable data-parallel and

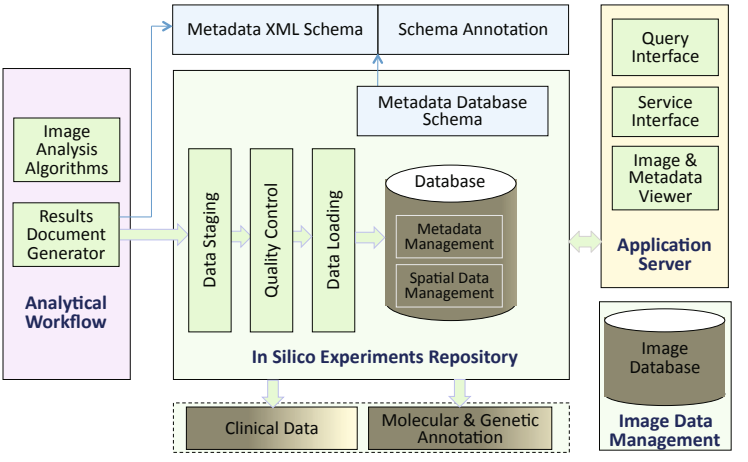

Fig. 2. Analytical microscopy imaging framework

task-parallel implementations of workflows that consist of a network of data processing operations. A subcomponent of this component is the results document generator. Each image analysis application or human annotation application generates the final result data in a format that conforms to the metadata model schema. Provenance information also is encoded in the document; the provenance information could include metadata about algorithm or workflow, analysis parameters, and input and output datasets. For example, in our implementation for the ISBTRC project, results and provenance information are submitted to the results repository as XML documents, conforming to the PAIS metadata model (see Sections 5.1 and 5.2).

Image Data Management. The image database provides the central repository for all microscopy images referenced in a study. To optimize data retrieval speeds for queries on large images and image regions, each image is partitioned into tiles or chunks. These chunks are distributed across multiple disks or storage systems to increase parallel I/O opportunities and are clustered on disks to reduce I/O seek overheads. The images or image tiles are stored in compressed form using a multi-resolution compression scheme in order to reduce storage and I/O costs. Multi-resolution spatial indices, such as R-trees, are employed to reduce the cost of searching the tile set of interest. An implementation of the image data management component is presented in [6].

Application Server. The application server component provides interfaces for query, algorithm invocation, data exchange and sharing, and data viewing. The query interface facilitates a flexible, convenient mechanism to search for and retrieve the data of interest. Additional user defined functions can also be created and run in the database engine, and executed from the query interface in order to provide improved performance. The service interface subcomponent supports Grid and Web Service interfaces for remote access to analyses and for sharing of

experiments and methods through well-defined interfaces. Tools and viewers for browsing and viewing image data and analysis results are part of the application server component as well.

In Silico Experiments Repository. This is the central component for management of analysis results, which are generated through computer algorithms or by human experts. The repository is anchored on a data model that consists of generalized data objects, comprehensive data types, and flexible relationships between data objects. In an implementation of this repository, the data model should be designed to capture metadata about in silico experiments, semantic metadata about segmented and classified structures, and provenance information about analyses. The repository instance should be able to allow access to information via a wide range of queries on metadata, spatial structures and relationships, and semantic annotations and relationships drawn from one or more domain ontologies. The in silico experiments repository is the focus of this paper and will be described in greater detail next.

4 Repository for in Silico Microscopy Imaging Experiments

In this section we discuss the requirements and design of repositories for in silico imaging experiments. These repositories first and foremost should enable a research team to efficiently carry out imaging experiments. That is, they should allow for efficient exploration of analysis results. They should also provide support for archiving analyses an investigator wants to save, share, and reference in other studies as well as for agile rapid prototyping and algorithmic exploration.

