

# Integração de Logs em Ambientes Distribuídos utilizando Gerenciamento OSI e Modelagem de Objetos \*

Guido Elwhin Peñaranda Silva Junior <sup>†</sup>      José Marcos Silva Nogueira

Departamento de Ciência da Computação  
Universidade Federal de Minas Gerais  
30161-970 CP: 701 Belo Horizonte, Minas Gerais  
E-mail: {guido,jmarcos}@dcc.ufmg.br  
<http://www.dcc.ufmg.br/~guido,~jmarcos>

## Resumo

Este artigo mostra a concepção e implementação de um Sistema de Integração de Logs (SIL). É uma ferramenta computacional que resolve o problema da integração de informações de monitoração em ambientes distribuídos. São explicados os conceitos da metodologia de desenvolvimento, chamada *Técnica de Modelagem de Objetos* e mostra como eles foram aplicados nas fases de análise e projeto do SIL. Além disso, é mostrado como foi feita a implementação do SIL usando tecnologias de ponta, tais como plataformas de gerenciamento de redes baseadas no modelo OSI, sistemas de interfaces gráficas baseados em eventos e ferramentas avançadas do sistema operacional UNIX.

## Abstract

This article explains the conception and implementation of a System for Integration of Logs (SIL). It's a computational tool that resolve the problem of integrating monitoring information on distributed environments. The concepts of the development methodology named *Modeling Object Technique* and how these concepts were applied to analys and project phases of SIL are explained. Besides, how modern technologies as network management plataforms based on OSI model, user interface systems based on events and advanced tools of UNIX operating system were used on the implementation of SIL.

## 1 Introdução

No gerenciamento de ambientes distribuídos é necessário analisar uma grande quantidade e variedade de informações de monitoração, tanto do próprio ambiente quanto das aplicações

\*Trabalho financiado pela TELEMIG S.A em convênio com o DCC/UFMG

<sup>†</sup>Aluno de Mestrado no DCC/UFMG e analista de sistema do projeto SIS

que estão sendo executadas sobre ele. Entretanto, a manipulação (localizar, acessar, correlacionar, etc...) dessas informações, nestes ambientes, é difícil. Isto se deve aos seguintes fatores, entre outros:

- As informações de monitoração estão espalhadas por toda a rede, sendo difícil localizá-las rapidamente. Isto prejudica um gerenciamento efetivo, onde é comum a necessidade imediata da informação para se tomar decisões rápidas.
- Cada ferramenta de monitoração pode utilizar uma modelagem particular da informação monitorada. Por exemplo, algumas reportam a hora em que a informação foi coletada, outras não. Uma consequência disto é que existe uma grande dificuldade para que estas ferramentas se interajam.
- Há uma falta de mecanismos efetivos para o gerenciamento de informações de monitoração referentes a épocas passadas, muito distantes da atual.
- A interação homem-máquina é, em geral, ruim, mostrando grande quantidade de informações desnecessárias, dificultando observar os aspectos relevantes. Além disso, as informações nem sempre são mostradas de uma maneira amigável, por exemplo, em forma gráfica.
- Faltam mecanismos para correlacionar informações de monitoração de natureza diferente.

Percebe-se, assim, a necessidade de uma ferramenta computacional que possibilite manipular facilmente as informações de monitoração em ambientes distribuídos.

Este tipo de ferramenta é importante para o gerenciamento de rede, pois com as informações de monitoração tem-se a **história** do comportamento do ambiente distribuído e das aplicações que executam sobre ele. Na gerência de desempenho, por exemplo, isto é muito útil na descoberta de recursos saturados. Observando-se, ao longo do tempo, parâmetros como taxa de utilização de CPU, atividade de I/O, tempo de resposta das aplicações, etc, pode-se descobrir se há uma tendência em direção à saturação.

O benefício desta ferramenta se dirige aos administradores de ambientes distribuídos, pois através dela poderão manipular facilmente informações de monitoração, obtendo-se, assim, rapidamente, subsídios para tomar decisões.

O problema da integração de informações de monitoração em ambientes distribuídos foi percebido no Laboratório de Redes de Computadores do Departamento de Ciência da Computação da UFMG. Neste laboratório, onde foi desenvolvido e está em processo de evolução a plataforma SIS (Sistema Integrado de Supervisão) [7] tem-se a responsabilidade de manter a integridade da rede para que o SIS execute normalmente suas funções 24 horas por dia. O SIS é um sistema desenvolvido pela parceria UFMG/TELEMIG, na área de telecomunicações, para a integração de sistemas de supervisão da planta da empresa no Estado de Minas Gerais.

Este texto apresenta o SIL (Sistema de Integração de Logs), cujo objetivo é solucionar o problema citado bem como o processo de concepção e implementação. Serão mostrados a metodologia e a tecnologia utilizadas no seu desenvolvimento e implementação.

A estrutura do restante deste texto está organizada da seguinte forma:

Capítulo 2 - descreve o problema a ser resolvido pelo SIL.

Capítulo 3 - introduz os conceitos da metodologia orientada a objetos e o modelo OSI de gerenciamento de redes utilizados no desenvolvimento do SIL.

- Capítulo 4 - define a especificação funcional do SIL.
- Capítulo 5 - mostra a fase de análise no processo de desenvolvimento do SIL.
- Capítulos 6 e 7 - mostram a fase seguinte de desenvolvimento, o projeto do SIL.
- Capítulo 8 - descreve como ambientes e ferramentas de desenvolvimento de software foram utilizados na implementação do SIL.
- Capítulo 9 - mostra um exemplo de aplicação do SIL na área de gerenciamento de desempenho.
- Capítulo 10 - finaliza o texto com conclusões e sugestões de trabalhos futuros.

## 2 Descrição do Problema

No contexto da atividade de monitoração, é muito comum o uso dos termos **Log** e **Registro de Log**. Log é o nome dado à entidade responsável por armazenar e gerenciar as informações de monitoração. Registro de Log é a unidade em que se armazena a informação dentro do log. A maneira como ambos são implementados (banco de dados, arquivo, memória, etc ...) não faz parte do conceito. Assim, a "história" do comportamento de uma entidade qualquer é formada pelo conjunto dos registros de log contidos em um determinado log.

A figura 1 caracteriza o ambiente de atuação do SIL. Ela mostra a existência de vários logs espalhados pelos nodos da rede (na figura, os logs são representados por losangos). Os métodos de monitoração extraem informações de um recurso qualquer<sup>1</sup> e a armazenam nestes logs. Assim, cada log possui um tipo diferente de informação de monitoração.

