Beyond DTDs: Constraining Data Content

José Carlos Ramalho
Pedro Rangel Henriques

Language Processing and Specification Group
Computer Science Department
University of Minho
Portugal

Work Background

- Project DAVID (1995) - "Algebraic Document Processing"
- Goal: Document Programming Environment
- Documents ≈ Programs
  - Both have to be processed
  - The steps towards processing are the same
- Platform for format conversion
- Definition of a Doc. Program. Language
Project DAVID

INPUT
SGML

DPE

OUTPUT
PS, HTML, RTF, ...

• Just taking advantage of syntax
• Putting aside all the work done about parsing

First Prototype

SGML Doc.
Translation Logic
Task Spec.
Processing Engine
Style Spec. - unpars.rules

NEW DOCS

Abstract Data Model
**Evolution**

- Good ideas - too academic!? 
- Lack of practice
- The Need to work with existing SGML models and Apps

To use this framework to process Semantics in SGML Documents

---

**What will we discuss?**

- Concerning SGML authoring, is Structural validation enough?
- Can we let the user to have full control of data?
- How can we Constrain more than Syntax?
- Is this problem a problem?
- What is its relevance?
What are we doing with SGML?

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

Document Essence

- What is a Document?
- What is its purpose?
- Who is its target audience?

- Must we have some special care regarding any of these items?
Document is ...

- “A paper with something written on” - 90% of the people
- “Something that teaches or can be used as an evidence” - from dic. encyc.
- “An information registry: a fact, an action, …
  - its structure can range from very simple to very complex: a letter; a book
  - it can have multiple forms

Existing SGML applications ...

Case Study 1: Parish register (XIII and XIV century)

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

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

Existing SGML applications ...

Case Study 1: Parish register (XIII and XIV century)

Case Study 2: Archaeology (Sites and Artefacts)
Case Study 2: Archaeology (Places and Artefacts)

Existing SGML applications …

Arch.Place record
Arch.Artefact record
SGML database
Internet (www)

Every coordinate (latitude and longitude) falls into the map in question

SGML and Semantics

- Can we just add constraints to SGML specifications in order to process semantic validations?
- What is missing?
- Do we need more than just constraints?
SGML: What ...

- SGML was conceived to specify structure
- Today, it is a very powerful specification tool
- We can not use its syntax to express constraints or invariants over element content

SGML: How ...

- To add extra markup
- To design a complete new language that could be embedded in SGML or coexist outside. Ex: database applications embed SQL
The Constraints

90% of the cases are very simple
- To restrict atomic element values
- To check relations between elements
- To perform a lookup operation with a value in some database

We feel the simplicity derives from the fact that SGML takes care of validations at higher levels (the structural ones).

The Semantic Validation Model

2 different steps:
- the definition: the syntactic part; the statements that express the constraints.
- the processing: the semantic part; the constraints interpretation.
The 2 Steps

- The definition implies:
  - the creation of a new language
  or
  - the adoption of an existent one
- The processing implies:
  - the creation of an engine capable of processing constraints written in the language above

Building a Semantic Val. Model

Definition Stage

Constraints

Text?
Structured Text?
Typed Information?

SGML Document

Processing Stage

?
What about types?

We are comparing an element against numeric values
Those values have an inherent type (integer or float)
The processing engine needs somehow to associate that type with the element being compared

Numeric types are easy to infer. What about other types?

What about other types?

But we still have more types like lists ...

There are more than 100 date formats
Here we have another problem:
  – Data Normalization
Building a Semantic Val. Model

Definition Stage

Constraints

SGML Document

? = Typed Information

+ 2 problems:
  - data normalization
  - type inference

Processing Stage

New SGML auth. and proc. model

Validation Process 2/2

Constraint Processor

Parsed SGML data

OK / errors

Validation Process 1/2

Parser

SGML Doc.

OK / errors

Valid SGML Doc.

Formatter

 OUTPUT

Style Specification

Constraint Editing

DTD

Editor

Design Process

Authoring Process

Formatting Process

“Beyond DTDs …”
Solutions

- Developing complex tools to do the job:
  - first the data normalization
  - then the type inference

- Adding some extra definitions into the DTD and some extra markup to the SGML data

The Solution

- To define 2 extra attributes optional to all elements:
  - `value` for data normalization (as in TEI DTD)
  - `type` for type inference

Examples:

```xml
... it happened in <date type="date" value="1853.10.05"> the fifth of October of the year 1853 </date> ...

... <latitude type="float">41.32</latitude> ...
```
Do we need to type every element?

**Archaeological sites Doc. Structure**

```
  arqsits
     |
     v
  arqelem
     |
     v
  identi
     |
     v
  latitu
  |
  quadr
  |
  interp
  |
  texto
```

Do we need to type every element?

A “no” answer implies:

- simplicity in the constraining language
- simplicity in the processing engine implementation
- an incomplete abstract model;
  - this would disable any manipulation of the document as an abstract model
Do we need to type every element?

Archaeological sites Doc. Structure

identi Type = “string”
latitu Type = “float”
Type = “string”
liga

texto Type = “text”

A “yes” answer would imply:
- a more complicated type system and processing engine
- probably the structured types are best inferred from the DTD
- we are able to process the document as an abstract model
**New SGML auth. and proc. model**

![New SGML auth. and proc. model diagram](image)

**Are we reinventing the Wheel?**

Looking for similarities ...

**SGML Documents and Programs**

- Program
- Program. Language
- Grammar Rules
- Terminals: chars and words (grammar)

- SGML Document
- Markup Language
- DTD
- Terminals: chars and words (SGML declaration and DTD)
Programming Languages Processing

- Lexical Analysis
- Syntactic Analysis
- Semantic Analysis
- Code Generation

PL Processing Models

Syntax Directed Translation (SDT)
- Lexical Anal. - specified and automatic
- Syntactic Anal. - specified and automatic
- Semantic A. - programmed by the developer

that is the scenery we have with SGML processing model
**PL Processing Models**

Semantics Directed Translation (based in Attribute Grammars - Knuth 1968)

- Lexical Analysis - …
- Syntactic Analysis - …
- Semantic A. - **specified and automatic**

*This is our goal in SGML context*

---

**Conclusion**

- We have specified a way to add Semantics to SGML Documents
- We have identified the problems: data normalization and type inference
- We did not specify a Constraining Language but we know that it is not a problem
- The problem will be the Constraint Processing
Future Work

- A simple constraint language is being studied/created.
- We are going to implement this semantic validation scheme (with the new language) in our prototype INES (“A Document Programming Environment”).

Questions …?

After Conference Contacts:
- Web: www.di.uminho.pt/~jcr
- Email: jcr@di.uminho.pt