

## 7

### Conclusão

Este trabalho apresentou uma infraestrutura para implantação distribuída de componentes de software multi-linguagem e multi-plataforma. Essa infraestrutura segue um processo de implantação compatível com Emmerich [4] com suporte à instalação de dependências estáticas [3] dos componentes através do encapsulamento do sistema de pacotes LUAROCKS. Para tanto foram criados dois serviços para implantação (*DeployManager* e *Packager*) e foram implementadas melhorias na infraestrutura de execução proposta por Augusto [42] no suporte à instalação dos componentes e à integração com repositórios de implementações. Reusamos o sistema de pacotes LUAROCKS para descrever, publicar e instalar as implementações de componentes e suas dependências estáticas. O LUAROCKS foi escolhido por possuir um resolvidor de dependências, usar repositórios remotos, operar em diferentes plataformas, instalar pacotes em diferentes versões e ser extensível. Essa escolha obrigou uma adaptação no LUAROCKS para torná-lo compatível com softwares desenvolvidos em diferentes linguagens.

A fim de simplificar a implantação distribuída, o serviço *DeployManager* permite que o usuário possa especificar a implantação de componentes em níveis gradativos de detalhamento. Esse serviço disponibiliza um mecanismo automático para mapeamento de componentes do usuário nas entidades do ambiente de execução e nas máquinas físicas. Contudo o ator da implantação possui um total controle e pode intervir nas partes do mapeamento que mais lhe interessar. Para viabilizar esse comportamento, o serviço *DeployManager* adota uma fase de planejamento, antes da implantação real, que facilita a manipulação do ambiente de execução e permite antecipar problemas comuns durante a implantação como: (i) incompatibilidade das conexões entre facetas e receptáculos; e (ii) indisponibilidade dos pacotes de componentes compatíveis com o contêiner e nó de execução.

Diferentemente de outras ferramentas como DANCE, DEPLOYWARE e ADAGE, este trabalho não adotou uma abordagem declarativa para implantação. Ao invés disso, adotamos uma abordagem imperativa pelo uso dos serviços *DeployManager* e *Packager* para implantar os componentes dis-

tribuídos. Contudo, essa escolha mostrou-se interessante por flexibilizar a manipulação das entidades implantadas em tempo de execução e por facilitar o reuso programático dos planos de implantação.

Por outro lado, este trabalho apresenta algumas limitações. No que se refere à análise de desempenho, não foi feita nenhuma avaliação sobre a sobrecarga imposta pelo resolvidor de dependências estáticas do sistema de pacotes LUAROCKS. Por sua vez, o LUAROCKS ainda precisa de melhorias para se equiparar, em qualidade, aos recursos providos por outros sistemas de pacotes populares, como DEB e RPM, no sentido de oferecer o empacotamento de artefatos desenvolvidos em outras linguagens além de Lua. Também não foi investigado como os tempos decorridos para concretizar a implantação distribuída oneraram a usabilidade da interface programática, ou mesmo se geram falhas de comunicação quando tenta-se implantar arquiteturas mais complexas e amplamente distribuídas.

Por fim, é possível identificar interessantes temas para investigação futura. Entre eles:

- Sistemas de componentes que suportem linguagens de descrição de arquitetura [19, 20], podem usar esta infraestrutura para implantar seus componentes de forma distribuída.
- Esta infraestrutura de implantação pode ser integrada à arquitetura de monitoramento *on-line* e transparente para componentes SCS proposta por Fonseca et al. [63], simplificando sua configuração e a implementação e uso de políticas de balanceamento de carga dos componentes durante a execução.
- A partir do monitoramento dos componentes em execução, pode ser necessário adaptar os planos de implantação para re-implantar parte ou todo o plano, ou ainda co-implantar novos componentes. Nesse contexto, vale investigar como reusar as estratégias propostas por Cudennec [18].
- Outra interessante alternativa é experimentar heurísticas inteligentes [64] para o mapeamento automático dos componentes SCS na infraestrutura de execução e nos recursos físicos, explorando os níveis gradativos de detalhamento permitidos por esta infraestrutura de implantação.
- Para permitir a descoberta de novos recursos físicos (ou de infraestrutura) a serem gerenciados por um *DeployManager*, pode ser interessante o reuso de ferramentas da Computação em Grade, como os *Grid Information Services* (GIS).

- É importante uma investigação sobre o desempenho dos mecanismos de mapeamento automático propostos neste trabalho, de forma que não inviabilizem seu uso prático para aplicações de grande escala.
- Para facilitar ainda mais o uso dos serviços apresentados neste trabalho, é interessante desenvolver uma interface gráfica que apresente os planos de implantação de forma visual, garantindo uma fácil interação com as etapas da implantação e uma melhor visualização da arquitetura da aplicação distribuída.