

XTDL — XML Tool Definition Language

Daniela da Cruz¹
Pedro Rangel Henriques¹

¹Departamento de Informática
Universidade do Minho

15 de Fevereiro de 2007

Contexto

- O desenvolvimento de ferramentas leva, tipicamente, à necessidade de as disponibilizar aos utilizadores;
- Estes normalmente começam por pedir uma cópia que o autor da ferramenta tende a ceder quer pelo envio por email quer pela gravação numa flash, cd-rom, ...;
- Surgem então duas questões:
 - A *mobilidade*: o tamanho da cópia da ferramenta excede a capacidade dos meios disponíveis;
 - A *instalação* da ferramenta tem dependências relativamente a outros pacotes de software ou a execução de procedimentos menos óbvios ao utilizador final.

Motivação

- Face a estes problemas urge encontrar uma solução que permita disponibilizar:
 - A ferramenta de forma a que os potenciais utilizadores a possam copiar facilmente;
 - A documentação básica sobre instalação e uso.
- A solução mais adequada para responder a estes 2 requisitos é a criação de um site WWW;
- Mas, com o passar do tempo e com o contínuo desenvolvimento da ferramenta, é necessário manter o site e produzir outros materiais com conteúdo semelhante mas fins diferentes.

- 1 Contexto
- 2 Motivação
- 3 Descrição de ferramentas
- 4 A linguagem XTDL
- 5 Os geradores XTS
 - SC — Site Generator
 - DC — Datasheet Generator
 - PC — Presentation Generator
 - CMG — Conceptual Map Generator
- 6 O site XLS
 - Esquema de processamento
- 7 Conclusão

Descrição de ferramentas

Na base de todo este trabalho está o conjunto de características que descreve uma ferramenta:

- Uma identificação: *designação* e *nome*;
- Breve descrição;
- *Autores* — nome, email — e instituição/empresa a que estão associados;
- Contexto de desenvolvimento — projecto, disciplina,...;
- Equipe de manutenção;

Descrição de ferramentas

- Porque foi concebida e Para que serve;
- Como está architectada e Como foi desenvolvida — estratégia de programação, tecnologia empregue,...;
- Documentação — manuais de instalação e utilização, exemplos, relatórios, artigos;
- Versões disponíveis, associando a cada uma delas uma *data* e o respectivo *URL* para download;

XTDL — XML Tool Definition Language

- XTDL, um dialecto XML, para auxiliar a distribuição e documentação de ferramentas;
- XTDL é constituída por um conjunto de **elementos** e **atributos** que expressam os *descritores* independentemente da transformação que se pretende fazer;
- XTDL permite escrever um único documento para a geração das diferentes documentações.

XTDL — XML Tool Definition Language

- O Schema XTDL terá as seguintes *tags*:
 - `tool`: elemento raiz constituído por *id*, *name*, *logo* e *lastUpdate*;
 - `team`: *authors*, *filiation*, *projectName*, *projectDate*;
 - `description`: *introd*, *context*, *tech-desc* (*explanation* e *arch*);
 - `distributions`: (*distribution*)⁺ — *version*, *file*, *samples*;
 - `docs`: (*doc*)⁺ — *label* e *url*;
 - `news`: (*new*)⁺ — *label* e *url*;
 - `links`: (*link*)⁺ — *label* e *url*;
 - `conceptualMap`: *from*, *assoc*, *to* e *url*.

XTDL — XML Tool Definition Language

- Cada *tag* permite 3 atributos: *site*, *presentation*, *datasheet*;
- Cada atributo é opcional e poderá ter 1 de 2 valores possíveis: *yes*, *no*;
- Valor do atributo \Rightarrow que documentação é gerada.
- Usa-se uma estratégia hierárquica para a propagação do valor dos atributos.

XTDL — XML Tool Definition Language

```
<tech-desc site="yes">
  <explanation>
    <paragraph site="yes"
      presentation="yes"
      datasheet="yes">
      A brief description of our tool ...
    <item presentation="no"
      site="no">...</item>
    <item site="yes">...</item>
  </paragraph>
  <paragraph site="no">
    Second paragraph
    <item>...</item>
  </paragraph>
</explanation>
</tech-desc>
```

- Nodo com atributo **no**: ignorado e todos os seus descendentes também;
- Nodo com atributo **yes**: gera informação enquanto nenhum dos seus descendentes tiver o respectivo atributo re-definido para **no**.

