Structured Reporting of Echocardiography Images in a DICOM Environment

S Nedevschi[1], F Rusu[1], I Popa[1], A Smeu[1],
D Olinic[2], C Homorodean[2]

[1]Technical University of Cluj-Napoca, Cluj-Napoca, Romania
[2]University of Medicine and Pharmacy, Medical Clinic no 1, Cluj-Napoca, Romania

Abstract

The paper presents a proposal and a solution for structured reporting of echocardiography images compliant and included into the DICOM environment we have developed. Our structured reporting method uses and extends the DICOM specifications having as starting points controlled and encoded dictionaries, context groups, and templates. The user-friendly report editor together with the DICOM visualization and processing modules offer to the referring physician the possibility to develop in an interactive and incremental way series of medical reports for the patients being studied. The environment architecture consists of a set of connected servers implementing the storage and retrieval DICOM services and of clients for image visualization, processing and report editing. The proposed methodology was tested for the echocardiography domain, but can be extended to other medical domains.

1. Introduction

Images play a very important role in the current medical process. They represent the starting point in reporting the structural and functional changes of the observed features, but also in diagnosis. In order to obtain interchangeable reports, the data has to be uniformly represented. DICOM Standard has defined structured reporting as a coherent representation modality of the data that can be extracted from medical images. Coded concepts and report templates represent the primitives of this reporting method. Reports are generated through the instantiation of corresponding template rows with data extracted from medical images or supplied by the referring physician.

During the work that was done for the DICOM project, a number of modules were designed and implemented. The archiving system permits the acquisition, storage, and communication of different types of medical images. The reporting environment contains tools for the management of coded concepts, templates, and reports. A way of integrating these two sub-systems for obtaining an environment with both image and report functionalities will be presented next.

The outline of the paper is as follows. Section 2 presents two structured reporting methods: a primary one that is based on templates, and one that takes into account the semantic relationships between report rows. Section 3 describes the architecture of the integrated system that has functions both for image and report management. Section 4 presents some results obtained by a group of echocardiography physicians that used the system we have designed and implemented. Conclusions are grouped in Section 5.

2. Reporting methods

Templates define the structure and the content of medical reports in terms of content items and the explicit relationships between them. In order to have a valid DICOM structured report, the constraints imposed by the corresponding templates have to be accomplished.

As we know, a template has a hierarchical tree structure that can be extended to a directed acyclic graph by the use of references [1]. A report based on a template consists of a valid sub-tree of the template tree. In other words, the creator of the report must search and select the template rows that will be instantiated and populated with values. For a template row to be instantiated in a report as a content item (name – value pair) [2], the required data has to be inserted. For instance, a coded concept is chosen from a context group and a numeric value needs to be specified for a template row with a numeric data type. A suitable structured reporting method must provide an easy way for searching and selecting template rows, and for populating them with values.

The primary reporting method follows the structure defined by templates. The partial report tree is always displayed on screen. On selecting a report row, the user can see all its possible descendants as given by the source template. The tree structure of the report is directly visible and the following operations are permitted:

- edit an existing row;
- navigate through the partial report tree;
- add a new descendant (additional data might be

needed);

- remove an existing row with all its descendants.

Template row presence conditions are evaluated continuously, thus the user can insert only valid rows. This primary method maps one-to-one with template rows and, thus, it provides the maximum possible degree of flexibility.

The primary interface is the result of an evolution process and still it is not perfect. Its limitations can be divided into three major categories:

- structural limitations. The physicians do not appreciate the hierarchical structure. They are used to table-like structures and form filling based interfaces. Another drawback comes from the increased nesting level. The standard templates are very large and consist of hundreds of rows and nesting levels higher than ten. They are impossible to manage without a great deal of focusing and familiarity.
- navigational limitations. It is difficult to locate indirect descendants. For that, doctors have to know the structure of the template and to instantiate the correct rows until reaching the target row. Also, in a large report tree it is hard to find the return points after completing a report module.
- semantic limitations. All templates are focused on structure and contain no data about which rows are semantically connected, or about the order in which they should be instantiated in a report. Our primary method suffers from this limitation that manifests itself by not treating adequately semantically related rows and by the impossibility of using structured editing patterns that originate from the reporting style of each individual physician.

