

# Interconexão Dinâmica de Objetos Distribuídos Java

Rogério Drummond  
rog@ahand.unicamp.br

Adrian C. Ferreira  
adrian@ahand.unicamp.br

11 de abril de 1999

## 1 Introdução

O desenvolvimento de aplicações distribuídas exige muito esforço de programação se comparado com o desenvolvimento de programas convencionais não distribuídos. Há grande necessidade de mecanismos que facilitem o trabalho do desenvolvedor e elevem substancialmente o nível de abstração para a criação, configuração e gerenciamento dos objetos remotos e de suas conexões. As tecnologias atuais não apresentam essas facilidades. Este documento apresenta a versão inicial da LegoShell, um ambiente para desenvolvimento de aplicações distribuídas onde os objetos remotos são criados com a mesma facilidade dos objetos locais. O processo de configuração pode ser realizado depois que esses objetos foram instanciados. A comunicação entre os objetos pode utilizar conectores que são vias previamente definidas para fornecer serviços associados a uma conexão.

## 2 LegoShell

Na LegoShell, o processo de configuração pode ser realizado depois que esses objetos foram instanciados, o que é chamado de interconexão dinâmica. A comunicação entre os objetos pode utilizar conectores que são vias previamente definidas para fornecer serviços associados a uma conexão. Como os conectores possuem infra-estrutura padronizada, é possível a implementação de funcionalidades mais genéricas.

Tanto a interligação de objetos como as operações de troca de dados e chamadas de métodos podem ser facilmente implementadas. Aplicações são conjuntos de objetos executados sob demanda e configuráveis de acordo com as necessidades do usuário. Os objetos LegoShell são executados sob demanda, portanto somente depois de instanciados é que os recursos da máquina servidora passam a ser utilizados. Desse modo, um servidor remoto não precisa estar em execução enquanto aguarda chamada a seus serviços.

Nessa proposta, o desenvolvedor tem a visão da rede como a de um grande provedor de objetos. Depois de criados e configurados para compor uma aplicação, os objetos podem terminar sua execução e ser reconfigurados para formar outra aplicação cuja semântica

resultante é diferente daquela anterior. Todos os objetos são potencialmente remotos e podem ser usados em qualquer localidade da rede.

O processo de construção de uma aplicação distribuída em LegoShell possui quatro etapas: implementação, instanciação, configuração e execução.

Durante a implementação, o programador utiliza diretamente os recursos do sistema de suporte. Nas três fases seguintes, o programador usa a interface da LegoShell para acessar os serviços do sistema de suporte. Todo o processo é bastante simples porque os recursos de baixo nível são gerenciados de modo transparente para o programador.

Os objetos locais ou remotos que irão integrar uma aplicação precisam ser instanciados. Isso é feito a partir da interface da LegoShell usando o comando `new` de interface de comandos.

Na LegoShell, os objetos se comunicam através de portas que são uma abstração de *sockets*. Configurar uma aplicação significa interligar portas de entrada de uns objetos às portas de saída de outros. Como a configuração é feita depois que os objetos foram instanciados, é possível conceber classes mais genéricas. No momento em que um objeto é implementado, não é preciso definir o número de portas de entrada ou saída. Uma mesma classe pode ser usada em várias aplicações e em cada uma delas, sua instância pode ser ligada a um número diferente de outros objetos. Por exemplo, é possível implementar uma classe genérica `Produtor` cujas instâncias atendem a diferentes números de consumidores, estes definidos apenas em tempo de execução.

Depois de criados e configurados, os objetos de uma aplicação devem ser executados através da chamada ao comando `start` da LegoShell.

### 3 Conclusão

O sistema LegoShell é formado por um conjunto de classes Java que oferecem recursos para facilitar a criação e interligação de objetos distribuídos. O objetivo da LegoShell é permitir que o desenvolvimento de uma aplicação distribuída seja realizada com alto nível de abstração. As principais vantagens apresentadas podem ser resumidas na facilidade para criação de objetos distribuídos, na capacidade de interconexão dinâmica desses objetos e na introdução do conceito dos conectores como padrão de comunicação.