XTS — XTDL Tool Set

- XTS processa a especificação XTDL;
- XTS inclui geradores para:
 - Site (SG — Site Generator);
 - Folheto (DG — Datasheet Generator);
 - Conjunto de diapositivos (PG — Presentation Generator);
 - Mapa de Conceitos (CMG — Conceptual Map Generator).

SC — Site Generator

- *Objectivo:* gerar a partir da descrição XTDL um site que permite a um público geral usar ou saber mais detalhes da ferramenta;
- *Linguagem destino:* HTML;
- *Atributo a considerar:* site.

SC — Site Generator

Para a geração do site WWW são necessárias várias travessias ao documento XTDL:

- 1 Para a geração dos links;
- 2 Para a geração da informação relativa aos links da travessia anterior.

SC — Site Generator

Exemplo: folha de estilo XSL responsável pela primeira travessia:

```
...
<table>
  <xsl:if test="//tool/description[@site='yes']">
    <xsl:apply-templates select="tech-desc" mode="first" />
  </xsl:if>
  <xsl:if test="//tool/distributions[@site='yes']">
    <xsl:apply-templates select="distributions" mode="first" />
  </xsl:if>
  ...
</table>
...

<xsl:template match="distributions" mode="first">
  <xsl:if test="(count(@site)=0) or (@site='yes')">
    <tr><td bgcolor="#F4FEFF">
      <a href="#distributions">Distributions</a></td></tr>
    </xsl:if>
  </xsl:if>
</xsl:template>
...
```

Caso de estudo — LissC

Logotipo

Nome

Coluna esquerda com os links

Technical Description
[Distributions](#)
[Documentation](#)

LISS COMPILER

Coluna central com a introdução e o contexto

Mapa de conceitos

Contactos dos autores

```

<description datasheet="yes" site="yes">
...
<tech-desc>...</tech-desc>
</description>
<documentation>...</documentation>
<distributions datasheet="no">...</distributions>
<news datasheet="no" site="no">...</news>
<conceptualMap>...</conceptualMap>
    
```

LISS (Language of Integers, Sequences and Sets) is a programming language that allows us to operate with integers (literals and variables), scalars, sequences (static arrays, and dynamic lists with pointers) and sets defined in comprehension.

Our initial goal, in the context of a Compilers Course, was to develop a complete compiler for a traditional imperative and structured programming language, LISS, just for teaching generators, parsing and code generation. In the first version, the compiler uses a top-down parsing strategy (pure recursive-descent, mixing LL-Conflicts with a K-lookahead approach), and is a syntax-directed translator (control-flow semantic rules associated during parsing), supporting inherited and synthesized attributes; it was implemented in C#, the compiler generator system CoCo/R (C# version) was used. In the version (under development) the compiler will be generated in Java by LISA tool.

For the verification/validation of the machine-code generated for a given program in LISS, we plan to use the tool LIESOM: ALMA, a visualization and annotation tool developed by us some time ago (in the context of FGVIA project), be used to inspect and understand LISS programs. This requires the generation of an ALMA front-end for LISS, what is done using, once again, LISA system.

```

graph LR
    LissC[LISSC] --> Compiler[Compiler]
    Compiler --> Tool[Tool]
    
```

danielehenriques.sty@fca.unicamp.br
 Copyright © 2016, Departamento de Informática, Universidade do Estado

DC — Datasheet Generator

- *Objectivo*: gerar um folheto para apresentar brevemente a ferramenta a um público, também genérico, que pretende conhecer a ferramenta de uma forma mais superficial;
- *Linguagem destino*: \LaTeX ;
- *Atributo a considerar*: datasheet;

