

3

Infra-Estrutura Utilizada

Para o desenvolvimento deste trabalho, foram utilizados alguns projetos do nosso grupo de pesquisa [42] como base, em especial o ORB OiL (*ORB in Lua*) [29] e a biblioteca LOOP [43]. Além do uso destes projetos, desenvolvemos a implementação Lua do modelo de componentes SCS [44] antes de iniciar os esforços para o desenvolvimento da infra-estrutura de execução. Este modelo conta também com uma implementação Java feita por outros integrantes do nosso grupo. Por implementarem o mesmo modelo, ambas as versões contam com as mesmas funcionalidades. Não entraremos em detalhes sobre as implementações Java.

O projeto OpenBus também já contava com alguns serviços, que foram desenvolvidos sobre uma versão simplificada do modelo SCS. Alguns destes serviços, implementados na linguagem Lua, foram integrados à infra-estrutura de execução.

Neste capítulo descreveremos brevemente as principais características do modelo SCS e dos serviços OpenBus já existentes.

3.1

Modelo de Componentes SCS

Esta seção não tem por objetivo descrever o modelo de componentes SCS detalhadamente, mas serão apresentados todos os conceitos necessários para o entendimento dos capítulos posteriores. Maiores detalhes podem ser encontrados em outros documentos, disponíveis em [44] e [42].

3.1.1

Componentes

O modelo SCS é baseado, primariamente, em interfaces providas e requeridas, que poderão ser descobertas em tempo de execução. Interfaces providas são denominadas facetas, e interfaces requeridas são denominadas receptáculos. Um componente SCS poderá disponibilizar diversas facetas, e ter zero ou mais receptáculos. Receptáculos podem ter duas versões: uma simples, que aceitará receber apenas um objeto implementador da interface requerida, e uma múltipla, que aceitará mais de um.

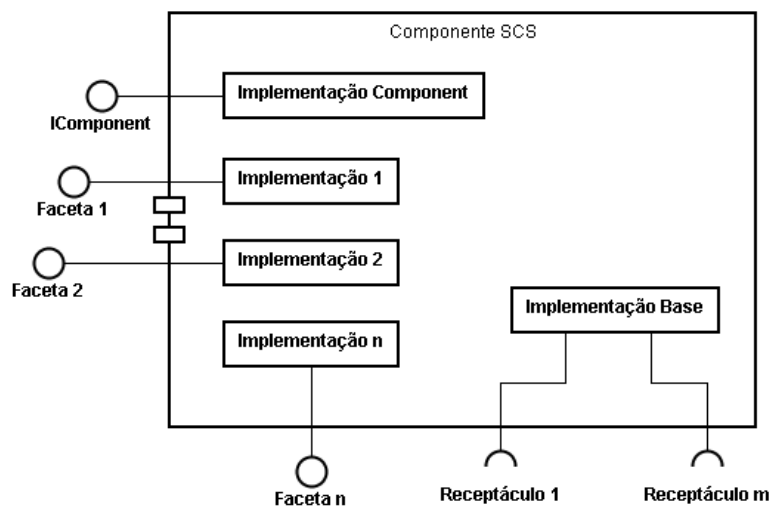


Figura 3.1: Exemplo de Componentes SCS

3.1.2

Facetas

Facetas são funcionalidades providas por um componente, na forma de métodos definidos em uma interface. Cada faceta é uma interface bem definida. Componentes SCS utilizam a arquitetura CORBA [16], tendo assim suas interfaces definidas através de arquivos de definição IDL, segundo o padrão definido pelo Object Management Group.

Todo componente SCS permite a obtenção de suas facetas, através de uma interface obrigatória, chamada IComponent. Desta forma, temos uma interface única a ser trocada entre componentes. De outra forma, a interação entre componentes ficaria mais complexa, e haveria casos em que

seria necessário que um componente conhecesse todas as facetas do outro, ainda que não as utilizasse.

3.1.3

Receptáculos

No modelo SCS, componentes especificam suas dependências externas através de estruturas chamadas receptáculos. Para definir uma dependência, basta criar um receptáculo e neste especificar a interface necessária. Assim, esta dependência será resolvida assim que uma faceta implementadora desta mesma interface seja conectada ao receptáculo. Receptáculos não são definidos em IDLs, e precisam ser especificados apenas no código-fonte.

O modelo almeja também facilitar a conexão entre componentes, e assim fornece outra faceta padrão, *IReceptacles*, que provê meios de conectar aos receptáculos componentes que implementem as interfaces especificadas. A interface *IReceptacles* não é obrigatória. Receptáculos podem permitir a conexão de apenas uma faceta implementadora de sua interface desejada, ou várias.

3.1.4

Conexões

Conexões entre componentes que seguem o modelo SCS são bastante simples: basta conectar uma faceta a um receptáculo, através do método apropriado para isto, que se chama *connect*. O único detalhe, já mencionado, é o fato de receptáculos poderem ser simples ou múltiplos. A possibilidade de se conectar e desconectar receptáculos a facetas efetivamente permite a (re)configuração dinâmica das dependências externas do componente. A Figura 3.2 mostra componentes conectados de diferentes formas.