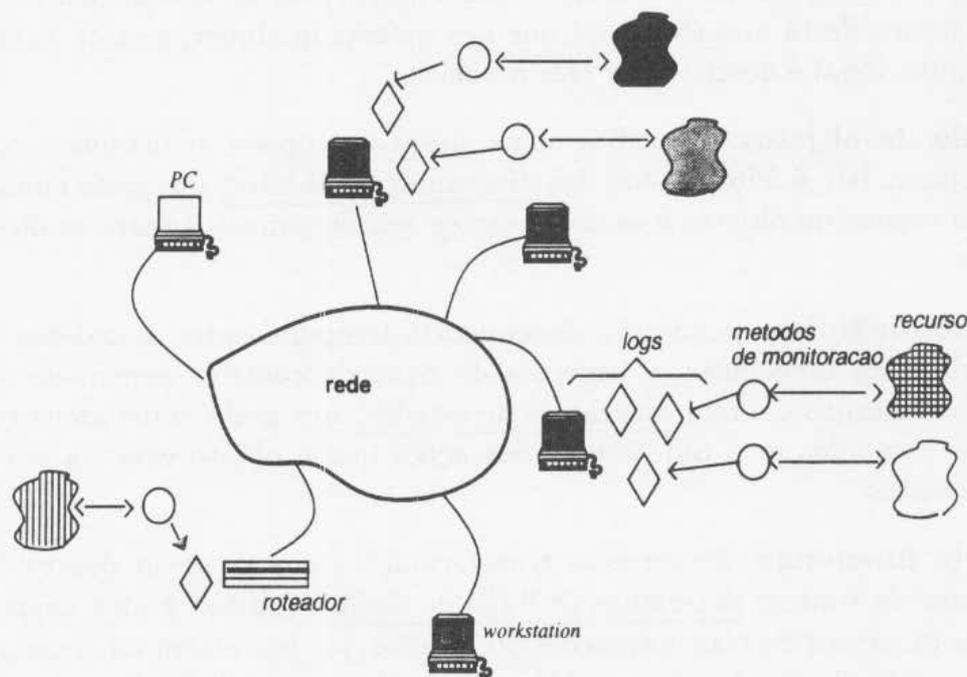


Figura 1: Ambiente do Sistema de Integração de Logs

O problema a ser resolvido pelo SIL é prover um ambiente computacional, que permita

<sup>1</sup>Nem sempre os métodos de monitoração são entidades independentes do recurso. O próprio recurso pode produzir informações de monitoração. Neste caso, pode-se considerar o método de monitoração como a parte do recurso dedicada à monitoração.

manipular, a partir de um único nó, os logs espalhados pela rede. Esta manipulação envolve:

- criar e remover um log em um nó qualquer da rede;
- consultar, remover e atualizar as informações contidas nos logs;
- alterar as características do comportamento dos logs;
- correlacionar as informações de diferentes logs;
- armazenar e recuperar em mídia as informações contidas nos logs;
- imprimir o conteúdo atual dos logs;

### 3 Conceitos

No desenvolvimento do SIL utilizou-se uma metodologia de desenvolvimento orientado a objetos e o modelo OSI de gerenciamento de rede.

#### 3.1 Metodologia de Desenvolvimento Orientado a Objetos

A metodologia empregada neste trabalho chama-se *Técnica de Modelagem de Objetos* [8]. O princípio básico desta metodologia é que um sistema qualquer, seja da natureza ou do mundo computacional é descrito por três modelos:

- **modelo de objetos:** identifica quais objetos compõem o sistema e como eles se relacionam. Isto é feito através dos diagramas de objetos, um grafo cujos nós representam classes de objetos e os arcos são os relacionamentos entre os objetos destas classes.
- **modelo dinâmico:** descreve a dependência temporal entre os objetos do sistema. É usado para especificar os aspectos de fluxo de controle dentro do sistema. O modelo dinâmico contém diagramas de estados, um grafo cujos arcos representam eventos recebidos pelo objeto e os nós ações que o objeto executa em resposta a estes eventos.
- **modelo funcional:** descreve as transformações que ocorrem dentro do sistema. Este modelo contém diagramas de fluxo de dados, que são grafos cujos nós representam processos de transformações ou objetos que produzem ou consomem dados. Os arcos são fluxos de dados. Além disso, descreve detalhadamente as ações do modelo dinâmico.

Os três modelos são partes ortogonais da descrição completa de um sistema sendo interligados através de referência um ao outro. O modelo funcional especifica *o que* acontece, o modelo dinâmico especifica *quando* acontece e o modelo de objetos, *o que* acontece a quem. O desenvolvimento, segundo a técnica, consiste em 3 etapas, todas elas tendo como base a modelagem acima descrita:

- **Análise:** partindo do enunciado do problema, mapeia-se a situação do mundo real em objetos, mostrando as propriedades relevantes. Os objetos deste modelo devem ser conceitos do domínio da aplicação e não conceitos de implementação computacional, como estruturas de dados. Um bom modelo poderá ser compreendido e criticado por peritos que não sejam programadores. O modelo de análise não deve conter quaisquer decisões de implementação. Ele é uma representação precisa e concisa do problema. As etapas subsequentes do desenvolvimento referem-se ao modelo de análise, em vez do original e, muitas vezes, vago enunciado do problema.
- **Projeto do Sistema:** é a estratégia de alto nível para resolver o problema. O projeto do sistema inclui decisões sobre a organização do sistema em subsistemas, a alocação dos subsistemas aos componentes de hardware e software, identificação de concorrências e decisões conceituais formando, assim, a infraestrutura do projeto.
- **Projeto de Objetos:** durante esta fase são elaborados, refinados e otimizados os modelos de análise para a produção de um projeto prático. Ocorre um deslocamento na ênfase dos conceitos da aplicação em direção aos conceitos computacionais. São acrescentados detalhes ao modelo de análise de acordo com a estratégia estabelecida durante o projeto do sistema.

Os objetos do domínio da aplicação e do domínio do computador são descritos através dos mesmos conceitos e notação baseados em objetos, embora pertençam a diferentes planos conceituais. Esta é uma das vantagens de se adotar uma metodologia baseada em objetos, a informação de uma etapa de desenvolvimento não é nunca perdida nem deve sofrer um processo de tradução para ser utilizada nas etapas seguintes. Isto é muito útil em programas sujeitos a mudanças de requisitos funcionais, pois se consegue propagá-las com facilidade pelas fases de desenvolvimento [8].

### 3.2 Modelo OSI de Gerenciamento de Rede

O modelo OSI de gerenciamento de redes, por demais conhecido não será aqui descrito. Entretanto, está mostrado na figura 2 um cenário de gerenciamento.