DC — Datasheet Generator

Template para geração do campo autores:

```
<xsl:template match="authors">
<xsl:if test="count(((author[@datasheet='yes']
  | (author[count(@datasheet)=0]))) &gt; 0">
  \author{<xsl:for-each select="(author[@datasheet='yes']
    | (author[count(@datasheet)=0]))">
    <xsl:apply-templates select="."/>
  </xsl:for-each>}
</xsl:if>
</xsl:template>

<xsl:template match="author">
<xsl:variable name="pos"
  select="count(following-sibling::author/name[@datasheet='yes']
    | following-sibling::author/name[count(@datasheet)=0])
    - count(following-sibling::author[@datasheet='no'])"/>
<xsl:choose>
<xsl:when test="$pos &gt; 0">
  <xsl:apply-templates select="name"/> \and </xsl:when>
  <xsl:otherwise><xsl:apply-templates select="name"/></xsl:otherwise>
</xsl:choose>
</xsl:template>
```

Caso de estudo — LissC

Resultado de processamento:

```
\section{Technical Description}
\subsection{Description}
LISSc was generated automatically from its
Attribute Grammar using the Compiler Generator
tool LISA.
LISSc generates pseudo-code for the virtual
machine VM.\\
\begin{itemize}
\item First version: was used CoCoR
(compiler generator) and virtual machine MSP.
\end{itemize}
```

Daniela da Cruz Pedro Rangel Henriques
February 12, 2007

1 Introduction

LISSC is a Compiler for the toy imperative (or procedural) programming language LISS, that stands for Language of Integers, Sequences and Sets.

2 Context

...

3 Technical Description

3.1 Description

LISSC was generated automatically from its Attribute Grammar using the Compiler Generator tool LISA. LISSC generates pseudo-code for the virtual machine VM.

- First version: using CoCoR

4 Development

...

- Version: 3.0

5 Site WWW

<http://www.di.uminho.pt/~gpl/LISS>

PC — Presentation Generator

- *Objectivo*: gerar um conjunto de diapositivos para apresentar a ferramenta a um público mais técnico com o intuito de conhecer e discutir mais detalhadamente a ferramenta;
- *Linguagem destino*: \LaTeX (beamer);
- *Atributo a considerar*: presentation.

PC — Presentation Generator

A estratégia para gerar este conjunto de diapositivos, foi:

- 1º diapositivo: informação acerca dos autores que irão realizar a apresentação, o tema (nome da *ferramenta*), local e data;
- 2º diapositivo: conteúdo global da apresentação organizado em tópicos;
- 3º Restantes diapositivos: no caso do elemento *description*, encontra-se dividido em 3 partes, que darão origem a diapositivos diferentes: introdução; contexto; descrição técnica;

Caso de estudo — LissC

The screenshot shows the 'Contents' page of the LissC website. The page has a dark blue header with a navigation menu containing 'Introduction', 'Context', 'Technical Description', 'Development', and 'Util Links'. The main content area is white and lists the following items:

- 1 Introduction
- 2 Context
- 3 Technical Description
 - Description
 - Architecture
- 4 Development
- 5 Util Links

At the bottom of the page, there is a footer with the text 'Daniela da Cruz, Pedro Henriques' and 'LissC Compiler'. Navigation icons are visible at the bottom right of the content area.

The screenshot shows the 'Context' page of the LissC website. The page has a dark blue header with a navigation menu containing 'Introduction', 'Context', 'Technical Description', 'Development', and 'Util Links'. The main content area is white and contains the following text:

At the beginning the language was conceived with pedagogical purposes, in the context of a Compilers Course. Nowadays it was recovered as a research project in the context of the Language processing group at University of Minho, to explore compiling techniques, generators and virtual machines.

At the bottom of the page, there is a footer with the text 'Daniela da Cruz, Pedro Henriques' and 'LissC Compiler'. Navigation icons are visible at the bottom right of the content area.

CMG — Conceptual Map Generator

- *Objectivo*: gerar um navegador conceptual sobre um MC, integrado no site WWW, que permita facilmente explorar o conhecimento na área relacionada;
- *Linguagem destino*: Prolog (a partir do qual se produz Dot com um compilador desenvolvido no seio do nosso grupo);

CMG — Conceptual Map Generator

Gerador:

- 1 Transforma relações descritas na *tag* `rel` do documento XTDL em factos Prolog (`map/4`);

Compilador:

- 2 Traduz os factos Prolog para `dot` do Graphviz — para ser visto posteriormente com o visualizador WebDot.

