5
A Ferramenta LawGenerator

Este capítulo apresenta a *LawGenerator,* a ferramenta de geração automática de código XMLaw desenvolvida para este trabalho. Nas seções a seguir será explicado o que ela é, como usá-la, qual a sua estrutura e como ela foi feita.

5.1.
O que é a ferramenta

A *LawGenerator* é uma ferramenta de transformação automática de leis, definidas em LawML para a linguagem XMLaw. Ela recebe como entrada um arquivo XML, que contém um modelo de lei especificado em LawML, o nome e o local do arquivo XML de saída, e gera, no local especificado, um arquivo XML que contém o modelo XMLaw no formato definido pela gramática XMLaw (veja o Anexo A).

Trata-se de um transformador de modelos representados em XML, que permite ao desenvolvedor de leis a geração automática do código XMLaw da lei, após especificar a mesma em LawML com uma ferramenta CASE que permita gravar os modelos UML no formato XML. O arquivo que contém o código XMLaw será posteriormente interpretado pelo mediador M-Law, conforme já foi explicado nas seções 1.3 e 2.2 desta dissertação. Neste trabalho a especificação das leis em LawML será feita através da ferramenta Visual Paradigm for UML (Visual Paradigm, URL, 2007), que é uma ferramenta CASE consolidada e que atende à versão 2.0 da UML. Maiores informações sobre a ferramenta CASE Visual Paradigm, usada para a especificação das leis, podem ser vistas no Apêndice I.

A Figura 5.1 apresenta uma visão geral do processo descrito nos parágrafos anteriores.

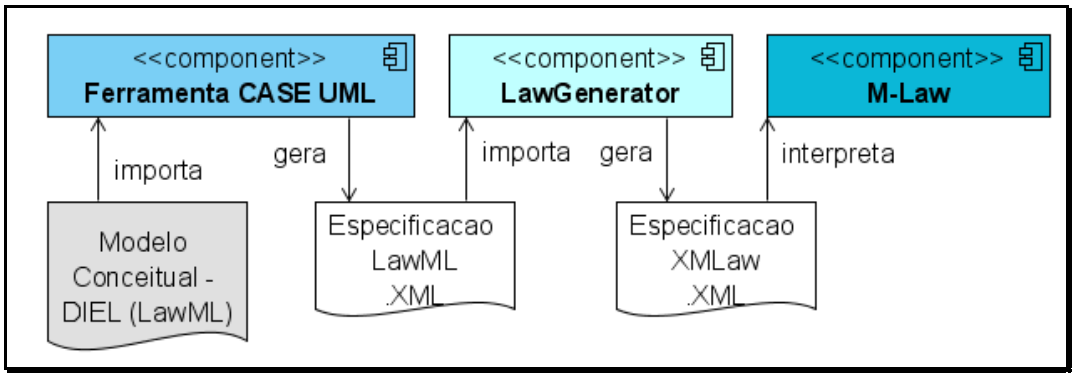

Figura 5-1 – Componentes da especificação de lei de interação

Por questões de escopo, a ferramenta LawGenerator não contempla a verificação completa de consistência dos modelos gerados. Essa tarefa ficará a cargo do desenvolvedor de leis, ao especificar a lei através de uma ferramenta CASE UML.

A *LawGenerator* também não realiza a engenharia reversa dos modelos gerados em XMLaw para o modelo LawML. No caso de necessidade de alteração de uma lei, o desenvolvedor de leis deverá fazer as alterações nos próprios diagramas produzidos com a ferramenta CASE escolhida.

5.2. Como usar a ferramenta

De posse do arquivo XML que contém o modelo da lei em LawML, o desenvolvedor de leis irá usar a ferramenta *LawGenerator* para realizar a transformação de modelos. Para tal, basta ele abrir o arquivo executável da aplicação, informar o caminho completo do arquivo XML que contém a especificação LawML, o caminho completo do novo arquivo XML que contém a especificação XMLaw a ser gerado e clicar no botão "Generate XMLaw specification file" (Figura 5-2).

