

## Especificação LOTOS de uma plataforma de apoio à gerência de redes

Mirela Sechi Moretti Annoni Notare<sup>1</sup>



Bernardo Gonçalves Riso<sup>2</sup>

Universidade Federal de Santa Catarina (UFSC)

Centro Tecnológico - Departamento de Informática e de Estatística (CTC-INE)

Curso de Pós-Graduação em Ciências da Computação - Grupo de Redes e Gerência (CPGCC-GRG)

Campus Universitário da Trindade - Caixa Postal 476 - Florianópolis SC Brasil CEP: 88040-970

fone: (048) 231-9738 / 231-9739 telex: (048) 2240 UFSC BR fax: (048) 231-9770 e-mail: mirella@inf.ufsc.br riso@inf.ufsc.br

**Resumo:** Neste trabalho apresenta-se a descrição formal de uma plataforma para auxiliar a gerência de redes heterogêneas. A descrição é realizada com o uso de um formalismo algébrico que permite especificar os aspectos de comportamento da plataforma. O formalismo utilizado é a Técnica de Descrição Formal LOTOS Básico, que é um padrão internacional da ISO. Inicialmente é especificado o sistema de gerência que reúne a plataforma, aplicações genéricas e aplicações comuns. Posteriormente é descrito o serviço oferecido pela plataforma e também o protocolo correspondente ao recurso de comunicação dessa plataforma.

**Abstract:** This work presents the formal description of a platform which supports the heterogeneous networks management. The description is made using an algebraic formalism that allows to specify the platform behaviour. The formalism used is the Basic LOTOS Formal Description Technique, an international standard from ISO. Initially the management system with the platform, generic applications and common applications is specified. Then the service offered by the platform and a protocol for a communication resource of this platform is described.

**Palavras-chave:** Sistemas distribuídos, redes de computadores, gerência de redes, plataformas de apoio à gerência de redes, especificação formal, LOTOS.

### 1 Introdução

*“O desenvolvimento de sistemas concorrentes e distribuídos requer que os mesmos sejam concebidos antes de serem implementados” [QuCu 94].*

As instalações de redes de computadores normalmente incluem as aplicações necessárias ao gerenciamento dessas redes. Quando, porém, interligam-se redes de diversos fabricantes, a tarefa de gerenciamento tende a ficar muito complexa, dificultando o uso de uma determinada aplicação de gerência que sirva para todas as redes interligadas. Tal complexidade é uma consequência da **difícil interoperabilidade** das aplicações de gerência proprietárias [ReFr 93a,b] [BaFi 94].

Este trabalho propõe a especificação formal de uma **plataforma de apoio a aplicações genéricas** de gerência. A plataforma de apoio a essas aplicações pode utilizar, por exemplo, o protocolo *SNMP* (*Simple Network Management Protocol*) da arquitetura *Internet*, e o *CMIS/CMIP* (*Common Management Information Service/Protocol*) [LePe 94] da arquitetura *OSI*.

As **especificações de serviço e de protocolo** [Viho 94] [KuMe 94] [KaSu 94] apresentadas neste trabalho são realizadas com a utilização da TDF LOTOS [IS 8807] [BoBr 87] [PiLo 90] [Lang 90] [Pire 94], que garante especificações exatas, seguindo a metodologia proposta em [Nota 93] [NoRi 94a,b,c,d,e]. Tal metodologia adota um desenvolvimento baseado em estilos de especificação, refinamentos sucessivos e transformação de especificações.

No item 2 do presente trabalho é apresentada a especificação do sistema de gerência que inclui a plataforma proposta. Nos subitens 2.1 e 2.2 são apresentados dois dos subsistemas que compõem o sistema de gerência. No item 3 é apresentada a especificação da plataforma. Seguem-se as conclusões (item 4), agradecimentos (item 5), anexo com os resultados do uso de ferramentas *LOTOS* e bibliografia (item 6).

<sup>1</sup> Professora substituta do INE/CTC/UFSC e aluna do Curso de Pós-Graduação em Ciências da Computação (UFSC).

<sup>2</sup> Professor adjunto do INE/CTC/UFSC e Doutor em Engenharia Elétrica (UFPB - Campina Grande, 1991).

## 2 Especificação do sistema MS (*Management System*)

*"The specification of a large system has to be abstract: otherwise it will not be comprehensible. (...) Unfortunately, the need for abstraction has not been always appropriately acknowledged both in system development and in standardization activities"* [Kilo 94].

