Project Semantic Web for Pathology
Report AP1

FU/NBI, Charite

March 15, 2004

Abstract

This report presents our achievements during the first work package of the project "Semantic Web in the Pathology", whose principal aim was to prepare the available and future project data for further project phases. At the end of AP1 the data consisting of text-based medical reports and digital images should be formalized in XML in order to serve as input for text processing tasks (AP3) or to be integrated (as a form of knowledge) in the knowledge base to be constructed in AP2.

1 Introduction: Main goals of AP1

The AP1 intends to offer a basic restructuralization of the existent data set at the Institute of Pathology, Charité. The data involved in the project consists of available and future materials, both medical reports, pictures and their annotations. The goal of AP1 is to guarantee a uniform formalization for the complete data stock, which can be subsequently used by the future components. In particular, XML-encoded medical reports and graphical material should be integrated in the Semantic Web knowledge base that is to be realized in AP2. Our efforts were concentrated therefore in two main directions: clearing up the actual pathology archive at the Charité followed by the XML transformation and the implementation of a tool for the direct generation of XML-based medical reports, respectively. Another central part of AP1 is the development of the XML schemes for text and images, which relate to medicine standards like HL7/CDA and DICOM.

This report gives an account of our work for the realization of these goals. Section **??** presents the current status of the data archives at the Charité and the steps we took for cleaning it up. The XML schemes are subject of Section **??** and **??**. The last section gives concrete insights about the technical infrastructure of the projects and the realization of the XML editing tool.

2 Available data

As emphasized in the introduction the data archives we have to deal with are medical reports and digital images related to them. Pathologists analyze histological slides by means of a microscope and subsume their observations in medical reports. These are in turn dictated in a dictation machine by the pathologist and subsequently typed in by typists in order to be stored in a ASCII database. The medical reports archive contains aproximately 15000 reports pro year. Each case corresponds to one or more medical reports, which in turn are associated to several slides (1 to 10), or digital images (for the Digital Virtual Microscope). Since our project focuses on lung pathology, we have to select from the complete archive the cases related to our application area (for details about these measurements, see **??**).

Digital images can be analyzed by means of the DVM, which offers a Web frontend for obervations and textual descriptions of slides. Every scanned slide has a aproximate size of 5GB, because of the high resolution requirements, which means up to 50 GB material for every case.

3 XML representation of medical reports

XML is known as a efficient and well structured method for storing data. The processing of XML documents is platform and software independent. However, for an frictionless data transfer a strict definition of the XML strukture is required. One goal of AP1 was a spezification for a interchangeable format for our pathological reports. Due to this goal, we focused on international standards such as Health Level Seven (HL7) and the Clinical Document Architecture (CDA).

HL7 is one of several ANSI-accredited Standards Developing Organizations (SDOs) operating in the healthcare arena. Its domain is clinical and administrative data. HL7 have several new and ongoing initiatives. The cornerstone of the HL7 Version 3 development process is the Reference Information Model (RIM). An other initiative is the Clinical Document Architecture. In September 2000, the HL7 membership ratified Version 1 of the CDA. The CDA defines an XML architecture for exchange of clinical documents. The encoding includes the specification and its semantics of the HL7 RIM and HL7 registered coded vocabularies. CDA Release 1 Level 1 is accredited by the ANSI as architecture for clinical documents.

In cooperation with the german HL7 usergroup, the Kassenrztlichen Bundesvereinigung (KBV), the Zentralinstituts fr die Kassenrztliche Versorgung (ZI), the Verein patienten-orientierter Informations- und Kommunikationssysteme (VHK), the Verband deutscher Arztpraxis-Softwarehersteller (VDAP), provider of clinical information systems and the Universities Gieen und Kln,

in 2000 the project SCIPHOX (Standardized Communication between Information Systems in Physician Offices and Hospitals using XML) was founded. The main goal of SCIPHOX is to deliver a spezification of an communication standard based on international standards such as XML. In doing so, the developments of the standards HL7 and xDT are to be integrated. As a main component, SCIPHOX use the CDA. For our pathological reports, we decided to use the CDA format. As document type definition we use the XML schemata from the SCIPHOX project.

3.1 Description of the XML structure

A CDA Release 1 Level 1 document consists of two main parts: the clinical document header and the body.

3.1.1 Clinical document header

The clinical document header contains all relevant information about the pathological case. Four header parts specify informations about:

- the document itself,
- the event (the pathological investigation),
- the actors of an procedure (the pathologist) and
- the acceptor of the procedure (the patient).

