

7 Conclusões

Ao longo desta tese, procurou-se analisar e propor soluções para algumas questões relacionadas a especificação, projeto e implementação de mecanismos de composição e de adaptação de software em sistemas de comunicação, em particular em redes programáveis. O trabalho, apresentado em linhas gerais no Capítulo 1, aplica conceitos oriundos de arquitetura de software e de desenvolvimento baseado em componentes de modo integrado. Com isso, consegue-se prover a projetistas e programadores de serviços em redes programáveis abstrações de alto nível, encorajar o reuso e dar suporte (nesta tese, explorado de modo ainda incipiente) à aplicação incremental de formalismos.

No Capítulo 2, estabeleceu-se uma terminologia de engenharia de serviços a partir da qual os cenários possíveis de aplicação da abordagem proposta foram delineados. Em particular, esse capítulo apresentou um modelo de composição de serviços (Colcher, 1999) que serviu como referência para essa abordagem. A carência, no modelo SCM, de abstrações para a representação da alocação de recursos a atividades de computação e comunicação demandou extensões ao mesmo, bem como uma “releitura” de algumas de suas abstrações. A introdução no modelo SCM da noção de ‘tarefa’ como contêiner de recursos computacionais e o tratamento do conceito pré-existente de ‘provedor’ como contêiner de recursos de comunicação são contribuições secundárias desta tese.

Em seguida, o Capítulo 3 apresentou a estrutura da linguagem de especificação LindaX, em que é centrado o trabalho. A linguagem LindaX constitui-se na principal contribuição desta tese. Seu principal aspecto inovador é o uso combinado de características de ADLs – em particular, a noção de estilos – com um arcabouço sintático único para DSLs com semânticas distintas. Nessa combinação, é fundamental o papel de ferramentas extensoras que detêm o conhecimento semântico de cada DSL. O fato de LindaX ter sido definida sobre esquemas XML lhe concedeu outra característica que, se não é um aspecto inovador, só há pouco tempo começou a ser explorada em arquitetura de software:

a extensibilidade. Em LindaX, os principais alvos de extensões são a sua integração com diferentes formalismos e a sua aplicação em plataformas diversas de programação baseada em componentes de software.

O Capítulo 4 foi dedicado à demonstração da aplicação da linguagem LindaX em duas funções distintas de sistemas de comunicação – encaminhamento de dados e provisão de QoS. Duas DSLs foram definidas para especificar essas funções – LindaRouter e LindaQoS. Embora o objetivo desse capítulo tenha sido de fato demonstrar a exequibilidade da abordagem proposta, as DSLs citadas constituem-se também em duas outras contribuições secundárias. Vale mencionar, nesse sentido, que uma versão anterior da DSL LindaQoS já havia sido aplicada anteriormente, com sucesso, na especificação de um serviço complexo de formatação de documentos hipermídia com QoS (Soares-Neto et al., 2003b; Soares-Neto et al., 2004).

No Capítulo 5, foram apresentadas as principais ferramentas, desenvolvidas dentro do ambiente LindaStudio, para manipular especificações em LindaX. Duas ferramentas principais – de refinamento e síntese – mereceram destaque nesse capítulo, tendo sido avaliadas em dois estudos de caso: (i) na integração de LindaX com a ADL Wright (Allen, 1997) e (ii) na aplicação de LindaX na plataforma de componentes OpenCOM (Coulson et al., 2004). Foi apresentado também nesse capítulo um *framework* para gerência de adaptações (instanciado, como prova de conceito, para o OpenCOM), desenvolvido a fim de simplificar a implementação de processos de síntese para diferentes plataformas. O *framework* modela genericamente mecanismos de configuração segura (por meio de um serviço transacional) e de controle de adaptações de software. Ele é, em si, independente de LindaX e constitui-se em outra contribuição desta tese. O principal aspecto inovador do *framework* para gerência de adaptações é que os mecanismos de configuração e de controle de adaptações instanciados a partir do mesmo são manuteníveis em tempo de execução. Especificamente no contexto do OpenCOM, a instanciação desse *framework* apresenta-se, por si só, como uma contribuição marginal para a plataforma. Ao tomar a forma de um ‘*metaframework*’ de componentes, a partir do qual diferentes *frameworks* de componentes OpenCOM podem ser construídos, o *framework* para gerência de adaptações confere um maior reuso ‘entre’ *frameworks* de componentes OpenCOM, englobando o reuso não só de componentes mas também das entidades controladoras desses *frameworks* de componentes.