O sistema de gerenciamento pode ser visto como uma caixa preta. Utilizando o estilo de especificação orientado para recursos [ViSc 88] pode-se observar o interior da caixa preta e visualizar os recursos que a compõem. O conjunto das aplicações genéricas de gerência constitui o recurso *GA* (generic applications). Para atender às solicitações dessas aplicações, a plataforma *PLAT* pode utilizar-se de aplicações comuns, que constituem o recurso *CA* (common applications). Reunidas as aplicações genéricas de gerenciamento com a plataforma de apoio a essas aplicações e as aplicações comuns tem-se o sistema completo de gerenciamento (*MS*). Veja a Figura 2.1

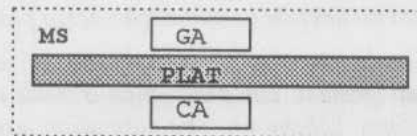


Figura 2.1 - O Sistema de Gerenciamento.

Os recursos *GA*, *CA* e *PLAT* podem ser vistos como processos executados em paralelo. Os processos *GA* e *CA* não se comunicam diretamente, mas sim através da plataforma *PLAT*.

Usando a TDF *LOTOS*, *GA* e *CA* são processos que podem ser combinados com o operador de composição independente (*|||*). Essa combinação constitui um novo processo (*GA|||CA*) que compartilha portas com o processo *PLAT*. Tal novo processo deve ser combinado com o processo *PLAT* através do operador de composição geral (*|[...]|*) que permite explicitar as portas compartilhadas (exemplificadas em alguns processos deste trabalho).

```
specification MS[...]:noexit
  behaviour (GA[...]||||CA[...])|[...]|PLAT[...]
  where
    process GA [...] :noexit:= ... endproc
    process CA [...] :noexit:= ... endproc
    process PLAT [...] :noexit:= ... endproc
endspec
```

### 2.1 Especificação do subsistema *GA* (*Generic Applications*)

O subsistema *GA* pode ser visto como uma caixa preta. Utilizando o estilo orientado para recursos, pode-se visualizar o interior da caixa preta *GA* e observar os recursos que a compõem. O processo *GA* reúne as aplicações genéricas de gerência. Tais aplicações constituem recursos que se distribuem por cinco áreas: contabilização, desempenho, falha, segurança e configuração. Veja a Figura 2.2.

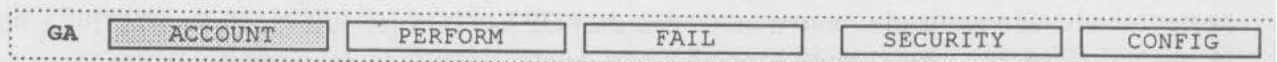


Figura 2.2 - Áreas das aplicações genéricas de gerência.

Os recursos correspondentes às áreas das aplicações são descritos pelos processos *ACCOUNT*, *PERFORM*, *FAIL*, *SECURITY* e *CONFIG* combinados com o operador *|||*, pois são executados independentemente.

```
process GA[...]:noexit:=
  ACCOUNT[...]||||PERFORM[...]||||FAIL[...]||||SECURITY[...]||||CONFIG[...] where ...
endproc
```

Cada uma dessas áreas pode conter diversas aplicações (recursos) de gerência compostas paralelamente. Na área *ACCOUNT*, por exemplo, duas aplicações possíveis são *USE* (para contabilizar o uso dos recursos da rede) e *LOG* (para contabilizar quem usa os recursos da rede). Veja a Figura 2.3.

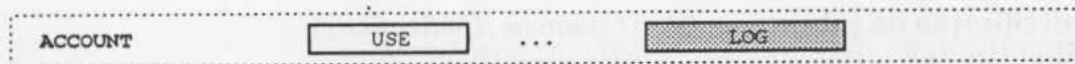


Figura 2.3 - Área de gerência ACCOUNT.

```
process ACCOUNT[...]:noexit:=
  USE[...] ||| ... ||| LOG[...] where ...
endproc
```

Tomando como exemplo a aplicação LOG, ela pode ser estruturada em gerentes e agentes. Em um caso simples, tal estrutura pode envolver a composição de um processo gerente (MANAGER\_OSI) e um processo agente (AGENT\_OSI) que seguem o modelo de gerência *OSI*, e um processo gerente (MANAGER\_INTERNET) e um processo agente (AGENT\_INTERNET) que seguem o modelo *Internet*. Para combinar tais processos, usa-se o operador |||, pois eles não se comunicam diretamente.