3.1.2 Body

The document body contains the text of the pathological report. In Level 1 of the CDA, the content is either none-XML or XML, organized in section tags. Each section can be labeled with a caption tag and can contain several paragraphs. The parts of the report named "Makroskopie", "Schnellschnitt", "Mikroskopie", "Kritischer Bericht" and "Kommentar" are stored in these sections. For the report text no further markup is used. For an example of an XML report see attachment A.

3.1.3 XML representation of the dictation path

The Digital Virtual Microscope (DVM), developed in the working group "Digitale Pathologie und EDV", is a web based client-server tool, allowing the examination of high resolution histological slides previously scanned from a conventional glass slide.
While dictating the pathological report, the examinator pass through the slide in an spezific path, called the examinaton path. This path is stored in the DVM. The sequence of the pathological report text itself is the dictation path.
The association between these two pathes is the diagnostic path, allowing the connection between histological image information and pathological description

in the report.
While the examination path is stored in the database of the DVM, the dictation path have to be represented in the XML stucture of the pathological report.
For this approach, the predefined CDA element "coded_entry" is used for each parameter of the dictation path.
The parameters are:

- UID, the unique identificator for a slide,
- PID, the unique identificator for the examinator,
- StartTime, time the dictation of a text passage beginns and
- StopTime, time the dictation of a text passege stops.

4 XML representation of digital images

5 Technical realization

5.1 Storage and formalization of the medical reports

Since the relevant thematic area of the project is lung pathology, a first step for the data preparation was the identification of the lung cases from the original ASCII data archive. First we transformed 12900 cases to store them in an SQL data base an building an frontend which offers a much more confortable user interface for data and query management compared to the original storage system. By executing a full text search by "lung" in this data base, we obtained about 400 results. Due to the ambiguity of the full text search we had to manually overview these cases in order to check their affiliation to lung pathology and selected finally about 320 content relevant hits. The remaining ones have not been eliminated from the data set, but the positives have been labeled as "lung case".

In a final step the complete data amount was tranformed in XML, according to the developed scheme (see 3).

5.2 Managing digital images

5.3 An XML editor for medical reports

The generation of medical reports in XML is assisted by a Java-based tool, which allows basic editing capabilities of the four major parts of a medical report. An important requirement for this tool is the coordination to the functionality and mode of operation of the Digital Virtual Miscroscope. The underlying scenario presumes that the pathologist analyzes virtual slides with the DVM and in the same time enters his obervations and conclusions into the system. The editor generates an XML document for every case and stores it in a Xindice XML database. Therefore the editor should not only record the textual input of the

user, but also additional events, which are relevant for search purposes. The correlation of analyzed slide segments and the corresponding input text plays a central role in this issue. Concretely the tool should register to each designated text fragment the time and the slide fragment the pathologist is supposed to be currently analyzing. The different types of information are part of the XML scheme and therefore included in every XML document.For this purpose we followed two directions for the development of XML tool: as extension of the DVM and as a standalone application.

5.3.1 Standalone application

In the standalone approach the Java-based tool should exchange data with the virtual microscope in order to gather picture-related informations. The editor gets information about the picture the pathologist is analyzing while typing microscopy report. The interface between editing and vizualization tool is not implemented at present. The editor connects to Xindice, requests the desired medical report and displays its content. New reports can be generated in this manner, editing existing reports is not supported yet, but an extension of the tool in this direction is straight forward. There is a support for the so-called "diagnostical path", storing information about the time interval, the picture and the fragment of the picture, the pathologist was observing while editing a part of the report. Therefore for every time interval, which is designated by "start" and "stop" buttons on the GUI, the start-time, the stop-time the person-id (of the report author), the picture id (for every case there are approximately ten scans of tissue) and the coordinates of the slide fragment on the screen are registered additionally to the textual input. The report can be stored in the database or locally as XML at the end of the session. A detailed description of the implementation is presented in Appendix A.

5.3.2 DVM extension

The second approach integrates the new functionality in the virtual microscope. In this case the correlation of the typed text to the digital slides and the corresponding information is immediate (see Figure **??**). For each of the four parts of a medical report the interface offers a text area with two buttons for the temporal delimitation. In contrast to the first approach, the typed information is stored in the DVM Miscrosoft SQL Server database in a path-oriented manner. For every path the pathologist follows in his slide observations (the paths are defined as a sequence of coordinates and time stamps) the corresponding text is stored separately in the database. The generation of the medical reports apart from the path perspective is made on the client-side by an additional application, which parses the ASCII/HTML ASCII text, transforms it in XML and stores it appropriately. The integration and development of the editing feature has been realized in collaboration with the developers in project "Meducase", authors of the digital virtual microscope.

