MEDOX

An XML-based Approach of Medical Data Organization for Segmentation and Simulation

Eduard Fried[1], Yan Geng[1], Sebastian Ullrich[2], Daniel Kneer[2], Oliver Grottke[3], Rolf Rossaint[3], Thomas M. Deserno[1], Torsten Kuhlen[2]

[1]Department of Medical Informatics, RWTH Aachen University
[2]VR Group, RWTH Aachen University
[3]Department of Anaesthesia, University Hospital Aachen
efried@mi.rwth-aachen.de

Abstract. Applications in simulation or visualization often rely on data from multiple sources (e.g., MRI scans, extracted anatomical objects in a 3D representation). However, structured data organization is either non-existent (e.g., simply stored in folders), proprietary or specific (e.g., DICOM archives). In this paper, a novel extensible approach to organize and access arbitrary medical data is proposed, called Medical Data Organization with XML (MEDOX). The concept is based on a lightweight XML database, complementary to recent MedX3D standard developments. Data files are stored in a file system and referenced along with additional meta data in an XML file. Basic versioning of data objects is also supported. A simple C/C++ library was implemented to abstract the access patterns to the database and to avoid the error-prone manual manipulation of the XML file. As a proof of concept, it has been applied to regional anesthesia simulation in virtual environments.

1 Introduction

In a project where subject specific datasets are used for segmentation, simulation and visualization, the data has to be organized in an efficient way. Usually, it is sufficient to work with the newest data objects, yet sometimes access to all intermediate files in various versions is required. The organization of data objects has to support the user in finding and selecting the required data.

The simple storage of data files in the file system without additional meta information is error-prone and also complicates data exchange as it requires explicit user interaction and knowledge of the locations of all files.

The storage of all data in a relational database (e.g., SQL-based [1]) hides the storage details and requires additional tools to access the data. A human readable representation of the data is the X3D format [2], which is based on the XML standard. This data format supports the organization of a 3D scene, including the definition of the 3D-objects itself. Additional data, like DICOM files, is not directly supported. There is a working group, MedX3D [3], that is developing new standards for medical usage. However, the work is still in

early drafts and mainly focuses on volume rendering. The Foundational Model of Anatomy (FMA [4]) defines an ontology for anatomical structures, but it provides no means to organize or reference data files. Working with versioned files is a core functionality of tools like Subversion [5]. They offer an efficient storage model (by saving differences between versions) and also provide operations for comparison and merging. The primary requirement in this case would be the version history of data and configuration files, though.

For a simple and efficient usage of the available data, the database has to meet certain criteria. As every data object results from a processing (or imaging) operation, it should be linked to this process in a unique way. Consequently, the processing model should be represented in the data organization scheme. Data access should not distract the user from working with the files. Ideally, the access operations can be directly integrated into the processing tools.

2 Materials and Methods

In this paper, the authors focus on the usage of subject specific datasets, which initially consist of imaging data and also contain data from subsequent processing steps. This data originates from medical imaging, segmentation algorithms etc. and it can be used for visualization, simulation or analysis purposes(Fig. 1). The data for each subject is structured by a database file based on a novel XML-specification described in the following paragraphs.

The basic structure of the XML database contains meta data about the patient, the data directory and a set of data nodes (Fig. 2). The data directory represents the application specific data organisation and its dependencies (Fig. 2). To allow maximum flexibility, it is stored as a hierarchical tree structure with `<element>` tags to declare data objects and `<structure>` tags to group elements and other structures. The element tag uses the `id` attribute to link the declaration to the data object definition and the `name` and `description` attributes to give the object an application specific name.

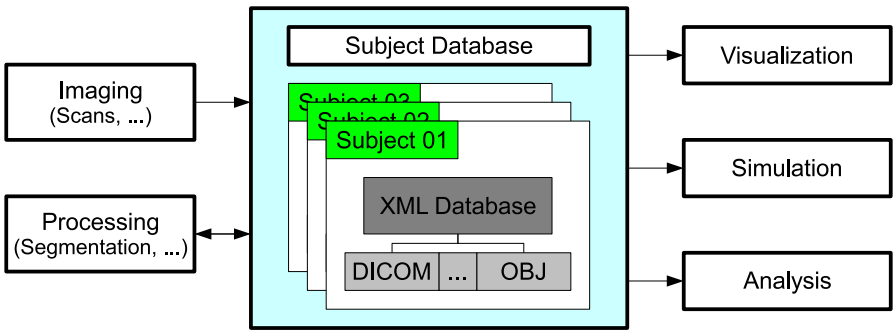

Fig. 1. Database architecture overview.

Inside the data node, there is a further distinction between image data and 3D-object data. The images can be raw data (e.g., medical scans) or processed images (e.g., denoising, segmentation). It is left to the user to extend the definition of the `<processed>` node with application specific processing parameters. As imaging data covers only a specific body region, this information is used to separate the data nodes.

