**Capítulo 4**

**Trabalhos Relacionados**

Neste capítulo será apresentado trabalhos acadêmicos relacionados à sistemas distribuídos modularizados. Em sua descrição será abordada uma breve apresentação dos trabalhosjunto com as características propostasneles. Além disso, será abordada na descrição uma breve comparação com o InteropFrame, destacando as principais diferenças e semelhanças.

4.1 HighFrame

Segundo {SantosSaulo2014} (HighFrame: Uma Ferramenta para o desenvolvimento em alto nível e deployment automático de sistemas distribuídos baseados em componentes), a HighFrame tem como objetivo simplificar o desenvolvimento de sistemas distribuídos baseados em componentes, permitindo ao desenvolvedor manter o foco no negócio do sistema. Ela propõe uma solução integrada que suporta ao desenvolvimento de componentes baseado em implementações genéricas. Como também a definição da arquitetura do sistema baseado em um modelo gráfico de alto nível. Além de suportar ao *deployment* automático da arquitetura nos nós distribuídos disponibilizando o sistema na sua forma funcional.

A HighFrame foi projetada com o objetivo de manter o foco do desenvolvedor no negócio. Para isso ela oferece uma camada de abstração para o desenvolvimento de componentes baseado no modelo de anotações Fraclet. Dessa forma torna possível o desenvolvimento de componentes independente da tecnologia e modelo de componente. Para isso as anotações feitas no código fonte representam um componente genérico que pode ser gerado automaticamente em qualquer modelo de componente.

A HighFrame se baseia em 3 camadas, como mostra a Figura \ref{011ArquiteturaHighFrame}.

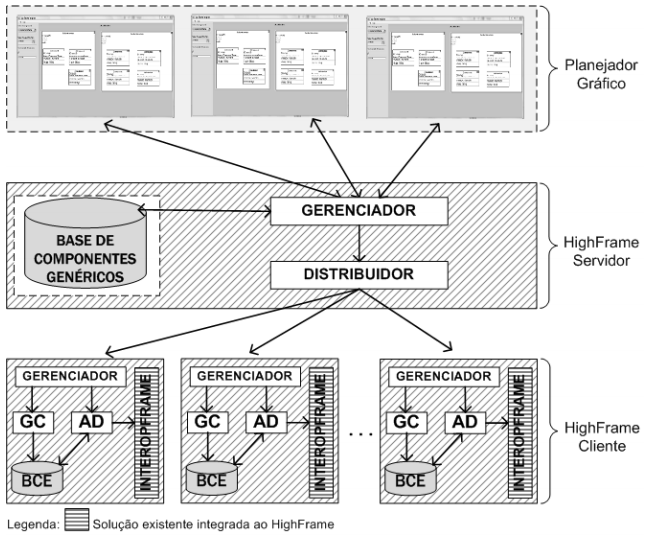


Figura 4.1 – Arquitetura da HighFrame. Fonte: {SantosSaulo2014}

No Planejador gráfico é possível difinir graficamente o porcesso de composição de um sistema. Com ele é possível definir os nós distribuídos e criar suas interconexões. Além disso é possível definir os modelos de componentes a serem utilizados, como também o binding remoto entre eles. A segunda camada é a HighFrame Servidor. Ela disponibiliza através do módulo gerenciador componentes genéricos existentes na base de componentes com o objetivo de que o desenvolvedor possa compor a arquitetura do sistema e o plano de deployment distribuído. Por fim, a camada HighFrame Cliente que é responsável por receber o plano de deployment e os componentes genéricos e assim gerar os componentes específicos e realizar suas interconexões.

O InteropFrame, que é a ferramenta utilizada no trabalho proposto, encontra-se na estrutura da HighFrame. Ele se localiza no módulo HighFrame Cliente e é utilizado para prover a interconexão automática entre componentes distribuídos heterogêneos. Assim como o InteropFrame, a HighFrame foi construída de forma a ser extensível. Entretanto, a extensibilidade dela depende do suporte aos modelos de componentes implementados no InteropFrame. Nesse trabalho, o InteropFrame foi incrementado com mais um modelo de componente, o OSGi, possibilitando assim uma nova extensão para a HighFrame.

4.2Vienna Component Framework

Segundo {OBERLEITNER2003}, o VFC (Vienna Component Framework) suporta a interoperabilidade e modularidade dos componentes através de modelos de componentes heterogêneos. Dessa forma, o VFC apresenta um sistema unificado de modelo de componente onde a abordagem é baseada na construção de componentes intermediários para cada modelo de componente. Sendo assim, o VFC abstrai as diferenças entre os modelos suportados e realiza a comunicação entre eles.

O VFC foi desenvolvido em JAVA com suporte a quatro modelos de componentes, assim como mostra a Figura \ref{012ArquiteturaVFC}: JavaBeans, EJB (Enterprise JavaBeans), CORBA (Common Object Request Broker Architecture) e COM (Microsoft’s Component Object Model).Esses modelos são suportados por meio de plugins. Logo novos modelos, assim como no InteropFrame e no HighFrame, podem ser adicionados com a implementação de novos plugins. Para isso o VFC fornece um modelo de programaçMicrosoft'omponente espec programaçodelos podem ser adicionados com a implementaç e COM (Microsoft'omponente especão generico que possibilita a construção de novos plugins.

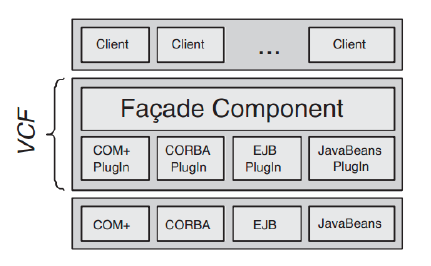


Figura 4.2 – Arquitetura do VFC {OBERLEITNER2003}.

Baseado no artigo de apresentação do VFC, nada em relação a comunicação entre componentes distribuídos foi apresentado. Ele apenas propôs uma solução que permite aos modelos de componentes diferentes se comunicarem de forma local. Diferentemente do InteropFrame que premite e realiza a comunicação desses componentes tanto de forma local como de forma distribuída. O VFC se assemelha muito ao InteropFrame em relação a sua arquitetura quando trata os modelos de componentes como plugins independentes. Além de gerar componentes intermediários que abstraem as diferenças entre modelos de componentes diferentes.