As a conclusion, both variantes support the generation of new medical reports and their formalization and storage in XML format. Future steps in this direction might include editing capabilities, assiciated automatically with an appropriate access rights policy and special considerations about user interface design or the integration of voice recognition software. These ideas go beyond the Semantic Web focus of the project, but are a possible subject for the future.

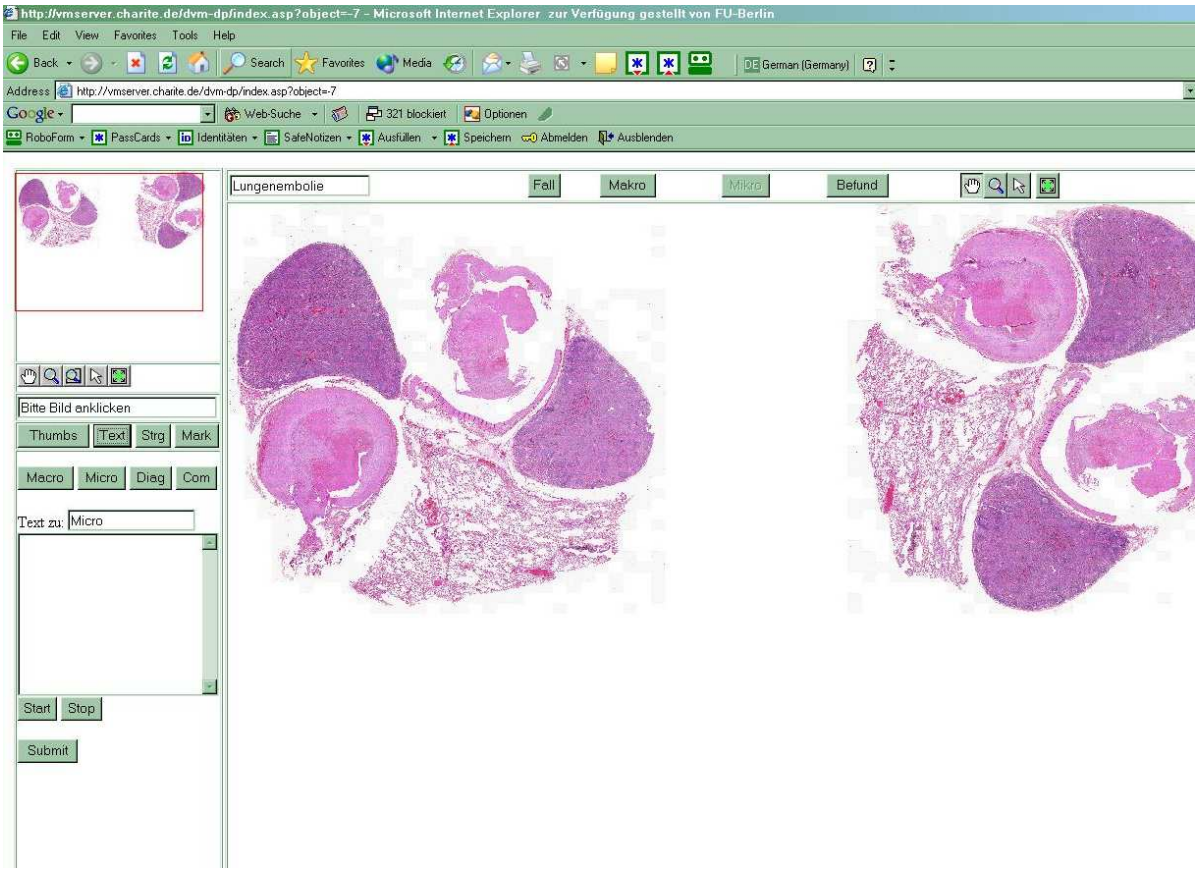

Figure 1: **E**xtension of the digital microscope

A CDA Release 1 Level 1 Document

A.1 Prolog

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<levelone xmlns="urn::hl7-org/cda"
  xmlns:sciphox="urn::sciphox-org/sciphox"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn::hl7-org/cda sciphox-cda.xsd"
  xmlns:swpatho="urn::swpatho-org">
```

A.2 Clinical Document Header

