

# RCCS – Rede de Compartilhamento de Componentes de Software

Kleber Bacili<sup>1</sup>, Ricardo Anido<sup>2</sup>, Marcílio Oliveira<sup>1-2</sup>

<sup>1</sup> Laboratório de Inovação em Software – Ci&T Labs / UNICAMP  
Inovasoft - Unicamp

<sup>2</sup> Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)  
Caixa Postal 6176 – 13.084-971 – Campinas – SP – Brasil

krbacili@cit.com.br, ranido@ic.unicamp.br, marcilio@cit.com.br

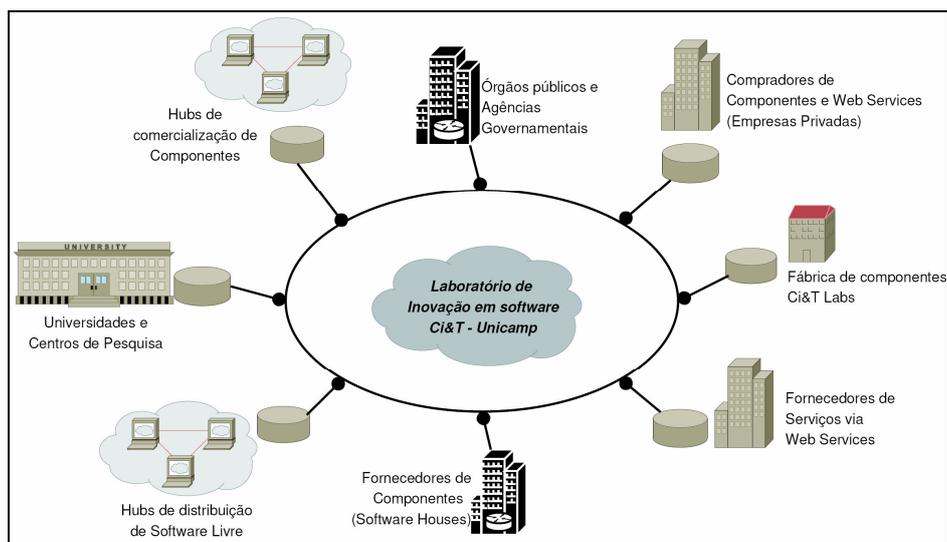
**Abstract.** *The Software Component Sharing Network is one of the results of a project of the Software Innovation Laboratory Ci&T-Unicamp. This is a Peer-to-Peer network which is based on open patterns and can be used by any person or any entity interested in sharing or searching for components. A reference implementation has been developed using the J2EE platform and its related technologies such as Web Services, hibernate, lucene, among others. This implementation has also been used as way of studying and validating new technologies.*

**Resumo.** *A RCCS, Rede de Compartilhamento de Componentes de Software, é fruto de um projeto do Laboratório de Inovação em Software Ci&T-Unicamp. Trata-se uma rede Peer-to-Peer, baseada em padrões abertos e que pode ser utilizada por qualquer pessoa ou entidade interessada em buscar ou compartilhar componentes. Foi desenvolvida uma implementação de referência, utilizando a plataforma J2EE e tecnologias afins como Web Services, hibernate, lucene, dentre outras, visando também o estudo e validação destas novas tecnologias.*

## 1. Introdução

A Rede de Compartilhamento de Componentes de Software (RCCS) é fruto de um projeto de pesquisa e desenvolvimento de solução tecnológica para viabilização de uma rede pública de distribuição e compartilhamento de componentes através de uma arquitetura independente de plataformas e da utilização dos padrões *Web Services*, em desenvolvimento no Laboratório de Inovação em Software Ci&T/Unicamp.

A RCCS é uma rede aberta, baseada em padrões abertos para comunicação, arquitetura e modelagem dos componentes e pretende atingir diversas categorias de entidades relacionadas a desenvolvimento e utilização de componentes de software. A figura 1 ilustra a forma com que a rede foi idealizada, com a participação de universidades e centros de pesquisa, órgãos governamentais, *hubs* de distribuição de software livre, fábricas de componentes, dentre outras entidades interessadas em compartilhar ou simplesmente buscar componentes na rede. Os pontos da rede podem se comunicar diretamente, ponto a ponto.