Além dos sítios correspondentes a tais gerentes e agentes da aplicação LOG, outros sítios ainda poderiam ser considerados. Veja a Figura 2.4.

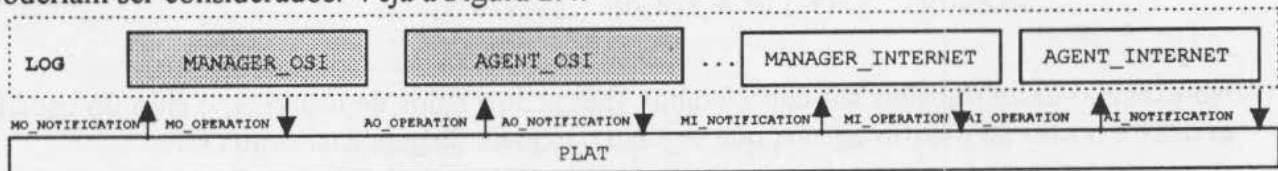


Figura 2.4 - Gerentes e agentes da aplicação LOG.

```
process LOG[...]:noexit:=
  MANAGER_OSI[...] ||| AGENT_OSI[...] ... ||| MANAGER_INTERNET[...] ||| AGENT_INTERNET[...]
  where ...
endproc
```

Em qualquer instante o gerente que segue o modelo de gerência *OSI* decide requisitar operações (representadas pelo evento MO\_OPERATION) ou receber notificações de gerência (representadas pelo evento MO\_NOTIFICATION). A ocorrência de um evento interno *i* [HiBo 94] [FrAu 89] define a escolha do gerente. A representação desse comportamento permite detalhar o processo MANAGER\_OSI que possui duas portas de comunicação (MO\_OPERATION e MO\_NOTIFICATION).

```
process MANAGER_OSI[MO_OPERATION,MO_NOTIFICATION]:noexit:=
  i;MO_OPERATION;MANAGER_OSI[...] [] i;MO_NOTIFICATION;MANAGER_OSI[...]
endproc
```

O comportamento do gerente que segue o modelo de gerência *Internet* é similar ao comportamento do gerente que segue o modelo de gerência *OSI*.

```
process MANAGER_INTERNET[MI_OPERATION,MI_NOTIFICATION]:noexit:=
  i;MI_OPERATION;MANAGER_INTERNET[...] []
  i;MI_NOTIFICATION;MANAGER_INTERNET[...]
endproc
```

De modo semelhante, o agente que segue o modelo de gerência *OSI* decide receber solicitações para executar operações (representadas pelo evento AO\_OPERATION) ou enviar notificações de gerência ao gerente (representadas pelo evento AO\_NOTIFICATION). A representação desse comportamento permite detalhar o processo AGENT\_OSI. O processo AGENT\_OSI possui duas portas de comunicação (AO\_OPERATION e AO\_NOTIFICATION).

```
process AGENT_OSI[AO_OPERATION,AO_NOTIFICATION]:noexit:=
  i;AO_OPERATION;AGENT_OSI[...] [] i;AO_NOTIFICATION;AGENT_OSI[...]
endproc
```

O comportamento do agente que segue o modelo de gerência *Internet* é similar ao comportamento do agente que segue o modelo de gerência *OSI*.

```
process AGENT_INTERNET[AI_OPERATION,AI_NOTIFICATION]:noexit:=
  i;AI_OPERATION;AGENT_INTERNET[...] [] i;AI_NOTIFICATION;AGENT_INTERNET[...]
endproc
```

## 2.2 Especificação do subsistema CA (*Common Applications*)

O subsistema CA pode ser visualizado como uma caixa preta. No estilo orientado para recursos, o processo CA pode ser definido pela **composição paralela de aplicações comuns** tais como o SNMP e o CMISE [IS 9595]. Uma vez que tais aplicações são independentes elas podem ser combinadas com a utilização do operador |||.

No estilo orientado para restrições [ViSc 88], cada uma das aplicações comuns pode ser definida a partir de **restrições locais e restrições fim-a-fim**, estabelecidas para essa aplicação. Esse é o caso, por exemplo, do CMISE, descrito a seguir. O processo CMISE pode ser definido pela composição dependente (usa-se o operador ||) de suas restrições do tipo local com suas restrições do tipo fim-a-fim, pois essas restrições compartilham portas.

```
process CA[...]:noexit:=
  SNMP ||| ...||| CMISE[...] where ...
  process CMISE[...]:noexit:=
    LOCAL_O[...] || FIM_A_FIM_O[...]where ...
  endproc
endproc
```

No exemplo deste trabalho são considerados apenas dois sítios de usuários: o sítio do usuário gerente e o sítio do usuário agente, que seguem o modelo de gerenciamento *OSI*.

