

7

Conclusões e Trabalhos Futuros**Resumo**

Este capítulo apresenta as conclusões obtidas a partir deste trabalho e especifica algumas contribuições consideradas relevantes, descrevendo como foram alcançadas. Finalmente, são ilustradas algumas orientações para trabalhos futuros decorrentes desta dissertação.

Na introdução desta dissertação foram levantadas três possíveis causas para insucesso da reutilização de software ou do excesso de adaptações necessárias ao desenvolvimento de soluções utilizando componentes de software. Foram elas: (1) problemas no projeto das unidades e das soluções, (2) problemas com relação à documentação, e (3) problemas com relação à utilização de técnicas inadequadas para o auxílio à reutilização de software.

Com relação aos problemas apresentados no projeto de soluções e unidades de software, associar diferentes conceitos na elaboração de uma única unidade implica em sucessivas alterações geradas pela evolução de poucos requisitos. Na tentativa de contornar este problema, foi proposta a separação e estruturação dos conceitos de coordenação, composição e artefatos de software em camadas arquiteturais de ACCA.

Com o objetivo de utilizar a documentação para auxiliar o entendimento, o aprendizado, a elaboração e a geração de uma solução foram utilizadas especificações prescritivas na camada de coordenação de ACCA. Escolheu-se a camada de coordenação, pois se acredita que ela apresente o conceito mais instável e mais sujeito a alterações.

Com relação a técnicas de reutilização, abordagens gerativas (Czarnecki & Eisenecker, 2000) são caminhos viáveis para auxiliar o processo de geração de soluções a partir de especificações geradas ao longo do processo de desenvolvimento. Neste trabalho, esta abordagem pode ser vista com a aplicação da proposta de (Gouveia et al., 2001) para a concretização da camada de coordenação.

Ainda com relação a técnicas de reutilização, foi proposto um *framework* orientado a objeto para a realização da camada de composição. *Framework* é uma técnica de reutilização utilizada em domínios de aplicação bem definidos (Johnson, 1997; Fayad, 1999), como foi o caso de composição de software. Ao elaborar uma solução com pontos flexíveis, é possível atender a um conjunto de peculiaridades necessárias a geração de diversas aplicações. Ao definir uma estrutura comum, é possível reutilizá-la em diferentes instâncias de aplicações, diluindo o seu custo de desenvolvimento em toda a família de aplicações.

7.1.

Lista de Contribuições

Arquitetura para a Coordenação e a Composição de Artefatos de Software: ACCA é uma proposta para a separação e a estruturação dos conceitos de coordenação, composição e componentes de software. Ao estruturar o desenvolvimento da solução com ACCA, questões específicas de conceitos, como coordenação ou composição de componentes, são tratadas em camadas isoladas, evitando assim que ocorram modificações intrusivas nos componentes ou em construções de outras camadas.

Framework para a Composição de Artefatos de Software: neste trabalho foi proposto um *framework* para a composição de artefatos de software, utilizado na realização da camada de composição de ACCA. Independentemente de ACCA, este *framework* pode ser utilizado para auxiliar a integração de diferentes tecnologias de componentes em uma solução.

Processo de Reificação e Desenvolvimento de Software a partir de Componentes: a partir de ACCA, foi possível elaborar um processo de desenvolvimento de software, que com o auxílio da estrutura proposta, organiza o conjunto de fases relevantes para a geração de uma solução a partir de componentes de software. Além disto, foi apresentado o processo de reificação de ACCA, útil para orientar o processo de concretização da arquitetura.

7.2.

Trabalhos Futuros

Outras técnicas de reutilização e geração de soluções (Krueger, 1992; Czarnecki & Eisenecker, 2000) poderiam ser avaliadas como alternativas de auxílio à realização dos conceitos de ACCA.

Com relação à utilização de outros catálogos de componentes, deve ser feita uma avaliação mais profunda de alternativas que poderiam atender melhor os requisitos levantados neste trabalho.

Para contribuir com o amadurecimento e a evolução de ACCA, outros estudos de caso mais complexos poderiam ser elaborados e desenvolvidos. No entanto, estes estudos são dependentes de problemas mais concretos e de interações com empresas ou a colaboração com outros grupos de pesquisa.

A partir do *framework* de composição, em um futuro breve, será possível especificar processos de composição a partir de uma linguagem de alto nível específica para a composição. Esta alteração visa facilitar a implementação, configuração e a realização das construções da camada de composição de ACCA. Com esta evolução, será possível promover a realização de mapeamentos entre a estrutura do *framework* e a linguagem de especificação de composições, permitindo que composições sejam instanciadas automaticamente a partir de um conjunto de regras de composição.

Visando um aumento de produtividade, pode-se desenvolver um ambiente de desenvolvimento de software que organize e facilite o desenvolvimento de soluções baseadas em componentes de software utilizando para isto diferentes perspectivas. Nesta abordagem, cada perspectiva corresponderia a uma camada proposta em ACCA. Alguns experimentos já foram feitos na plataforma Eclipse (Eclipse, 2003), e foi possível observar a viabilidade desta iniciativa.

A partir das experiências obtidas por meio desta dissertação de mestrado, será possível avaliar a possibilidade de identificar e utilizar diferentes conceitos de desenvolvimento de software em uma ontologia para estruturar a geração de soluções. Possivelmente, a elaboração desta ontologia se basearia em iniciativas como *Model Driven Architecture* (MDA, 2003), utilizando para a sua realização provavelmente abordagens gerativas (Czarnecki & Eisenecker, 2000) para auxiliar a realização dos conceitos identificados. A princípio, a organização dos conceitos em uma ontologia seria feita de acordo com o grau de evolução, ou instabilidade, que cada conceito geralmente apresenta ao ser utilizado. Entende-se por instabilidade, um fator⁴ que indica a probabilidade de que determinada característica sofra alterações, isto é, evolua ao longo de seu ciclo de vida.

⁴ Fator neste caso pode ser entendido como alguma métrica.