3.1.5

Introspecção

O modelo fornece suporte nativo à introspecção. Através de uma faceta chamada *IMetaInterface*, pode-se descobrir remotamente quais as facetas e

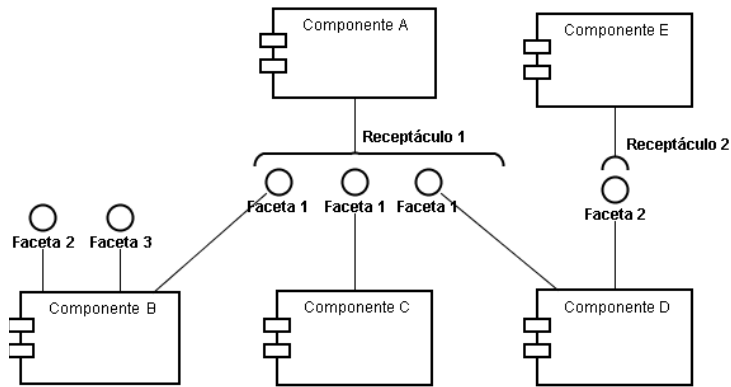


Figura 3.2: Componentes SCS conectados

receptáculos estão definidos em um componente. Esta faceta não é obrigatória e conta com uma implementação padrão. Seus métodos permitem a obtenção de todo o conjunto de descrições de facetas ou receptáculos, enquanto a faceta `IComponent` só permite a obtenção de facetas específicas. Receptáculos não são objetos CORBA, e assim sendo apenas sua descrição será disponibilizada.

3.1.6

Descrição das Facetas do Modelo

As facetas `IComponent`, `IReceptacles` e `IMetaInterface` estão brevemente descritas no Apêndice B.1.

3.2

Interface de Programação da Aplicação em Lua

Para facilitar o desenvolvimento de componentes SCS, é fornecida uma pequena biblioteca com implementações-padrão das facetas principais, além de algumas funções utilitárias. As interfaces são implementadas na forma de classes, e desse modo a biblioteca segue o padrão de programação orientada a objeto definido pelo projeto LOOP [43]. Os componentes não são obrigados a seguir o mesmo padrão, mas o encorajamos por facilitar bastante a programação no paradigma de orientação a objeto e aumentar a produtividade e clareza de código.

Além do suporte a facetas e receptáculos, o modelo também encoraja

que seus componentes forneçam um mínimo de introspecção, que se traduz na possibilidade de inspeção de suas facetas e receptáculos em tempo de execução. Isto facilita a criação de serviços de descoberta, e também a programação no paradigma reflexivo para os desenvolvedores de componentes que assim desejem.

Através destas funcionalidades mencionadas, é possível configurar e reconfigurar dinamicamente as dependências do componente. É possível, por exemplo, trocar o componente implementador de uma interface requerida em tempo de execução, sem a necessidade de parar e reiniciar a aplicação.

Para a criação de uma nova instância de componente, fornecemos uma função chamada `newComponent()`. Esta função recebe como parâmetros a fábrica LOOP, a descrição e o identificador do componente, e devolve uma nova instância de componente SCS. A fábrica LOOP é responsável por criar a estrutura básica da nova instância do componente, enquanto que a descrição é utilizada para a criação de facetas, receptáculos e estruturas de dados internas necessárias. Esta descrição nada mais é que o conjunto das descrições de todas as facetas e receptáculos do componente.

A instância recebida já conta com todos os receptáculos e facetas especificadas pelo programador do componente, inclusive as facetas providas pela biblioteca que tenham sido requisitadas.

3.3

OpenBus

O projeto OpenBus é desenvolvido no TecGraf [25], e objetiva permitir a integração de aplicações científicas, possivelmente legadas, através de um barramento comum. O OpenBus utiliza o modelo de componentes SCS, e utilizará os serviços criados neste trabalho. Da mesma forma, este trabalho se aproveita de alguns serviços disponibilizados pelo OpenBus. Por isso, foi iniciado um processo de integração, e os serviços utilizados pela infra-estrutura de execução serão brevemente descritos a seguir. Maiores informações podem ser encontradas na documentação do projeto.

3.3.1

Serviço de Controle de Acesso

O controle de acesso provido pelo OpenBus é realizado pelo Serviço de Acesso, que atua como um ponto de entrada para o barramento. A autenticação pode ser feita através de um par [usuário , senha] ou através de um certificado. Tipicamente, aplicações cliente se autenticam com um par [usuário , senha] - a identificação e senha do usuário que iniciou a aplicação. Servidores tipicamente se autenticam através de um certificado. Os servidores, em nosso caso, seriam os serviços da infra-estrutura de execução SCS, ou quaisquer componentes / serviços criados por outros desenvolvedores.

Após autenticar o usuário ou servidor, o Serviço de Acesso emite uma credencial que deverá ser usada pelo mesmo para acessar os demais serviços do barramento (básicos ou não). A credencial possui um identificador e o nome da entidade à qual está associada, que pode ser um usuário ou um serviço. Essa credencial tem um ciclo de vida e deve ser renovada para que não expire (mecanismo de *leasing*).

3.3.2

Serviço de Registro

O Serviço de Registro é responsável por controlar as ofertas de serviços disponíveis no barramento. Um componente que queira oferecer um serviço deve explicitamente registrar sua oferta no Serviço de Registro.

Componentes que desejem utilizar um serviço disponível devem obter a localização e as características/propriedades de provedores desse serviço, através de consultas ao Serviço de Registro. Os componentes que não oferecerem um serviço, mas que quiserem ser encontrados por outros componentes - por exemplo, para permitir sua participação em um ambiente de colaboração - devem também se registrar.

A referência para o serviço de registro deve ser obtida através do Serviço de Acesso. Para obter esta referência, é necessário que o componente possua uma credencial válida.

3.3.3

Serviço de Sessão

Este serviço serve para agrupar componentes através de uma mesma credencial. Ele assume a responsabilidade de renovar a credencial deste grupo, que só expira quando não houver mais membros. Ao receber uma credencial, ao invés de se autenticar no Serviço de Controle de Acesso, o componente pede sua adição à sessão. A saída do barramento ocorre de forma análoga. Este serviço oferece também um mecanismo de troca de mensagens entre seus componentes por *broadcast*.