

# Uma Abordagem para Objetos Distribuídos Aplicada ao Simulador de Satélites

Luciana Akemi Burgareli

Orientador: Dr. Prof. Mauricio Gonçalves Ferreira

[luciana@iae.cta.br](mailto:luciana@iae.cta.br), [mauricio@ccs.inpe.br](mailto:mauricio@ccs.inpe.br)

## Resumo

*Na tentativa de melhor aproveitar as características da distribuição de objetos, técnicas de como melhor distribuir os objetos são apresentadas e aplicadas aos subsistemas do Simulador de Satélites do INPE. A computação distribuída vem rapidamente ganhando espaço no campo da informática, entretanto, existem poucas técnicas a fim de projetar e modelar os sistemas de objetos distribuídos. É observada uma deficiência no aspecto de como modelar corretamente esta distribuição, ou seja, como melhor distribuir os objetos de acordo com as características da aplicação e necessidades do usuário, explorando-se mais ativamente as vantagens desta tecnologia e alcançando-se assim, maior qualidade. Com o objetivo de melhorar a organização da distribuição dos objetos, são apresentadas técnicas de modelagens baseadas na análise do comportamento dos objetos, na aplicação e nas necessidades dos usuários.*

## Abstract

*The purpose of this paper is to present a study of a distributed object application to the Satellites Simulator, present in INPE. The distributed computing is rapidly gaining more importance in the computer field, however, there are a few tools available to design and model the distributed object system. A deficiency is noted in the process of correctly modeling this distribution, that is, how to improve the distribution according to the application characteristics and the user's necessity, exploiting more actively this technology advantages and obtaining a better quality this way. Therefore, besides exploring characteristics of object distribution, the intention is to build up an analysis through the object distribution modeling, noting the user needs and aspects of the application. For this purpose, some techniques that will establish criteria and different methods of modeling will be proposed.*

## 1. Introdução

Atualmente, as metodologias voltadas para os objetos distribuídos apenas se preocupam com a distribuição, e não

apresentam métodos de modelagens para melhor projetar esta distribuição e assim organizar estes objetos.

Percebe-se que, analisando o comportamento dos objetos, pode-se realizar um planejamento para distribuí-los de forma mais adequada, fazendo com que a modelagem traga benefícios à aplicação. O objetivo é criar possibilidades de se obter melhores resultados apresentados pela aplicação, advindos desta nova organização dos objetos.

Na tentativa de buscar uma melhor organização da distribuição, são apresentadas técnicas de modelagens que são baseadas na análise do comportamento dos objetos, na aplicação e nas necessidades dos usuários.

O caso de uso considerado é um dos serviços apresentados pelo *Software* de Controle de Satélites Distribuído, também denominado SICSD, do Instituto Nacional de Pesquisas Espaciais – INPE. O SICSD explora os recursos da distribuição, aplicando uma arquitetura flexível e dinâmica para objetos distribuídos.

Um dos subsistemas da arquitetura SICSD, ao qual serão empregadas as técnicas de modelagens, é o Simulador de Satélites. Esta ferramenta permite criar um ambiente operacional realístico, reproduzindo com fidelidade cada fase prevista de vida útil do satélite [10], auxiliando no desenvolvimento e validação de procedimentos de operação e controle do mesmo.

Transparência, flexibilidade, confiabilidade, performance, disponibilidade, economia no custo da implantação e facilidade de expansão são as características apresentadas pela distribuição que eliminam as limitações, antes encontradas no Simulador de Satélites, já que o mesmo se apresentava atuando em sistema centralizado.

No intuito de se obter melhorias nestas características proporcionadas pela distribuição, serão aplicadas aos objetos do Simulador de Satélites, as técnicas de modelagens aqui apresentadas.

## 2. Trabalhos Relacionados

Nos últimos anos, empresas desenvolvedoras de *software* vêm lançando no mercado vários novos padrões a serem aplicados aos sistemas de objetos distribuídos. Com relação à distribuição de objetos, estes padrões se tornam, a cada ano, mais abrangentes e eficientes.

