MEDINFO 2001
V. Patel et al. (Eds)
Amsterdam: IOS Press

XML-Based Application Interface Services – A Method to Enhance Integrability of Disease Specific Systems

Udo Altmann, Ali Gerhard Tafazzoli, Frank Rüdiger Katz, Joachim Dudeck

Institute of Medical Informatics
University of Gießen, Gießen, Germany

Abstract

Disease specific systems usually offer excellent functionality for the management of the covered diseases. But the restriction to a certain disease also often hampers their wide spread use since they aren't optimised for clinical workflow. The Gießener Tumordokumentationssystem (GTDS) is such a disease specific system. It is not only designed for the use in tumour registries but also to support clinical care. In order to integrate it into hospital information systems, we implemented standard communication interfaces, but this measure is not sufficient since it doesn't consider aspects of the normal workflow of a clinical user. Therefore, we developed a strategy that should ease the access to the system in the environment of existing systems. From the technical point of view, XML with its capabilities to represent even complex data in a rather simple way helped to implement this strategy. We use it to communicate with API-like services and also created a WWW environment to demonstrate the access to these services. Since this environment itself is a means to integrate systems, we intend to expand this environment to an appropriate region based means to improve the communication with registries.

Keywords:

XML; API; Tumour Documentation; Tumour Registry; Clinical Workstation

Introduction

Disease specific systems usually offer excellent functionality for the management of the covered diseases. The Gießener Tumordokumentationssystem (GTDS) is such a system. In Germany, we developed the paradigm, that tumour documentation should be integrated as seamless as possible into the process of care to avoid retrospective and redundant data entry with its well-known methodical problems [1]. Therefore, GTDS was not only designed for the use in tumour registries but also for the support of clinical care [2]. Many functions support management of follow-up care, the calculation and tracking of therapy plans and the calculation of chemotherapy dosages, and, communication, that is very important in such multidisciplinary environments [3]. The large amount of useful functions and its adaptability has made GTDS (http://www.gtds.de) a successful system in more than 30 various registries.

But the restriction to a certain disease also often hampers the wide spread use of systems since they aren't optimised for clinical workflow. People are more used to departmental systems where they have their order entry and results reporting systems and refuse to deal with too many different systems, especially when they only cover a small percentage of their patients.

According to this experience, GTDS is being used online in oncology outpatient clinics or special wards, but for the large rest, the communication with the registry still is paper-based. Therefore, there is a risk, that important information is not transferred to the registry, because there is no possibility to enter data at the right moment. Such problems can only be answered with integrated solutions. In order to integrate GTDS into hospital information systems or clinical workstations, we first implemented standard communication systems. This measure was very useful to check completeness with regard to the tumour patients treated in the hospital and to avoid redundant data entry but it was not sufficient, because such available data normally is not detailed enough. Bringing the system itself to the physicians' desk is no solution, since it doesn't consider the already mentioned workflow aspects [4].

Therefore, it became clear, that the existing clinical workstations should be extended with a functionality that allows for the documentation of specific items needed in the registry and also provides services that support oncology care. At the time when we developed this strategy, hospital information systems often were rather monolithic and didn't support component based solutions at an affordable price. The emerging techniques of Internet and XML have improved the perspective of such integration efforts.

