

## 6 Conclusão

Esta tese teve como objetivo principal o estudo da aplicação de transformações para manter a rastreabilidade de um sistema de software. Esta abordagem permite a captura automática das informações das ligações de rastreamento de Evolui-Para e de Justifica, a utilização de vários níveis de granularidade e sofisticação das tarefas de rastreamento, a disponibilização de mecanismos de inferência sobre as informações de rastreamento capturadas, a utilização de representações formais e informais para as informações de rastreamento, sendo ainda totalmente aderente ao processo de desenvolvimento utilizado pelo desenvolvedor.

Na implementação da abordagem proposta foi utilizada uma arquitetura baseada no sistema transformacional implementado pela Máquina Draco-PUC, estendido de maneira a poder tratar aplicações XML, procurando automatizar a maior parte possível das tarefas de rastreamento.

O mecanismo de rastreamento especificado a partir deste estudo foi utilizado para apoiar a evolução automática de cenários de um sistema.

### 6.1. Contribuições da Tese

Nesta tese realizamos a especificação e implementação de um mecanismo de rastreamento baseado na tecnologia transformacional. Este mecanismo permite que sejam incorporadas informações sobre a semântica associada a cada ligação, adicionadas as informações de rastreamento capturadas e armazenadas pelos sistemas de rastreamento usuais. A obtenção das informações de rastreamento relacionadas com a evolução de um artefato original para um novo artefato é basicamente um problema resolvido, realizado normalmente através da utilização de matrizes de rastreamento. Nosso diferencial reside na disponibilização das informações que explicam de que maneira se deu esta evolução, ou seja, além do estado inicial e final de um artefato, são capturadas as informações sobre como foi realizada esta evolução. A nomeação de qual a operação realizada pelo

desenvolvedor sobre os artefatos agrega um conhecimento semântico sobre a transformação de evolução do artefato. No caso específico dos cenários, a indicação das operações realizadas, em conjunto com os relacionamentos existentes entre os cenários da configuração inicial, permite identificar os cenários da nova configuração que serão sofrerão o impacto pelas operações realizadas. A existência de uma série de heurísticas pré-definidas associadas a cada operação, que guiam o desenvolvedor na escolha de qual operação usar na evolução dos artefatos, permitem ainda que, a partir do momento em que uma determinada operação foi identificada, seja inferido o porque de sua aplicação no contexto atual, fornecendo desta maneira a justificativa para sua aplicação.

No contexto do desenvolvimento de software utilizado nesta tese, cenários são utilizados ao longo de todo o ciclo de vida do sistema, desde os requisitos até a codificação. Desta maneira, cenários são criados na fase de requisitos e sobre eles são aplicadas transformações até a obtenção dos cenários de codificação. Nosso mecanismo captura as transformações realizadas sobre cada cenário ao longo de todo seu ciclo de vida, permitindo que se identifique qual a operação aplicada, seus resultados e impactos sobre os demais cenários.

O uso do mecanismo de rastreamento baseado na tecnologia transformacional permite minimizar o problema da falta de atualização das informações de rastreamento, aumentando o valor destas informações para o desenvolvedor; sistematizar o processo de rastreamento; e diminuir o custo de obtenção, atualização e validação das informações de rastreamento.

A principal contribuição desta tese reside, portanto, na constatação da viabilidade da utilização da tecnologia transformacional no rastreamento dos artefatos do processo de software de uma maneira geral, e mais especificamente no rastreamento de requisitos descritos na forma de cenários. Esta abordagem transformacional para o rastreamento de artefatos abre novas perspectivas para a automatização do processo de captura das informações de rastreamento. Alguns artigos publicados [31][32] já abordam parcialmente os principais conceitos relacionados com a utilização da tecnologia transformacional no rastreamento de artefatos.

O mecanismo proposto pode ser utilizado no rastreamento de outros tipos de artefatos descritos em XML. O requisito necessário a esta utilização é a existência

de uma taxonomia de evolução do artefato, na forma de um conjunto de operações que podem ser aplicadas sobre o artefato durante o processo de desenvolvimento.

A pesar de não estarem diretamente relacionadas com o objetivo principal desta tese, foram realizados uma série de trabalhos relacionados à construção da infra-estrutura, à implementação de ferramentas de suporte e à utilização da Máquina Draco-PUC, indispensáveis para que se pudesse implementar e validar o mecanismo de rastreamento proposto. Devido a estes motivos e ao caráter inovador dos mesmos, consideramos ainda como contribuições desta tese:

- A integração entre as Redes de Domínio Draco-PUC e XSL, permitindo que qualquer aplicação XML possa ser tratada como um domínio Draco-PUC e, desta maneira, favorecendo a disseminação do uso da Máquina Draco-PUC;
- uma ferramenta para visualização da aplicação das transformações. Esta ferramenta facilita o entendimento dos conceitos de sistemas transformacionais, a depuração do código das transformações e o próprio projeto das transformações, contribuindo para uma maior facilidade de uso da Máquina Draco-PUC;
- a especificação de uma notação para descrever transformações que facilita o seu entendimento e a comunicação entre desenvolvedores;
- a caracterização e implementação de duas formas de uso da Máquina Draco-PUC relativas à geração e aplicação de transformadores sem a necessidade de realizar transformações para linguagens intermediárias

## **6.2. Comparação com Propostas existentes**

A principal característica que difere o mecanismo de rastreamento proposto nesta tese das ferramentas existentes reside na maneira pela qual as ligações relacionadas ao processo (Evolui-Para e Justifica) são capturadas, armazenadas e manipuladas. Enquanto que na maior parte dos mecanismos existentes estas ligações são armazenadas na forma de relacionamentos entre o artefato original e o novo artefato, o nosso mecanismo armazena a própria ação executada pelo desenvolvedor na forma de transformações que podem ser aplicadas ao artefato original para que se obtenha o novo artefato. O armazenamento das transformações carrega um maior valor semântico sobre as intenções, contexto,

restrições e conseqüências da realização das ações correspondentes pelo desenvolvedor na evolução dos artefatos.