Assim, a computação distribuída vem rapidamente ganhando espaço no campo da informática, entretanto, existem poucas técnicas a fim de projetar e modelar os sistemas de objetos distribuídos. É observada uma deficiência no aspecto de como modelar corretamente esta distribuição, ou seja, como melhor distribuir os objetos de acordo com as características da aplicação e necessidades do usuário, explorando-se mais ativamente as vantagens desta tecnologia e alcançando-se assim, maior qualidade.

Uma larga escala destes sistemas vem sendo afetada por esta deficiência [2], fazendo com que autores se preocupem com este fato, como pode-se evidenciar em [2],[5] e [11]:

Butler [2] introduz sua modelagem dividida por domínios computacionais, onde cada domínio possui um ambiente executável independente. Os objetos são caracterizados pela quantidade de recursos requeridos por seus atributos e métodos. A técnica possibilita um objeto requisitar um dado de um domínio e um outro objeto receber o resultado. O objetivo é suportar projeto e desenvolvimento em larga escala de sistemas complexos utilizando a tecnologia de objetos e possibilitar estudos específicos através deste modelo.

Kalogeraki [5] propõe em sua modelagem um algoritmo que determina o grau inicial de replicação dos objetos, a fim de alocar recursos de forma eficiente, melhorando assim, o desempenho do sistema. O algoritmo baseia-se na importância da replicação do objeto e recursos como memória e CPU, tendo como meta evitar prejudicar a qualidade do serviço quando os recursos são limitados e o número de réplicas de objetos tendem a aumentar.

Sessions [11] apresenta regras básicas para distribuição, por exemplo, a escolha do padrão a ser utilizado, considerações importantes para distribuir os objetos, regras de escrita de código, etc.

### 3. Sistemas Distribuídos

Atualmente, em um ambiente corporativo, a utilização do paradigma Cliente/Servidor se torna cada vez mais indispensável para as aplicações modernas. O cliente, hoje, possui processador com maior capacidade de processamento, provendo total autonomia ao usuário, podendo atender perfeitamente as solicitações de um segundo cliente sempre que necessário, agindo, desta forma, como um servidor. Por outro lado, o Servidor, para prover resultados a uma determinada chamada do cliente, pode buscar informações em outros servidores da rede, representando assim, um cliente. Portanto, observa-se que a filosofia Cliente/ Servidor presente neste ambiente é marcada pelo fato de clientes e servidores não assumirem papéis únicos, ou seja, todos os computadores da rede podem tornar-se clientes e servidores, dependendo da aplicação.

O ambiente descrito acima é denominado de Sistema Distribuído, onde duas ou mais máquinas interligadas cooperam entre si, sem memória compartilhada, e executam tarefas em paralelo, oferecendo um melhor desempenho às aplicações, e aparentando a seus usuários, um único computador, proporcionando, desta forma, um ambiente cuja distribuição física poderá ser transparente ao usuário final.

Sistemas distribuídos, integração de máquinas em diferentes arquiteturas e sistemas operacionais, Internet e outras tecnologias modernas são realidades evidentes na informática atual. Para se desenvolver softwares em novos ambientes como estes, uma tecnologia muito empregada é a Orientação a Objetos [8].

A orientação a objetos vem se consagrando na computação moderna, onde o software é organizado como uma coleção de objetos nos quais armazenam-se seus dados e estrutura lógica que dita seu comportamento.

A união destas duas tecnologias: Sistemas Distribuídos e Orientação a Objetos origina a área de Objetos Distribuídos.

### 4. Objetos Distribuídos

Objeto distribuído é a evolução do objeto convencional. Para que estes objetos possam ser acessados por outros objetos locais ou remotos, estes devem possuir uma interface específica, onde os compiladores geram um código especial. O programa que solicita o objeto desconhece o local onde este objeto chamado está instanciado, detalhes de implementação do mesmo ou o sistema operacional que estiver sendo utilizado.

A “ponte” que interliga estes códigos especiais, a fim de estabelecer a comunicação dos objetos, é denominada Middleware.