O modelo usa o paradigma gerentes e agentes, os quais ficam espalhados pela rede. Os agentes têm sob seu controle objetos gerenciados que modelam o ambiente a ser gerenciado. Os gerentes acessam as informações destes objetos emitindo operações de gerenciamento dirigidas aos agentes (serviço CMISE). Informações detalhadas sobre o modelo são encontradas em [1].

## 4 Especificação Funcional do Sistema de Integração de Logs

Alguns **requisitos** para o sistema de integração de logs foram inicialmente definidos como sendo necessários:

- **extensibilidade:** o SIL deve ser *extensível*, pois *a priori* não se sabe todos os tipos de informações de monitoração necessárias ao gerenciamento. Gradualmente surge a necessidade de monitorar novos tipos de informação.

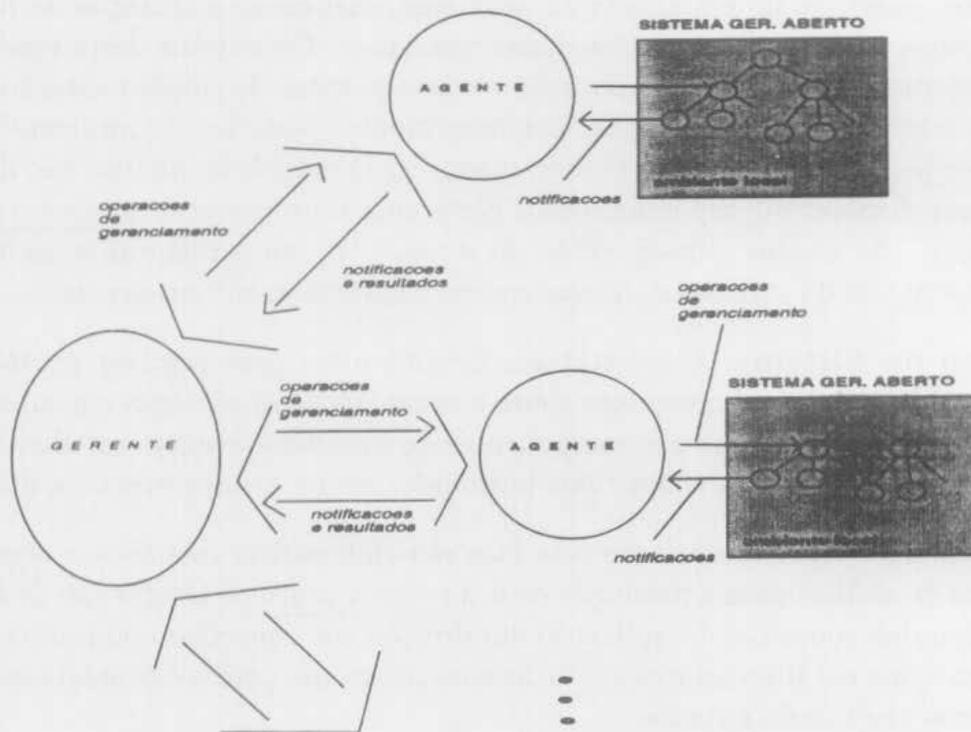


Figura 2: Cenário OSI de gerenciamento

- **escalabilidade:** também, não se sabe, *a priori*, a quantidade de informação de monitoração necessária às atividades de gerenciamento. Portanto o SIL deve ser escalável.
- **interfaces gráficas de alta qualidade:** a atividade de gerenciamento envolve uma grande quantidade e variedade de informações. Assim, o SIL deve ter interfaces gráficas amigáveis voltadas para o usuário final, facilitando a manipulação dessas informações.

As funcionalidades do Sistema de Integração de Logs são divididas em:

- **Criação dos Logs:** prover mecanismos de criação ou remoção de logs em um ou mais nodos da rede. Para isto, é necessário especificar (configurar) as características desejadas para o log, tais como localização, título, estado administrativo, período de monitoração, nome do programa de monitoração que irá criar registros de log dentro dele, etc . Outros dados, como o nome do criador do log, a data da última alteração na configuração do log e o responsável por esta alteração, são preenchidos automaticamente pelo SIL. Fornecidos estes dados, o SIL os armazena para futuras alterações e/ou referências.
- **Manipulação dos Logs:** prover serviços de manipulação das informações contidas nos logs, tais como, consulta, remoção e impressão dos registros do log. Além disso, pode-se atuar no log propriamente dito, como alterar os seus dados de configuração ou removê-lo. Como efeito colateral desta última operação, se o log removido não estiver vazio, os seus registros de log também serão removidos.
- **Correlacionamento entre Logs:** mecanismos para o correlacionamento das informações de monitoração contidas em diferentes logs. Esta correlação pode ser feita

através de gráficos do tipo x,y (*plot*), do tipo “pedaço de torta” (*pie chart*), do tipo área (gráfico do tipo x,y, porém a área entre a curva e o eixo x é preenchida) ou através de uma tabela cronológica (ordenação das informações dos logs em ordem cronológica). Além disso, deve ser possível escolher quais informações dentro dos registros de log participaram da correlação.

- **Criação de registros nos Logs:** mecanismos que possibilitem qualquer método de monitoração criar registros de log dentro de um determinado log.

## 5 Análise do Sistema de Integração de Logs

Esta seção mostra a modelagem do sistema para a fase de análise, em termos dos modelos de objetos, dinâmico e funcional.

### 5.1 Modelo de Objetos

A construção do modelo de objetos baseou-se na figura 1. O resultado da modelagem é mostrado na figura 3. Abaixo segue a descrição dos objetos desta figura<sup>2</sup> e de suas relações:

- **usuário:** usuário do SIL. Possui o atributo *login-name* que o identifica unicamente em toda a rede. A relação entre o objeto “usuário” e o objeto “nodo da rede” representa o fato de que um usuário pode abrir uma sessão em alguns nodos da rede (*login*).
- **nodo da rede:** representa qualquer tipo de equipamento que pode ser conectado à rede. Em particular, em alguns destes equipamentos é possível o usuário abrir uma sessão. Possui o atributo *host-name*, que o identifica unicamente em toda a rede. Um nodo da rede pode ter zero ou mais usuários logados nele, zero ou mais métodos de monitoração executando sobre ele, zero ou mais logs contidos nele e possuir zero ou mais recursos. Além disto, os nodos da rede podem se comunicar.
- **log:** entidade que gerencia as informações de monitoração contidas nela. Os atributos deste objeto coincidem com os atributos da classe padronizada Log [2] e [3]. Um objeto log está contido em um único nodo da rede, pode conter zero ou mais objetos registro de log e é atualizado por um único método de monitoração (veja abaixo).
- **registro de log:** é a unidade básica que compõe um log. Os seus atributos são padronizados pelas normas [2] e [3]. Um registro de log só pode estar contido em um único log e só pode ser criado por um único “método de monitoração”.
- **método de monitoração:** entidade que coleta informação de monitoração de algum recurso. O seu atributo é *nome do software*. Um método de monitoração

<sup>2</sup>A convenção para as extremidades das relações entre as classes de objetos (os arcos) é: círculo preenchido, “um ou mais”; círculo não preenchido, “zero ou mais”, e sem círculo “um”. Por exemplo, um objeto usuário pode estar logado em vários objetos do tipo nodo da rede (círculo preenchido na extremidade do lado nodo da rede). Por outro lado, em um objeto nodo da rede, pode haver zero ou mais usuários logados nele (círculo não preenchido na extremidade do lado usuário).

executa em um único nodo da rede, atualiza um único log, entra em contato com um único recurso e cria zero ou mais registros de log.