Egyed [2] apresenta uma proposta de rastreamento dirigida por cenários, onde as informações são capturadas a partir da observação de cenários de teste sendo executados pelo sistema. Esta proposta necessita que tanto os modelos quanto uma versão executável do sistema estejam disponíveis. A informação gerada mostra apenas qual os componentes do sistema que são utilizados na execução dos cenários de teste (ligação Satisfaz). Nossa proposta pode ser utilizada em qualquer fase do desenvolvimento pois não depende da análise dinâmica do sistema, sendo ainda utilizada para tratar as ligações de Evolui-Para e Justifica.

Pinheiro e Goguem [78] utilizam uma abordagem formal para manter a rastreabilidade de um sistema na forma de relacionamentos entre objetos. Estas informações precisam ser informadas manualmente pelo desenvolvedor. Nossa proposta permite que grande parte da informação seja capturada automaticamente.

Uma abordagem que apresenta algumas similaridades com a nossa é apresentada por Antonioli et. al. [79] que calcula a diferença entre versões para auxiliar o desenvolvedor a tratar inconsistências entre as versões, apontando para regiões do código onde as diferenças estão concentradas. É utilizada uma representação intermediária, a Abstract Object Language (AOL), e computado o delta analisando este código. A principal diferença em relação a nossa proposta reside nesta característica, pois utilizamos diretamente as árvores de sintaxe abstratas e desta maneira não perdemos nenhuma informação ao fazermos a transformação entre o programa original e sua representação. Outra diferença é que, enquanto a proposta de Antonioli trata somente as informações existentes no Diagrama de Classes, nossa proposta pode ser aplicada em qualquer tipo de modelo. O uso das informações capturadas também difere, pois nosso objetivo é utilizá-las para manter os diversos tipos de ligações e não somente mostrar inconsistências entre versões.

Uma proposta que também utiliza transformações é apresentada por Baxter [43] [80], que propõe que os refinamentos executados durante o desenvolvimento devem ser armazenadas na forma de transformações. Estas transformações poderiam então ser utilizadas no processo de engenharia reversa de outros sistemas a partir do código fonte, com o objetivo de obter os modelos do sistema.

A implementação desta proposta depende da existência de uma especificação formal do sistema sobre a qual são aplicadas transformações para derivar o código do sistema, obrigando que o desenvolvedor utilize métodos formais no desenvolvimento. Nossa proposta permite a utilização de qualquer método ou ferramenta já conhecidos pelos desenvolvedores, tendo em vista que a geração das transformações é realizada diretamente sobre os resultados das ações realizadas pelo desenvolvedor. Finalizando, o enfoque de Baxter & Mehlich é na engenharia reversa de sistemas não tratando a questão da rastreabilidade.

O sistema GASE proposto por Holt & Pak [81] permite realizar a visualização das diferenças de arquitetura entre versões de um sistema. O computo das diferenças é realizado através da procura de trechos do código que caracterizam a arquitetura, tais como chamadas de funções e declarações *include* em C++. Nosso mecanismo de identificação de diferenças pode ser aplicado para identificar diferenças relacionadas a qualquer aspecto descrito nos artefatos além de disponibilizar uma técnica de reconhecimento de planos que permite a obtenção de informações em um nível de abstração mais alto.

### **6.3. Trabalhos Futuros**

Nossas sugestões de trabalhos futuros são relacionadas ao refinamento e extensão do mecanismo de rastreamento e a otimização de sua implementação.

Quanto ao refinamento e extensão do mecanismo, podem ser realizados estudos que permitam responder as seguintes questões levantadas nesta tese e que permanecem em aberto:

- Qual deve ser a frequência de aplicação do processo de captura? Esta frequência deve ser contextualizada a cada processo de desenvolvimento? Quais são os parâmetros necessários para definir esta frequência?

- Como tratar a questão da evolução das transformações de rastreamento? Qual seu relacionamento com a evolução dos artefatos correspondentes?

- É viável armazenar apenas a versão inicial e as transformações de cada artefato? Devemos armazenar versões intermediárias dos artefatos para verificações e otimizações?

Os trabalhos de melhoria da implementação do mecanismo foram apresentados na seção 5.4 e tratam da otimização da técnica de reconhecimento de

planos e do refinamento das heurísticas utilizadas para reconhecer relacionamentos e operações entre cenários.

Finalmente, sugerimos a realização de estudos sobre a evolução de outros artefatos, tais como os Diagramas de Classes ou cartões de CRC, com o objetivo de definir uma taxonomia para sua evolução, nos moldes da taxonomia de cenários utilizada nesta tese. Desta maneira tornar-se-á possível a instanciação do mecanismo de rastreamento baseado em transformações para outros tipos de artefatos.