Caso de estudo — LissC

(XTDL) \Rightarrow (Prolog)

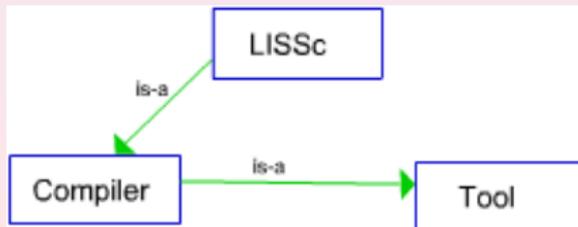
```
<conceptualMap>
<rel>
  <from>LISSc</from>
  <assoc>is-a</assoc>
  <to>compiler</to>
  <url>www.di.uminho.pt/~gepl/LISS</url>
  map('LISSc',is-a,'compiler', 'www.di.uminho.pt/~gepl/LISS')
</rel>
<rel>
  <from>compiler</from>
  <assoc>is-a</assoc>
  <to>tool</to>
  map('compiler',is-a,'tool', '')
</rel>
</conceptualMap>
```

Caso de estudo — LissC

(Prolog) \Rightarrow (Dot)

```
map('lissc',is-a,'compiler',  
    'www.di.uminho.pt/~gepl/LISS')  
map('compiler',is-a,'tool', '')
```

```
digraph "LanguageProcessing@di.um.pt"  
{  
  node[shape=box,color="blue"];  
  "LISSc" [ URL = "www.di.uminho.pt/~gepl/LISS" ]  
  "Comppiler" [ URL = "" ]  
  "Tool" [ URL = "" ]  
  "LISSc" -> "Compiler" [ label="is-a" color="green"]  
  "Compiler" -> "Tool" [label="is-a" color="green"]  
}
```



XLS — XTDL Laboratory Site

XLS é um site criado para disponibilizar o conjunto de geradores XTS para permitir a edição online do ficheiro XTDL e o uso das ferramentas à distância; após a edição do ficheiro e geradas as documentações, é permitido o seu *download* para, de forma independente, proceder à respectiva publicação;

XLS — XTDL Laboratory Site

Tecnologias utilizadas:

- Altova StyleVision 2007 e Altova Authentic 2007 — para permitir a edição do XML online;
- Microsoft Visual Studio .Net — aplicação Web desenvolvida em ASP.NET na linguagem C# recorrendo às classes: `System.Xml`, `System.XSL`, `System.XPath`;
- MySQL (registo de utilizadores e respectivo histórico);