### 2.2.1 Restrições locais do provedor CMISE

O provedor CMISE provê serviços aos usuários gerente e agente através da plataforma. O CMISE pode ser descrito a partir de suas restrições locais e de suas restrições fim-a-fim. As restrições locais do provedor CMISE são uma combinação independente das restrições locais ao sítio do gerente (representadas pelo processo LOC\_MO) e das restrições locais ao sítio do agente (representadas pelo processo LOC\_AO). Veja a Figura 2.5.

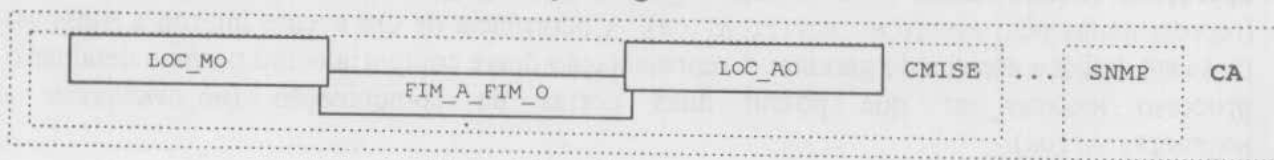


Figura 2.5 - Aplicações comuns CA.

```
process LOCAL_O[...]:noexit:=
  LOC_MO[...] ||| LOC_AO[...] where ...
endproc
```

#### a) Restrições locais do provedor CMISE no sítio do gerente

O comportamento do provedor CMISE no sítio do gerente (descrito pelo processo LOC\_MO) pode ser definido considerando o recebimento de requisições de operações do usuário gerente (descrito pelo processo MO\_OPERATION) e a entrega de notificações ao usuário gerente recebidas de um usuário agente (descrito pelo processo MO\_NOTIFICATION).

Para ser possível tal fluxo de operações e notificações, é necessário que uma associação tenha sido estabelecida previamente. O estabelecimento de tal associação pode ser descrito pelo processo MO\_ASSOCIATION.

Os processos MO\_TERM1 e MO\_TERM2 descrevem as formas possíveis de encerramento de associação. A possibilidade de estabelecimento de uma nova associação é representada pela chamada recursiva ao processo LOC\_MO. Quando uma associação é encerrada, não é mais possível o fluxo de operações e notificações.

```
process LOC_MO[...]:noexit:=
  ((MO_ASSOCIATION[...] [> MO_TERM1[...]]>>
  (((MO_OPERATION[...]|||MO_NOTIFICATION[...]) [> MO_TERM2[...]) [> MO_TERM1[...]] ))
  >>LOC_MO[...] where ...
endproc
```

O provedor CMISE no sítio do gerente (descrito pelo processo LOC\_MO) recebe uma requisição para o **estabelecimento de associação** [IS8649] [IS8650] que o usuário gerente decidiu enviar. Tal recebimento de requisição pelo provedor CMISE é descrito pelo evento MO\_ASSOCIATE\_cf\_rqt.

Como o estabelecimento de associação é um serviço confirmado, o provedor CMISE no sítio do gerente deve enviar uma confirmação ao usuário gerente (positiva ou negativa, de acordo com a resposta recebida do usuário agente). Tal envio de confirmação é descrito pelo evento MO\_ASSOCIATE\_cf\_cnf. Até que tal confirmação seja entregue ao usuário gerente, o provedor CMISE não pode receber outra requisição do usuário gerente (exceto uma requisição de término abrupto de associação, que pode ocorrer a qualquer momento, descrito pelo evento MO\_ABORT\_ncf\_rqt, através do processo TERM1).

O provedor CMISE pode não ser capaz de suportar a associação requisitada. Nesse caso, uma confirmação (negativa) é entregue ao usuário gerente, descrita pelo evento MO\_ASSOCIATE\_cf\_cnf.