- **recurso:** representa a entidade que se está monitorando. Os seus atributos representam as características intrínsecas do recurso. Assim, são específicos para cada tipo de recurso. Um recurso localiza-se em um único nodo da rede e pode ser monitorado por um ou mais métodos de monitoração, cada um observando aspectos diferentes do mesmo recurso.

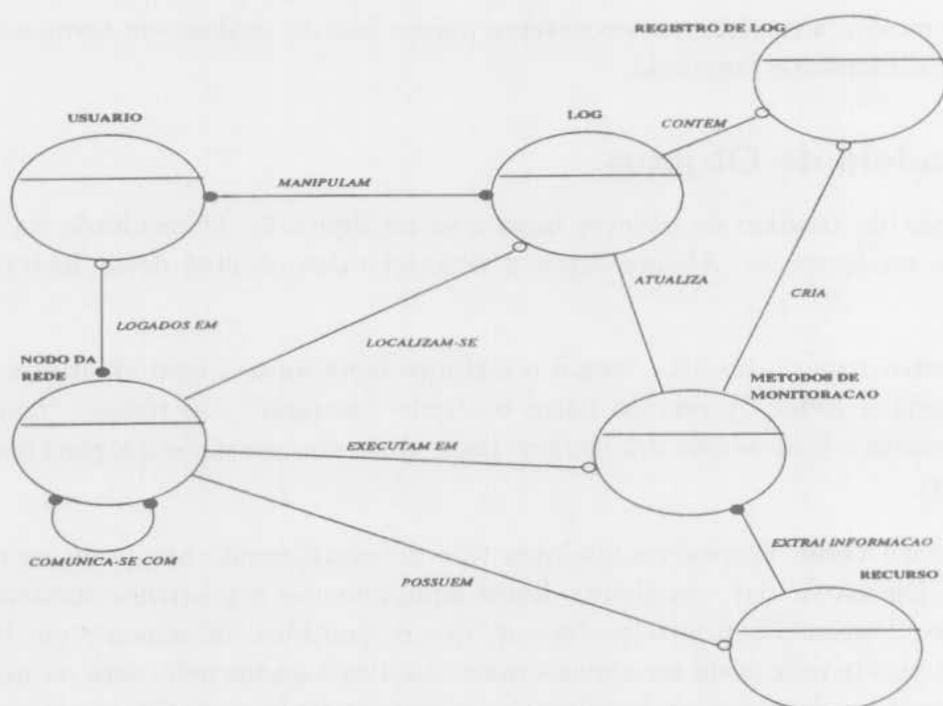


Figura 3: Modelo de Objetos da Fase de Análise

## 5.2 Modelo Dinâmico

Feita a modelagem das entidades do mundo real em objetos, a próxima fase consiste em se determinar o comportamento temporal dos objetos. A figura 4 mostra os principais diagramas de eventos.

Existem duas sequências principais na figura. A primeira mostra o desencadeamento das atividades dentro do sistema, em resposta às requisições de operações de logs feitas pelo usuário (cenário 1). A segunda mostra o desencadeamento provocado no sistema pelo método de monitoração na criação de registros de log dentro de um log (cenário 2). Destas sequências, derivam-se os diagramas de estados dos objetos mostrados na figura 5, e descritos com detalhes na seção seguinte.

