# 3
# Metamodelo

Um modelo é uma representação abstrata de um sistema e um metamodelo é uma descrição abstrata de um modelo. Ele consiste em um conjunto de conceitos e relacionamentos dentro de um determinado domínio [Gholizadeh, 2010].

Nossa abordagem sobre metamodelo consiste em um conjunto de regras léxicas e sintáticas da linguagem. Elas fazem o controle sobre os seus elementos, evitando que o usuário utilize itens lexicais que não pertençam ao vocabulário da linguagem.

As regras léxicas descrevem as combinações válidas para a formação dos elementos da linguagem. Elas definem os elementos que podem ser usados na elaboração do modelo referente a uma linguagem específica. As regras sintáticas descrevem como os elementos podem ser combinados para formar conjuntos válidos da linguagem.

Para definirmos as regras léxicas e sintáticas do metamodelo foi elaborada uma linguagem de metamodelagem em XML estruturada através de um documento XSD [XML Schema, 2010] apresentado no Apêndice C dessa dissertação.

Obedecendo essa estrutura definida para a linguagem, podemos dividir a sua definição em duas partes: os elementos da linguagem e as suas regras.



## 3.1
## Elementos

Na estrutura do arquivo XML, a definição dos elementos do modelo é dividida em duas partes: nós (*nodes*) e arestas (*edges*). Na área chamada *nodes* é especificado cada nó da linguagem e na área chamada *edges* é especificada cada aresta da linguagem. Cada uma dessas áreas possui suas próprias características de formação que serão explicadas a seguir.

### 3.1.1
### Nós

Dentro da marcação “nodes” do XML do metamodelo é encontrada a lista de elementos (*node*) da linguagem. Cada elemento é definido através de uma

série de atributos que ditam as suas características e relacionamentos. Esses atributos são definidos a seguir:

O atributo **ID** define o nome do elemento nó. Este nome é utilizado na busca do arquivo de imagem PNG do ícone e do elemento em arquivo SHAPE que deverão estar na pasta do metamodelo. Os dois arquivos devem ter o mesmo nome que o atributo ID. O identificador também serve para fazer a ligação entre os elementos e as suas regras. Todas as verificações realizadas em cima do elemento usam esse atributo como referência.

O atributo **WIDTH** define a largura *default* do elemento nó, em *pixels*. Toda vez que se adicionar um elemento novo da biblioteca, ele virá com essa largura.

O atributo **HEIGHT** define a altura *default* do elemento nó, em *pixels*. Toda vez que se adicionar um novo elemento ao modelo vindo da biblioteca, ele virá com essa altura.

O atributo **STYLENODE** define a opção "*shape*" ou "*image*". Caso seja atribuída a opção *shape*, o programa entenderá que existe na pasta do metamodelo um arquivo .SHAPE e .PNG com o mesmo ID (nome do elemento). Esses arquivos devem ser criados pelo programa Dia. Caso seja atribuída a opção *image*, obrigatoriamente deverá ser adicionado o atributo *imagename* seguido do nome da imagem que representará o elemento. Mais detalhes na explicação seguinte do atributo *imagename*.

O atributo **IMAGENAME** define o nome da imagem contida na pasta referente a esse elemento. Essa imagem não é vetorial e pode ser no formato .PNG, .JPG e .GIF. Ela ficará no lugar do .SHAPE e será a sua representação após o elemento ser inserido no canvas.

O atributo **VALUE** define o valor do elemento que é o texto *default* apresentado dentro ou abaixo do elemento inserido no diagrama.

O atributo **FONTSIZE** define o tamanho da fonte do texto do elemento.

O atributo **FONTCOLOR** define a cor da fonte do texto do elemento, no formato #RRGGBB.

O atributo **ALIGNTEXT** neste atributo colocará o alinhamento do texto em relação ao elemento. Existem três opções: *left*, *center* e *right*.