O provedor CMISE no sítio do gerente pode também receber uma requisição de estabelecimento de associação do usuário agente, a qual deve ser entregue ao usuário gerente. Tal entrega é descrita pelo evento MO\_ASSOCIATE\_cf\_ind. O provedor CMISE no sítio do gerente aguarda resposta (positiva ou negativa) do usuário gerente para enviar confirmação ao usuário agente. O recebimento de tal resposta é descrito pelo evento MO\_ASSOCIATE\_cf\_rsp. O provedor CMISE pode receber do usuário agente uma requisição de encerramento abrupto de associação interrompendo o processo MO\_ASSOCIATION a qualquer instante (descrito pelo evento MO\_ABORT\_ncf\_ind, do processo TERM1). O processo exit habilita um processo subsequente.

```
process MO_ASSOCIATION[...]:exit :=
    MO_ASSOCIATE_cf_rqt; (MO_ASSOCIATE_cf_cnf; exit
                        [] MO_ASSOCIATE_cf_cnf; MO_ASSOCIATION[...])
    [] MO_ASSOCIATE_cf_ind; (MO_ASSOCIATE_cf_rsp; exit
                           [] MO_ASSOCIATE_cf_rsp; MO_ASSOCIATION[...])
endproc
```

O **encerramento de associação** pode ser negociado (envolvendo os eventos MO\_RELEASE\_cf\_rqt, MO\_RELEASE\_cf\_cnf, MO\_RELEASE\_cf\_ind, MO\_RELEASE\_cf\_rsp) ou não negociado (envolvendo os eventos MO\_ABORT\_ncf\_rqt e MO\_ABORT\_ncf\_ind). Os eventos envolvidos no modo não confirmado de encerramento de associação são descritos no processo MO\_TERM1. Os eventos envolvidos no modo confirmado de encerramento de associação são descritos no processo MO\_TERM2. As requisições de encerramento de associação podem partir do usuário gerente, do usuário agente, ou então, do provedor CMISE.

Os eventos descritos no processo MO\_TERM1 não requerem confirmação ou resposta e podem ocorrer mesmo antes de chegar uma confirmação de requisição de associação. O provedor CMISE recebe uma requisição de encerramento abrupto de associação do usuário gerente ou entrega uma indicação de encerramento abrupto de associação ao usuário gerente (por ter recebido uma requisição do usuário agente). O recebimento da requisição do usuário gerente é descrito pelo evento MO\_ABORT\_ncf\_rqt e a entrega da indicação ao usuário gerente é descrita pelo evento MO\_ABORT\_ncf\_ind. O provedor CMISE pode solicitar encerramento de associação por erros internos, enviando indicação de encerramento (MO\_ABORT\_ncf\_ind) para ambos os usuários.

```
process MO_TERM1[...]:noexit :=
    MO_ABORT_ncf_rqt; MO_TERM1[...]    [] MO_ABORT_ncf_ind; MO_TERM1[...]
    [] MO_ABORT_ncf_ind; MO_TERM1[...]
endproc
```

No processo `MO_TERM2` estão os eventos responsáveis pelo término de associação de forma negociada. O provedor `CMISE` no sítio do gerente recebe do usuário gerente uma requisição para encerrar uma associação de forma negociada. O recebimento de tal requisição é descrito pelo evento `MO_RELEASE_cf_rqt`.

Após essa requisição, nenhuma outra requisição pode chegar ao provedor (exceto `MO_ABORT_ncf_rqt`) até que uma confirmação seja entregue ao usuário gerente (descrita pelo evento `MO_RELEASE_cf_cnf`). Tal confirmação pode ser positiva ou negativa, dependendo da resposta do usuário agente). Se a confirmação é positiva, o usuário gerente pode estabelecer uma nova associação, tal possibilidade é descrita pela chamada ao processo `LOC_MO`. Se a confirmação é negativa, o usuário gerente pode tentar novamente o encerramento da associação (tal comportamento é descrito através da chamada recursiva ao processo `MO_TERM2`).

O procedimento de encerramento negociado pode ser interrompido a qualquer momento com a requisição de encerramento abrupto de associação solicitado pelo usuário gerente ou agente. Tal interrupção é descrita no processo `TERM1` pelos eventos `MO_ABORT_ncf_rqt` e `MO_ABORT_ncf_ind`.

```
process MO_TERM2[...]:noexit:=
  [] MO_RELEASE_cf_rqt;(MO_RELEASE_cf_cnf;LOC_MO
                        [] MO_RELEASE_cf_cnf;MO_TERM2[...])
  [] MO_RELEASE_cf_ind;MO_RELEASE_cf_rsp;LOC_MO
                        [] MO_RELEASE_cf_rsp;MO_TERM2[...])
endproc
```