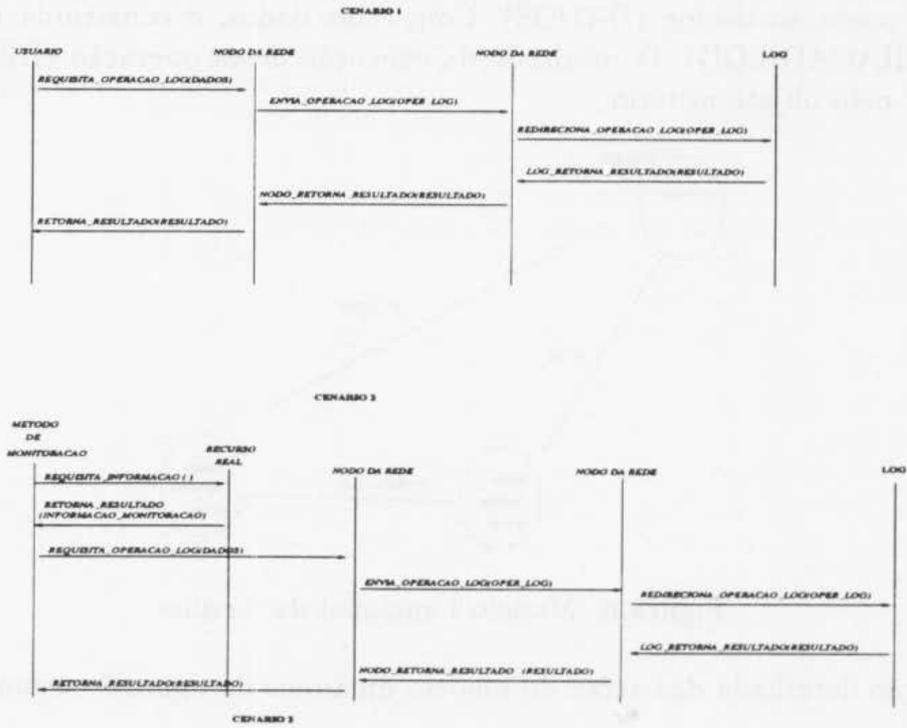


Figura 4: Modelo Dinâmico da Análise

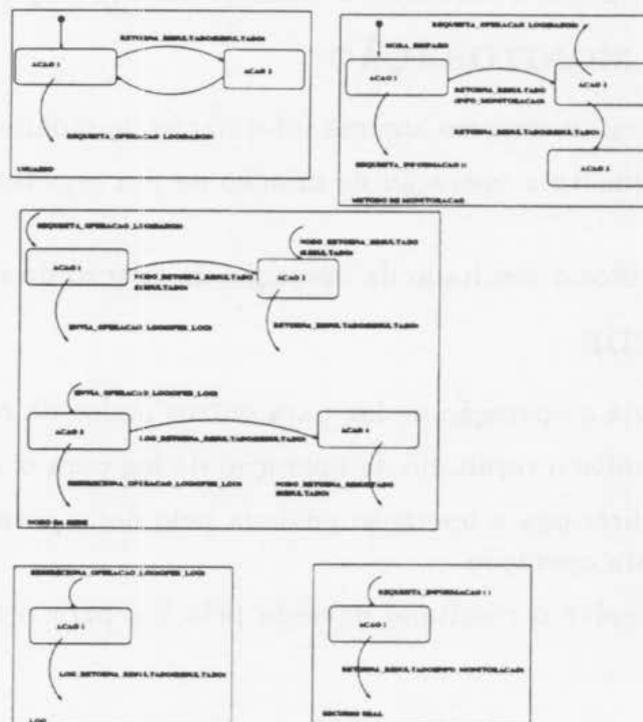


Figura 5: Diagrama de Estados da Análise

### 5.3 Modelo Funcional

A transformação dos dados, no modelo de análise, é mostrada na figura 6. As setas indicam um fluxo de dados de um processo de transformação para outro. O objeto usuário fornece os dados da operação de log (DADOS). Com estes dados, é construída uma operação de log (OPERACAO\_LOG). O resultado da execução desta operação (RESULTADO) é "consumido" pelo objeto usuário.

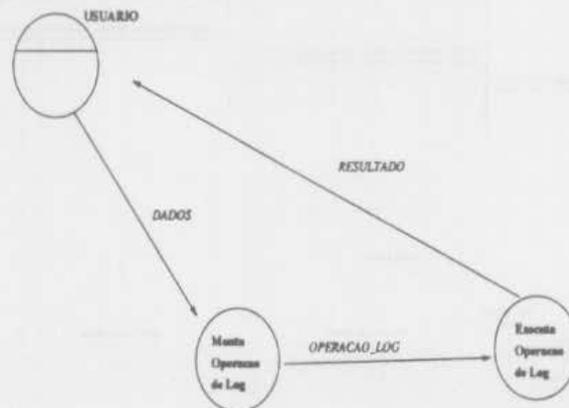


Figura 6: Modelo Funcional da Análise

A descrição detalhada das ações do modelo dinâmico da figura 5, segue abaixo:

#### • USUÁRIO:

- **ação 1:** requisita uma operação envolvendo log. Para isto fornece o tipo de operação e os parâmetros necessários.
- **ação 2:** processa o resultado da operação de log. O usuário observa o resultado da operação, pensa e requisita novamente uma operação envolvendo log.

#### • MÉTODO DE MONITORAÇÃO:

- **ação 1:** extrai do recurso alguma informação de monitoração.
- **ação 2:** requisita a operação de criação de um registro de log dentro de um log.
- **ação 3:** verifica o resultado da operação de criação do registro de log.

#### • NODO DA REDE:

- **ação 1:** envia a operação de log para outros nodos da rede.
- **ação 2:** devolve o resultado da operação de log para o usuário.
- **ação 3:** redireciona a operação enviada pelo nodo para os logs que estão envolvidos nesta operação.
- **ação 4:** devolve o resultado enviado pelo log para o nodo que requisitou a operação.

#### • LOG:

- **ação 1:** executa a operação enviada pelo nodo da rede retornando o resultado.

- **RECURSO:**

- **ação 1:** fornece a informação de monitoração requisitada pelo método de monitoração.

## 6 Projeto do Sistema de Integração de Logs

Esta seção mostra as decisões do projeto de alto nível do SIL.

### 6.1 Subsistemas

A figura 7 mostra os subsistemas identificados no SIL. Os subsistemas que se encontram na direção para onde a flecha aponta fornecem algum tipo de serviço para os subsistemas que se encontram na outra extremidade da flecha. Abaixo, segue a descrição dos serviços que os subsistemas oferecem:

- **Subsistema de Controle de Software:** responsável pelo controle dinâmico do sistema computacional, ou seja, pela sequência de procedimentos que devem ser invocados em resposta a uma ação do usuário na interface gráfica.
- **Subsistema de Conversão de Dados:** provê serviços que transformam os resultados fornecidos pelo subsistema de gerenciamento de rede em um formato adequado para o subsistema de interface gráfica.
- **Subsistema de Interface Gráfica:** provê serviços relacionados à interface gráfica, como gráficos, tabelas, mensagens de ajuda, menus, caixas de diálogo, etc ...
- **Subsistema de Gerenciamento de Rede:** *framework* que implementa as abstrações do modelo OSI de gerenciamento: MIT, notificação, objeto gerenciado, primitivas CMISE, etc ...
- **Subsistema de Controle de Concorrência:** implementa serviços de controle de concorrência no acesso ao subsistema de gerenciamento de rede.
- **Subsistema de Métodos de Monitoração:** métodos de monitoração contendo mecanismos oferecidos pelo SIL para incorporarem as informações coletadas por eles em um determinado log.
- **Subsistema de Banco de Dados de Objetos Gerenciados:** implementa a abstração de um banco de dados orientado a objetos que armazena instâncias de objetos gerenciados. Oferece serviços de acesso, atualização, criação e remoção de objetos gerenciados dentro do banco de dados.
- **Subsistema de Comunicação de Rede:** infra-estrutura de comunicação de rede, ou seja, as primitivas de serviço dos protocolos de comunicação.
- **Subsistema de Classes de Objetos Gerenciados:** classes que implementam a funcionalidade das instâncias dos objetos gerenciados contidos no banco de dados de objetos gerenciados. Eles são acessados pelo subsistema de gerenciamento de rede.



Figura 7: Subsistemas do Sistema de Integração de Logs

- **Subsistema de Sincronização de Tempo:** sincroniza o tempo no ambiente distribuído que se deseja gerenciar. Em particular, este subsistema não está relacionado com nenhum outro, pois é um módulo muito independente que trata de sincronizar o relógio das máquinas na rede.
- **Subsistema de Temporização:** escalona o disparo dos métodos de monitoração.

