SGML Documents: Where Does Quality Go?

José Carlos Ramalho
Jorge Gustavo Rocha
José João Almeida
Pedro Rangel Henriques

Language Processing and Specification Group
Computer Science Department
University of Minho
Portugal
What will we discuss?

- When information increases, when information sources increase and vary, what happens to quality?
- How can we ensure/preserve quality?
- What is quality (what are we talking about)?
- In what contexts is quality more relevant?
- Can we measure it? ...
What are we doing with SGML?

- Constructing document DBs
- Publishing books on the Internet
- Converting parish registers (XIII and XIV century) to SGML
- Publishing from SGML DBs: Internet, CDROM, paper, …
- Connecting SGML Documents to GIS
Quality?

- Quality is good.
- Quality is important.
- Quality is when something is good and achieves to remain good for a period of time.
- Attribute, class, category (from dic.).
- Specific attribute that distinguishes a person, a thing or an entity (from encyclopedia).

Lots of Subjectivity
Quality (in our context)?

- Interface
- ...
- Data relevance
- ...
- Data correctness

There is a lot less subjectivity in this item
Aims of this work

- We want to minimize *Data Incorrectness*
- We don’t want to change existing models
- We want to extend them
- In the end we want to eliminate information revision cycles
SGML authoring and processing model

Diagram:
- Editor
  - DTD
  - Design Process
- Editor
  - SGML Doc.
  - Authoring Process
- Parser
  - Validation Process
  - OK / errors
- Formatter
  - Style Specification
  - Valid SGML Doc.
  - Formatting Process
- OUTPUT
Data (in)correctness

Example 1: Portuguese History

- Kings
- CD ROM
- ... ???
- Kingdoms
- Wars
Data (in)correctness

Example 1: Portuguese History

What went wrong?

- Kings with inexistent kingdoms
- Wars happening in the wrong era
- Characters that died before they were born
- ...
Data (in)correctness

Example 2: Parish register (XIII and XIV century)

- Death certificate
- Marriage articles
- Baptism certificate
- Family Database

...???
Data (in)correctness

Example 2: Parish register (XIII and XIV century)

Problems:

- negative ages
- death before baptism
- marriages between people with age differences higher than 100
- ...

Birth certificate

Death certificate
What do we propose?

- An extra validation task:
  - we need an additional level of abstraction separating information content from document structure.
- Implemented over an external functional system (in the moment …)
- Capable of expressing invariants and pre-conditions over data contents
- Invisible from the user point of view
Special Comment Sections: embedding code in DTDs

```xml
<!DOCTYPE king [
<!ELEMENT king -- (name, coname, bdate, …)><!-- INV:
inv_king(k) = …
-->
<!DOCTYPE king [ … ]>
```

Throught an anchor to an external file

```xml
<!-- INV: king.cam -->
<!DOCTYPE king [ … ]>
```
Example: kings and decrees

<!-- INV: king.cam -->
<!DOCTYPE king [
<!ELEMENT king -- (name, coname, bdate, ddate, decree+)>
<!ELEMENT decree -- (date, body)>
<!ELEMENT (name, coname, bdate, ddate, date) -- (#PCDATA)>
<!ELEMENT body -- (#PCDATA)>]

Inv_king(k) =
{ if( k notin famous_personsDB → k ++ “not in FPDB”),
  if( bdate_(k) > ddate_(k) → k ++ “died before he has born”),
  if( ddate_(k) - bdate_(k) > 120 → k ++ “lived more than 120”),
  if( !all( x ← decree_l(k) : bdate_(k) < date_(x) /\ date_(x) < ddate_(k) ) → k ++ “made a decree outside his life” )
};

king.dtd

king.cam
Example: kings and decrees

```
<king>
  <name>D.Dinis</name>
  <coname>Farmer</coname>
  <bdate>1270.09.23</bdate>
  <ddate>1370.09.23</ddate>
  <decree>
    <date>1300.07.15</date>
    <body>From this day only bicycles are allowed to circulate.</body>
  </decree>
  <decree>
    <date>1389.11.03</date>
    <body>McDonald’s will sell green wine instead of COCA-COLA.</body>
  </decree>
</king>
```

ERRORS:
D.Dinis must be inserted in FPDB.
D.Dinis made a decree outside his life.
Other Examples

- Tying an Archaeological Database to a GIS:
  - archaeological SGML documents have geographical coordinates.
  - we must ensure that every one of those coordinates is within a certain range.

- City Council Elections
  - each voting section produces a final report with the results (an SGML document).
  - we must ensure that the number of votes matches the number of subscribed voters minus the absent ones.
New SGML auth. and proc. model

Validation Process 2/2

CAMILA

DTD2CAM

DTD

Editor

Design Process

Validation Process 1/2

Parser

ESIS

SGML Doc.

OK / errors

Valid SGML Doc.

Formatter

Style Specification

Formatting Process

OUTPUT

Authoring Process

OK / errors

SGML Doc.
Camila Validation Process

Designer

DTD

<kimg>

...</kimg>

esis2cam

ESIS

nsgmls

validate

OK / errors

User

DTD

<king>

...</king>

Types

Invariants

LOAD aux. Func.

Data flow

Control flow
**Camila Validation Process**

- **Designer**
- **Types**
- **Invariants**
- **LOAD**

**Data flow**

**Control flow**

```xml
<!ELEMENT king - - (name, coname, bdate, ddate, decree+)>

dtd2cam

**Types**

```c
TYPE
  king = name_ :name
  coname_ :coname
  bdate_ :bdate
  ddate_ :ddate
  decree_l :decree-seq

ENDTYPE
```

```c
inv_king( k ) = true;
```

SGML/XML’97 - 8..11Dec - Washington - José Carlos Ramalho

19
Conclusion

- The new proposed model enables us to put some kind of data constraints associated with DTD element contents.
- We can avoid many errors given by a distracted user.
- We can improve information quality and reduce information revision cycle.
In the case studies we have dealt with so far we didn’t find complex invariants.

Structural correctness imposed by SGML already enforces some validation over element contents.

Most of needed invariants are very simple: domain range validation, relationship validation, ...
Future Work

- A simple constraint language is being studied/created to optimize the proposed system.
- We are going to implement this validation scheme (with the new language) in our prototype INES ("A Document Programming Environment").
INES: Document Programming Env

Designer
Context Rules
Style Specification

DOCTYPE "X"

INES

Texto "Y"
Texto "X"
Texto "Z"

Utilizador A
Utilizador B
Utilizador C

Doc X
Doc Y
Doc Z
INES: inside

Designer

Errors

Context Conditions; Invariants

Style Specification

DTD Editor

DTD

DTD

Context Editor

DTD

Editor Generator

SGEN

Código Scheme

“X” Editor

DTD

DSSSL Editor

DSSSL

RTF

PostScript

Doc X

Graphical representation of the INES system.