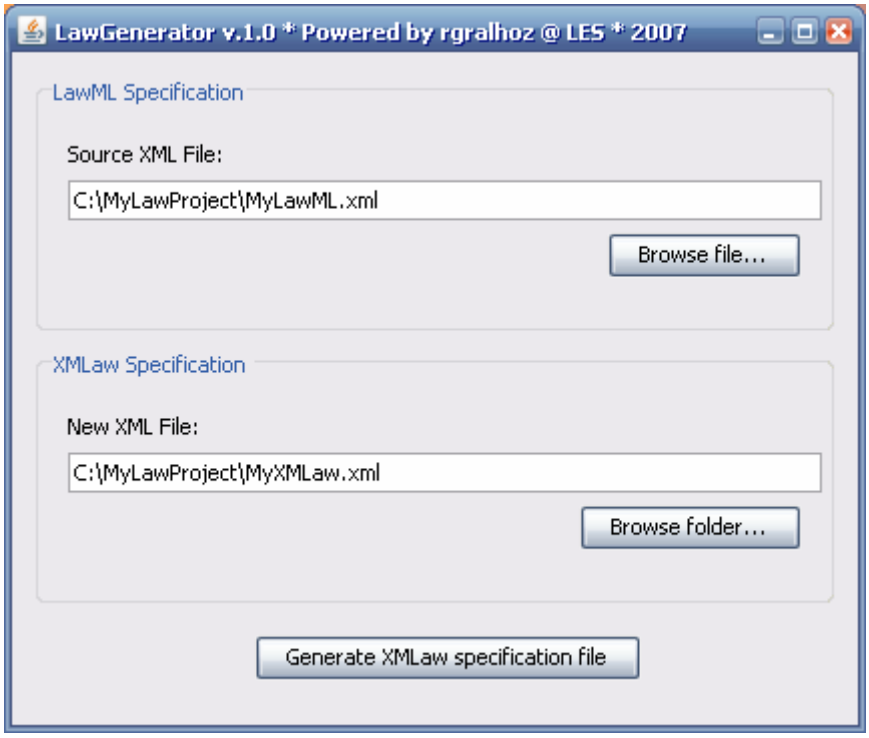

Figura 5-2 – Tela da aplicação *LawGenerator*

Ao solicitar a geração do arquivo com a especificação XMLaw da lei, a ferramenta irá executar a leitura do arquivo XML, a transformação do modelo LawML contido no arquivo origem em modelo XMLaw e a geração do arquivo XML que contém o último.

5.3.
Estrutura da ferramenta

A ferramenta *LawGenerator* pode ser dividida em quatro camadas: Camada de Visão, Camada de Controle, Camada de Modelo e Camada de Persistência (Figura 5-3).

