2.1.1.1 Modelo OpenCOM

O OpenCOM é um modelo de componentes de baixo peso projetado para o desenvolvimento de middlewares em dispositivos de computação com poucos recursos (processamento, memória, armazenamento). Além de ser um modelo de baixo peso, o OpenCOM provê a capacidade de reconfiguração dinâmica de middlewares tanto no domínio estrutural quanto no comportamental {rocha2008}.

O OpenCOM é fundamentado em três tecnologias {nascimento2013}:

- Componentes - O modelo permite a especificação da estrutura de sistemas através do uso de componentes e conexões entre componentes (ROCHA, 2008).

Os conceitos fundamentais no OpenCOM são interfaces, receptáculos e conexões. Uma interface representa uma unidade de provisão de serviços, enquanto que um receptáculo representa uma unidade de requerimento de serviços e é usado para tornar explícita a dependência de uma interface de um componente com outra. Uma conexão representa uma ligação entre um serviço fornecido por uma interface de um componente e um serviço requerido por um receptáculo de outro componente {clarke2001};

- Reflexão computacional - O modelo OpenCOM foi projetado para suportar a reflexão computacional - que é a capacidade que um sistema tem de observar sua própria representação/estrutura e modifica-la em tempo de execução;

- Frameworks de componentes – Uma característica chave do OpenCOM é o uso da noção de frameworks de componentes. Um framework de componentes é definido no OpenCOM como um conjunto fortemente acoplado de componentes que coopera para resolver alguma área de interesse. O OpenCOM também fornece um protocolo de extensão bem definido para a aceitação de componentes adicionais que modificam ou estendem o comportamento do framework de componentes, além de restringir o modo como os componentes são organizados {coulson2008}.

Cada componente OpenCOM implementa quatro interfaces {grace2007 apud nascimento2013}:

- ILifeCycle - fornece operações de inicialização e finalização que são chamadas quando um componente é criado ou destruído, ou seja, define ações que devem ser tomadas no início e no final do ciclo de vida do componente;

- IConnections (opcional) - oferece métodos para modificar as interfaces ligadas aos receptáculos de um componente. Esta interface deve ser implementada por todos os componentes que possuem receptáculo;

- IMetaInterface - suporta inspeção dos tipos de interfaces e receptáculos declarados pelo componente;

- IUnknown - é equivalente à interface do mesmo nome no Microsoft COM, isto é, ele é usado para obter a referência para a interface solicitada na instância do componente.

O OpenCOM implanta um substrato padrão em *runtime* que está disponível em todo o espaço de endereçamento. Insto é implementado através de um componente *singleton* chamado "OpenCOM” que exporta uma interface chamada IOpenCOM. O papel do *runtime* do OpenCOM é o de gerir um repositório de componentes disponíveis e, assim, permitir a criação e exclusão de componentes. Além disso, a interface IOpenCOM serve como um ponto centralizado para a submissão de todas as solicitações de conexão ou desconexão entre receptáculos e interfaces no seu espaço de endereçamento. Para facilitar a reconfiguração, o *runtime* registra cada criação e exclusão de cada componente ou conexão em um espaço de meta-estrutura chamado de *system graph* (grafo do sistema). Isto permite que o OpenCOM suporte consultas que, através de um identificador de conexão, fornece detalhes sobre o receptáculo e as interfaces participantes da conexão, juntamente com detalhes de seus componentes que os implementam {clarke2001}.

2.1.1.2 Modelo Fractal

O Fractal é apresentado em (OW2, 2004) como um modelo de componentes modular e extensível, que pode ser usado para projetar, implementar, implantar e reconfigurar sistemas e aplicações (OW2, 2004). O objetivo geral do Fractal é reduzir os custos de implantação, desenvolvimento e manutenção de softwares, e em particular os projetos do consórcio ObjectWeb (OW2, 2004). Segundo Bruneton et al. (2006), o Fractal é um modelo de componentes de baixo peso, hierárquico e reflexivo com compartilhamento de componentes que proporciona os benefícios da programação baseada em componentes para o desenvolvimento de sistemas distribuídos. Este modelo tem sido utilizado na construção de uma ampla gama de plataformas de middleware (COULOURIS et al., 2011). O modelo de componentes Fractal suporta várias linguagens de programação, como por exemplo, Java e C, e de forma experimental .NET, SmallTalk e Python.

2.1.1.3 Modelo OSGI

2.9 Discussão