**Figura 1. Arquitetura da RCCS**

Conceitualmente, *Peer-to-Peer* (P2P) é uma alternativa aos modelos computacionais do tipo cliente-servidor. Através desta tecnologia, um sistema pode acessar diretamente recursos disponibilizados por outro, independente de terceiros atuando como servidores centrais.

## 2. Objetivo e justificativa

O principal objetivo do projeto é a **criação de uma rede pública de busca e compartilhamento de componentes de software através de uma arquitetura distribuída e independente de plataforma.**

Frente à necessidade de se construir software mais rápido e com mais qualidade em períodos cada vez mais curtos, o reuso de software pode ser visto como uma forte tendência para solucionar tais problemas. Reuso pode ser considerado uma denominação genérica, envolvendo uma variedade de técnicas utilizadas desde as etapas de design até a implementação. O principal objetivo é evitar o re-trabalho no desenvolvimento de um novo projeto, capitalizando trabalhos anteriores, fazendo com que as soluções já desenvolvidas sejam imediatamente implementadas em novos contextos. Desta forma, tem-se melhores produtos em um menor intervalo de tempo, com redução nos custos de manutenção pois as partes do sistema são independentes e facilitando a inclusão de novas funcionalidades. Além disso, a qualidade aumenta devido ao reuso de componentes previamente bem testados [D'Souza 99] [Jacobson97].

Uma das principais motivações na utilização de desenvolvimento baseado em componentes é exatamente a premissa de reutilização. Isto implica no reaproveitamento de partes já desenvolvidas. Esta abordagem, por reduzir o tempo de desenvolvimento e aumentar a qualidade do produto final é, de fato, bastante atraente [Krueger92].

Para que seja possível reusar algo, é necessário saber onde está. É sob esta perspectiva que este trabalho foi desenvolvido. O principal desafio foi desenvolver uma arquitetura distribuída visando o compartilhamento de componentes de software, envolvendo também a concepção do repositório, dos serviços de compartilhamento, mecanismo de busca e descoberta de repositórios.

### 3. Metodologia de execução

Para atingir seu propósito, a rede foi idealizada considerando-se algumas premissas fundamentais (ou requisitos iniciais). E foi em torno destas premissas que as pesquisas e o desenvolvimento se realizaram. A seguir, são listados esses aspectos pré-definidos:

- ✓ **Utilização de padrões.** A rede deve ser modelada seguindo padrões abertos [RAS06] e [IBM04] para facilitar e incentivar o surgimento de novas implementações e participações na rede.
- ✓ **Interoperabilidade.** É importante que a rede não esteja restrita a uma plataforma de desenvolvimento. Como a comunicação deve ser possível entre pontos com diferentes implementações, é muito importante que a proposta proporcione interoperabilidade.
- ✓ **Repositórios distribuídos e independência de entidade central.** Não deve existir um conjunto de repositórios centrais onde os componentes são armazenados. Estes repositórios estarão de forma distribuída, descentralizando os recursos da rede. Deste modo, a RCCS não deve ser dependente de nenhuma entidade específica e a queda de um ponto (ou simplesmente sua ausência na rede) não torne a rede indisponível momentaneamente.
- ✓ **Escalabilidade.** Tendo em vista que a rede é aberta, é imprescindível que sua arquitetura suporte escalabilidade em relação ao número de participantes.

Vale ressaltar também que uma das diversas atividades executadas no projeto foi a validação de novas tecnologias na implementação da RCCS. As principais tecnologias pesquisadas e utilizadas no desenvolvimento da rede foram Arquiteturas Peer-to-Peer, SOA (*ServicesOriented Architecture*), RAS (*Reusable Asset Specification*) [RAS06], Lucene – *Text Search Engine*, AJAX (*Asynchronous Javascript And XML*) e JSF (*Java Server Faces*). Foram escritos *papers* sobre tais tecnologias e disponibilizados no site do Ci&T Labs (<http://labs.cit.com.br>) para acesso público.