```
 <clinical_document_header>
 <!=====================================================
       information about the document
  ====================================================-->
  <!-- unique identifier of the document -->
  <id EX="154182"/>
  <!-- document type -->
  <document_type_cd V="11529-5" S="2.16.840.1.113883.6.1" DN="Pathologischer Bericht"/>
  <!-- date of origination -->
  <origination_dttm V="9.4.2001 00:00:00"/>
 <!=====================================================
       information about the event
  ====================================================-->
  <patient_encounter>
   <!-- unique case identifier -->
   <id EX="133470"/>
   <!-- threating institution -->
   <practice_setting_cd V="HOSP" S="2.16.840.1.113883.5.10588" DN="Pathologie"/>
   <!-- date of procedure -->
   <encounter_tmr V="6.4.2001 00:00:00" />
  </patient_encounter>
 <!=====================================================
       information about actors of the procedure
  ====================================================-->
  <!-- signatory of the document -->
  <authenticator>
   <authenticator.type_cd V="VRF" />
   <participation_tmr V="???" />
   <signature_cd V="S" />
   <person>
    <id EX="BR" />
    <person_name>
```

```
   <nm>
    <PFX V="???" QUAL="AC"/>
    <GIV V="???"/>
    <FAM V="???"/>
    <SFX V="???" QUAL="PT"/>
   </nm>
  </person_name>
 </person>
</authenticator>
<!-- signatory of the document, legal responsible -->
<legal_authenticator>
 <legal_authenticator.type_cd V="SPV"/>
 <participation_tmr V="???"/>
 <signature_cd V="S" />
 <person>
  <id EX="MD" />
  <person_name>
   <nm>
    <PFX V="???" QUAL="AC"/>
    <GIV V="???"/>
    <FAM V="???"/>
    <SFX V="??? z.B. Chefarzt" QUAL="PT"/>
   </nm>
  </person_name>
 </person>
</legal_authenticator>
<!-- intended recipient -->
<intended_recipient>
 <intended_recipient.type_cd V="TRC"/>
 <person>
  <id EX="XXX-MNP-134A" />
 </person>
</intended_recipient>
<!-- originator of the document, e.g. who has dictated -->
<originator>
 <originator.type_cd V="AUT"/>
 <participation_tmr V="9.4.2001 00:00:00"/>
 <person>
  <id EX="nobody" />
  <person_name>
   <nm>
    <PFX V="???" QUAL="AC"/>
    <GIV V="???"/>
    <FAM V="???"/>
    <SFX V="???" QUAL="PT"/>
   </nm>
```

```
  </person_name>
 </person>
</originator>
<!-- originating organization -->
<originating_organization>
 <originating_organization.type_cd V="CST"/>
 <organization>
<organization.nm V="Institut fr Pathologie, Charite Universittsmedizin Berlin"/>
 </organization>
</originating_organization>
<!-- e.g. typist -->
<transcriptionist>
 <transcriptionist.type_cd V="ENT"/>
 <person>
  <id EX="SGJX" />
  <person_name>
   <nm>
    <PFX V="???" QUAL="AC"/>
    <GIV V="???"/>
    <FAM V="???"/>
    <SFX V="???" QUAL="PT"/>
   </nm>
  </person_name>
 </person>
</transcriptionist>
<provider>
 <provider.type_cd V="PRF"/>
 <person>
  <id EX="???" />
 </person>
</provider>
<!==================================================
       information about acceptor of the procedure
 ==================================================-->
 <patient>
  <patient.type_cd V="PATSBJ"/>
  <person>
   <id EX="42539"/>
   <person_name>
    <nm>
     <GIV V="???"/>
     <FAM V="12ddb756d6ccdeb97a3816f57fc4c28b"/>
    </nm>
    <person_name.type_cd V="L" S="2.16.840.1.113883.5.1xxx"/>
   </person_name>
   <addr USE="WP RES">
```

```
    <STR V="???"/>
    <HNR V="???"/>
    <ZIP V="???"/>
    <CTY V="???"/>
   </addr>
  </person>
  <birth_dttm V="69"/>
  <administrative_gender_cd V="M" />
  <local_header ignore="all" descriptor="swpatho">
  <!-- cost unit of the procedure -->
   <swpatho:swpatho-ssu type="Kostentraeger" country="de" version="v1">
    <swpatho:Kostentraegerbezeichnung V="CHA"/>
   </swpatho:swpatho-ssu>
  <!-- the receipt number of the case -->
   <swpatho:swpatho-ssu type="E-Nummer" country="de" version="v1">
    <swpatho:E-Nummer V="E13327-01"/>
   </swpatho:swpatho-ssu>
  </local_header>
 </patient>
</clinical_document_header>
```

A.3 Document Body