O atributo **DASHED** é utilizado quando o atributo **STYLENODE** é definido como shape para um elemento nó. Esse arquivo .shape é carregado da pasta do

metamodelo criado no programa Dia. Porém, quando esse elemento é desenhado no Dia [Dia, 2011] utilizando uma linha tracejada ele não é representado corretamente no editor de metamodelos. Para que o elemento apareça com os seus contornos em tracejado é preciso ativar este atributo. Nele, é possível atribuir o valor true e false: true para linha tracejada e false para linha contínua.

### 3.1.2
### Arestas

Dentro da marcação “edges” do XML do metamodelo é encontrada a lista de elementos (*edge*) da linguagem. Cada elemento é definido através de uma série de atributos que ditam as suas características e relacionamentos. Esses atributos são definidos como:

O atributo **ID** define o nome do elemento aresta. Diferente do atributo ID para nó, este não faz referência a um arquivo da pasta do metamodelo. Isto porque a aresta não necessita de um arquivo gráfico para a sua visualização. O identificador também serve para fazer a ligação entre os elementos e as suas regras. Todas as verificações realizadas em cima do elemento usam esse atributo como referência.

O atributo **WIDTH** define a largura *default* do elemento aresta, em pixels. Toda vez que se adicionar um elemento novo da biblioteca, ele virá com essa largura.

O atributo **HEIGHT** define a altura *default* do elemento aresta, em pixels. Toda vez que se adicionar um elemento novo da biblioteca, ele virá com essa altura.

O atributo **IMAGEICON** define o nome da imagem contida na pasta do metamodelo. Essa imagem não é vetorial e deve ser no formato .PNG. Ela será utilizada como ícone na biblioteca que referencia a aresta.

O atributo **VALUE** define o valor do elemento. Valor este que é o texto apresentado por *default* dentro ou abaixo do elemento inserido no diagrama. Esse atributo é muito útil para a criação de arestas da linguagem CTT, onde os caracteres inseridos alteram a semântica da aresta.

O atributo **FONTSIZE** define o tamanho da fonte do texto. Texto este inserido no atributo *value* explicado anteriormente.

O atributo **FONTCOLOR** define a cor da fonte do texto do elemento, no formato #RRGGBB.

O atributo **ALIGNTEXT** define o alinhamento do texto em relação ao elemento. Texto este inserido no atributo *value* explicado acima. Existem três opções *left*, *center* e *right*.

Dentro da definição da aresta foi criada uma divisão espacial de suas características nomeadas da seguinte forma: *start arrow*, *body* e *end arrow*. As áreas *start arrow* e *end arrow* são semelhantes trocando apenas de posicionamento. Já a área *body* é diferente das outras duas por se tratar do corpo da aresta. Segue uma breve explicação dessa formação:

**StartArrow**

O atributo **TYPE** define o formato da seta de inicio da aresta. Existem os seguintes tipos de seta: *none, classic, diamond, oval, open, block*. A Figura 14 exemplifica esses tipos.

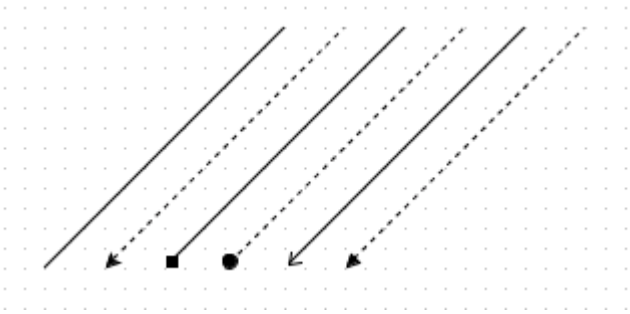

**Figura 14 - Os tipos de seta de início de aresta disponíveis: none, classic, diamond, oval, open, block.**

**Body**

O atributo **TYPE** define o corpo da aresta do metamodelo. Existem os seguintes tipos disponíveis: *straight, horizontal, vertical, entity, arrow*. A Figura 15 exemplifica esses tipos.