The 3D-objects are referenced inside a set of segmentation nodes. As these objects result from processed images, a modified image will yield different 3D-objects. This process can be considered as *branching* and its resulting objects can be stored in a different segmentation node. Examples for branches could be a manual Ground Truth segmentation or automatic segmentations with different

```
<subject_database>
  <meta_data>... </meta_data>
  <data_directory> ... <data_directory>
  <data region="inguinal">
    <images>
      <raw id="" type="Angiography"> ... </raw>
      <processed id=""> ... </processed>
    </images>
    <objects>
      <segmentation id=""> ... </segmentation>
      <segmentation id=""> ... </segmentation>
    </objects>
  </data>
  <data region="spinal"> ... </data>
</subject_database>
```

Fig. 2. Basic XML database structure. All data definition nodes use an `id` attribute that is referenced in the data dictionary.

```
<data_directory>
  <structure type="">
    <structure type="">
      <element id="" mode="" name="" description=""> ... </element>
    </structure>
    <element id="" mode="" name=""  description=""> ... </element>
  </structure>
</data_directory>
```

Fig. 3. Generic data directory structure. `type` and `name` attributes of structures and elements are application-specific. The `id` attribute creates a bijective mapping to the actual data object definition later in the document. The `mode` attribute specifies the available representations of a data object, e.g., for visualization and simulation. The `description` attribute stores a string that can be presented to the user, while the `name` attribute is used internally as a part of the file name of that object.

algorithms. A mere correction or update of an image also results in modified 3D-objects, which can now be stored with a new version number (Fig. 2). In order to organize changes of data, the authors propose a versioning of the XML file with a tool like Subversion. Thus, a technical user can benefit from the human readable format of the file while it does not grow with changes and still allows unrestricted access to its history.

3 Results

As a proof of concept, the approach was applied in a project for simulation of regional anaethesia. Datasets consist of imaging data (DICOM series), 3D polygonal data (Wavefront OBJ files), other data (e.g., spline-based nerve cords hierarchy) and meta-information files (XML). The database is accessed by segmentation tools [6] and a VR-based simulator [7]. As a standardized naming convention for anatomical objects in the data directory the authors use entries from the FMA ontology.

Currently, the database holds five patient datasets, each with two original MR images (morphology, angiography) and about 50 anatomical structures (bones, muscles, etc.) as segmented image data and extracted 3D-objects.

The authors implemented a C/C++ library which abstracts the access patterns to the database and can be used with our segmentation, simulation and visualization tools to access data from within the application.

4 Discussion

A novel approach for medical data organization, called MEDOX, was described. The data objects are stored as files in the file system and referenced via an XML database file. This file contains a data directory that can be used to describe the application specific data dependencies by an application to interactively determine the available data objects without parsing the complete database. The XML file can easily be processed by established tools and APIs and integrated

```
<segmentation id="">
  <object id="" ref_id="" version="">
    <filename type="obj"> file_name_version.ext </filename>
    <properties> ... </properties>
  </object>
  <object id="" ref_id="" version=""> ... </object>
</segmentation>
```

Fig. 4. Structure of a segmentation node. The `id`s of the contained objects correspond with the `id`s in the data directory. `ref_id` corresponds with the `id` attribute of the image node the 3D-object is extracted from. The `version` number of an object is added as a suffix to the object's filename.

into own processing tools. As XML is an open standard, the user does not depend on proprietary third party tools. He can also add own extensions to the XML structure without rendering existing implementations useless.

The versioning of data objects is handled in two ways. The XML database that references the data files is versioned via SVN. The data files itself are not versioned. Storing the version history of a data file in the XML database would unnecessarily increase the database size and slow down its parsing. Instead, the versioned files are saved as copies with the according version number as a suffix in its filenames. Every version of the XML database references the most recent data file versions at that time.

Further development can concentrate on synchronisation of databases. The global database can be exported into a local copy (for performance reasons) that contain only a subset of the data objects. The local XML database is adjusted accordingly to reference only the available files. It is planned to support the import of new data from a local copy into the global database.

Additionally, new emerging standards like X3D and its MedX3D extension are observed and will be adapted into the approach once their drafts are finalized and getting certified.

Acknowledgement. This work was developed under the auspices of the German Research Foundation (DFG, RO 2000/7-1, KU 1132/4-1, LE 1108/8-1).

References

1. ISO. ISO/IEC 9075-1:2008: Information technology – Database languages – SQL – Part 1: Framework (SQL/Framework). Nederlands Normalisatie Institut: pub-ISO; 2008.
2. Daly L, Brutzman D. X3D: extensible 3D graphics standard. In: Proc SIGGRAPH. New York, NY, USA: ACM; 2008. p. 1–6.
3. John NW, Aratow M, Couch J, et al. MedX3D: standards enabled desktop medical 3D. Stud Health Technol Inform. 2008;132:189–94.
4. Rosse C, Mejino JLVJ. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. J Biomed Inform. 2003;36(6):478–500.
5. Subversion. Subversion; 2006. http://subversion.tigris.org.
6. Teich C, Liao W, Ullrich S, et al. MITK-based segmentation of co-registered MRI for subject-related regional anaesthesia simulation. Proc SPIE. 2008;6918:2M–1–10.
7. Grottke O, Ntouba A, Ullrich S, et al. Virtual reality-based simulator for training in regional anaesthesia. Br J Anaesth. 2009;103(4):594–600.