```
<body>
 <section>
  <caption>Befund</caption>
   <section>
    <caption>Makroskopie</caption>
    <paragraph>
     <content>
       Lungen-PE: 25 x 8 x 6 mm groe Lungengewebsprobe mit 23 mm
       langer Klammernaht und 5 mm groem, eingeschnittenem Tumor
       mit solider Schnittflche.
     </content>
    </paragraph>
   </section>
   <section>
    <caption>Mikroskopie</caption>
    <paragraph>
     <content>
       Tumorzellen aufgebaut aus relativ groen Tumorzellen
       mit rund-ovalem, teils blasig aufgetriebenem Kern mit
       zumeist deutlichem Nukleolus. Eosionphiles Zytoplasma.
       Zellen teils in trabekulrer Lagerung. Mig bis hohe
       Kernpleomorphie, mitotische Aktivitt. Keine sichere
```

```
        Gallepigmentbildung, teils jedoch Ausbildung von
        gallegangshnlichen Strukturen zwischen den Leberzellbalken.
        Immunhistologisch keine wesentliche
        Expression von Pan-Zytokeratin und von AFP, dagegen
        Positivitt fr den
        Zytozytenmarker sowie fr Alpha-1-Antitrypsin.
     </content>
    </paragraph>
   </section>
   <section>
    <caption>Kritischer_Bericht</caption>
    <paragraph>
     <content>
        Lungenmetastase eines hepatozellulren Karzinoms,
        nach Angabe Lungen-PE.
     </content>
    </paragraph>
   </section>
   <section>
    <caption>Kommentar</caption>
    <paragraph>
     <content>
        Zustand nach Entfernung eines HCC im September 1999.
        Immunhistologisch besttigt sich der bereits
        im Schnellschnitt geuerte Verdacht auf das
        Vorliegen einer Metastase des
        hepatozellulren Karzinoms.
     </content>
    </paragraph>
   </section>
  </section>
 </body>
</levelone>
```

A.4 Dictation Path

```
  <section>
   <caption>Mikroskopie</caption>
   <paragraph>
    <content>
        <coded_entry>
                <coded_entry.value V="123" S="UID"/>
        </coded_entry>
        <coded_entry>
                <coded_entry.value V="56" S="PID"/>
        </coded_entry>
```