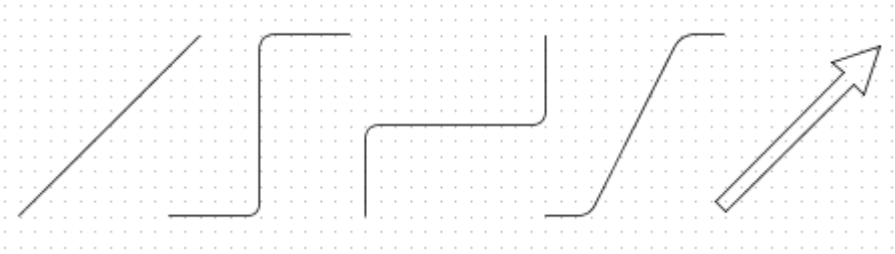

**Figura 15 - Os diferentes tipos de arestas na sequência: straight, horizontal, vertical, entity, arrow.**

O atributo **COLOR** define a cor da aresta. Ele deve ser preenchido com o valor em hexadecimal no formato #RRGGBB. A Figura 16 exemplifica essa opção.

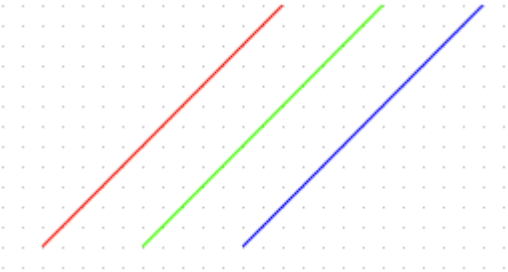

**Figura 16 - Controle sobre as cores das arestas.**

O atributo **STROKEWIDTH** define a espessura de linha de uma aresta. A espessura da linha varia de acordo com o valor do número inteiro atribuído. A Figura 17 exemplifica essa opção.

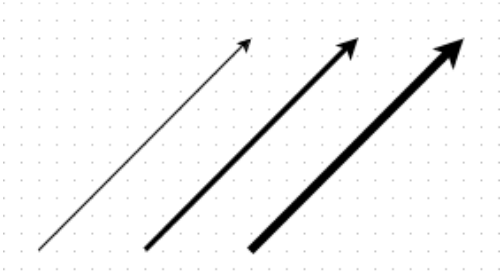

**Figura 17 - Controle de espessura da aresta.**

Para algumas arestas de um metamodelo será necessária a utilização de uma linha tracejada, como por exemplo: a fala de recuperação de ruptura do

modelo MoLIC. Para estes casos foi desenvolvido o atributo **DASHED** que informa se a linha é tracejada ou contínua. Para isso, é necessário atribuir os valores *true* para linha tracejada ou *false* para linha contínua. A Figura 18 exemplifica essa opção.

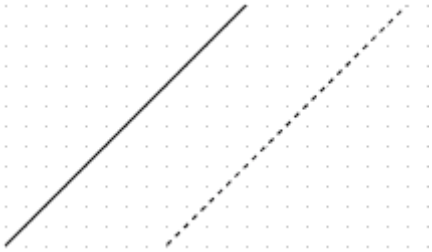

**Figura 18 - Aresta com linha tracejada.**

### EndArrow

O atributo **TYPE** é parecido com o existente no StartArrow, porém ele atua do lado oposto. Ele define o formato da seta final da aresta. Existem os seguintes tipos: *none, classic, diamond, oval, open, block*. A Figura 19 exemplifica esses tipos.