Middleware é um software de conectividade que consiste em um conjunto de serviços que permite a interação, através da rede, de múltiplos processos executando em uma ou mais máquinas [3].

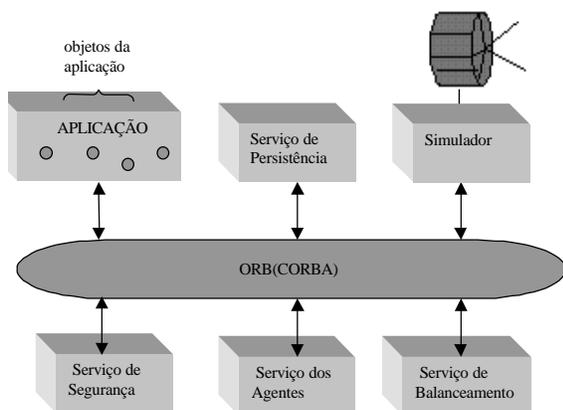
As características apresentadas pela distribuição de objetos como: transparência, tolerância a falhas, disponibilidade, recuperabilidade, concorrência no processamento, concorrência de acesso aos objetos e melhor desempenho; comprovam a importância desta tecnologia.

Atualmente, vários estudos e sistemas estão sendo desenvolvidos baseados em objetos distribuídos, utilizando ferramentas como: CORBA (Common Object Request Broker Architecture) da OMG (Object Management Group), DCOM (Distributed Component Object Model) da Microsoft e o RMI (Remote Method Invocation) da Sun Microsystems que são padrões que oferecem a arquitetura necessária para se trabalhar com esta tecnologia.

## 5. Aplicação : A arquitetura SICSD

O objetivo da arquitetura SICSD consiste em apresentar uma aplicação distribuída, aplicada ao *software* de controle de satélites, onde seus objetos se apresentam de forma dinâmica, ou seja, podem migrar de um nó para outro, se adaptando às solicitações de serviços, melhorando um conjunto de características, como desempenho, flexibilidade, confiabilidade e utilização dos recursos computacionais disponíveis. Esta arquitetura apresenta seus objetos se comunicando através de um “*middleware*”, que implementa a especificação CORBA.

A figura 1 apresenta uma visão da arquitetura SICSD e seus serviços disponíveis.



**Figura 1 - Uma visão dos serviços da arquitetura SICSD. - Fonte: Ferreira [4].**

- Serviço dos agentes: Gerencia as informações sobre objetos e nós. Monitora os estados mantendo atualizadas e distribuídas as informações necessárias para se desenvolver uma aplicação dinâmica e flexível para controle de satélites.

- Serviço de Balanceamento: Este serviço é capaz de levantar os pontos críticos do ambiente configurado para controle de satélites, por exemplo, identificar os nós que contêm uma sobrecarga de trabalho, tomando decisões que podem resultar na migração ou na replicação dos objetos da aplicação para controle de satélites.

- Serviço de Persistência: Este serviço tem como objetivo armazenar/recuperar os objetos persistentes da aplicação.

- Serviço de Segurança: Responsabiliza-se por garantir o acesso ao sistema somente de pessoas previamente autorizadas, e garantir que somente objetos autorizados possam migrar para máquinas remotas.

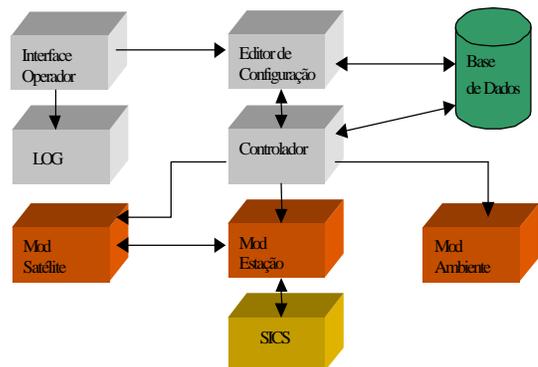
- Simulador de Satélites: Objeto responsável por simular os possíveis estados de um satélite, ou seja as

possíveis condições internas que ele apresenta em um determinado instante.