```
        <coded_entry>
            <coded_entry.value V="10:23:34" S="StartTime"/>
      </coded_entry>
        <coded_entry>
            <coded_entry.value V="10:25:12" S="StopTime"/>
      </coded_entry>
      Lungenteilresektat mit Manifestation einer malignen Neoplasie,
      teilweise von einer Pseudokapsel umgeben, teilweise diffus
      das angrenzende Lungenparenchym infiltrierend, in soliden
      Komplexen liegend, daneben ausgedehnte nichtkohsive Verbnde,
      hochgradige Dysplasien, vesikulre Zellkerne, prominente
      Nukleolen, zahlreiche, auch atypische Mitosen,
      ausgedehnte Nekrosen, Einblutungen.
  </content>
 </paragraph>
 <paragraph>
  <content>
      <coded_entry>
              <coded_entry.value V="124" S="UID"/>
      </coded_entry>
      <coded_entry>
              <coded_entry.value V="56" S="PID"/>
      </coded_entry>
        <coded_entry>
            <coded_entry.value V="10:25:13" S="StartTime"/>
      </coded_entry>
        <coded_entry>
            <coded_entry.value V="10:28:56" S="StopTime"/>
      </coded_entry>
      Im Randbereich frischer Lungeninfarkt.
      Der Resektionsrand ist tumorfrei.
  </content>
 </paragraph>
</section>
```

B XML-Editor as Standalone Application

B.1 Class Dependencies

The main()-Method is in the class "TextEditClass" (see Figure **??**). It instantiates TextEditFrame, which is the central class containing most of the graphical elements and functions, which control the whole program. The database interface is implemented in the class "BefundDatabase" using xmlDB. When an error occurs, the class "DBException" contains the Exceptions to be thrown. The class "XmlEdit" is also used by "TextEditFrame": It contains functions to open, display and save the XML-documents. For every finding within a medical case a new BefundTab is created, representing a tab in the central JTabbedPane. The class contains just graphical elements. When a new finding is created, an "EditableBefundTab" is inserted into the "JTabbedPane". This class inherits from "BefundTab", but additionally contains information about the diagnostical path. The feature "diagnostical path" is realized using a stack containing an instance of the class "DataElem", which stores the necessary information of a single time interval. The open dialog ("OpenDialog") and the about box (TextEditFrame_AboutBox) are also implemented as standalone classes.