2.1. Structured reporting wizard

Structured Reporting Wizard is an intelligent add-on to the primary reporting method. It is actually an imperative scripting language that assists the user and helps him to eliminate or strongly reduce the described limitations. The user will no longer be forced to manually navigate through the report tree that will be used only for showing the current position in the partially filled report. Semantic relationships between template rows will be used as this scripting language allows specifying the order of template row instantiation.

A SRW program is created for a specific structured reporting template. The creator of the template writes one or more SRW programs for that template. Each program defines a way to complete a report based on that template. The report editor only uses the SRW program. The wizard will guide the user through the template instantiating rows specified by the program. Where it is needed, the user can make decisions that will influence the behavior of the wizard. The user doesn't need to know the template underlying structure. Relieved of navigational tasks, filling a report can be done in a much shorter time and the physician can now focus on the findings not on how to encode the findings.

The SRW language provides easy but powerful constructs. These are the macro and the choice block. A macro is similar to a procedure. It consists of a list of instructions that will be executed in sequence. Currently we have a row insertion instruction and a tree navigation instruction. Variables and row aliases are permitted. A macro provides the linear part of the assisted report editing. The user has no choice but to fill in the data required by the instantiated rows. The rows are usually semantically linked and are inserted in the order decided by the SRW programmer. The other construct is the choice block that gives the user the power to take decisions on how to fill the report. When a choice block is invoked, a list of options is displayed to the user. By choosing one such option a suitable macro is called that will take the user through another linear report filling part. The SRW programmer decides upon the displayed options and the macro associations by using specific language constructs.

Macros can call choices and other macros (nesting is allowed). Choices can call a certain macro, given the user decision. When the macro ends execution (after calling other macros and choices), the choice options are redisplayed, so the user can choose other, or finish. On finishing, the choice block ends execution and returns control to the macro that called it.

To implement this behavior, a virtual machine has been implemented that can parse the source code and act upon primary user interface. The machine holds the parsed program into a semantic tree and executes its instructions. A stack is used for nested calls. Space for local variables is reserved on each macro entry.

3. System architecture

The integrated system that has capabilities both for image and report management is presented in Figure 1. As can be seen, it has a client – server architecture that consists of multiple distributed servers and clients with functions for image processing and report editing. The communication between these components is realized using the DICOM protocol.

The archiving system contains servers for images and reports. They implement the DICOM services for storage, query and retrieval (C-STORE, C-FIND, C-GET). The application server intermediates the communication between the above servers and the database server that realizes the storage service. The security of the communication is other service supported by the application server.

The integrated environment has modules for domain structuring, report editing, and image processing. Each of these will be extensively covered next.

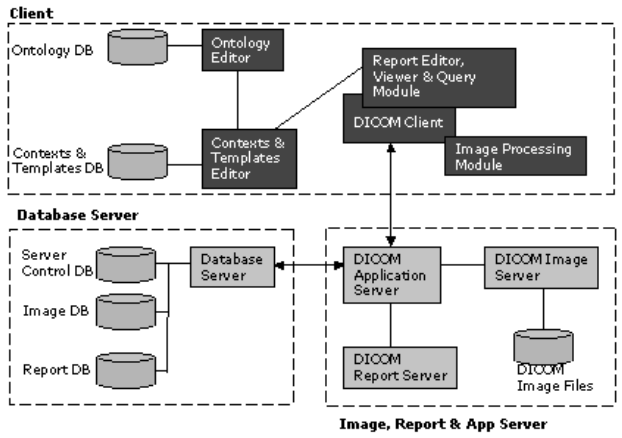


Figure 1. System architecture.

3.1. Medical domain structuring module

This module contains a set of tools used to manage DICOM primitives for structured reporting – coded concepts, context groups and templates – stored in corresponding databases. The main purpose of the module is to realize independence of continuous standard modifications and to enable specialists with powerful tools for extending the standard. Ontology Editor is the application that works with our SNOMED CT-like ontology and has text, language, position and code retrieval capabilities. Contexts & Templates Editor is the other tool in this module and it manages context groups and templates, elements that define the structure and the content of medical reports. The underlining characteristic of these end-applications is their user-friendliness.

3.2. Report editor module

The tasks related to reports – editing, management, rendering and communication – are supported by this module and the associated servers.