4.1 Metadata

Rich metadata plays a crucial role in sharing, reusability, and reproducibility of in silico imaging experiments. Metadata should be able to precisely and unambiguously describe an in silico experiment and its components. One of the reasons that information derived from biomedical images is underused can be attributed to lack of efficient and flexible data models to support the modeling, managing, querying and sharing of analysis results and derived data. The Annotation and Image Markup (AIM) model is a caBIG® standard[7] developed to provide standardization for image annotation and markup for radiology images. However, microscopy and pathology images have their unique characteristics.

The immediate challenge is that the metadata model should be efficient to support large volumes of result sets. For instance, one of the ISBTRC experiments involving 213 whole-slide images has segmented and annotated more than 90 million nuclei. In addition, a single XML-based results document, which contained markups for all nuclei and the 23 features associated with each nucleus on a single slide, reached 7GB in size. Another challenge is the complexity of analysis results. The metadata about an in silico experiment can be semantically complex. The metadata model should be able to represent slide related image,

markup, feature, and annotation (e.g., classification of anatomic structures) information. This information includes a) context relating to patient data, specimen preparation, special stains, etc, b) human observations involving pathology classification and characteristics, and c) algorithm and human-described segmentations (markups), features, and annotations. Markups can be either geometric shapes or image masks; annotations can be calculations, observations, disease inferences or external annotations. The relationships between data elements can also be complex. For example, additional annotations can be derived from existing annotations. As a result, generic and extensible metadata models are required to support different types of experiments and applications.

The metadata model should also include a semantic description of the computation being carried out. At a minimum, the model should allow a user to express algorithm metadata, parameters, and semantic and concrete identification of input and output datasets. A more advanced model could support ontology-driven semantic descriptions of workflow templates and instances as well as concrete provenance information about an execution of a given workflow. Ontology-driven semantic representations provide a richer system of searching and reasoning about workflows. An example of semantic workflow systems is WINGS [8]. It provides a core ontology for generic components and data types to express workflows. This core ontology can be extended to support data types and data processing components in an application domain [13]. WINGS allows a user to describe an application workflow using semantic properties associated with workflow components and data types at a high-level of abstraction referred to as a workflow template. The workflow template and the semantic properties of the components and data types are expressed using the Web Ontology Language(OWL)[2].

4.2 Query Support

The repository should provide support for metadata based queries (e.g., count nuclei where their grades are less than 3), spatial queries (e.g., find density of nuclei where their grades are between 1 and 3 in selected region of interest), and semantic queries based on reasoning on spatial relationships and/or ontology relationships. The types of queries include: i) retrieval of image data and metadata to obtain data for analytical procedures, ii) queries to compare results generated from different approaches, and validate machine generated results against human observations; iii) spatial queries on assessing relative prevalence of features or classified objects, or assessing spatial coincidence of combinations of features or objects; and iv) queries to support selection of collections of segmented regions, features, objects for further machine learning or content based retrieval applications.

Many of the analytical imaging results are anatomic objects such as lesions, cells, nuclei, blood vessels, etc. Spatial relationships among these objects are often important to understanding the biomedical characteristics of biology systems. Common spatial relationships include containment, intersection or overlap,

[2] http://www.w3.org/TR/owl-ref

distance between objects, and adjacency relationships. Besides spatial relationships, another common requirement is to support calculation of coordinate and measurement information (such as computing the area, centroid, perimeter, minimal bounding box) of a markup object. The ISBRTC in silico experiments, for example, generate large volumes of results in the form of segmented regions, markups, annotations, and features. These data elements are stored, managed, and queried for algorithm validation as well as integration with genomics, clinical, and radiology data. Examples of the types of queries are: (1) Find the number of nuclei, which are classified by observer A and whose feature f is within the range of a and b; (2) Which nuclei types preserve nuclei features (distance, shape, etc) between two images; and (3) Which brain tumor nuclei classified by observer A and brain tumor nuclei classified by observer B exhibit spatial overlap in a given region of interest.