## 6.2 Arquitetura

A figura 8 mostra a arquitetura do SIL, semelhante a citada em [9]. O componente chamado NME (*network management element*) é o conjunto de software devotado para tarefas de gerenciamento<sup>3</sup>. Ele contém os subsistemas de gerenciamento de redes, controle de concorrência, banco de dados de objetos gerenciados, comunicação de rede e classes de objetos gerenciados. O componente NMA (*network management application*)<sup>4</sup> recebe comandos de um operador via interface gráfica, contacta os NME's espalhados pela rede e a seguir mostra o resultado para o operador. Ele possui os seguintes subsistemas: controle de software, conversão de dados, interface gráfica, gerenciamento de redes, controle de concorrência, comunicação de redes e classes de objetos gerenciados.

Na figura, estão representados ainda o sistema operacional da máquina (SO) onde são executados os NMA's e NME's, os serviços de comunicação da máquina (COM) e outras entidades que podem estar executando (APPL).

<sup>3</sup>Na terminologia OSI conhecido como agente.

<sup>4</sup>Na terminologia OSI conhecido como gerente.

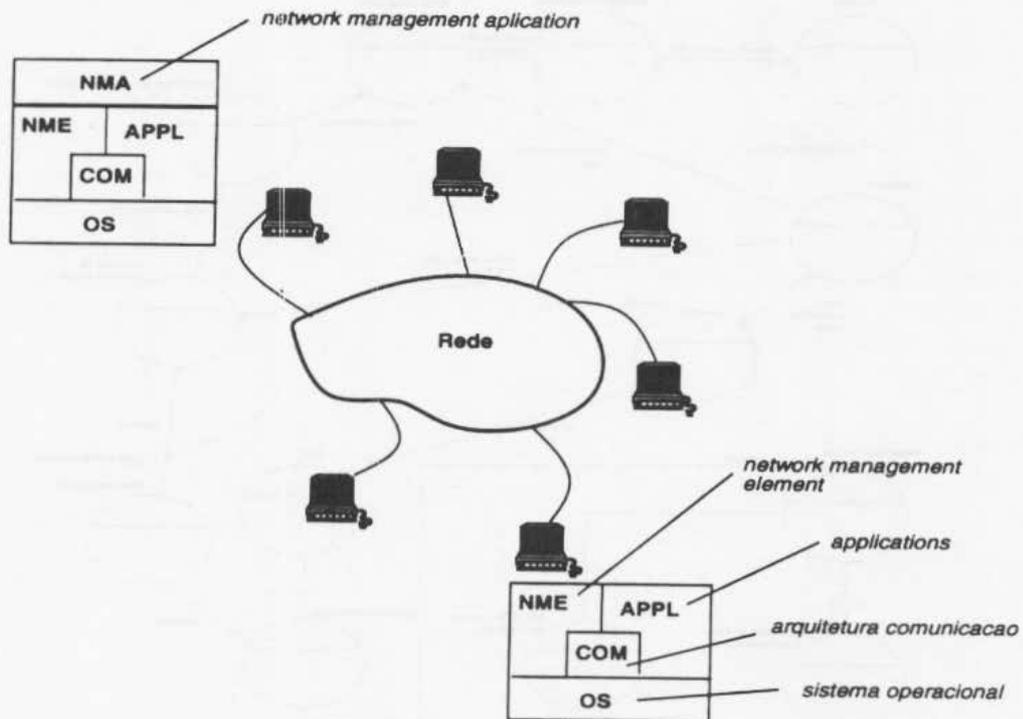


Figura 8: Arquitetura do Sistema de Integração de Logs

## 7 Projeto de Objetos

Esta seção mostra a modelagem do sistema para a fase de projeto, em termos dos modelos de objetos, dinâmico e funcional.

### 7.1 Modelo de Objetos

O objeto “Nodo da Rede” da fase de análise foi desmembrado nos objetos gerente (envia operação CMISE) e agente (recebe operação CMISE). Isto porque, para enviar uma operação de log são necessários dois passos. O primeiro obtém do usuário os dados necessários para a execução da operação e o segundo monta uma operação CMISE. Assim, surgiram os objetos “Interface Gráfica” e “Intermediador”, como mostra a figura 9. Os objetos “Log” e “Registro de Log” foram transportados integralmente para a fase de projeto. Porém, como não possuíam todas as características necessárias para atender a especificação funcional do SIL, foram propostos os novos objetos “LogMonitoração” e “RegistroLogMonitoração”. O objeto “LogMonitoração” é derivado do objeto “Log” acrescentando a este novos atributos tais como, título do log, nome do programa de atualização, nome do operador que criou o log. O objeto “RegistroLogMonitoração” acrescenta ao objeto “Registro de Log” o atributo hora de coleta da informação.

Considera-se, ainda, que o universo da atividade de monitoração constitui-se de:

- **software:** monitoração de aspectos intrínsecos de software. Como em um SGBD, o número de transações pendentes, como em um programa gerenciador de filas, o tamanho médio da fila, etc ...
- **hardware:** monitoração de aspectos intrínsecos de hardware. Como em uma central telefônica, o número de linhas ocupadas, como em um *no-break*, a carga atual da bateria, e assim por diante ...



- “Intermediador”: funciona como um intermediador entre as classes interface gráfica e gerente. Ele recebe os dados do usuário vindos do objeto interface gráfica e constrói uma operação CMISE que é repassada ao objeto gerente. Por outro lado, quando recebe os resultados do gerente converte-os em um formato adequado para a interface gráfica e os repassa para ela.
- “Controle de Concorrência”: controla a concorrência dos objetos “Intermediador” e “Métodos de Monitoração”, no envio de operações CMISE ao objeto “Gerente”.
- “Temporizador”: refere-se à questão de disparar o(s) método(s) de monitoração no(s) período(s) que o usuário deseja.
- “Sincronizador”: sincroniza o tempo no ambiente distribuído a ser gerenciado.

## 7.2 Modelo Dinâmico

Assim como na fase de análise, aqui também será mostrado o comportamento temporal dos objetos da fase de projeto. Nesta fase, porém, os objetos são do mundo computacional e fazem parte da solução do problema.

A construção dos diagramas de eventos segue o mesmo raciocínio da fase de análise e não será mostrado. Apenas os diagramas de estados derivados deles (figura 10).

## 7.3 Modelo Funcional

A transformação dos dados nesta fase é mostrada na figura 11. O objeto interface gráfica captura os dados fornecidos pelo usuário (DADOS\_USUARIO). A seguir, o objeto intermediador monta uma operação CMISE (OPERACAO\_CMISE), que se transforma em uma primitiva CMISE (PRIMITIVA\_CMISE) enviada ao agente.

Este, por sua vez, fornece o resultado da operação CMISE (RESULTADO), que é convertido em uma forma adequada para a interface gráfica (RESULTADO\_CONVERTIDO). Assim, o resultado pode ser mostrado de um forma amigável ao usuário.