Report editing starts from a template that specifies the structure and the type of report content items. Simplified, editing a report is equivalent with the traversal of the template hierarchical structure and the instantiation of each content item with a corresponding value. Unfortunately, this scenario is complicated by the presence of some standard elements – row multiplicity, requirement type and condition, template parameters – that imply a continuous report structure evaluation during editing. Our solution consists of an interpreter that evaluates syntactic expressions for presence conditions and template parameters passing. The recognised syntax contains arithmetical and logical operations, references to other report rows, and template parameters. The allowed expressions incorporate symbolic representations of concepts and contexts. Delegation is used as a primary method in our object-oriented interpreter. As the user advances through the process of report editing, the interpreter evaluates its input and updates the structure such that it satisfies all the constraints. At any point, user choices maintain the partial report in a consistent state. The user is unaware of the interpreter presence and all she/he has to do is just selecting the desired items and supply them with the necessary information through a set of context sensitive dialogs that match the type of data.

In the process of editing we use a framework representation of reports. This representation has the form of a degenerated tree and is closely related with the template representation. Its advantage is flexibility. Practically, reports represented this way can easily be mapped to other formats (XML, DICOM, and text).

We chose to store report-objects in a relational database and thus using the advantages of such an engine. For that, we had implemented a module at the framework level that realises the database transfer in both directions. One of the benefits of using a relational database is the enormous number of possible interrogations. We exploited that feature at report retrieval. The report editor module implements three different methods for locating medical reports. The first method uses patient name or identifier as interrogation key. A more complicated method takes as input parameters concept name – coded value pairs and returns the reports that contain such pairs. The last retrieval method we propose locates medical reports that satisfy a specified content structure. The user has to specify the interest data and also its structure, following the same steps as in the process of report editing. What results in the process of retrieval is a set of reports that satisfy the imposed constraints, but also the attached data to reports – images, waveforms. In the light of this, we could interpret the retrieval method based on content structure as a way of finding images that are defining for some specified concepts. The possibility of storing partial reports and updating them, as the necessary data becomes available, gives flexibility to the report editing process that can extend now over a long period of time and can involve the participation of more persons.

Intuitive rendering of the content of reports is an important feature that eases the understanding and the interpretation of data. A modality of describing the synthetic elements of DICOM standard, such as relationships and value types, into clear text is what has to be found. We have decided to use XML for this purpose because of its extensibility and its large utilisation. XSD and XSLT technologies are used to enhance the graphical aspect of medical reports. Viewing and printing are possible by using ActiveX modules for a variety of web browsers. Rendering of the attached information – images, waveforms – is strictly linked with the notions it best characterises and is done by invoking an applet that extracts it from the DICOM Image Server and visualises it on the client host.

DICOM communication implies standard file format and transmission syntax. Unfortunately, the file format is defined in a way that makes impossible – or, at least, very difficult – the recovery of the structure that generated the report. In order to import structured reports into our framework representation and to include them in the retrieval mechanism, we had to make some extensions to the standard file format.

3.3. Image processing module

This module implements functions for image acquisition from various modalities, for conversion of images to and from different formats, and for processing of single and multi-frame images. The main use of this module is the enhancement of the quality of images in order to ease the extraction of the parameters that have to be reported.

4. Results

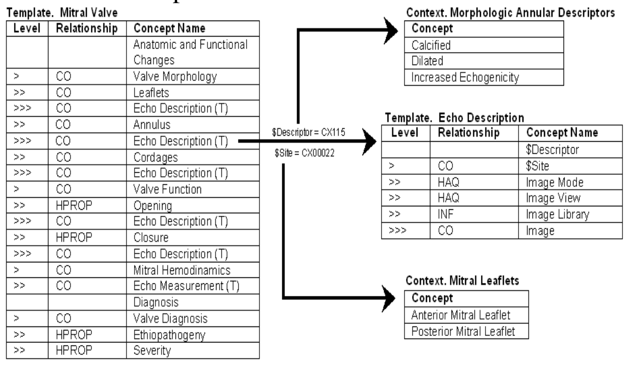
Having the integrated system for use, the physicians in our team could realize a complete and extensive test of the structured reporting methodology. After more iterations through the steps of the methodology, a coherent set of context groups and templates were obtained for the domain of echocardiography. In Figure 2 is presented the final template for mitral valve and some associated contexts.