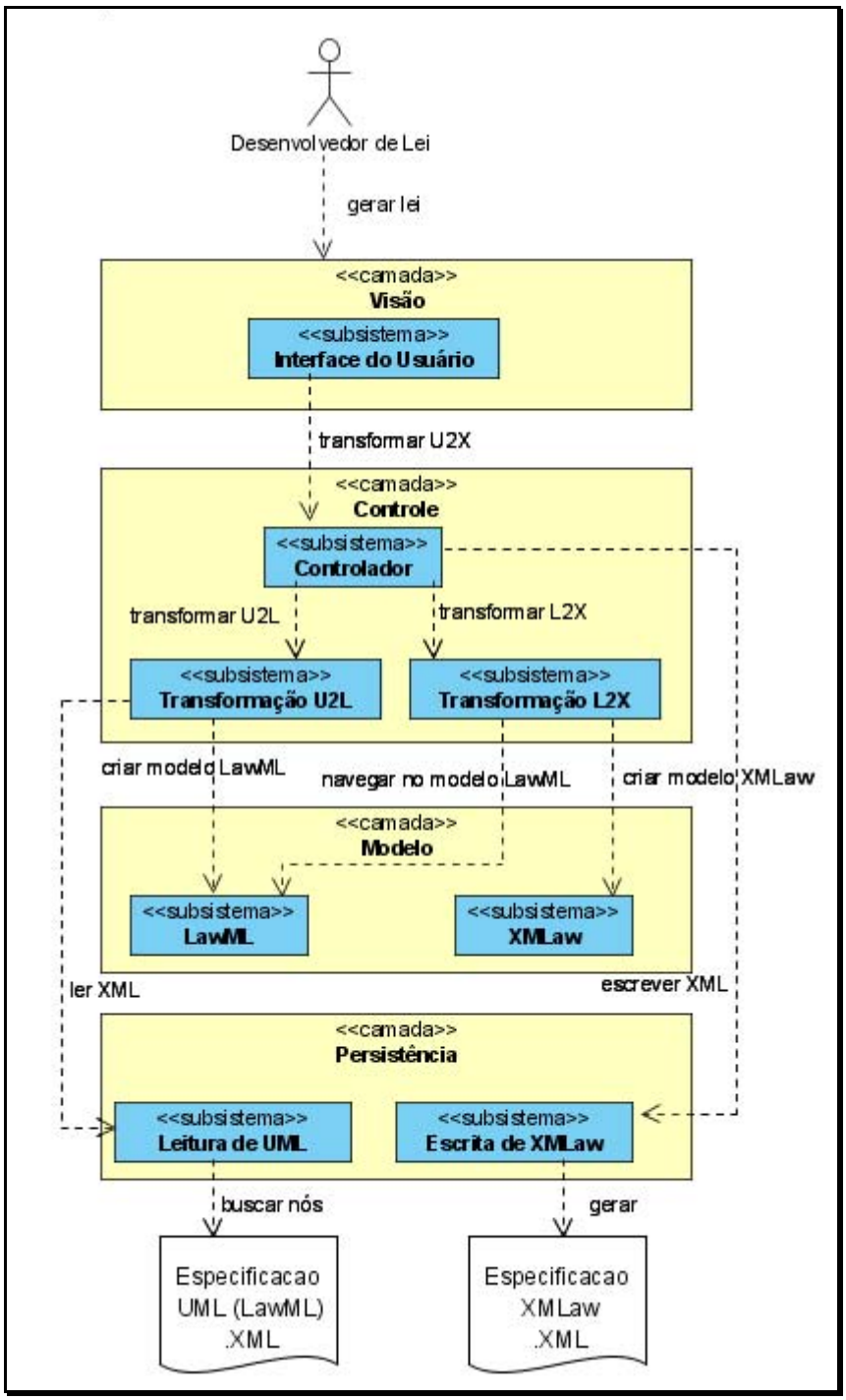

Figura 5-3 – Estrutura da *LawGenerator*

A **Camada de Visão** é formada pelo subsistema Interface do Usuário, que recebe a requisição do usuário pela geração da lei, junto com os parâmetros

necessários – os nomes completos dos arquivos de entrada e de saída. Essa requisição é feita a partir de um formulário, que pode ser visto na Figura 5-2.

A **Camada de Controle** é formada por três subsistemas: Controlador, Transformação U2L e Transformação L2X.

O subsistema Controlador, que implementa o padrão de projeto Comando, recebe a solicitação de transformação de modelos da Interface do Usuário e solicita a transformação do modelo UML do arquivo de entrada em modelo LawML ao subsistema Transformação U2L. Em seguida, solicita ao subsistema Transformação L2X a transformação do modelo LawML em um modelo XMLaw, de acordo com as regras de transformação propostas no Capítulo 4. Por fim, o subsistema Controlador solicita a escrita do XML de saída, com o modelo XMLaw gerado, ao subsistema Escrita de XMLaw, da **Camada de Persistência**.