#### 3.1. Modelagem e arquitetura da rede

Um dos aspectos mais importantes é a modelagem da rede. Como dito anteriormente, trata-se de uma rede P2P, sem um servidor central. Mas isto não é suficiente. É necessário identificarmos a arquitetura, determinarmos como os pontos da rede se comunicarão, qual a tecnologia utilizada pra fazer a troca de informações e como estas informações serão transmitidas na RCCS.

Em cada repositório de componentes deve haver um servidor onde os serviços fiquem disponíveis e a comunicação é feita na forma de *Web Services*. A arquitetura permite que os pontos da rede comuniquem-se diretamente, sem a interferência ou dependência de um servidor central. A inexistência deste servidor central proporciona à rede uma característica interessante de independência de qualquer entidade.

#### Protocolo de comunicação e modelagem de objetos da RCCS

O protocolo de comunicação determina quais são e como serão representadas as informações transmitidas na rede. Como a comunicação é feita através de *Web Services*,

as informações serão trocadas através de objetos, ou seja, o protocolo trata de quais objetos são transportados.

O modelo do componente foi baseado no RAS [RAS06], que é uma especificação proposta por um consórcio de empresas de tecnologia e adotada pela OMG (*Object Management Group* - <http://www.omg.org>) como padrão para especificação de componentes. É uma representação totalmente baseada em meta modelos. O principal objetivo do RAS é especificar a quantidade mínima de meta informações que um componente deve ter. Além disso, ele determina quais artefatos um componente possui, desde diagramas e casos de uso, até os arquivos de teste, código fonte e binário. Possibilita também classificar estes artefatos. Podemos citar também alguns outros objetivos do RAS, como estimular o desenvolvimento baseado na utilização de componentes e aumentar a comunicação entre os produtores e os consumidores de componentes.

### **3.2. Funcionalidades principais da rede**

Foi vislumbrada uma rede que pudesse trazer, de fato, funcionalidades que impulsionem e incentivem a utilização de componentes. Para isso, algumas funcionalidades são básicas e, sobretudo, essenciais. Pensando na forma mais natural de utilizarmos um sistema de compartilhamento, vimos que a busca de material é o primeiro passo.

Além de encontrar um componente, o usuário deve ter a opção por fazer o *download*. Para isso, a funcionalidade de *download* também se faz necessária. Visando agilizar a busca e diminuir o tráfego de dados, uma etapa é inserida entre a busca e o *download*: o *Asset Info*. Ou seja, uma funcionalidade para captura de informações sobre o *asset*/componente. A busca retornará informações básicas sobre os componentes encontrados, e, a partir disso, o usuário poderá requisitar maiores informações sobre o componente. E então, o *Asset Info* retornará todas as informações. A partir das informações fornecidas, o usuário poderá decidir por fazer ou não o *download* do componente. A busca e descoberta de repositórios é a parte mais relevante e é apresentada a seguir.

#### **Busca de componentes**

A busca é o primeiro passo para o funcionamento da rede. Podemos considerá-la como a parte mais importante do sistema, uma vez que ela determinará, muitas vezes, a qualidade do serviço. Uma busca deve ser simples e rápida, porém eficiente.

Tendo em vista que temos uma arquitetura *Peer-to-Peer* para a rede de compartilhamento e baseados nas duas principais arquiteturas existentes (pura e híbrida) [Walkerdine02], a arquitetura de busca deveria utilizar o que consideramos como pontos fortes de cada arquitetura existente.