A especificação das ações do modelo dinâmico da figura 10 serve de base para a última fase do desenvolvimento: a implementação do SIL. Abaixo seguem as especificações das ações:

### • INTERFACE GRÁFICA:

- **ação 1:** quando o usuário faz uma requisição de operação envolvendo log, captura os dados fornecidos por ele e os envia para o objeto intermediador.
- **ação 2:** recebido o resultado da operação mostra-o de uma forma amigável na interface gráfica (tabela, gráfico, etc ...)

### • INTERMEDIADOR:

- **ação 1:** recebido os dados do usuário, pede autorização ao objeto controle de concorrência para enviar uma operação cmise ao gerente.
- **ação 2:** recebida a autorização do objeto controle de concorrência, monta uma operação CMISE e a envia para o gerente.

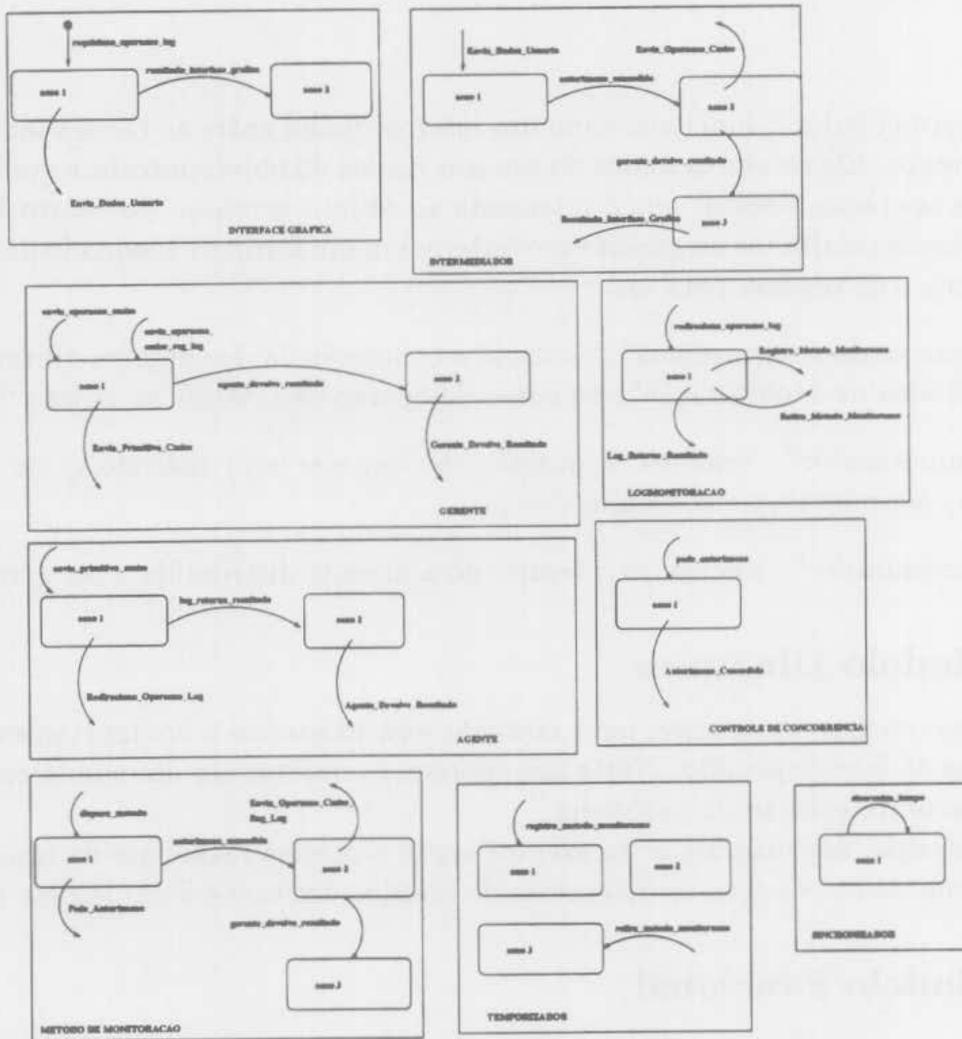


Figura 10: Diagrama de Estados da Fase de Projeto

- **ação 3:** converte e envia o resultado recebido do gerente em um formato adequado para a interface gráfica enviando-o para esta.

#### ● GERENTE

- **ação 1:** envia as primitivas correspondentes as operações CMISE recebidas.
- **ação 2:** devolve o resultado para quem requisitou a operação CMISE.

#### ● AGENTE:

- **ação 1:** redireciona a primitiva CMISE recebida para os logs envolvidos na execução desta primitiva.
- **ação 2:** devolve o resultado recebido pelo log para o gerente, que enviou a operação CMISE.

#### ● LOGMONITORAÇÃO:

- **ação 1:** executa a primitiva CMISE enviada pelo agente, devolvendo o resultado. Se a operação for a criação de um novo log de monitoração, registra no objeto temporizador o escalonamento do método de monitoração. Se a

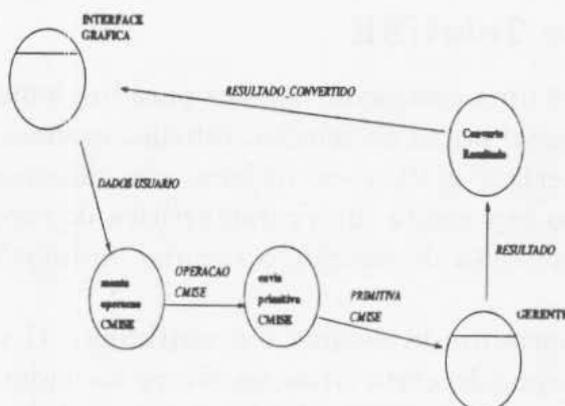


Figura 11: Modelo Funcional do Projeto

operação for a remoção de um log de monitoração retira do objeto temporizador o escalonamento.

- **CONTROLE DE CONCORRÊNCIA:**

- **ação 1:** recebido o pedido de autorização, verifica se alguém está no momento utilizando o gerente. Se estiver, bloqueia quem pediu a autorização até que o gerente esteja livre e então possa ser concedido o pedido. Senão, concede o pedido imediatamente.

- **MÉTODO DE MONITORAÇÃO:**

- **ação 1:** pede autorização ao objeto controle de concorrência para enviar uma operação cmise ao objeto gerente.
- **ação 2:** concedido o pedido, monta e envia uma operação cmise de criação de registros de log ao gerente.
- **ação 3:** processa o resultado da operação de criação do registro de log.

- **TEMPORIZADOR:**

- **ação 1:** escalona a ativação do método de monitoração que está sendo registrado.
- **ação 2:** chegada a hora, dispara o método de monitoração.
- **ação 3:** retira o escalonamento da ativação do método de monitoração.