Annotations on objects may draw from one or more domain ontologies (e.g., cell ontology to describe different cell types, genome ontology to represent genomic characteristics), creating a semantically rich environment. The repository should allow for querying of data using semantic information. An example query from the ISBTRC studies is "Search for objects with an observation concept (astrocytoma), but also extend it to include all its subclass concepts (gliosarcoma and giant cell glioblastoma)." An important aspect of semantic information systems is the fact that additional assertions (i.e., annotations and classifications) can be inferred from initial assertions (also called explicit assertions) based on the ontology and the semantics of the ontology language. This facilitates a more comprehensive mechanism for exploration of experiment results in the context of domain knowledge. In some cases, it is desirable to extend an ontology with new concepts and properties. That is, a researcher may want to define and add new concepts and classes to the ontology using axioms and rules on existing classes and computable attributes, such as spatial relationships based on distance or relationships between computed features. This would allow incorporation of new knowledge to the system, and might result in new set of inferred annotations (assertions). Combined use of semantic stores/reasoners [11,22,5] and rule engines [10] can offer a repository system capable of evaluating spatial predicates and rules [19,15]. In such a system, the rule engine and the semantic store/inference engine interact to compute inferred assertions based on the ontology in the system, the set of rules, and the initial set of explicit assertions (annotations). Rules that utilize the spatial-temporal relationships might generate new instances of ontological concepts based on the evaluation of the rules. These instances are fed into the semantic inference engine to compute new assertions. The new assertions are input to the rule engine to compute new instances based on rules. This process continues iteratively until no more assertions/instances can be generated.

4.3 High-Performance Computing for Large Data Volumes

In order to scale to large volumes of data, the repository should take advantage of parallel computation power and I/O access. This can be achieved through data

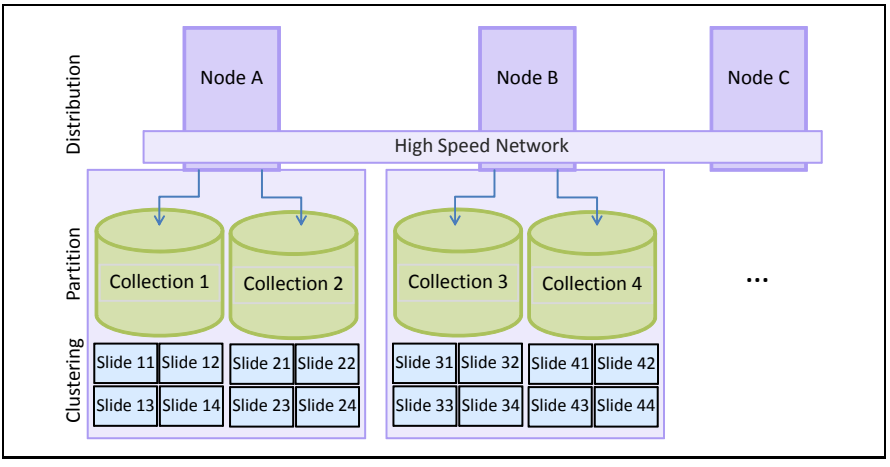

Fig. 3. High performance computing for managing large scale image metadata

distribution and partitioning techniques to take advantage of high performance computing resources (Figure 3) and cluster computing extensions in database management systems,

Data distribution and clustering to reduce I/O costs. Databases can be physically partitioned into multiple physical nodes on cluster based computing infrastructure, which consists of multiple physical servers, where each node has its own CPUs, disk controllers and disks (shared-nothing architecture). Physical database partitions across multiple nodes connected through high speed connections can scale quickly with the power of clusters. Multi-dimensional clustering, on the other hand, provides a method for automatic physically clustering of data along multiple dimensions on more than one key (or dimension) simultaneously. This reduces seek overheads when accessing the data along one or more dimensions. Database logical partitions reside on the same physical node can take advantage of symmetric multiprocessor (SMP) architecture. Having a partitioned database on a single machine with multiple logical nodes is also known as a shared-everything architecture, where the partitions use common memory, CPUs, disk controllers, and disks. Logical partitioned database can then take advantage of the parallelism support for both queries and I/O on a single SMP machine. In addition, *table partitioning* provides another way of dispersing data across multiple storage objects. For example, we can partition data in a table based on slide IDs, or range of dates. This can effectively constrain the search space to boost query performance.