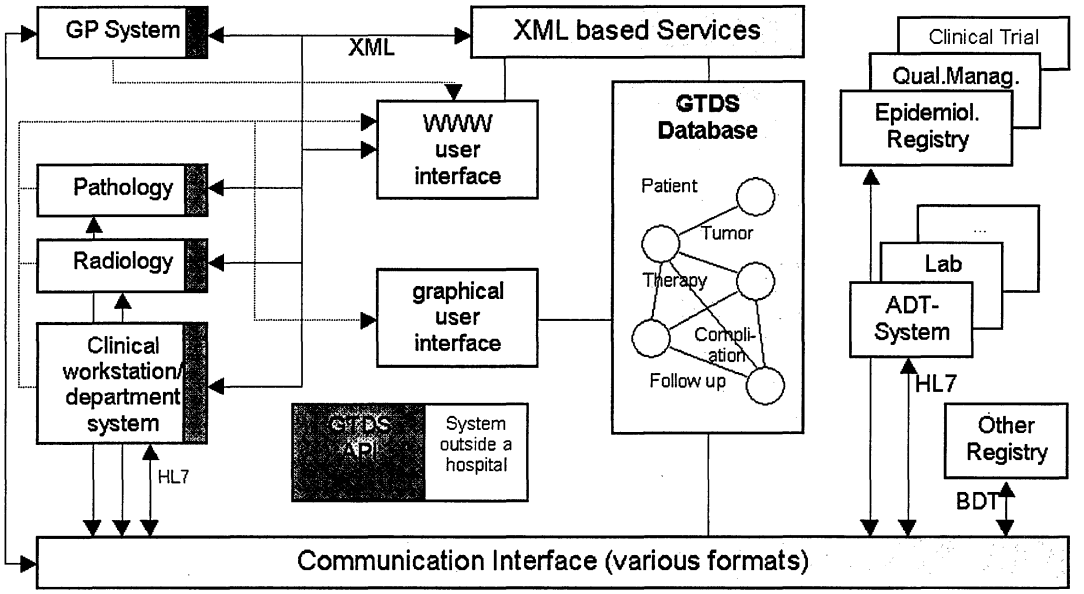

Figure 1 - Overview of existing and possible access methods to GTDS

Specific Requirements of a Registry System

Some aspects distinguish a registry system from other systems in a hospital information system

- Registry systems (at least as used in Germany) often also cover patients that are never treated in the hospital where the registry is located. This fact implies, that the systems have to have their own administrative part for patient demographic data and cannot completely rely on patient identification mechanisms used in the hospital.
- It is not sufficient to regard patient data only in relation to patient and time. It is necessary to be able to record and to distinguish multiple tumour diseases, and, there exist other information objects, that might be tracked separately like the treatment of recurrent disease, metastasis, side effects, or complications caused by the therapy.

These requirements are another reason, why communication of a clinical workstation system with a registry system cannot be completely automated. The task of the registry is to maintain a shared patient record in which information from various care providers has to be integrated. Somebody has to decide, whether a patient is the same patient as found in the registry via comparison of names or a new patient. Is it a new disease or can already some information be retrieved from the registry? Is the complication, that the patient reports, already known? Can it be attributed to a known therapy?

Materials and Methods

Figure 1 gives an overview of the overall design, showing the various methods to communicate with GTDS. Full functionality can be obtained via the graphical user interface. A rather simple method to combine two systems is to start the registry system from an other system (dotted line). But this is not a real integration because both systems share only few data.

Some data can be transferred via communication interfaces. We support various formats, e.g., HL7 mainly for administrative data. BDT is a German standard for physician office systems, for which an oncology extension based on a common basic data set has been defined [5]. Often, it is also necessary to support proprietary interfaces, especially in environments with restricted resources where standard communication interfaces could not yet be implemented. As mentioned above, these communication interfaces work well in settings, where ambiguous situations about patient data cannot occur, for example when all patients of a registry come from one provider. In these settings, identification problems cannot occur and data from other sub systems (mainly surgery codes) can be integrated with rather few efforts. All what is to be done is to implement filter mechanisms that guarantee, that only tumour patient data are processed. These filter mechanisms can be completely automated, but sometimes they can also be supported by user input or be implemented using explicit requests.