7.1. Sumário de contribuições

As contribuições para o estado-da-arte da área-tema desta tese, mencionadas no início deste capítulo, podem ser sumariadas na lista que se segue:

1. A concepção da estrutura de estilos e do arcabouço sintático de DSLs da linguagem LindaX (Gomes et al., 2003);
 - a) A análise e especificação de DSLs (e estilos correspondentes) para a arquitetura genérica de provisão de QoS definida em (Gomes, 1999; Gomes et al., 2001a) e de uma arquitetura de plano de dados de roteadores programáveis concebida a partir das idéias introduzidas em (Coulson et al., 2003);
 - b) Análise e detalhamento do modelo SCM proposto em (Colcher, 1999).
2. A modelagem do *framework* para gerência de adaptações (Gomes e Soares, 2005);

7.2. Resultados decorrentes

Do trabalho presente decorre um conjunto de especificações e produtos de software:

1. Especificação dos esquemas XML que compõem o núcleo de LindaX;
2. Projeto e implementação do ambiente LindaStudio, englobando os *drivers* das ferramentas `Expander`, `Generator`, `Translator`, `StyleEditor` e `ConfEditor`.¹
3. Implementação de *plug-ins* para edição textual (sem *tags* XML) de estilos (`StyleEditor`) e configurações (`ConfEditor`);
4. Implementação de *plug-ins* da ferramenta `Expander` para as DSLs `LindaRouter` e `LindaQoS`;

¹Conforme descrito no Apêndice A, o *driver* da ferramenta `Critic` já era oferecido no ambiente de desenvolvimento de software `ArchStudio 3`, sobre o qual `LindaStudio` foi construído.

5. Instanciação do *framework* para gerência de adaptações no OpenCOM (Coulson et al., 2004);
6. Implementação de um *plug-in* da ferramenta `Generator` que gera *scripts* de verificação de regras de adaptação para a instância do *framework* para gerência de adaptações no OpenCOM;
7. Implementação de um *plug-in* da ferramenta `Translator` que refina configurações em DSLs LindaX para a ADL Wright (Allen, 1997);²
8. Implementação de um *plug-in* da ferramenta `Translator` que refina configurações em DSLs LindaX para *scripts* de configuração para a instância do *framework* para gerência de adaptações no OpenCOM.

7.3. Trabalhos futuros

A estrutura extensível da linguagem LindaX lhe permite abranger, a princípio, outros formalismos e plataformas de programação além daqueles utilizados nesta tese. Adicionalmente, apesar do foco da abordagem proposta em sistemas de comunicação adaptáveis, uma das características que tentou-se manter em LindaX foi sua generalidade em relação a outros domínios de aplicação. Ambos os aspectos de extensibilidade e generalidade só podem, contudo, ser validados de forma contínua. Nesse sentido, outras possibilidades que vêm sendo investigadas como continuação do presente trabalho, explorando a extensibilidade da linguagem, incluem: (i) a especificação comportamental de componentes, *pipes*, portas e pontos de acesso através da álgebra de processos $R\pi$ proposta por De Souza e Cunha (2003)³ e (ii) a síntese de configurações e de mecanismos de controle de adaptações (envolvendo novas instâncias do *framework* para gerência de adaptações) em Java. Para explorar a generalidade da linguagem, vislumbra-se também a definição de estilos e DSLs associados a outros domínios. Um domínio de aplicação particular, de interesse do autor desta tese, são as ‘grades de próxima geração’ (*Next Generation Grids* – NGGs). NGGs podem ser vistas como uma extensão do paradigma de redes ativas para grades computacionais (Int’l Workshop on Middleware for Grid Computing, 2004), propiciando um maior dinamismo na introdução de novos serviços a recursos em grade.

²Implementação feita em colaboração com o Professor Carlos de Salles Soares Neto, da Universidade Federal do Maranhão.

³Meta prevista dentro do subprojeto GIGA “Engenharia de Serviços e Aplicações com QoS: Rede Cooperativa de Vídeo”, financiamento CPqD/RNP – FUNTTEL.

Um aspecto que merece maior atenção em possíveis continuações deste trabalho é a representação formal de LindaX. Isso propiciaria processos de refinamento formalmente fundamentados ou mesmo validações de descrições diretamente a partir da linguagem. Esse último ponto está ligado à proposta de trabalho de integração com a álgebra $R\pi$, mencionada anteriormente.

Como extensão a esta tese, vislumbra-se também a possibilidade de tratamento conjunto de especificações de DSLs LindaX distintas, como LindaQoS e LindaRouter. Em especial, é uma contribuição futura prevista tratar de modo integrado adaptações arquiteturais envolvendo diferentes aspectos. Nesse processo, acredita-se que o estabelecimento de uma semântica de adaptabilidade mais precisa à propriedade `Level` (em realidade, pouquíssimo explorada nesta tese) será fundamental. O mesmo se aplica ao tratamento conjunto de descrições nas visões arquitetural e de recursos oferecidas por LindaX. Nesse sentido, é necessário um melhor detalhamento da descrição de aspectos ligados à gerência de recursos por meio de tarefas e provedores. A maior dificuldade encontrada foi estabelecer cenários de avaliação coerentes com aqueles desenvolvidos ao longo da tese, haja vista que a implementação atual da plataforma OpenCOM ainda não é dotada de um metamodelo de recursos bem definido (Durán-Limón, 2001; Parlavantzas et al., 2002). Ainda com relação a tarefas e provedores, tem-se por objetivo estudar a aplicação dessas abstrações na descrição (não-arquitetural) de outros aspectos, além de localidade e gerência de recursos.