Semantic Query Execution. With very large datasets, semantic query execution and on-the-fly computation of assertions may take too long on a single processor machine to be useful in exploration of datasets. Pre-computation of inferred assertions, also referred to as the materialization process, can reduce the execution of subsequent queries. Materialized assertions can be stored in the system and

optimization techniques including indexing can be utilized. However, the process of materialization may take very long for large datasets. Execution strategies leveraging high-performance parallel and distributed machines can reduce execution times and speed up the materialization process [14,19,15]. One possible strategy is to employ data parallelism by partitioning the Euclidean space in which the spatial objects are embedded. Another parallelization strategy is to partition the ontology axioms and rules, distributing the computation of axioms and rules to processors. This partitioning would enable processors to evaluate different axioms and rules in parallel. Inter-processor communication might be necessary to ensure correctness. This parallelization strategy attempts to leverage axiom-level parallelism. It will likely benefit applications where the ontology contains many axioms with few dependencies. A third possible strategy is to combine the first two strategies with task-parallelism. In this strategy, N copies of the semantic store engine and M copies of the rule engine are instantiated on the parallel machine. The system coordinates the exchange of information and the partitioning of workload between the semantic store engine instances and the rule engine instances. The numbers N and M will depend on the cost of the inference execution as well as the partitioning of the workload based on spatial domain and/or ontology axioms.

5 An Implementation of in Silico Imaging Experiments Repository

We have developed an implementation of the in silico experiments repository component (Figure 2) using relational database technology. The database schema is composed of a set of tables based on the Pathology Analytical Imaging Standards (PAIS) model [21]. We describe this implementation in this section.

5.1 PAIS Data Model

The PAIS model is designed to provide an object-oriented, extensible, semantically enabled data model to support pathology analytical imaging and human observations. PAIS provides highly generalized data objects, comprehensive data types, and flexible relationships between data objects. PAIS is also storage and performance efficiency oriented, and supports alternative implementations. Based on an object-oriented design, PAIS is easily extensible. The logical model of PAIS is designed in UML, and consists of 62 classes and interclass associations. The major components (main classes and relationships, not including attributes) are shown in Figure 4. These classes can be categorized as:

- Image reference information – the reference and metadata of the images. These include the ImageReference class with subclasses DICOMImageReference, and MicroscopyImageReference. The later has two subclasses WholeSlideImageReference and TMAImageReference. The Region class specifies which area (e.g., a tile) in the original image is used for the annotation.

