SIEMENS

Algorithm Validation Toolkit
AVT2EXT

Algorithm Validation Toolkit
AVT2EXT

Improvement Plan
Revision 1.0

Last Change: May 14, 2010

Robert W. Schwanke
Fabian Moerchen

History

Document History

Version/ Status	Date of Issue	Author	Change and Reason of Change
0.1 Draft		Robert W. Schwanke	Final draft prior to AVT2EXT release
1.0 Released	14-May-2010	Robert W. Schwanke	Released

History of released Versions

Version	Release date	Product Version

Table of Contents

1. History 2
2. Table of Contents 3
3. Introduction 4
4. Additional use case modeling 4
   4.1. Management of AVT experiments 4
   4.2. Data curation 4
5. Features 4
   5.1. Collections 5
   5.2. Persisting results 5
   5.3. AIM Templates 5
   5.4. Worklists 5
   5.5. Outlier Identification and Highlighting 5
   5.6. Comparing Markups in Image Reader 5
   5.7. Annotation of Annotation 5
   5.8. Multiple Image Observations Comparison 6
6. Architectural challenges 6
   6.1. Annotation database 6
   6.2. Image management 6
   6.3. Series size 6
   6.4. Image Reader launch latency 6
   6.5. XIP Host™ interface to Annotation Database 6
   6.6. AVT services 7
      6.6.1. Web-based AVT 7
7. Performance Engineering 7
8. Incremental, iterative development 7

1. Introduction

This report lists potential improvements and additional features for AVT based on the experience with and feedback on the current version of AVT. The discussed topics should be considered for inclusion in the requirements for the next development cycle.

Analytical tasks in AVT revolve around images and image annotations, using the AVT toolset. We will refer to such tasks as *AVT experiments*.

> **Note: This memo will be revised and re-issued when the CDRH evaluation team has completed its assessment and provided its feedback.**

2. Additional use case modeling

The AVT project has previously generated two use-case models:

- an informal use-case model developed in AVT Phase II, broadly covering the whole topic of measurement variability analysis with approximately 100 use-case categories and sub-categories, and
- a detailed use-case model developed in AVT2EXT, defining the functionality that was implemented.

The following use-case categories were deferred from the development in AVT2EXT and should be considered in the next cycle of analysis and development.

2.1. Management of AVT experiments

In order to document an experiment and make it reproducible, the AVT toolkit needs to capture information from the complete life cycle of an experiment such as selection of images and annotations used, algorithm and software version used, independent variables used, statistical results generated, etc. This use case needs to address topics like:

- Defining an AVT experiment
- (Re-)running an AVT experiment
- Comparing results from AVT experiments
- Sharing an AVT experiment

2.2. Data curation

Data curation is the cluster of activities that select, collect, organize, and harmonize the data that is being analyzed. Curation is needed to prepare data to be used in AVT experiments and to manage data generated by AVT experiments. This use case needs to address topics like:

- Raw vs. derived data
- Local vs. caGrid data storage
- Browsing support
- Dataset-specific AIM templates
- Tagging anomalous data points (for conditional exclusion)

3. Features

During the development of AVT2 and AVT2EXT many additional useful features were suggested by testers and the development team. The main features are discussed below in no particular order. Additional feature ideas can be found in the Feature Request Tracker and the Bug Tracker of the SCR public GForge site for AVT.

3.1. Collections

AVT should offer the user the capability to model several collections of DICOM series, AIM annotations, and possibly additional information not stored in the DICOM and AIM object such as (de-identified) clinical data associated with the patient, subject, study, series, course of the disease, etc. This feature would directly .support the use cases 'Management of AVT experiments' and 'Data curation'.

3.2. Persisting results

Currently the final result of an AVT experiment is displayed on the screen and can only be exported as figures etc. AVT should offer the user the ability to store results of an experiment for later resuse. This feature would directly support the use cases 'Management of AVT experiments' and 'Data curation'.

3.3. AIM Templates

AIM is a very general representation for image annotations. The use of AIM for a specific use case often requires additional restrictions on the types of information or allowed vocabulary each AIM object may contain or be associated with. Currently such specific instances of AIM are directly build into the AVT applications such as the Image Reader and the query interface in XIP Host. In order to support many different use cases in the future a format for AIM templates needs to be developed in coordination with the AIM development team and the AVT applications should be made capable of understanding such templates to dynamically adjust the user interfaces. For the Image Reader this concept was demonstrated in TCGA phase 0, in a limited way.

3.4. Worklists

One piece of an experimental design is the allocation of reading tasks to readers. Storing each reader's assignments as a worklist would be a convenient way to reduce bookkeeping errors that otherwise would arise from manual task assignment methods. The XIP Host™ currently has a rudimentary worklist facility which may be a useful starting point for what AVT needs. Using worklists also suggests the addition of a workflow manager, likely as a separate service.