The new method accesses the GTDS database via a set of services. Like an application programming interface, the

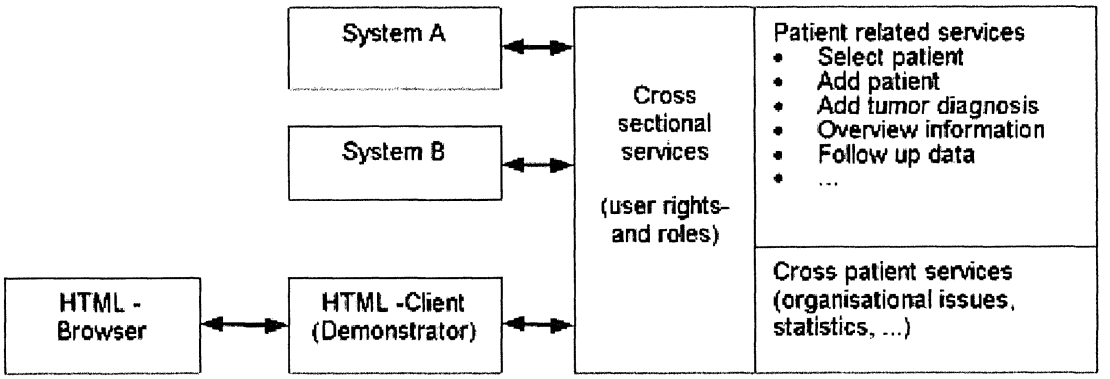

Figure 2 – Overview of the service architecture

services guarantee the integrity of the database and protect the database from unauthorised access.

Unlike other programming interfaces, the information contained in the calls to the interface is rather complex. For sending information about a tumour to the registry, highly structured and coded data, e.g., ICD-O codes, TNM-staging information, metastasis and information about the diagnostic procedures have to be transported. Some of the data, for example metastasis, have a many to one relationship with a tumour. Classical APIs require many calls to the interface with many attributes to be processed sequentially. Since XML is regarded as a medium for processing documents and for implementing communication interfaces, we chose the XML language for the encoding of requests and the delivery of results.

Applications, that need to interact with the registry system, have to implement a small component, that prepares data for requests and displays the results in the application specific look and feel. To show the access to the services by examples, we realised an HTML-based demonstrator, that could also serve as a real application in an appropriate environment.

The objects in the database box in figure 1 symbolise the different objects, on which the services operate.

The technical basis consists of ORACLE RDBMS with stored procedures, Java servlets and, for the HTML demonstrator, PERL scripts that are now replaced also by Java servlets for better performance. For the processing of XML, we could use publicly and freely available modules for Java as well as for PERL. Figure 2 shows an overview of the architecture.

The services consist of cross sectional services that handle user identification, session management, access rights and accessible services. All specific services are handled through these services. The specific services can be divided into patient related services, that process requests to the different aspects of the disease, and services, that operate on many patients, like statistics or services for organisational issues (e.g., reminders to follow up care or interdisciplinary case conferences).

As mentioned before, tumour documentation deals with various types of objects. Therefore, object identification for relating information to the right objects is a major issue when data from different sources has to be integrated. Since object identification often can not only rely on algorithms but has to use human knowledge (e.g., is this really a new metastasis?), communication interfaces can sometimes not be completely automated. Since services have a closer integration with the clinical workstation, services that establish a dialog in such cases have also been implemented.

Java servlets are mainly used for the technical part of the communication. They implement some cross sectional services and break the requests down to atomic components, services on objects, that are themselves realised in stored procedures.

HTML demonstrator

The use of XML is not restricted to the communication with the services. Especially for the HTML-demonstrator, we used XML also to describe the design of the HTML entry forms as shown in figure 3.

```
- <html:tr>
    <html:td>Immuntherapie</html:td>
  - <html:td>
      <immun type="list" list_id="BD906" />
    </html:td>
  </html:tr>
- <html:tr>
    <html:td>Knochenmarktransplantation</html:td>
  - <html:td>
      <kmt type="list" list_id="BD906" />
    </html:td>
  </html:tr>
```