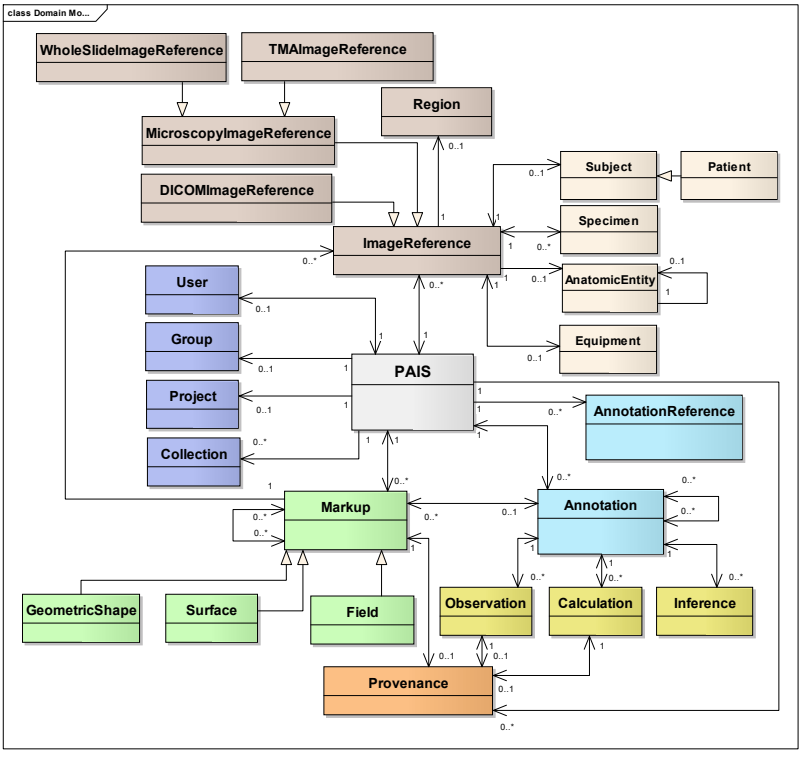

Fig. 4. Overview of PAIS model with a subset of major classes

- Image target information – who, where, and how the images are generated. These include Subject (such as Patient), Specimen, AnatomicEntity and Equipment classes.
- Organizational information – who performs the study and annotation and for what purpose. There are four classes: User, Group, Project and Collection. A collection is a group of items of the same type, gathered for display or study. For example, results from the same algorithm with different parameters on the same image belong to the same collection.
- Markup. Markup delineates a spatial region in the images and represents a set of values derived from the pixels in the images. Markup symbols are associated with one or multiple images, and can be in form of GeometricShape, Surface, or Field. Geometric shapes can be Point, Line, Polyline, Polygon, MultiPoint, MultiLine, MultiPolygon, Rectangle, Circle and Ellipse.
- Annotation. Annotation associates semantic meaning to markup entities through coded or free text terms that provide explanatory or descriptive information. There are three types of annotations: Observation, Calculation, and Inference. Observation holds information about interpretation of a markup or another annotation entity. Observations can be quantified based on different measure scales such as ordinal and nominal scales. Calculation stores

information about the quantitative results from mathematical or computational calculations, represented in CalculationResult, such as Scalar, Array, Histogram, and Matrix. Inference is used to maintain information about disease diagnosis derived by observing imaging studies and/or medical history.
– Algorithm provenance information. The Provenance class, as illustrated in Figure 4, helps to determine the derivation history of a markup or annotation, including algorithm information, parameters, and inputs. Such information is critical for validating approaches and comparing algorithms.

5.2 Results Documents and Data Loading Protocol

To enable convenient data sharing and exchanging, we use XML based representation of the PAIS model to represent result documents. The PAIS logical model is mapped and adjusted into an XML Schema. This schema is used by analysis workflows to generate compatible PAIS XML documents. To reduce the document size for processing, PAIS documents are often generated on partitioned regions such as tiles, and different PAIS document instances from different regions of the same image will share the same document UID. For example, we generate a couple of hundred tile based PAIS XML documents for a single whole slide image. These partitioned PAIS XML documents are further compressed into zipped files.

The XML representation of PAIS facilitates exchange and verification of documents in a standards-based manner. However, for very large result sets, it is not an efficient representation, even with compression of the documents. For exchanging and storing large static data (considered relative to the metadata that will be generated), self-describing structured container technologies (such as HDF5) could provide a more efficient alternative. Such container technologies provide more efficient storage than text-based file formats like XML, while still making available the structure of the data for query purposes. For instance, segmented regions and spatial data structures corresponding to multi-dimensional, multi-resolution data subsets can be stored in HDF5 files.