- **SINCRONIZADOR:**

- **ação 1:** sincroniza de tempos em tempos o ambiente distribuído.

## 8 Ambientes e Ferramentas de Desenvolvimento

Esta seção descreve como os ambientes e ferramentas de desenvolvimento foram usados na implementação do SIL.

## 8.1 O Ambiente TeleUSE

Este ambiente [10] provê uma coleção de objetos para implementar a interface gráfica com o usuário (menus, diálogos, botão de seleção, tabelas, gráficos, etc).

No contexto de interface gráfica, os objetos são chamados **widgets** e os atributos **recursos**. Cada recurso representa uma característica do propósito específico do widget. Por exemplo, no widget botão de seleção, o recurso *messageString* significa a mensagem contida dentro dele.

Um outro tipo importante de recurso é o **callback**. O seu valor é o nome de uma função, que é invocada quando certa situação ocorre no widget. No caso do widget botão de seleção, quando o usuário aperta o botão, é invocada a função definida no recurso *activateCallback*.

Em particular, o conjunto de widgets do TeleUSE segue o padrão Motif [4]. Para este trabalho foi necessário, ainda, integrar ao TeleUSE os widgets de tabela (XRT-table) e gráficos (XRT-graph), desenvolvidos pela empresa canadense KLG Group Inc, pois não se encontravam no conjunto Motif.

Assim, o subsistema de interface gráfica foi implementado com os widgets do TeleUSE e o de controle de software pelos recursos callback.

## 8.2 A Plataforma Q3ADE

O ambiente Q3ADE [11] (Q3 Agent Development Environment), desenvolvido pela empresa dinamarquesa UH Consulting, é um *framework* que implementa as abstrações do modelo OSI de gerenciamento e provê exatamente a arquitetura da figura 2. Este ambiente implementa os subsistemas de gerenciamento de redes e de banco de dados de objetos gerenciados.

No Q3ADE os objetos gerenciados são representados por classes C++. Neste ambiente é possível desenvolver novos objetos gerenciados e incorporá-los aos agentes. Para isto é necessário redefinir métodos padrões das classes C++ que representam os objetos gerenciados. Estes métodos são invocados pelo Q3ADE quando uma operação CMISE é enviada por um gerente.

Assim foram desenvolvidos os métodos das classes C++, que representam os novos objetos propostos neste trabalho (LogMonitoração, RegistroLogMonitoração, RegistroLogMonitoraçãoSoftware e RegistroLogMonitoraçãoHardware). As classes padronizadas pela OSI [2] já vieram implementadas dentro do Q3ADE. Isto completa a implementação do subsistema de classes de objetos gerenciados.

## 8.3 Outros Ambientes e Ferramentas

Na implementação dos outros subsistemas do SIL foram usadas várias ferramentas do UNIX: **CRON** [6] no subsistema de temporização, usado pela classe LogMonitoração para escalonar a ativação do programa de monitoração que irá atualizar o log. **Lex&Yacc** [5] no subsistema de conversão dos resultados fornecidos pelo subsistema de gerenciamento de rede em um formato adequado para os recursos dos widgets de interface gráfica. **Semáforo** no subsistema de controle de concorrência na disputa pelo envio de operações CMISE aos gerentes. C++ no subsistema de classes de objetos gerenciados propostos neste trabalho. C para que os métodos de monitoração pudessem

criar registros de log dentro de um log. **rdate** no subsistema de sincronização do ambiente distribuído. **TCP/IP** no subsistema de comunicação de rede.

## 9 Exemplo de Aplicação

O Sistema de Integração de Logs foi usado na área de gerência de desempenho. Foram configurados logs para a monitoração do número de pacotes que saem das interfaces de rede das máquinas e do tempo de atraso de comunicação na rede sentido por uma aplicação que a utiliza intensamente.

Depois de um certo tempo de monitoração, correlacionou-se os logs e pode-se observar que, à medida que aumentava o número de pacotes que saíam da interface de rede de uma determinada máquina (intencionalmente foi feito nesta máquina um FTP de um arquivo muito grande), o atraso de comunicação na rede sentido pela aplicação aumentava. Assim, utilizando o SIL, pode-se descobrir qual nodo foi o responsável pelo atraso (tarefa que sem o SIL poderia ser bastante demorada em uma rede com muitas máquinas).

## 10 Conclusões

Este texto mostrou a aplicação prática dos conceitos do modelo OSI de gerenciamento de redes e de uma metodologia de desenvolvimento orientado a objetos na solução de um problema específico de gerenciamento de redes: a integração de logs em ambientes distribuídos.

Na implementação do Sistema de Integração de Logs integrou-se diversas tecnologias como linguagens de programação orientadas a objetos (C++), sistemas de interface gráficas (widgets), plataformas de gerenciamento de redes baseadas no modelo OSI, ferramentas do sistema operacional UNIX, etc ...

Como trabalhos futuros, pode-se conceber e implementar um mecanismo de segurança das informações contidas nos logs. Pois, em geral, não é conveniente que todos os usuários tenham acesso a todas as informações de monitoração do ambiente distribuído.

## Referências

- [1] CCITT. Recommendation X.701 (1992) — ISO/IEC 10040: 1992, Information technology – Open Systems Interconnection – Structure of management information: Systems management overview.
- [2] CCITT. Recommendation X.721 (1992) — ISO/IEC 10165-2 : 1992, Information technology – Open Systems Interconnection – Structure of management information: Definition of management information.
- [3] CCITT. Recommendation X.735 (1992) — ISO/IEC 10164-6 : 1992, Information technology – Open Systems Interconnection – System management: Log Control Function.
- [4] Dan Heller and Paula M. Ferguson. *Motif Programming Manual*. first edition, 1994.
- [5] Mason Tony Hohn Levien and Brown Doug. *Lex Yacc*. first edition, 1992.

- [6] Evi Nemeth. *UNIX System Administration Handbook*. first edition, 1995.
- [7] José Marcos Silva Nogueira and Dilmar Malheiros Meira. A distributed platform for the integration of telecommunication management systems. *NOMS 96*, 1996.
- [8] James Rumbaugh. *Object-Oriented Modeling and Design*. second edition, 1991.
- [9] William Stallings. *SNMP, SNMPv2 and CMIP - The Practical Guide to Network-Management Standards*. second edition, 1993.
- [10] Thomson Software Products Inc., San Diego, CA. *TeleUSE System Overview*, 1994.
- [11] UH Consulting, Balerup, Denmark. *The Q3ADE Agent Development Enviroment: Implementers Guide*, 1995.