No momento da busca, o usuário seleciona, entre um conjunto de repositórios conhecidos, onde deseja realizar a busca e, além disso, pode optar por buscar por novos repositórios em servidores *yellow page*, utilizando o protocolo de publicação de serviço dos *Web Services*, o UDDI (Figura 2a). Conhecendo os demais participantes da rede a busca é disparada para todos os repositórios – exceto os pesquisados (Figura 2b). Desta forma, cada *RAB* pode montar sua própria base de conhecimento, selecionando quais os repositórios serão utilizados para uma consulta futura.

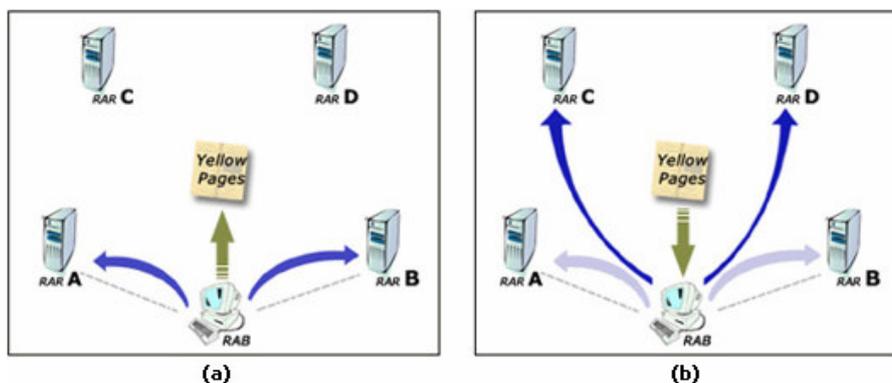


Figura 2. Arquitetura de busca na RCCS. [Oliveira05a]

O resultado da busca é apresentado com alguns detalhes. Os componentes retornados na busca são listados, apresentando informações relevantes, como repositório de origem, categoria, distribuição e descrição. Mais informações sobre as demais funcionalidades podem ser encontrados nos artigos publicados [Oliveira05b] ou no site do Ci&T Labs (<http://labs.cit.com.br>).

## 4. Resultados obtidos

### 4.1. Produtos de software gerados

A RCCS implementa uma ferramenta de compartilhamento de componentes de software, que podem ser instalados localmente e utilizados para disponibilizar e buscar componentes na rede. A figura 3 apresenta algumas interfaces da RCCS.



Figura 3. Interfaces RCCS (busca, resultado e detalhamento).

Foi implementado um plug-in para eclipse ([www.eclipse.org](http://www.eclipse.org)) integrado à RCCS. Através do plug-in é possível buscar, utilizando o eclipse, componentes de software distribuídos e integrá-los a um projeto ou biblioteca, fazendo *download* e instalando componentes instaláveis na aplicação.

O projeto da rede foi finalizado e integrado ao repositório de ativos digitais DigitalAssets Manager – DAM (<http://www.digitalassets.com.br>) possibilitando o compartilhamento de componentes entre implantações do DAM e da RCCS.

#### 4.2. Outros produtos gerados (que foram disponibilizados para o mercado)

Foi desenvolvido e publicado um protocolo para compartilhamento de componentes de software, baseado no RAS. O documento de especificação está disponível no site do Ci&T Labs (<http://labs.cit.com.br>).

#### 4.3. Artigos publicados

Além da realização de apresentação em eventos técnicos, como o III Fórum de Inovação Ci&T e na IX Semana de Informática UFV, o trabalho originou publicações nacionais e uma internacional:

- ✓ O mecanismo de descoberta de repositórios (*Resource Discovery*) foi publicado no **I WorkShop de P2P**, em Fortaleza-CE, sob o título de "Resource discovery em redes Peer-to-Peer aplicado à busca e compartilhamento de componentes de software" [Oliveira05a].
- ✓ A rede de compartilhamento de componentes de software foi apresentada no **Salão de Ferramentas do 23º SBRC** [Oliveira05b]. Neste salão, além de uma apresentação formal, a RCCS foi instalada em diferentes computadores para realização de demonstrações práticas.
- ✓ A integração da RCCS com o DigitalAssets Manager foi publicada no salão de demonstrações do **OOPSLA (21st Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications)** [Bacili06]. Este evento é considerado um dos mais importantes do mundo na área de orientação de objetos e aplicações. Foi possível realizar exibições práticas do mecanismo da RCCS integrado a outra ferramenta, e foi possível demonstrar ambas realizando buscas distribuídas, uma vez que o protocolo de compartilhamento é o mesmo.