Data Submission and Staging. PAIS documents generated from image analysis applications can be either submitted to the database server directly by the application on the fly, or grouped for batch submission. PAIS documents are compressed and then submitted to the database server for data staging, where they are stored in a staging table. The database is populated by mapping each XML document into tables. The database internally loads the documents as XML typed column in the database. The temporary XML data enables efficient retrieval for mapping purpose. To map data from XML to the tables, we take advantage of the XMLTABLE function provided by XML databases, which queries the XML column, and generates table like representation of results. These values are then inserted into the tables. To make sure the data loading process works for large transactions, we keep track of the status of each XML document. Initially each document is assigned an "incomplete" status. If a document has been mapped successfully, the status of the document is changed to *complete*. At that point, the XML document can be removed from the database.

5.3 Relational Database Implementation

The database is currently implemented with IBM DB2 Enterprise Edition 9.7.1 with DB2 Spatial Extender, running on PowerEdge T410 (four quadcore CPUs, 16GB memory, and a 15K rpm hard drive) with CentOS 5.5. The database includes (1) *Image target and reference tables* for accessing specimen and image metadata information; (2) *Markup tables* implemented as one spatial table (using the DB2 spatial extensions) that stores geometric shapes in a spatial database, another table that manages association relationship between markups and annotations, and a third table for managing human markups, which are often at a different scale; (3) *Annotation tables* consisting of a table for scalar based calculation results, a table for quantified ordinal scale observations, and another table for quantified nominal scale observations; and (4) *Provenance tables* for managing metadata about algorithms, algorithm parameters, and input datasets.

To support queries on spatial relationships, we model and manage markup objects as spatial objects, supported by spatial databases. In the PAIS data management component, we support the following spatial data types: Point, Line, Polyline, Polygon, Rectangle, Circle, Ellipse, MultiPoint, Multiline, and Multipolygon. The most commonly used spatial type is polygon. These spatial data types are represented as vector graphics based format – SVG[3], so they can be represented as text format for convenient data exchange and visualization. We leverage the spatial extension of DB2 for efficient management and query of spatial information. The spatial table in our implementation is defined as a ST_Polygon spatial data type provided by IBM DB2. We also employ in queries several spatial functions implemented in DB2 such as spatial relationship functions (ST_Contains, ST_Touches, etc) and functions that return information about properties and dimensions of geometries (ST_Area, ST_Centroid, etc). Many of our spatial queries are different from traditional GIS queries. An initial study we have carried out shows that optimizations can be implemented to reduce query execution times. For example, the performance of spatial joins between two algorithms on the same image can be much improved by divide-and-conquer based approach. By dividing a region into four partitions, the cost of spatial overlap queries can be immediately reduced to less than half.

Our current implementation of the application server (see Section 3) offers a SQL interface and a caGrid data service interface [20]. We have developed a caGrid service layer on top of the database to enable data sharing. caGrid is a Grid middleware infrastructure with a service oriented architecture, where researchers can share both their data and analytical resources as grid services, and perform federated queries across distributed databases. We are also building a Google Map like image and metadata viewer, which can quickly zoom into different resolutions of images through identifying and retrieving tiled image portions at specific resolution.

For an initial evaluation of the implementation, we selected 18 slides, and loaded image analysis results from two different algorithm parameter sets and

[3] http://www.w3.org/Graphics/SVG

human annotated results. These generate around 18 million markups, and 400 million features. We are able to perform most queries efficiently – the current implementation does not support semantic queries. To support large scale, high performance data management, we plan to use IBM InfoSphere Warehouse Server to manage our data. InfoSphere Warehouse Server uses DB2 with database partitioning features which can effectively support Cluster based and SMP based computing infrastructures.

6 Related Work

Digital microscopy has become an increasingly important biomedical research tool as hardware instruments for rapid capture of high-resolution images from tissue samples have become more widely available. There are several projects that target creation and management of microscopy image databases and processing of microscopy images. The Virtual Microscope system [6] developed by our group provides support for storage, retrieval, and processing of very large microscopy images on high-performance systems. The Virtual Slidebox project [4] at the University of Iowa is a web-based portal of a database of digitized microscopy slides for education. The users can search for virtual slides and view them through the portal. The Open Microscopy Environment project [9] develops a database-driven system for analysis of biological images. The system consists of a relational database that stores image data and metadata. Images in the database can be processed using a series of modular programs. These programs are connected to the database; a module in the processing sequence reads its input data from the database and writes its output back to the database so that the next module in the sequence can work on it. OME provides a data model of common specification for storing details of microscope set-up and image acquisition. OpenCCDB [18,17] is a data model developed to ensure researchers can trace the provenance of data and understand the specimen preparation and imaging conditions that led to the data.

