

## SERVIÇO DE PERSISTÊNCIA PARA AMBIENTES DISTRIBUÍDOS EXPLORANDO OS RECURSOS DO REPOSITÓRIO DE INTERFACES

Patrícia Maria Pereira<sup>\*,1</sup>, Tatu Nakanishi<sup>\*\*,2</sup>, Maurício G. V. Ferreira<sup>\*\*,3</sup>

(1)Área de Sistemas de Informação

Laboratório Associado de Computação e Matemática Aplicada  
Instituto Nacional de Pesquisas Espaciais (INPE)

(2)Centro de Ciências Exatas

Universidade de Mogi das Cruzes (UMC)

(3)Centro de Controle e Rastreo de Satélites

Instituto Nacional de Pesquisas Espaciais

(\*)Mestrado, Bolsa CAPES, e-mail: [patricia@nucleo.inpe.br](mailto:patricia@nucleo.inpe.br); (\*\*)Orientadores, e-mails:  
[tatu@umc.br](mailto:tatu@umc.br), [mauricio@ccs.inpe.br](mailto:mauricio@ccs.inpe.br)

### Resumo

Este artigo mostra uma nova abordagem para persistência de objetos em ambientes distribuídos. O serviço aqui apresentado tem como base o serviço de persistência especificado pelo padrão CORBA, mas não adere a todas as suas concepções. O serviço proposto tem o propósito facilitar o gerenciamento de mudanças, além de reduzir o tempo de desenvolvimento e a complexidade da aplicação. Explorando o conteúdo do Repositório de Interfaces, recurso já disponível em implementações da especificação CORBA, a camada de persistência toma conhecimento, em tempo de execução, das interfaces de objetos lá contidas o que torna o sistema flexível e extensível a novos objetos de aplicação.

*Palavras-Chave: CORBA, Sistemas distribuídos, Persistência de Objetos, Repositório de Interfaces.*

### 1. Introdução

O serviço de persistência apresentado neste artigo procura se adequar às necessidades da arquitetura SICSD [1]. Esta é uma arquitetura flexível e dinâmica para objetos distribuídos que se aplica ao software de controle de satélites do departamento CRC (Centro de Controle e Rastreo de Satélites) do INPE (Instituto Nacional de Pesquisas Espaciais).

A arquitetura SICSD disponibiliza os serviços necessários a qualquer aplicativo destinado ao controle de satélites: processamento de telemetria e envio de telecomandos. É composta por objetos da aplicação e por serviços como o serviço de persistência, foco da pesquisa aqui exposta, para armazenamento dos objetos persistentes da aplicação. Estes objetos da aplicação migram de um nó para o outro de acordo com a demanda de solicitações de serviços, permitindo que a execução de tais serviços seja distribuída de forma balanceada a fim de otimizar o uso dos recursos computacionais disponíveis para o controle de satélites. A comunicação neste ambiente distribuído é feita através de um *middleware* que implementa a especificação CORBA (*Common Object Request Broker Architecture*).

O padrão CORBA especifica um serviço de persistência para objetos distribuídos denominado *Persistent State Service* (PSS)[3]. O PSS estabelece uma linguagem, a PSDL (*Persistent State Definition Language*), para definição dos objetos persistentes e dispositivos de armazenamentos. Duas etapas são necessárias para empregar esta linguagem no processo de armazenamento dos objetos persistentes. A primeira consiste em criar, para cada classe persistente da aplicação, um arquivo com extensão PSDL contendo a descrição dos atributos que devem ser armazenados no banco de dados. A segunda caracteriza-se pela compilação e geração dos códigos necessários para armazenar/recuperar os objetos persistentes da aplicação. Com o serviço de persistência CORBA é necessário um controle rígido de versões das classes do sistema, a fim de garantir que mudanças nos atributos de classes, que contenham objetos persistentes, sejam também refletidas nos respectivos arquivos PSDL.

A camada de persistência proposta segue algumas concepções do PSS especificado pelo padrão CORBA, mas procura facilitar o gerenciamento de mudanças (alteração do código de um objeto, adição de novos objetos, etc.) evitando a necessidade de recompilação do serviço. O serviço de persistência acessa dinamicamente a interface dos objetos a fim de identificar os seus métodos e atributos. Para isso utiliza-se de um meio já disponível em implementações CORBA, o repositório de interfaces [4].