O subsistema Transformação U2L, responsável pela transformação UML-LawML, solicita a leitura do XML de entrada ao subsistema Leitura de UML, da **Camada de Persistência** e, conforme busca elementos na Persistência, solicita a criação de elementos do modelo LawML ao subsistema LawML, da **Camada de Modelo**. Depois de concluída a transformação UML-LawML, ele retorna ao subsistema Controlador o modelo LawML criado.

O subsistema Transformação L2X, responsável pela transformação LawML-XMLaw, navega no modelo LawML criado e solicita a criação do modelo XMLaw ao subsistema XMLaw, da **Camada de Modelo**. Depois de concluída a transformação LawML-XMLaw, ele retorna ao subsistema Controlador o modelo XMLaw criado.

A **Camada de Modelo** é formada pelos subsistemas LawML e XMLaw, e é responsável pela manipulação dos elementos de modelo, assim como de seus relacionamentos e de suas propriedades.

A **Camada de Persistência** é formada pelo subsistema Leitura de UML, que busca os nós XML do arquivo de entrada com a especificação UML (LawML), e pelo subsistema Escrita de XMLaw, que gera o arquivo XML de saída com a especificação do modelo XMLaw.

5.4. Como foi feita a ferramenta

A ferramenta *LawGenerator* foi projetada, construída e testada de maneira incremental, com o uso da tecnologia Java (Java, URL, 2007). A divisão em camadas e subsistemas permite um maior desacoplamento, de maneira que os subsistemas possam ser reutilizados de maneira independente. O subsistema Transformação L2X, por exemplo, transforma um modelo LawML em um modelo XMLaw, independente de como o primeiro foi criado e de como o segundo vai ser usado.

O desenvolvimento foi feito com o importante auxílio de dois ambientes integrados de desenvolvimento (IDE's) de código aberto: Eclipse 3.2.1 (Eclipse, URL, 2007) e NetBeans 5.5.1 (NetBeans, URL, 2007).

NetBeans foi escolhido para a **Camada de Visão** por possuir um excelente construtor de Interface Gráfica de Usuário, chamado GUI Builder ou Projeto Matisse, que utiliza elementos das bibliotecas SWT (SWT, URL, 2007), AWT (AWT, URL, 2007) e Swing (Swing, URL, 2007), e que permitiu, de maneira simples e rápida, a criação da tela da aplicação (com formulário, botões etc.).

As outras camadas foram produzidas no ambiente Eclipse, por possuir uma melhor integração com outras tecnologias através de *frameworks* que são usados como *plugins* (ECLIPSE - HELP, URL, 2007).

A **Camada de Modelo** foi programada com o uso de um poderoso plugin do Eclipse baseado na MDA (MDA, URL, 2007) – o Eclipse Modeling Framework (EMF, URL, 2007), que permitiu a geração automática de código Java dos modelos através da especificação UML (no caso do modelo LawML) ou da especificação XSD (XML Schema, URL, 2007), no caso do modelo XMLaw (veja o Anexo A). O código gerado para o modelo XMLaw permitiu a geração quase completa do subsistema Escrita de XMLaw, da **Camada de Persistência**, capaz de persistir os modelos na sintaxe correta de XMLaw, em arquivos XML (XML, URL, 2007).

Para o subsistema Leitura de UML, da **Camada de Persistência**, foi utilizada a API Xalan-Java (Xalan, URL, 2007), baseada na tecnologia XPath (XPath, URL, 2007), uma linguagem que permite a formulação de expressões de pesquisas por nós e/ou atributos em arquivos XML.

Para garantir uma correta transformação de modelos e geração do arquivo no formato XMLaw, foram elaborados cenários de teste que abrangem todos os elementos, atributos e relacionamentos de LawML e, portanto, de XMLaw. Esses cenários foram especificados na ferramenta CASE e exportados para arquivos XML. Classes de testes baseadas no framework JUnit (JUnit, URL, 2007) garantiram a execução automática da leitura desses arquivos e da geração de respectivos arquivos de saída, que foram conferidos conforme a gramática de XMLaw.