Na versão atual do ambiente LindaStudio, os *plug-ins* da ferramenta `Expander` são implementados sem nenhum auxílio ferramental. Idealmente, esses *plug-ins* deveriam ser sintetizados a partir dos próprios estilos aos quais eles estão relacionados. Contudo, uma abordagem desse tipo demanda, no mínimo, a inclusão de novas formas de predicados nas descrições de estilos. Mais especificamente, além de predicados de restrição, estilos devem poder ser povoados com predicados “causais”, em que ações são tomadas em resposta a determinadas condições. Em um exemplo típico, a instanciação de um componente do tipo `A` dispararia automaticamente a instanciação de um componente do tipo `B` e a criação de uma associação entre eles (por meio de um *pipe* específico). O desenvolvimento de predicados causais em LindaX e, em particular, seu mapeamento para plataformas de componentes também dará à abordagem proposta nesta tese um primeiro suporte à descrição de redes autogeridas (Konstantinou, 2003). Esse ponto já vem sendo analisado em conjunto

com o grupo de pesquisa DMRG (Distributed Multimedia Research Group, 2004), que desenvolve a plataforma OpenCOM, como parte de uma colaboração iniciada através de estágio de doutorado⁴ do autor desta tese na Universidade de Lancaster, Inglaterra.

Embora LindaX permita a composição aninhada de configurações, essas não são refinadas, nesta tese, para composições aninhadas em Wright, sendo primeiramente planejadas pela ferramenta de refinamento. Isso decorre, primordialmente, da falta de suporte em Wright a configurações com compartilhamento de constituintes, conforme discutido no Capítulo 5. Contudo, LindaX apresenta também um complicador nesse processo, mascarado pelas limitações de Wright, que é o fato de estilos em LindaX não poderem ser aninhados, uma vez que no vocabulário de LindaX só é possível definir componentes primitivos. Não se pretende, porém, resolver essa questão simplesmente introduzindo tipos de componentes compostos em estilos LindaX. Vislumbra-se também que estilos LindaX permitam a composição (ou recursão) de estilos, ou seja, que a configuração interna de um tipo de componente composto não precise ser prescrita, podendo, no entanto, ser restringida a partir de outro (ou do mesmo) estilo. A ausência dessa facilidade em LindaX obriga a criação de ‘estilos de cola’ que permitem a composição de configurações seguindo diferentes estilos. Os estilos `GenericForwarder` e `HierarchyNqoS` são exemplos típicos em LindaRouter e LindaQoS, respectivamente. Recursividade de estilos é um aspecto ainda não devidamente abordado na área de engenharia de software – em grande parte, devido à dificuldade de tratamento formal – e seu estudo seria, portanto, uma contribuição futura em potencial para essa área.

Por fim, o *framework* para gerência de adaptações proposto nesta tese apresenta três limitações principais. Primeiro, adaptações incrementais não são consideradas. Isto é, o *framework* dá suporte à configuração inicial a partir de uma linguagem de especificação arquitetural de alto nível (configuração em lote), bem como ao controle de adaptações disparadas a partir da plataforma de componentes (adaptações “*ad-hoc*”). Adaptações incrementais disparadas a partir de uma linguagem de especificação arquitetural de alto nível (adaptações “programadas”) não são previstas. Segundo, o *framework* não dá suporte ao controle de adaptações distribuídas. No contexto de LindaQoS, essa é uma limitação severa, uma vez o estilo `DistributedNqoS` é definido para representar exatamente configurações

⁴Financiamento CAPES/MEC.

distribuídas. Um serviço de controle de adaptações distribuídas demanda a implementação de associações não-locais que viabilizem a interceptação ‘remota’ de eventos de adaptação. Terceiro, o mapeamento de configurações e restrições em LindaX para código Lua demanda a existência de interpretadores dessa linguagem embutidos nos componentes do *framework*. Em ambientes onde memória é escassa (telefones celulares, sensores) o uso de Lua pode ser inadequado. Em particular, o mapeamento das restrições para máquinas de estados pode ser uma possibilidade interessante. Esses são três pontos desta tese que também vêm sendo analisados em conjunto com o grupo de pesquisa DMRG da Universidade de Lancaster, Inglaterra.