### 5.1. O Simulador de Satélites

A finalidade principal do Simulador de Satélites é permitir treinar os operadores quanto ao controle e rastreamento do mesmo, e prover um ambiente real que possa ser utilizado para elaboração dos testes de aceitação do *software* de sistema de controle de satélites [10].

A arquitetura do Simulador de Satélites, pode ser observada na figura seguinte:



**Figura 2 - Arquitetura do Simulador de Satélites  
Fonte Adaptada: - Rozenfeld, Miguez, Orlando [10].**

O Simulador de Satélites é operado através da *Interface Operador* em um terminal exclusivo onde o Simulador de Satélites é inicializado e, a partir dele, o responsável pela simulação deve poder acompanhar o andamento do processo, monitorando seu estado e interagindo com o mesmo a fim de exercer ações de controle ou provocar falhas. É através do *Editor de Configuração* que se edita a *Base de Dados* do sistema. Nesta base de dados se encontram: itens de identificação, senha, e atributos do usuário; telas que permitem visualizar e monitorar estados correntes da simulação, parâmetros que definem este estado e ainda as falhas simuladas.

As ocorrências de eventos importantes são registrados num arquivo de *LOG*, que pode ser consultado durante o processo de simulação.

Os *Modelos Satélite, Estação e Ambiente* são responsáveis respectivamente por simular as funções dos subsistemas do satélite, as estações de rastreamento e controle do satélite e o ambiente no qual o satélite opera.

A troca de mensagens com o SICS ocorre na forma padrão, não havendo, por parte do controlador de satélites do SICS como distinguir a operação simulada da real.

## 6. Trabalho Proposto

O Simulador de Satélites atual apresenta algumas limitações decorrentes do fato de ser uma aplicação atuante em um sistema centralizado. Um dos objetivos deste trabalho é utilizar o recurso da computação distribuída através de suas propriedades inovadoras, aplicando-as ao Simulador, tentando assim, eliminar as limitações por ele apresentadas.

Um sistema complexo de software que apresenta qualidade, além de funcionalidade, exibe uma harmonia que o torna flexível a modificações. Esta harmonia surge, muitas vezes pelo fato de seguir alguns padrões já consagrados. Assim, constata-se que, a maioria dos projetos bem sucedidos é semelhante em diversos aspectos, apresentando muitos elementos que contribuem para este sucesso, entre eles, a utilização de padrões e modelagens.

Neste contexto, observa-se a importância em se obter um estudo focando a modelagem da distribuição, ou seja, uma tentativa de se estabelecer técnicas de como melhor distribuir os objetos, a fim de se acentuar os benefícios conseguidos com a distribuição.

Para buscar uma melhor organização dos objetos na aplicação, alguns aspectos importantes devem ser analisados, como por exemplo:

- a função exercida e o comportamento dos objetos diante da aplicação;
- fatores individuais de cada objeto ou quando relacionados a outros objetos;
- aspectos determinantes da aplicação;
- necessidades dos usuários.

A partir desta análise, idealiza-se o levantamento de algumas técnicas. As técnicas aqui propostas estabelecem critérios e diferentes formas de modelagens. A iniciativa não tem como objetivo apontar qual a melhor técnica a ser utilizada. Os esforços se concentrarão apenas em se levantar as vantagens e desvantagens das técnicas apresentadas, explorando fatores qualitativos e quantitativos das mesmas e classificando-as de acordo com resultados obtidos num estudo comparativo, que envolve desempenho, disponibilidade e diminuição de tráfego, etc.