O perfil prático do projeto pôde ser afirmado através das publicações, que foram, em sua maioria, em salões de demonstração. Nestes salões foi possível instalar e permitir a iteração dos participantes do evento com a ferramenta.

#### 4.4. Recursos humanos capacitados (especialistas, mestres, doutores, etc.)

O projeto contou com pesquisadores do Instituto de Computação da Unicamp, consultores e gerentes de projeto da Ci&T e com alunos de graduação e mestrado. Um dos principais resultados do projeto foi a capacitação de pessoal, gerando especialistas em nas tecnologias relacionadas ao projeto, sobretudo em SOA (*Service Oriented Architecture*), arquiteturas Peer-to-Peer, além de um mestre em ciência da computação, que teve a RCCS como tema de projeto de mestrado (Busca e Compartilhamento de Componentes de Software em Redes Peer-to-Peer).

#### 4.5. Dissertações e/ou teses geradas

Foi gerada uma tese de mestrado, desenvolvida pelo membro do laboratório Marcílio da Silva Oliveira, no Instituto de Computação (IC) da Unicamp, sob a orientação da Prof<sup>a</sup>. Dr<sup>a</sup>. Islene Garcia. A dissertação foi defendida em 15 de dezembro de 2006, no IC.

#### 4.6. Eventuais parcerias ou programas de transferência de tecnologia efetuados

O Ci&T Labs tem parceria com diversas instituições de pesquisa, e vêm realizando projetos em conjunto com C.E.S.A.R., UFPE, UNICAMP, UFPB, IBTA, CenPRA e UFV. O projeto da RCCS foi desenvolvido juntamente com a Unicamp, e algumas ferramentas (como o plug-in) foram desenvolvidas no laboratório da UFV, em Viçosa/MG. Na equipe da UFV, além das ferramentas, foram realizados trabalhos acadêmicos sobre a Rede de Compartilhamento. Objetiva-se a aplicação de resultados da RCCS em novos projetos de parceria entre instituições.

#### 5. Aplicabilidade dos resultados

O projeto desenvolveu, além de um protocolo de compartilhamento, uma ferramenta para manter localmente e compartilhar componentes de software. Atualmente os resultados já são aplicados na prática. A RCCS está integrada à principal ferramenta nacional de gerenciamento de ativos digitais (Prêmio B2B de qualidade: [http://www.digitalassets.com.br/release\\_12.html](http://www.digitalassets.com.br/release_12.html)). Além disso, a RCCS já possui pontos remotos na Unicamp (Ci&T Labs), UFV (Depto. de Informática) e Ci&T.

O modelo de repositório definido também pode ser aplicado em outras ferramentas. Foi definido um padrão de banco de dados baseado no RAS e pode ser implementados em diferentes bases de dados. Isso já vem acontecendo, e ferramentas como DigitalAssets Manager já utilizam parte do repositório da RCCS para representar a flexibilidade de atributos de ativos digitais.

Também foi implementado um mecanismo de empacotamento de ativos em modelo RAS (Export-RAS). Este mecanismo gera o pacote “.ras”, contendo as informações os artefatos do componente, conforme definido na especificação [RAS06]. Os principais gerenciadores de ativos internacionais utilizam este padrão para intercâmbio de componentes, e essa funcionalidade alavanca as ferramentas nacionais a um padrão internacional de representatividade de ativos, que muitas vezes é requisito obrigatório para escolha de ferramentas.