Figure 3 - Description of entry form

This part of the description file has references ("list_id") to a description that can be found in another document. Since our working group also was involved in the development of

the data standard used in hospital cancer registries, we showed the advantages of the formulation of data standards in XML by converting the word processor document to XML [6]. Figure 4 shows the referred section of the standard XML document:

```
<ueberschrift>Art der durchgeführten Maßnahmen</ueberschrift>
- <abschnitt FileID="BD906w.xml">
  - <wert>
      <code>J</code>
      <beschreibung>Ja (diese Maßnahme wurde durchgeführt)</beschreibung>
    </wert>
  - <wert>
      <code>N</code>
      <beschreibung>Nein (diese Maßnahme wurde nicht durchgeführt, war nicht
        vorgesehen)</beschreibung>
    </wert>
  - <wert>
      <code>A</code>
      <beschreibung>Abgelehnt (diese Maßnahme war vorgesehen, wurde aber
        vom Patienten abgelehnt)</beschreibung>
    </wert>
```

Figure 4 – Example from the standard document

Figure 5 shows sample patient data:

```
- <verlauf>
    <patientenId>44</patientenId>
    <verlaufIfdnr>4</verlaufIfdnr>
    <unters_datum>28.04.1993</unters_datum>
    <freitext>Kolonnachsorge nach DKG/ADT, kleines Programm</freitext>
    <erfass_anl>L</erfass_anl>
    <quelle />
    <operation>N</operation>
    <bestrahlung>N</bestrahlung>
    <chemo>N</chemo>
    <hormon>N</hormon>
    <kmt>N</kmt>
    <immun>N</immun>
    <nachsorge>J</nachsorge>
    <ahb>N</ahb>
```

Figure 5 – Sample patient data

All these components are merged by a PERL script to an appropriate HTML entry form (Figure 6).

Zwischenzeitlich durchgeführte Therapie

Operation	Nein (diese Maßnahme wurde nicht durchgeführt v
Bestrahlung	Nein (diese Maßnahme wurde nicht durchgeführt v
Chemotherapie	Nein (diese Maßnahme wurde nicht durchgeführt v
Hormontherapie	Nein (diese Maßnahme wurde nicht durchgeführt v
Immuntherapie	Nein (diese Maßnahme wurde nicht durchgeführt v
Knochenmarktransplantation	Nein (diese Maßnahme wurde nicht durchgeführt v
Nachsorge	Ja (diese Maßnahme wurde durchgeführt)
Anschlußheilbehandlung	Nein (diese Maßnahme wurde nicht durchgeführt v

Figure 6 – Resulting HTML entry form

The HTML variables are named in a way

<SELECT NAME="ergebnis[1]verlauf[1]kmt"> ...

that allows, that the names can be used to reconstruct an XML document with hierarchically structured elements and nested one to many relationships. This XML document contains the possibly modified data and can be sent to an appropriate GTDS service.

The first merging mechanisms were implemented using methods of the W3C DOM standard. With the recent availability of the W3C standard for Transformation (XSLT) and related tools, the same functionality can and is now be realised with XSLT style sheets in a more declarative way. More information is available at

http://www.gtds.de/web

Results

Due to limited resources and some problems with performance using PERL CGI scripts, the project's status is still prototype but several customers see big advantages in the future use of simple HTML interfaces for users with a limited spectrum of tasks and also in the easy availability of clients (browsers) with no need for the installation of runtime environment. This is especially true for the access of general practitioners, where the installation and maintenance of GTDS clients would be very sumptuous. The introduction of health professional cards will provide secure network solutions for this access method. First contacts with industrial partners who also use XML techniques increasingly shall prepare implementations of the XML interface in clinical workstations.

Discussion

The project is part of the result of ongoing work that aims at the development and implementation of methods for a better integration of tumour documentation and clinical routine processes together with the easy implementation of standards in documentation systems. Those standards are needed for communication processes as well as for the comparability of data.

Modern object oriented programming techniques have the aim to define reusable objects from which applications can easily be built, since complex program alorithms are hidden in the objects, and only interfaces allow to interact with the objects. In a similar way, we offer services that operate on the objects of tumour documentation with the aim, that clinical workstation applications integrate functionality of a tumour documentation system at appropriate opportunities in their workflow.