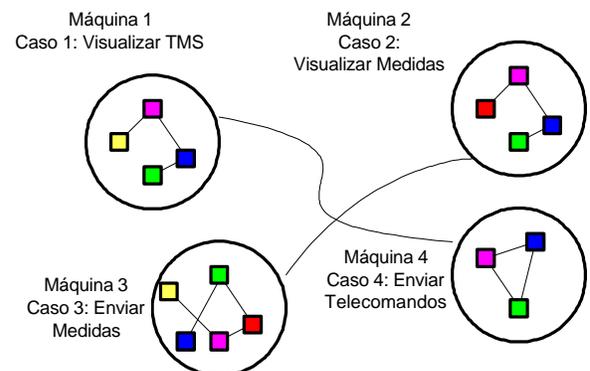
### 6.1. Eliminação das limitações do Simulador de Satélites

- **O aumento da disponibilidade de serviços:** A tecnologia de objetos distribuídos provê mecanismos que asseguram a disponibilidade dos objetos, independente de falhas nos computadores [4].  
**Tolerância à falhas:** A falha de um computador ou objeto, em um ambiente distribuído representa apenas uma falha parcial do sistema, que pode ser superada,

através de novas conexões com objetos que prestam serviços equivalentes.

- **Interoperabilidade:** Esta importante característica provê que objetos distribuídos possam invocar outros métodos mesmo quando plataformas, sistemas operacionais e linguagens de programação forem heterogêneas. Isto solucionaria as restrições impostas pela homogeneidade existente no Simulador de Satélites.
- **O aumento da concorrência e desempenho:** A capacidade de instanciar cópias de um mesmo objeto, Satélites em máquinas distintas, proporciona um melhor desempenho ao atendimento a requisições de múltiplos usuários. Um fator que deve ser levado em consideração para explorar o recurso de concorrência em sistemas distribuídos, ou seja, dois ou mais usuários podem solicitar o mesmo serviço ao sistema, mas sendo atendidos por objetos instanciados em diferentes nós.
- **Flexibilidade para atender às diferentes situações de controle:** Com a aplicação de objetos distribuídos no sistema do Simulador de Satélites, pode-se distribuir partes do processo de simulação, ou até mesmo replicá-lo totalmente em máquinas distintas.

### 6.2. Modelagens da Distribuição baseada em objetos que se colaboram para realizar um Caso de Uso



**Figura 4 –Objetos Distribuídos por modelagem baseada em objetos que se colaboram para realizar um Caso de Uso**

Na figura 4, observa-se os objetos que se colaboram para a realização de um caso de uso, a circunferência está delimitando esta colaboração. As cores equivalentes caracterizam instância de um mesmo objeto.

Em muitas aplicações, em determinados momentos, percebe-se uma interdependência existente entre objetos que não deve ser quebrada. É o que ocorre com alguns

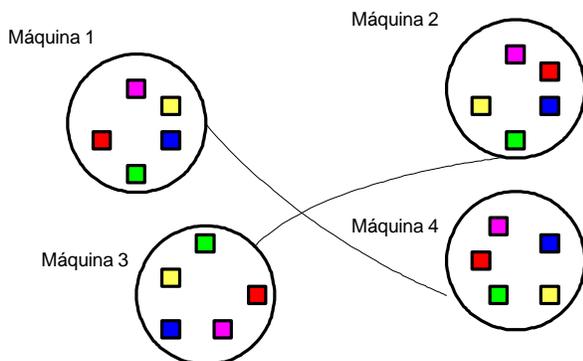
objetos observados no Simulador de Satélite, como os objetos Telecomando e Telemetria.

O objeto Telecomando representa mensagens que podem ser enviadas para o satélite, com finalidade de corrigir ou mudar posições de chaves, ligar e desligar sensores, já o objeto Telemetria demonstra o estado interno do satélite, voltagens, temperatura, ou seja, condições dos equipamentos do satélite. Quando um objeto telecomando é recebido do cliente, ao ser tratado, reflete quase que imediatamente no frame de telemetria enviado ao SICSD. Observa-se então, que freqüentemente um objeto telecomando atua e afeta o objeto telemetria, assim, deixá-los trabalhando próximos ou até mesmo na mesma máquina, pode trazer algumas vantagens, se obedecendo a um determinado critério.