The Allen Reference Atlas (ARA) [1], which is funded by Paul Allen of Microsoft, has a high-resolution anatomical 3-D atlas of the mouse brain. It provides anatomical information for every voxel (at various resolutions of 100×100, 50×50 down to 25×25) in the 3D coronal atlas made up of 130 coronal mouse slices. The ARA provides a fixed vocabulary of regions names. The Bisque system [16] and associated tools like the Digital Notebook allow a biologist to capture the image experimental data and metadata and store these in a relational database. The eXtensible Imaging Platform (XIP) project is an open source framework for fostering medical imaging algorithm developments [3]. However, this platform is mainly designed and used for radiology image analysis. Additionally, this system lacks a systematic approach for building application workflows, high performance computation and management of image analysis results. DICOM WG 26 is developing a DICOM based standard for storing microscopy images [2], where headers will store metadata such as patient, study and equipment information. Tiles are managed as series and the mapping relationship is represented in an

XML format. However, the metadata is limited and could not be extended for analysis information, and DICOM itself is a storage and data exchange format and not suitable for queries.

7 Conclusion

Availability of an increasing array of high-throughput and high-resolution instruments has given rise to large datasets of omics data – such as genomics, proteomics, metabolomics – and imaging data – such as radiology and microscopy imaging. There are an increasing number of research projects that either primarily focus on in silico experiments or involve them as a significant component of their studies. Microscopy imaging holds tremendous potential for highly detailed in silico examination of morphology of biological systems. In this paper we argue that software for in silico imaging experiments will need to implement more comprehensive support than management, viewing, and annotation of slides. To fully realize the potential of in silico imaging studies, software will be required to support rich, semantic metadata models to represent complex analysis results, databases capable of supporting metadata, spatial, and semantic queries, and high-performance computing techniques for execution of expensive analysis operations and queries.

Acknowledgement. This research is supported in part by PHS Grant UL1RR025008 from the CTSA program, by R24HL085343 from the NHLBI, by Grant Number R01LM009239 from the NLM, by NCI Contract No. N01-CO-12400 and 94995NBS23 and HHSN261200800001E, by NSF CNS 0615155, 79077CBS10, and CNS-0403342, and P20 EB000591 by the BISTI program.

References

1. The allen reference atlas, `http://www.brain-map.org`, `http://mouse.brain-map.org/api/`
2. Dicom wg-26, `http://medical.nema.org/DICOM/minutes/WG-26/`
3. The extensible imaging platform project, `https://collab01a.scr.siemens.com/xipwiki/`
4. The virtual slidebox, `http://www.path.uiowa.edu/virtualslidebox/`
5. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A generic architecture for storing and querying rdf and rdf schema. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 54–68. Springer, Heidelberg (2002)
6. Çatalyürek, Ü.V., Beynon, M.D., Chang, C., Kurç, T.M., Sussman, A., Saltz, J.H.: The virtual microscope. IEEE Transactions on Information Technology in Biomedicine 7(4), 230–248 (2003)
7. Channin, D., Mongkolwat, P., Kleper, V., Sepukar, K., Rubin, D.: The caBIG Annotation and Image Markup Project. Journal of Digital Imaging (2009)
8. Gil, Y., Ratnakar, V., Deelman, E., Mehta, G., Kim, J.: Wings for pegasus: Creating large-scale scientific applications using semantic representations of computational workflows. In: AAAI, pp. 1767–1774. AAAI Press, Menlo Park (2007)