3.5. Outlier Identification and Highlighting

AVT outlier analysis should be enhanced to support additional statistical methods for outlier detection. The outliers should be highlighted in the results table to help identify correlations among error measures.

3.6. Comparing Markups in Image Reader

The image reader should facilitate comparing different markups of the same tumor, either by displaying them simultaneously in different colors or by rapidly cycling between them (in different colors).

3.7. Annotation of Annotation

The image reader should be able to load an existing annotation, edit it, and save it as a new annotation. This is needed, for example, for creating nominal ground truth annotations and for adding experiment-specific information to imported annotations.

3.8. Multiple Tumors per Series

The image reader should be enhanced to support the annotation of multiple tumors in one series in the same session. MVT should be enhanced to perform statistical analysis on annotations of different tumors in the same series.

4. Architectural challenges

In addition to features that directly affect the interaction of the user with the application, additional topics concerning the architecture of AVT were identified. The following topics each have unique issues, but there are also many interactions between them, which will require comprehensive, holistic analysis.

4.1. Annotation database

Even among the limited set of applications of AVT it became clear, that the systems needs to be capable of scaling up to large collections of images and annotation. For example, the CDRH Phantom collection has 1000 series. Each series contains about 5 tumors. Each tumor may be read by 6 readers, + some number of algorithms. Each reader may read each tumor twice, by each of several methods (e.g. RECIST, WHO, volumetric contours). So even for this limited set of images, 100k annotations could be generated easily. The AD storing the annotations will need to be optimized to support efficient queries against such datasets.

4.2. Image management

The storage and transfer of images from AD through XIP Host™ to the AVT applications need to be optimized to minimize copying and include data caching, memory management, and smart trade-offs between storing objects in files and in blobs.

4.3. Series size

The larger series in the CDRH data sets are straining the capacities of various libraries used in AVT, as well as the available physical memory of mid-range laptops. Investigations are needed to more efficiently stream only parts of the data to the AVT tools and use shared memory among processed running on the same computer.

4.4. Image Reader launch latency

Since the Image Reader will be used by radiologists not otherwise involved in the AVT-based experiments, launch time should be good enough not to frustrate the user unduly. What is the *de facto* standard for acceptable launch time on an image reader, and how much can we exceed that for an experimental image reader?

4.5. XIP Host™ interface to Annotation Database

The XIP Host™ has a clearly defined interaction with the AVT applications (IA, AE, and MVT) through the DICOM Supplement 118 Application Hosting standard. The interface to AD, however, is currently directly compiled into the XIP Host™ rather than connected by a plugin interface. The XIP Host™ should define and implement such an interface to query data sources. In particular, this interface likely needs:

- Database query, retrieve, and store functions with generic DICOM and AIM query capabilities and custom extension capabilities,
- Plug-in interface for application-specific query UIs,

- Opportunistic data loading, based on a combination of worklist anticipation, caching, and pre-fetch hints coming through the application hosting interface.

4.6. AVT services

In order to integrate AVT (or parts thereof) as services into caGRID, the code needs to be refactored to provide de-coupled services that can operate independently on different machines. The service oriented architecture of AVT should include

- An AD service that can store and query DICOM and AIM objects based on existing caGRID service implementations such as AIME but that also offers additional functionality to query DICOM and AIM objects conjunctively as needed for AVT, as well as storage, query, and retrieval of additional AVT data such as experiments and results,
- An AE service that can execute image annotation algorithms on selected dataset storing the results back to an AD service,
- An MVT service that can performs statistical analysis on images and annotations storing the results back to an AD service.

In order to ensure sufficient performance, the architecture may need to support combined services running on the same node to take advantage of collocation of image and annotation databases. In order to utilize large scale computing resources, support for clusters (for example, Hadoop), high performance computing (HPC) environments, and/or cloud deployment should be investigated.

4.6.1. Web-based AVT

Currently AVT is a standalone application with connections to remote data sources through caGRID. In order to reduce the footprint, web-based access to an AVT server should be investigated, as well as whether or not such a server would be based on the proposed web-based XIP Host™.

5. Performance Engineering

Although several of the other topics in this report suggest performance implications, performance engineering entails a pragmatic analysis of the actual performance characteristics and, in particular, bottlenecks, of the entire running system considered as a whole.

6. Incremental, iterative development

The sequence of increments for developing AVT should be organized according to supported experiments. The first three increments have been organized around MICCAI'08 Grand Challenge, TCGA/Vasari, and CDRH Thoracic Pilot and Pivotal studies. Candidates for subsequent increments could be based on the collections that are being added to the NBIA. *Potential* requirements should be solicited from any and all interested parties, but a relatively small Change Control Board should *qualify* and *prioritize* the requirements.