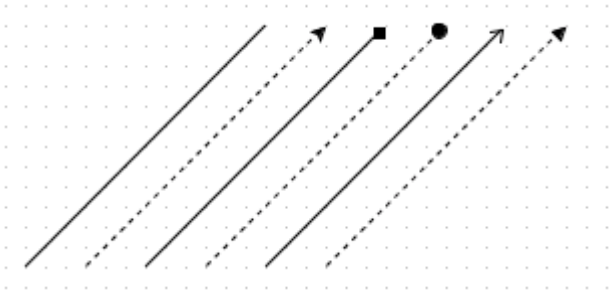

**Figura 19 - Os tipos de seta final de aresta disponíveis: none, classic, diamond, oval, open, block.**

## 3.2
## Regras

As regras descrevem a sintaxe da linguagem dentro de uma gramática, que começam em um símbolo inicial, derivando diversas sentenças válidas da linguagem [Minas, 2006]. Uma regra define o relacionamento entre um elemento e seus correlacionados, com a propriedade de inclusão ou exclusão.

Dentro da marcação das regras, existe o atributo **TYPE** que define o tipo de diagrama que utilizará as regras. Nele, é possível escolher as opções *graph* para diagramas que possuem uma estrutura de grafo ou *tree* para diagramas que possuem uma estrutura de árvore. Para as três linguagens de IHC abordadas nessa dissertação, é utilizada a opção de grafo. Isto porque, mesmo a linguagem CTT que possui uma estrutura hierárquica pode ser representada como grafo, já que os nós de mesma altura possuem ligações entre si, fugindo da definição estrutural de uma árvore.

No atributo **WARNINGICON** é configurada a imagem que indica um erro no grafo. Atualmente ele apresenta uma figura de alerta com as opções *true* para piscar e *false* para não piscar. Futuramente esse atributo poderá ser usado para escolher a imagem que representa o erro.

Na área onde é definida uma regra, é utilizada a marcação **RULE**. Nela, alguns atributos podem ser configurados para especificar a regra de um elemento, como o atributo **ELEMENT,** que é usado para definir o nome do elemento que vai ser analisado na regra. Esse nome tem que ser o mesmo colocado na parte dos elementos para haver uma ligação entre as duas informações.

Para diferenciar o uso de regras entre nós e arestas foi criado o atributo **TYPE**, que define o tipo de elemento para o qual a regra será usada: nó (*node*) ou aresta (*edge*).

Interno ao marcador de regra, existe o marcador **OPTION** que permite a inclusão ou exclusão de uma lista de elementos. Por isso a necessidade do atributo **NEGATIVE**, que define se a regra será de adição (valor *false*) ou exclusão (valor *true*). Caso seja de adição, todos os elementos relacionados serão aceitos pela regra. Caso seja de exclusão, todos os elementos relacionados serão negados pela regra.

Dentro do **OPTION** foi definido o marcador **SOURCE** que agrupa todos os nós iniciais de uma aresta. Esse grupo pode ser uma listagem de adição ou exclusão dependendo do que foi decidido no atributo **NEGATIVE** explicado anteriormente.

Cada elemento desse grupo é chamado de **ELEMENT**. Ele se refere a um nó ou aresta definido anteriormente no metamodelo. Ele possui um atributo **ID** que é o identificador do elemento que deve ter o mesmo nome do elemento nó ou aresta definido no marcador **ELEMENTS**.

A outra opção dentro do marcador **OPTION** é o **TARGET**. Ele agrupa todos os nós finais de uma aresta. Esse grupo poder ser uma listagem de adição ou exclusão semelhante ao marcador que já foi explicado anteriormente no marcador **SOURCE**. Cada elemento inserido dentro de **TARGET** possui o marcador **ELEMENT** que se refere a um nó ou uma aresta previamente definida.

Além dos marcadores **SOURCE** e **TARGET** utilizados dentro do marcador **OPTION**, existe uma terceira opção que é o marcador **CHILDREN**. Nele, todos os elementos adicionados ao marcador são considerados filhos permitidos do nó aninhado. Para a listagem dos elementos é utilizado o marcador **ELEMENT** com o atributo **ID**, como já explicado anteriormente.