Este critério para distribuição de objetos pode ser conseguido analisando-se os subsistemas do Simulador de Satélites e fazendo com que objetos que se comunicam mais freqüentemente entre si, participem da realização de um mesmo Caso de Uso. Estes objetos então, relacionados por Caso de Uso, serão criados numa mesma máquina. Com isso, se diminuem os relacionamentos entre os Casos de Uso e conseqüentemente entre as máquinas. Este fato poderá proporcionar vantagens, reduzir o tráfego na rede e se obter maior disponibilidade. Por exemplo, se ocorrer uma falha em um determinado objeto de um Caso de Uso, este não prejudicará um segundo caso.

Não existe impedimento em se ter duplicidade de Casos de Uso em máquinas distintas, pois, sendo cópia de um Caso de Uso, o conjunto de objetos proporcionará as mesmas características que o Caso de Uso de origem.

### 6.3. Modelagem da Distribuição baseada em Tolerância a Falhas



**Figura 5 –Objetos Distribuídos por modelagem baseada em Tolerância a Falhas**

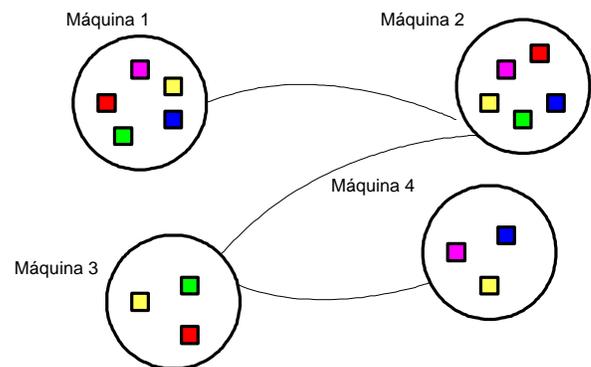
Um segundo aspecto a ser abordado, na tentativa de se encontrar opções de modelagem para uma melhor distribuição, é proporcionar tolerância a falha de objetos. Focando nesta propriedade, o novo critério estabelecido, é replicar os objetos, independente do tipo de serviço

executado, em todas as máquinas existentes no sistema. Neste caso, a sobrecarga das máquinas se torna inevitável, porém se assegura fortemente a disponibilidade de determinado objeto. A figura 5 demonstra como este sistema se apresenta.

### 6.4 –Modelagem da Distribuição baseada na Forma Aleatória

A terceira forma apresentada consiste na modelagem aleatória.

Esta técnica é equivalente ao método tradicional, exceto por obedecer a um único critério imposto pela modelagem, que estabelece a existência de pelo menos uma cópia de um mesmo objeto no sistema. A figura a seguir ilustra esta distribuição:



**Figura 6 –Objetos Distribuídos por modelagem baseada na Forma Aleatória.**

### 6.5. Modelagem Baseada em uma Distribuição Dinâmica

Uma outra modelagem apresentada é a baseada em uma distribuição dinâmica, onde os objetos podem migrar ou replicar automaticamente de uma máquina para outra. Esta atividade é executada pelo serviço de Balanceamento, proposto pelo SICSD, já citado anteriormente. O serviço de Balanceamento realiza a distribuição dinâmica de acordo com a disponibilidade da CPU ou com o número de conexões entre objetos e máquinas.

É responsabilidade deste serviço garantir os recursos de mobilidade, a flexibilidade e o dinamismo da arquitetura proposta. Por exemplo, se há um número elevado de objetos instanciados em um determinado nó, o serviço de balanceamento é responsável pela migração de objetos de máquinas saturadas para máquinas ociosas. Um segundo caso seria se, objetos instanciados em determinado nó estiverem recebendo várias solicitações de serviços de outros nós da rede, o serviço de balanceamento deve replicar os objetos de um nó saturado em nós ociosos.

O objetivo desta modelagem se concentra em observar como se comportam os objetos distribuídos dinamicamente.

Pretende-se, também desenvolver uma análise comparativa a respeito do desempenho das formas de modelagem apresentadas.

## 7. Implementação do Protótipo

Um protótipo do Simulador de Satélite já está sendo implementado, utilizando para a distribuição dos objetos, o padrão Java RMI (*Remote Method Invocation*), da *SunMicrosystem*. Nos primeiros subsistemas desenvolvidos do Simulador de Satélites, é possível observar o relacionamento dos objetos, como já ocorre nas atividades de Envio de Telecomando e Recebimento de Telemetria.