### 2. Aspectos gerais do serviço proposto

O Serviço de Persistência proposto apresenta-se como o mediador entre a aplicação e o sistema de armazenamento de dados, disponibilizando para a aplicação o acesso a este sistema que pode ser um arquivo, um BDR (Banco de Dados Relacional) ou um BDOO (Banco de Dados Orientado a Objetos). Os objetos da

aplicação solicitam ao serviço de persistência o acesso ao sistema de armazenamento. O serviço faz o mapeamento deste objeto para um registro se o meio de armazenamento for um arquivo, para uma tupla caso seja um BDR e não tem transformações, se for um BDOO. Os objetos persistentes da aplicação não necessitam conter código para as operações de armazenamento ou recuperação.

O serviço mantém-se fixo em cada nó da rede, aguardando as solicitações dos objetos para acessarem o sistema de armazenamento. O serviço de persistência é responsável por gerenciar onde, ou seja, em qual nó da rede os objetos da aplicação devem ser armazenados/recuperados [2].

Esse serviço contém uma base de dados denominada “base de armazenamento”, a qual deve ser previamente configurada pelo Administrador de Banco de Dados, contendo a relação dos objetos persistentes da aplicação e o nome da base de dados onde eles devem ser armazenados. A base de dados de um objeto pode estar replicada em vários nós, como por exemplo, a base de dados de um objeto X pode estar presente tanto no nó1 quanto no nó2. Em caso de replicação, como na suposição feita acima para o objeto X, o serviço de persistência atualiza apenas um dos nós e atribui ao Sistema Gerenciador de Banco de Dados Distribuído (SGBDD) a responsabilidade de manter a atualização das cópias dos outros nós.

A intervenção do administrador de banco de dados faz-se necessária no processo de definição da localização da base de dados de cada objeto persistente por dois motivos. O primeiro caracteriza-se por definir quais serão os servidores de dados para a arquitetura SICSD. Ou seja, apesar da arquitetura ser distribuída, os nós que se apresentam como servidores de dados devem ser configurados da maneira mais adequada ao papel que vão desempenhar. Normalmente, possuem maior capacidade de processamento, memória e armazenamento em disco. O segundo motivo objetiva mensurar qual a disponibilidade desejável da base de dados de um determinado objeto persistente.

### 2.1. O funcionamento do serviço proposto

O serviço proposto acessa em tempo de execução à interface dos objetos. Desta maneira obtém-se, dinamicamente, conhecimento de como fazer uso dos métodos e atributos dos objetos. Este acesso dinâmico é feito através do repositório de interfaces, recurso já existente em implementações do padrão CORBA, responsável por gerenciar as interfaces de todos os objetos distribuídos pelo sistema [4].

O funcionamento do serviço proposto para armazenamento do estado de um objeto pode ser descrito, de forma sucinta, do seguinte modo: o serviço recebe uma mensagem de um objeto da aplicação para acessar o banco de dados (operação 1, da figura 1), então um PID (*Persistent Identification*) é gerado (operação 2). Acessando a “base de armazenamento”, recupera-se a identificação da base de dados na qual o objeto deve ser armazenado (operação 3). Este acesso a “base de armazenamento” possibilita disponibilizar para o serviço de persistência a relação de todos os possíveis nós onde este objeto pode ser armazenado. Faz-se uma análise do melhor nó para armazenar tal objeto que depende da taxa de disponibilidade de I/O em cada nó. Em seguida, recupera do repositório de interfaces os atributos do objeto solicitante, seus respectivos tipos e também os métodos de acesso a ele na memória (operação 4). Assim, o serviço de persistência é capaz de acessar o objeto na memória, recuperar o estado atual do objeto (operação 5) e armazená-lo no sistema de armazenamento correto (operação 6).