B.2 Class Description

B.2.1 TextEditClass

Contains the main()-method, instantiates the class "TextEditFrame". The control is relayed to "TextEditFrame".

B.2.2 TextEditFrame

The main window controlling the program (see Figure **??**). Most of the graphical elements, as well as parsing feature related to the particular XML format are contained in this class.

Important functions:

- **void jbInit()**: Graphical elements are initialised and painted.
- **void paintPatientenTab()**: All elements of the patient data tab are painted.
- **String getCaption(Node)**: Gets a ¡section¿-node and returns the cdataSection of the child node named ¡caption¿.
- **void insertBefund(Node)**: A ¡section¿-node containing a medical finding as a substructure, is being painted in a newly created tab.
- **void insertSection(String caption, Node)**: used by insertBefund(), inserting the content of a section (e.g. "Makroskopie") into the right place.

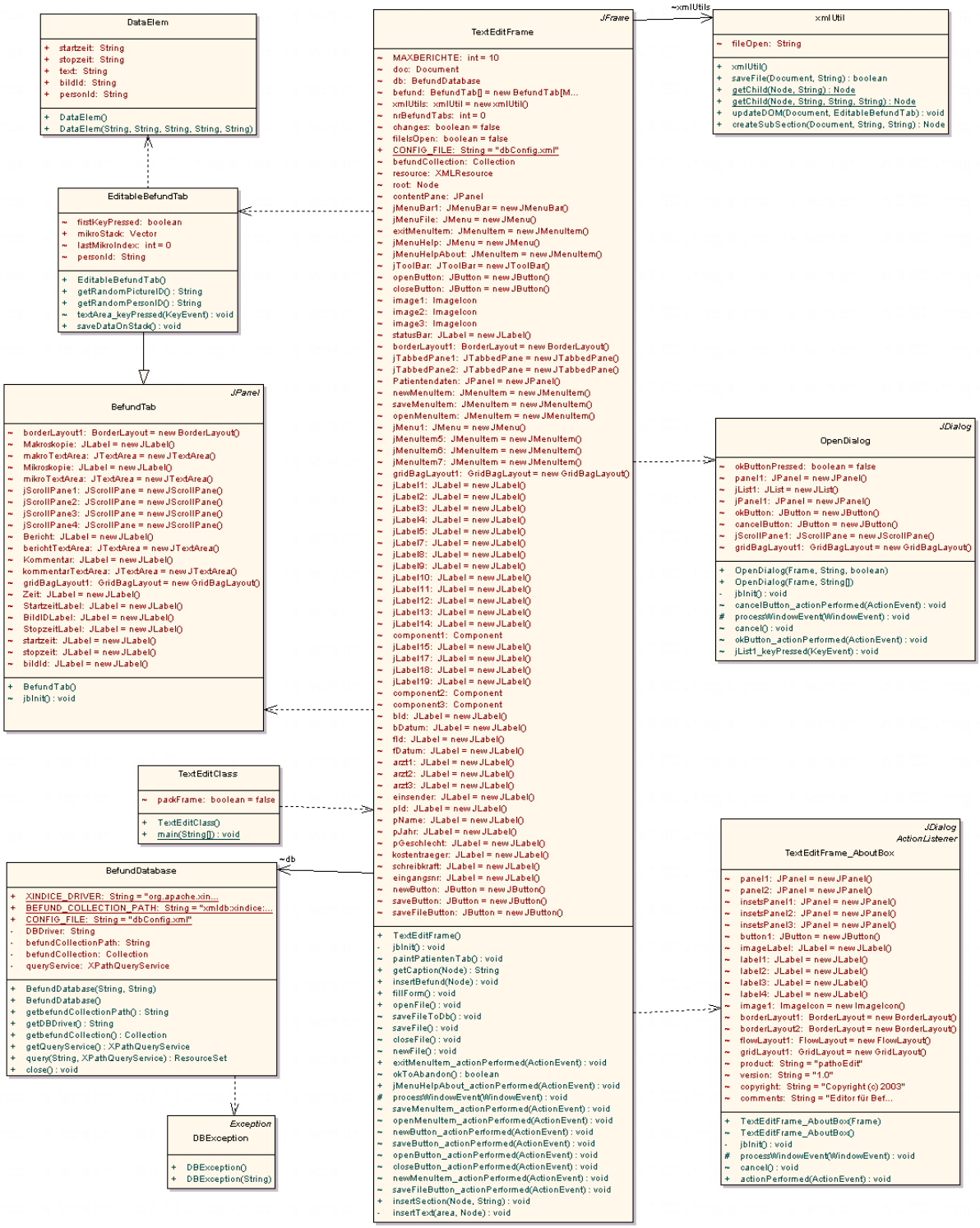

Figure 2: Class Dependencies

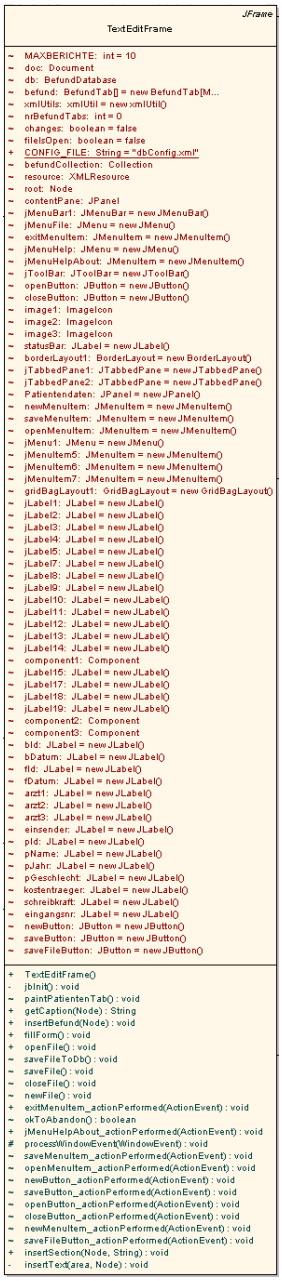

Figure 3: **C**lass TextEditFrame

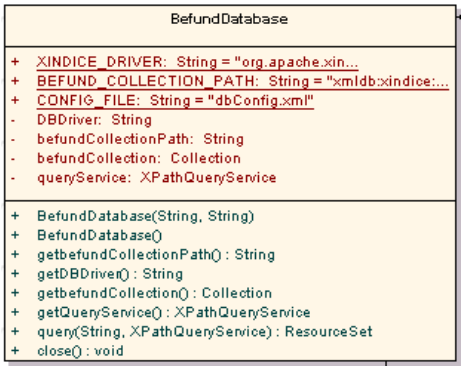

Figure 4: **C**lass BefundDatabase

- **void insertText(Node, JTextArea)**: used by insertSection(), concatenates the content of all ¡paragraph¿-nodes and inserts the resulting text into the corresponding JTextArea. This is essential when inserting the microscopy part, which normally contains several paragraphs because of the diagnostical path.
- **void fillForm()**: Reads a finding and shows the content in a JTabbedPane.
- **boolean okToAbandon()**: When exiting the program, this function is called to ask whether a changing should be saved or not.

Other functions:open, save, close findings

- void openFile()
- void saveFileToDb()
- void saveFile()
- void closeFile()
- void newFile()

B.2.3 BefundDatabase

Constructor: Automatically connects to the database containing the XML-documents. For this purpose a configuration file named "dbconfig.xml" is opened from the working directory. The file contains information about the database driver and the path inside the database leading to the medical findings (Figure **??**).

Example:

```
<?xml version="1.0" encoding="ISO-8859-1"?> <DBConfig>
    <databaseDriver>
        org.apache.xindice.client.xmldb.DatabaseImpl
    </databaseDriver>
    <befundCollectionPath>
        xmldb:xindice:///db/Befunde
    </befundCollectionPath>
</DBConfig>
```

Additionally the QueryService is initialised. If an error occurs, a DBException is thrown.
Important functions:

- **query(String xpath,XPathQueryService service):** With this function a XPath-Query can be performed. A ResourceSet with the matching nodes in the virtual DOM is returned.

Other Functions:

- getBefundCollectionPath()
- getDBDriver()
- getBefundCollection()
- getQueryService()

Those functions return the components of the database constructed by the constructor.

B.2.4 DBException

Class contains the exception thrown by the class BefundDatabase.

B.2.5 DataElem

Data structure including the data that has to be known to remember the diagnostical path for the microscopy. At this stage of implementation the following data is stored (Figure **??**):

- start-time,
- stop-time,
- picture-id,
- person-id,
- text (content)

For every paragraph such a data structure is created and put on the stack in the class EditableBefundTab. When saving a case, this stack is transformed into XML and added to the microscopy section of this report.

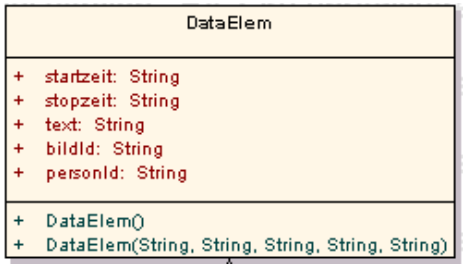

Figure 5: **C**lass DataElement

B.2.6 BefundTab

A BefundTab is a Tab showing a medical finding of a case. It contains the graphical elements, primarily the four text areas and a kind of status bar containing the start- and stop-button, the attached system times and the picture-id (Figure **??**).

B.2.7 EditableBefundTab

This class inherits from "BefundTab" (Figure **??**). In contrary to the "BefundTab" the text areas are editable, there is a stack describing the diagnostical path and there are ActionListeners helping to edit the Text. Normally the user is expected to press the start-button before typing his microscopy part to remember the time he started typing, an ActionListener automatically enables the write mode, if the start button is not activated. When the stop-button is pressed, the function saveDataOnStack() is called. It registers the stop-time, the picture-id and saves everything on the stack.

B.2.8 XmlUtil

This class provides functions concerning XML-specific tasks, like parsing, updating and creating the DOM (see Figure **??**).

Important functions:

- **boolean saveFile(Document,String filename):** This function transforms the content of the Document to XML and writes it into filename. Existing files are overwritten.
- **static Node getChild(Node, String s):** Returns the first found child node having the name "s".
- **static Node getChild(Node,String s, String attr, String val):** Returns the first child node having the name s and an attribute attr with the value val.

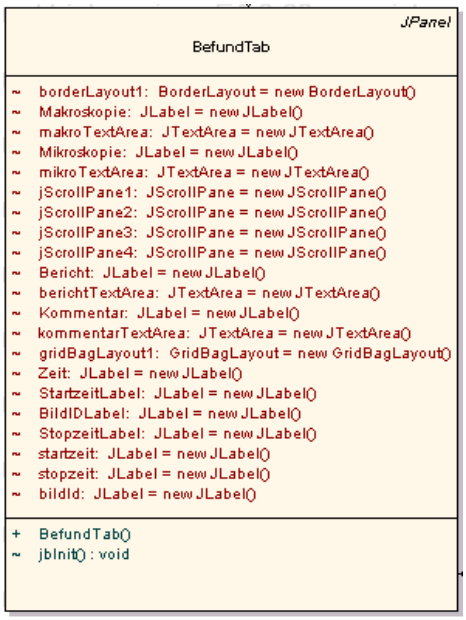

Figure 6: **C**lass BefundTab

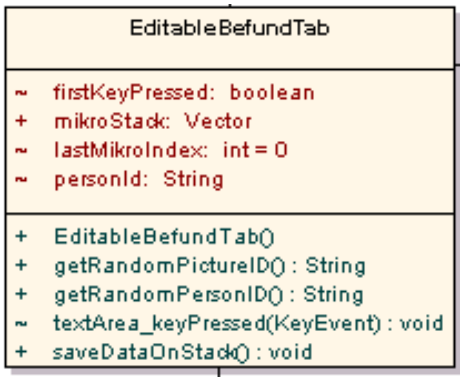

Figure 7: **C**lass EditableBefundTab

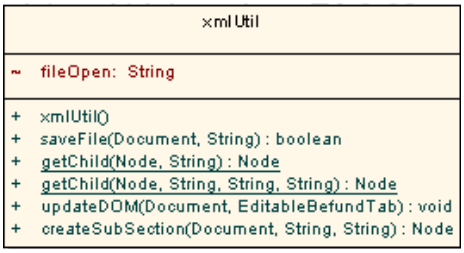

Figure 8: **C**lass XmlUtil

- **void updateDOM(Document doc, EditableBefundTab tab):** The newly created finding is being read, transformed into XML and inserted into the document.
- **Node createSubSection(Document, Stringcaption, String content):** used by updateDOM(), creates the skeleton of a new ¡section¿-node in this form:

```
<section>
    <caption></caption>
    <paragraph>
        <content></content>
    </paragraph>
</content>
```

B.2.9 Other classes

Additionally there are two more classes, OpenDialog and TextEditFrame_AboutBox to open dialog windows.

C XML-Editor as Extension of the DVM