Esquema de processamento

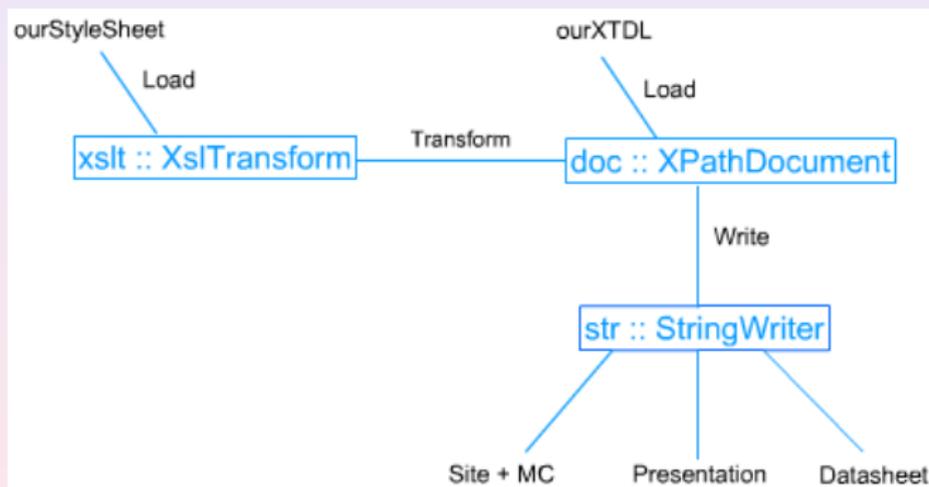


Figura: Esquema de processamento do documento XTDL

XLS — XTDL Laboratory Site

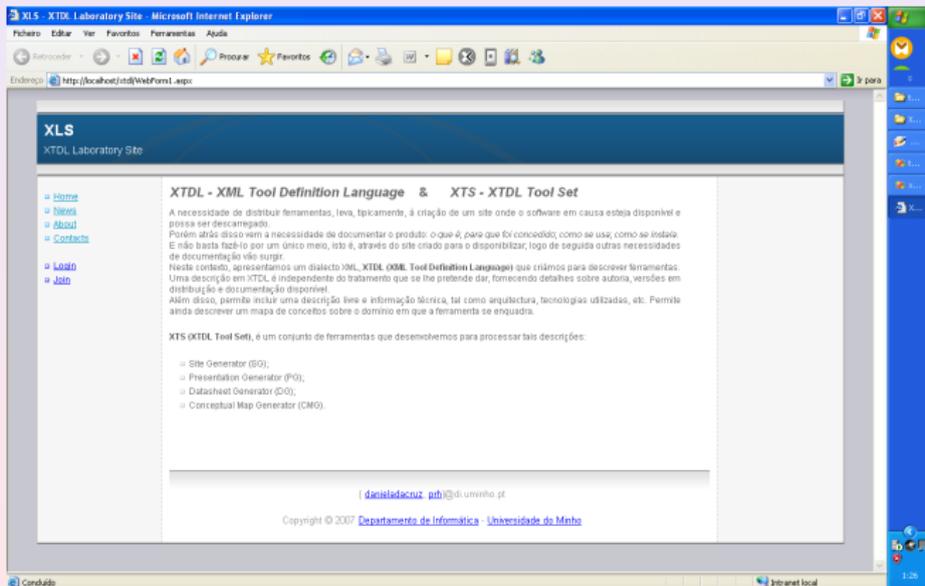


Figura: Página inicial do XLS

XLS — XTDL Laboratory Site

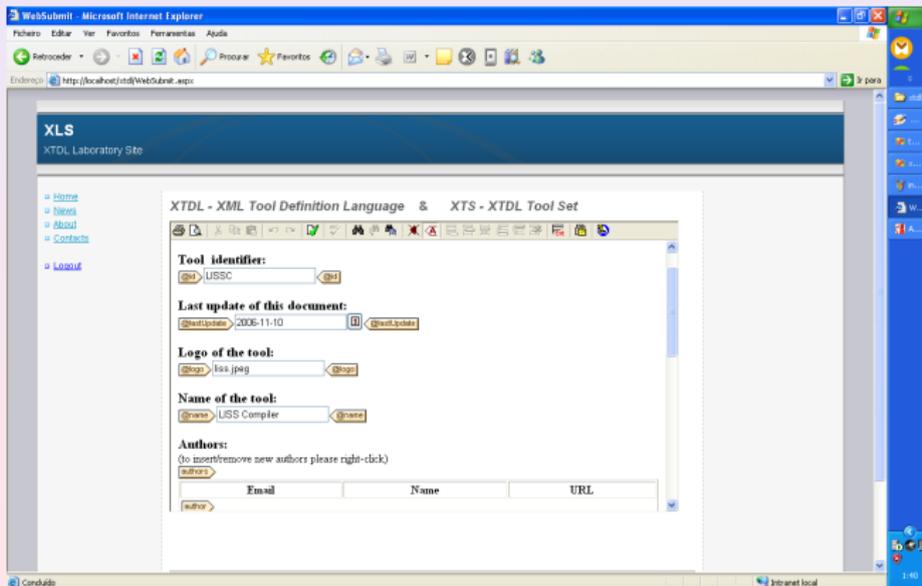


Figura: Edição online do documento XTDL

Conclusão

- Embora o sistema desenvolvido seja um contributo para a documentação de software, o nosso principal objectivo residiu na automatização muito pragmática do processo de disponibilizar e documentar ferramentas.
- Trabalho futuro: Gerador de artigos.