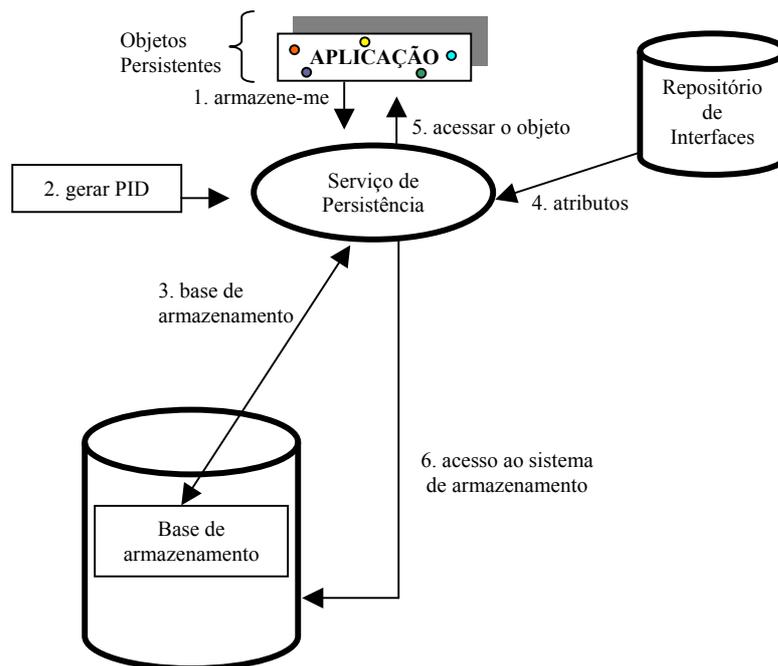


Figura 1. Funcionamento do Serviço Proposto  
Fonte [2]

### 3. Importância do serviço de persistência proposto

Atualmente as equipes de desenvolvimento de software das empresas concentram seus esforços na análise, projeto e implementação dos “objetos de negócios”. Estes são objetos específicos de uma aplicação para atender aos requisitos dos usuários. No caso de uma aplicação para controle de satélites, uma telemetria com o seu código, sua descrição, seu limite inferior e limite superior pode ser considerada um objeto de negócio. Os outros objetos necessários ao bom funcionamento da aplicação, como por exemplo, objetos que fazem a interface com o usuário ou acessam um sistema de armazenamento de dados, podem ser encontrados mais freqüentemente no mercado.

O serviço proposto reduz o trabalho do desenvolvedor evitando que ele tenha que mapear cada classe para um arquivo PSDL e permitindo que ele concentre seus esforços na especificação dos objetos de negócio. Os objetos persistentes da aplicação não precisam conter código para operações de armazenamento ou recuperação. Cada objeto de negócio possui métodos de acesso ao sistema de armazenamento de dados, mas o código interno desses métodos contém simplesmente uma solicitação de serviço ao serviço de persistência proposto.

Um aspecto importante desta proposta de trabalho é a utilização do repositório de interfaces. Explorando o conteúdo do repositório de interfaces todas as alterações em atributos ou métodos de um objeto são imediatamente refletidas em suas interfaces e, em consequência, são automaticamente visíveis para o serviço de persistência. Desta maneira, uma recompilação do serviço de persistência não se faz necessária, o que não ocorreria no caso de uma implementação do PSS.

No entanto, tem-se consciência de que o acesso dinâmico à interface do objeto demanda um certo tempo, o que pode tornar o serviço proposto mais lento quando comparados aos serviços de persistência que acessam objetos pré-definidos no processo de compilação. É objetivo deste trabalho analisar este desempenho através de comparações entre o tempo de acesso estático e o acesso dinâmico, todavia é válido lembrar que, com a tecnologia de processamento disponível atualmente no mercado, afirmar algo sobre performance é muito relativo.

### 4. Conclusão

Este artigo relatou o projeto para desenvolvimento de um serviço de persistência que se adapte à arquitetura SISCD. Este projeto vale-se de algumas idéias da especificação PSS, como a transparência quanto ao tipo de sistema de armazenamento utilizado, mas procura reduzir a carga de trabalho dos desenvolvedores e facilitar o gerenciamento de mudanças. O serviço foi concebido visando suprimir a responsabilidade do mapeamento de cada classe persistente para um arquivo PSDL, conforme proposto na especificação do PSS, e eliminar a necessidade de geração de código para os objetos persistentes em caso de alteração. Como foi mostrado, pretende-se utilizar o repositório de interfaces, recurso já disponível em implementações CORBA. Enfim, esta é uma nova abordagem de persistência que pode ser uma alternativa para o armazenamento dos objetos em ambientes distribuídos.

### 5. Bibliografia

[1] Ferreira, M. G. V. “Uma arquitetura flexível e dinâmica para objetos distribuídos aplicada ao Software de Controle de Satélites”. Tese de Doutorado. Computação Aplicada. INPE. 2001.

[2] Ferreira, M. G. V. “Um componente gerenciador de dados”. Tese de Mestrado. Computação Aplicada. INPE. 1996.

[3] OMG. “Persistent State Service 2.0”. OMG Document orbos/99-07-07. Agosto, 1999.

[4] Visibroker – “*Visigenic- Programmer’s Guide*”. Version 3.2. 1998. 235 p.