Além da utilização da arquitetura da rede em outras ferramentas, o repositório desenvolvido também é utilizado em outros projetos. Atualmente, o modelo de repositório e banco de dados é utilizado no projeto de construção de um **Identificador Automático de Componentes de Software (IACS)**. O foco central do projeto consiste em montar um mecanismo capaz de reconhecer, em parques de aplicações já desenvolvidas, grupos de artefatos que componham componentes. Visando, sobretudo, a extração destes componentes para catalogação e reuso.

#### 6. Características inovadoras

Relacionado às características inovadoras, podemos citar:

- ✓ Modelagem e implementação de um protocolo específico para transmissão de componentes utilizando serviços independentes de plataforma (*Web Services*)
- ✓ Implementação de arquitetura híbrida Peer-to-Peer de compartilhamento de componentes de software, utilizando bases de conhecimento locais e servidores de referência (*Yellow Pages*).

- ✓ Desenvolvimento de ferramenta de gerenciamento e compartilhamento P2P de componentes. Apesar de existirem muitos exemplos de softwares de compartilhamentos de conteúdos, esta é a primeira solução específica para componentes de software, integrando mecanismos de busca, comunicação e transferência de componentes.
- ✓ Modelagem e implementação de um modelo extensível de repositório compatível com o padrão OMG (RAS). Este modelo pode ser replicado em diferentes ferramentas/contextos.

## 7. Conclusão e Perspectivas futuras

As ferramentas de compartilhamento de conteúdo sempre foram vistas como ferramentas para *download* de música e vídeos. A proposta deste projeto foi de utilizar o conceito de aplicações P2P em prol da produtividade dos desenvolvedores, visando facilitar o intercâmbio de informações úteis, neste caso componentes de software. Ou seja, foi possível utilizar tecnologias e arquiteturas já existentes, em um contexto diferente, proporcionando uma solução específica para novas necessidades.

Além de criar uma rede para compartilhamento de componentes, o projeto apresentou uma série de efeitos positivos, desde o fomento de novas tecnologias nacionalmente à promoção do reuso de componentes no desenvolvimento de software.

## Referências

- [Bacili06] Bacili K. R. and Oliveira M. S. *Digitalassets manager, managing and sharing software components*. OOPSLA - 21st Annual ACM Conference on Object-Oriented Programming, Systems,
- [D'Souza99] D. F. D'Souza and A. C. Wills. *Objects, Components, and Frameworks With UML: The Catalysis Approach*. Addison Wesley. USA., 1999.
- [IBM04] *Technology options for application integration and extended enterprise patterns*. IBM Red books, 2004. Disp. Em: <http://www.redbooks.ibm.com/redpapers/pdfs/redp3840.pdf>. Último acesso em 10/02/2006.
- [Jacobson97] L. Jacobson, M. Griss, and P. Jonsson. *Software Reuse: Architecture, Process and Organization for Business Success*. Addison Wesley. USA., 1997.
- [Krueger92] C. W. Krueger. Software Reuse. *ACM Computing Surveys*, 24(02):131{182, 1992.
- [Oliveira05a] M. S. Oliveira, I. Garcia, e A. Nunes. *Resource Discovery em uma rede Peer-to-Peer aplicado a busca e compartilhamento de componentes de software*. I Workshop de Peer-to-Peer-XXIII Simpósio Brasileiro de Redes de Computadores - Fortaleza, CE - maio de 2005, 2005.
- [Oliveira05b] M. S. Oliveira, I. Garcia, e A. Nunes. *RCCS, Rede de Compartilhamento de Componentes de Software*. X Salão de Ferramentas - XXIII Simpósio Brasileiro de Redes de Computadores - Fortaleza, CE - maio de 2005, 2005.
- [RAS06] RAS. *Reusable Asset Specification, OMG Available Specification*. Object Management Group, 2006. Disponível em <http://www.omg.org/technology/documents/formal/ras.htm>. Último acesso em 01/12/2006.
- [Walkerdine02] Walkerdine, J., Melville, L., Sommerville, I., *Dependability Properties of P2P Architectures*, In the Second International Conference on Peer-to-Peer Computing (P2P'02)