XML and especially the related techniques and tools have the role to integrate the different related systems. Both aspects of XML are used in our approach, the aspect of the usage for communication and the aspect of building documents. The techniques, especially the transformation technique, are a means to convert knowledge, e.g., from data standards, to an appropriate form for the usage in applications. The integration of data standards is an important aspect since the degrees of freedom for the application of XML is very high. In recent publications [7] we discussed this aspect in more detail. This contribution demonstrates the feasibility of the usage of components of a data standard written in XML in an application.

A similar approach with a similar technical basis that we used for the generation of HTML from XML (and vice versa) in the HTML-demonstrator has recently been published in a German computer magazine (X4Dialog) [8].

In earlier publications [9] we already described our model of format independent and format specific components of a

communication interface. The format independent component in principle has the same tasks as the inner part of the GTDS services. It guarantees the observance of access rights and data integrity. Although the technical basis has changed with the introduction of stored procedures, XML can be regarded as another format used by format specific modules. The advantage of data base procedures is that they can be formulated in a high level programming language. Like XML, they ease the reusability of knowledge and code.

Conclusion

Services are an effective method to integrate systems with an optimum of functionality and reduction of programming efforts. We developed application services that aim at a better integration of a disease specific system into a clinical workstation. The development is motivated by the necessity to offer tumour documentation during clinical routine processes to get more complete data. XML and the related techniques have eased the development in various ways. The HTML-demonstrator developed to show the access to the services will be used in situations, where the integration of services is too sumptuous or where only some restricted functions shall be available.

Acknowledgments

Parts of the project were sponsored by the Federal Ministry of Health

References

[1] Dudeck JW, Wagner G, Grundmann E, Hermanek P. Basisdokumentation für Tumorkranke. 4th ed., Berlin: Springer, 1994

[2] Altmann U, Katz FR, Haeberlin V, Willems C, Dudeck J. Concepts of GTDS - An Oncology Workstation. Proceedings of the 8th World Congress on Medical Informatics (1995), 759-762

[3] Altmann U, Katz FR, Tafazzoli AG, Haeberlin V, Dudeck JW. GTDS - a Tool for Tumor Registries to Support Shared Patient Care. Proc AMIA Annu Fall Symp; 1996 Oct 26-30; Washington; Philadelphia: Hanley & Belfus; 512-516,

[4] Altmann U, Katz F, Tafazzoli A, Haeberlin V, Dudeck J. Aspects of Integrating a Disease Specific System into a Hospital Information System. J. Brender et al. (Eds.) Medical Informatics Europe '96, IOS Press, 1996

[5] Altmann U, Wächter W, Müller A et al. Datenaustausch in der Onkologie mittels BDT. Proceedings 43. Jahrestagung der GMDS 1998; Bremen, Germany; pp 247 - 250

[6] Altmann U, Wächter W, Tafazzoli AG, Katz F, Dudeck J. XML as Format for Representation of Data Standards: A Case Study Using the "Basisdokumentation für Tumorkranke". Proceedings 44. Jahrestagung der GMDS 1999, Heidelberg, pp 282 - 285

[7] Altmann U, Tafazzoli AG, Noelle G, Huybrechts T, Schweiger R, Wächter W, Dudeck JW. Combining Dictionary Techniques With Extensible Markup Language (XML) - Requirements To A New Approach Towards Flexible And Standardized Documentation. Proc AMIA Symp 1999;:12-16

[8] Brüning O. Dokumententausch - X4 Dialog: Hilfe für dynamische Webseiten. iX Magazin für professionelle Informationstechnik 10/2000; Heise, Hannover, pp 102-108

[9] Altmann U, Katz FR, Haeberlin V, Tafazzoli AG, Dudeck J. Application of a Standard Methodology for the Development of Messages and Aspects of Realization in the Area of Tumour Documentation. C. Pappas et al. (Eds.) Medical Informatics Europe '97, IOS Press, Amsterdam etc., pp 776-780

Address for correspondence

Dr. med. Udo Altmann
Institute of Medical Informatics
Heinrich-Buff-Ring 44
35392 Gießen, Germany

e-Mail: Udo.Altmann@informatik.med.uni-giessen.de
WWW: http://www.akkk.de, http://www.gtds.de