O ambiente utilizado constitui-se de 5 computadores nos quais os objetos são distribuídos com o auxílio de um Gerenciador de Cenários. Este Gerenciador de Cenários consiste em um *software*, criado também para este ambiente, onde cria-se ou destrói os objetos nas máquinas disponíveis de acordo com o tipo de modelagem estabelecida (tolerância a falhas, aleatória, por caso de uso, etc). Através dele, pode-se gerenciar também a disponibilidade da CPU e o número de conexões entre máquina e objetos.

No ambiente dinâmico, atividades como a de migração e replicação dos objetos já foram desenvolvidas. Estas atividades, que fazem parte do Serviço de Balanceamento, proposto pela arquitetura SICSD, possibilita observar como os objetos se comportam também neste ambiente.

## 8. Considerações Finais

Ao empregar objetos distribuídos aos subsistemas do Simulador de Satélites, pretende-se estender a ele, propriedades que eliminarão diversas limitações existente atualmente, no que diz respeito à disponibilidade, transparência, tolerância a falhas, etc.

Algumas situações não tão positivas, como interferência em desempenho e sobrecarga de máquinas, decorrentes da aplicação da distribuição, devem ser previstas. Estas possibilidades deverão ser analisadas e caracterizadas também como uma contribuição ao estudo a ser realizado. Desta forma, a análise do comportamento do sistema do Simulador de Satélites, a partir da introdução destes novos conceitos, resultará em dados significativos que colaborarão para o estudo dos objetos distribuídos nesta aplicação.

Atualmente, ao se decidir trabalhar com distribuição de objetos, não é considerado nenhum método de como melhor organizar e distribuir estes objetos entre as máquinas da aplicação. Em geral, as empresas apenas

fornecem padrões para distribuição, e não apresentam técnicas de modelagem para este fim.

Portanto, pretende-se aplicar ao Simulador de Satélites, as técnicas de modelagens para a distribuição de objetos que foram apresentadas neste texto. O objetivo é prover melhorias às características advindas da distribuição

Será possível assim, após a aplicação destas técnicas, realizar uma análise qualitativa e quantitativa dos resultados. Espera-se, desta forma, que esta análise aponte a necessidade de se modelar corretamente a distribuição dos objetos para que se obtenha ganhos importantes com relação a eficiência da aplicação.

## 9. Referências

- [1] - Ahuja, S.; Quintao, R.; "Performance Evaluation of Java RMI: A Distributed Object Architecture for Internet Based Application", IEEE, 2000.
  - [2] - Butler, J. M.; "Quantum Modeling of Distributed Object Computing", IEEE, 1995.
  - [3] - Costa, S. R.; "Objetos Distribuídos: Conceitos e Padrões", Dissertação de Mestrado, Computação Aplicada, INPE, 2000.
  - [4] - 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.
  - [5] - Kalogeraki, V.; Moser, L. E.; Melliar-Smith, P. M.; "Dynamic Modeling of Replicated Objects for Dependable Soft Real-Time Distributed Object System", IEEE, 1999.
  - [6] - Kono, K.; Masuda, T.; "Efficient RMI: Dynamic of Object Serialization", IEEE, 2000.
  - [7] - Minetti, S, Jr.; "Objetos Distribuídos", Visionnaire, Curitiba, junho de 1997.
  - [8] - Mowbray, T. J.; Ruh, W. A., *Inside Corba - Distributed Object Standards and Applications*, 1997.
  - [9] - Orfali, R.; Harquey, D.; Edwards, J.; *The Essential Distributed Objects Survival Guide*, 1996.
  - [10] - Rozenfeld, P.; Miguez, R.; Orlando, V.; "Proposta de um Simulador para o Satélite SCD1", INPE, 1990.
  - [11] - Sessions, R.; "Ten Rules for Distributed Object Systems", OS/2 Magazine, October, 1996.
- [http://www.objectwatch.com/articles\\_by\\_staff.htm](http://www.objectwatch.com/articles_by_staff.htm)