9. Goldberg, I., Allan, C., Burel, J.M., Creager, D., Falconi, A., Hochheiser, H., Johnston, J., Mellen, J., Sorger, P., Swedlow, J.: The open microscopy environment (ome) data model and xml file: Open tools for informatics and quantitative analysis in biological imaging. Genome Biol. 6(R47) (2005)
10. Hill, E.F.: Jess in Action: Java Rule-Based Systems. Manning Publications Co, Greenwich (2003)
11. Kiryakov, A., Ognyanov, D., Manov, D.: Owlim - a pragmatic semantic repository for owl. In: WISE Workshops, pp. 182–192 (2005)
12. Kumar, V.S., Rutt, B., Kurç, T.M., Catalyurek, U.V., Pan, T.C., Chow, S., Lamont, S., Martone, M., Saltz, J.H.: Large-scale biomedical image analysis in grid environments. IEEE Transactions on Information Technology in Biomedicine 12(2), 154–161 (2008)
13. Kumar, V.S., Kurç, T.M., Ratnakar, V., Kim, J., Mehta, G., Vahi, K., Nelson, Y., Sadayappan, P., Deelman, E., Gil, Y., Hall, M., Saltz, J.H.: Parameterized specification, configuration and execution of data-intensive scientific workflows. Cluster Computing (April 2010)
14. Kumar, V.S., Narayanan, S., Kurç, T.M., Kong, J., Gurcan, M.N., Saltz, J.H.: Analysis and semantic querying in large biomedical image datasets. IEEE Computer 41(4), 52–59 (2008)
15. Kurç, T.M., Hastings, S., Kumar, V.S., Langella, S., Sharma, A., Pan, T., Oster, S., Ervin, D., Permar, J., Narayanan, S., Gil, Y., Deelman, E., Hall, M.W., Saltz, J.H.: Hpc and grid computing for integrative biomedical research. IJHPCA 23(3), 252–264 (2009)
16. Kvilekval, K., Fedorov, D., Obara, B., Singh, A., Manjunath, B.S.: Bisque: A platform for bioimage analysis and management. Bioinformatics 26(4), 544–552 (2010)
17. Martone, M.E., Tran, J., Wong, W.W., Sargis, J., Fong, L., Larson, S., Lamont, S.P., Gupta, A., Ellisman, M.H.: The cell centered database project: An update on building community resources for managing and sharing 3d imaging data. Journal of Structural Biology 161(3), 220–231 (2008)
18. Martone, M.E., Zhang, S., Gupta, A., Qian, X., He, H., Price, D.L., Wong, M., Santini, S., Ellisman, M.H.: The cell-centered database: a database for multiscale structural and protein localization data from light and electron microscopy. Neuroinformatics 1(4), 379–395 (2003)
19. Narayanan, S.: Efficient Virtualization of Scientific Data. PhD thesis, Ohio State University, Columbus, OH (2008)
20. Oster, S., Langella, S., Hastings, S.L., Ervin, D.W., Madduri, R., Phillips, J., Kurç, T.M., Siebenlist, F., Covitz, P.A., Shanbhag, K., Foster, I., Saltz, J.H.: cagrid 1.0: An enterprise grid infrastructure for biomedical research. Journal of the American Medical Informatics Association, 138–149 (December 2007)
21. Wang, F., Pan, T., Kurç, T., Sharma, A., Saltz, J.H., Chen, W., Chu, V., Hu, J., Yang, L., Foran, a.D.J.: Unified modeling of image annotation and markup. In: APIII: Advancing Practice, Instruction & Innovation Through Informatics, Pittsburgh, PA (September 2009)
22. Zhou, J., Ma, L., Liu, Q., Zhang, L., Yu, Y., Pan, Y.: Minerva: A scalable owl ontology storage and inference system. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) ASWC 2006. LNCS, vol. 4185, pp. 429–443. Springer, Heidelberg (2006)