Using the above template and relevant images, series of reports were generated on real cases. An extract of such a report is represented in Figure 3.

A SRW program was also developed for the above template. The program was based on the reporting methodology practiced in a particular hospital. Later, reports were created using the wizard. A significant speed increase was measured (filling an average report took about two minutes), competing with the free text approach. The SRW approach contributed to achieving a more intelligent, user-oriented interface.

5. Conclusions

The paper presents a DICOM compliant system that integrates functionalities for both medical images and structured reports. Starting from a template, a referring physician generates reports using data extracted from series of images. Two medical reporting methods were presented: a primary one that is based on template structure, and one that takes into consideration the semantic of templates.



Template. Mitral Valve

Level	Relationship	Concept Name
		Anatomic and Functional Changes
>	CO	Valve Morphology
>>	CO	Leaflets
>>>	CO	Echo Description (T)
>>	CO	Annulus
>>>	CO	Echo Description (T)
>>	CO	Cordages
>>>	CO	Echo Description (T)
>	CO	Valve Function
>>	HPROP	Opening
>>>	CO	Echo Description (T)
>>	HPROP	Closure
>>>	CO	Echo Description (T)
>	CO	Mitral Hemodinamics
>>	CO	Echo Measurement (T)
		Diagnosis
>	CO	Valve Diagnosis
>>	HPROP	Ethiopathogeny
>>	HPROP	Severity

Context. Morphologic Annular Descriptors

Concept
Calcified
Dilated
Increased Echogenicity

Template. Echo Description

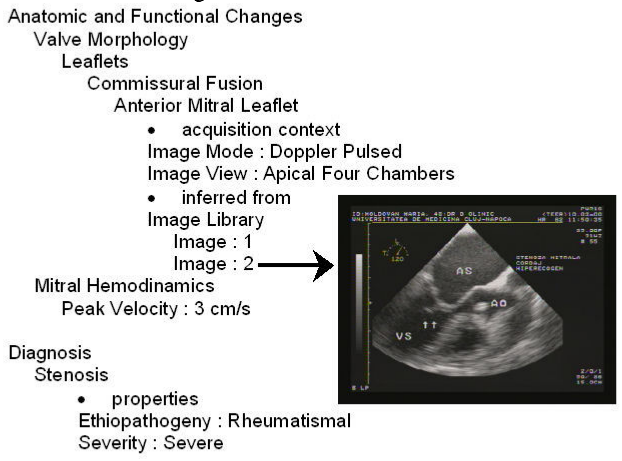
Level	Relationship	Concept Name
		$Descriptor
>	CO	$Site
>>	HAQ	Image Mode
>>	HAQ	Image View
>>	INF	Image Library
>>>	CO	Image

Context. Mitral Leaflets

Concept
Anterior Mitral Leaflet
Posterior Mitral Leaflet

Figure 2. Mitral Valve template and contexts.

The resulted system can be considered a prototype. It has the basic functions of a structured reporting environment that uses images. In the future we intend to extend the image processing module with functions for automated extraction of parameters and the reporting module with a diagnosis inference mechanism.



Anatomic and Functional Changes
Valve Morphology
Leaflets
Commissural Fusion
Anterior Mitral Leaflet
- acquisition context

Image Mode : Doppler Pulsed
Image View : Apical Four Chambers
- inferred from

Image Library
Image : 1
Image : 2
Mitral Hemodinamics
Peak Velocity : 3 cm/s

Diagnosis
Stenosis
- properties

Ethiopathogeny : Rheumatismal
Severity : Severe

Figure 3. Mitral Valve sample report.

References

[1] Nedevschi S, Olinic D, Popovici R, Rusu F, Smeu A, Homorodean C. DICOM Compliant Environment for Structured Reporting in Echocardiography. Computers in Cardiology 2003;30:24-27.

[2] Clunie DA. DICOM Structured Reporting. Bangor, Pennsylvania, USA: PixelMed Publishing, 2000.

Address for correspondence

Sergiu Nedevschi
Technical University of Cluj-Napoca
15 C. Daicoviciu Street, Cluj-Napoca, RO-3400, Romania
Sergiu.Nedevschi@cs.utcluj.ro