No processo `MO_OPERATION` são descritas as operações que o provedor `CMISE`, no sítio do gerente pode receber do usuário gerente, para entregar ao usuário agente, que as executará: `GET` (para obter um valor de atributo de objeto gerenciável), `CANCEL_GET` (para cancelar uma operação `GET` prévia), `CREATE` (para criar uma instância de objeto gerenciável), `DELETE` (para apagar uma instância de objeto gerenciável), `SET` (para atribuir um valor a um atributo de objeto gerenciável) e `ACTION` (para realizar uma ação sobre um objeto gerenciável). As operações `SET` e `ACTION` podem ser realizadas em modo confirmado ou em modo não confirmado. As demais somente em modo confirmado. Essas operações são combinadas com o operador `|||`, pois são executadas independentemente.

```
process MO_OPERATION[...]: noexit:=
  MO_GET[...]|||MO_CREATE[...]|||MO_DELETE[...]|||MO_SET[...]|||MO_ACTION[...]|
  where...
endproc
```

O recebimento de uma operação `GET` do usuário gerente pelo provedor `CMISE` no sítio do gerente é descrito pelo evento `MO_GET_cf_rqt`. O comportamento posterior à ocorrência desse evento é descrito pelo processo `MO_GET1`.

```
process MO_GET[...]: noexit :=
  MO_GET_cf_rqt;MO_GET1[...] where ...
endproc
```

Após a operação `GET` ser recebida pelo provedor `CMISE`, existem três possibilidades. O provedor pode enviar ao usuário gerente uma confirmação positiva ou negativa (descrita pelo evento `MO_GET_cf_cnf`), ou o provedor pode receber do usuário gerente outra requisição (descrita pelo evento `MO_GET_cf_rqt`), ou ainda, o provedor pode receber do usuário gerente a requisição de cancelamento de uma operação (descrito pelo evento `MO_CANCEL_GET_cf_rqt`).

```
process MO_GET1[...]:noexit:=
  MO_GET_cf_cnf;MO_GET[...]|
  [] MO_GET_cf_rqt;MO_GET1[...]|
  [] MO_CANCEL_GET_cf_rqt;MO_CANCEL_GET1[...] where ...
endproc
```

No caso de ser requisitado um cancelamento de operação, duas confirmações são necessárias: uma para confirmar a operação de cancelamento e outra para confirmar a operação GET. Antes dessas confirmações serem entregues ao usuário gerente, há a possibilidade de chegar no sítio do gerente do provedor CMISE a requisição de outro cancelamento.

```
process MO_CANCEL_GET1[...]:noexit:=
    MO_CANCEL_GET_cf_cnf;MO_CANCEL_GET2 [...]
    [] MO_CANCEL_GET_cf_rqt;MO_CANCEL_GET1[...]
endproc
```

Após o provedor CMISE enviar a confirmação de cancelamento ao usuário gerente, pode ser enviada confirmação de operação GET, ou pode ser enviada confirmação de cancelamento, ou ainda, pode chegar requisição de outro cancelamento.

```
process MO_CANCEL_GET2[...]:noexit:=
    MO_GET_cf_cnf;MO_GET1[...]
    [] MO_CANCEL_GET_cf_cnf;MO_CANCEL_GET2 [...]
    [] MO_CANCEL_GET_cf_rqt;MO_CANCEL_GET1[...]
endproc
```

As operações CREATE, DELETE, SET e ACTION apresentam comportamentos similares. A operação ACTION é descrita pelo processo MO\_ACTION.

```
process MO_ACTION[...]: noexit :=
    MO_ACTION_ncf_rqt;MO_ACTION[...]
    [] MO_ACTION_cf_rqt;MO_ACTION1[...] where ...
endproc
```

Quando em modo não confirmado, o comportamento posterior ao recebimento de requisição de uma operação ACTION pelo provedor CMISE é descrito pela chamada recursiva ao processo MO\_ACTION que possibilita novos recebimentos dessa operação do usuário gerente. Quando em modo confirmado, o comportamento posterior pode envolver outra requisição ou então uma confirmação de uma operação ACTION. Tal comportamento é descrito pelo processo MO\_ACTION1.

```
process MO_ACTION1[...]:noexit:=
    MO_ACTION_cf_rqt;MO_ACTION[...] [] MO_ACTION_ncf_rqt;MO_ACTION[...]
    [] MO_ACTION_cf_cnf;MO_ACTION[...]
endproc
```

No processo MO\_NOTIFICATION é descrito o comportamento do provedor CMISE, no sítio do gerente, relativo à entrega de **notificações** e ao recebimento de respostas correspondentes ao usuário gerente.

O provedor CMISE no sítio do gerente entrega uma notificação ao usuário gerente (recebida de um usuário agente que relata um evento ocorrido em um objeto gerenciado). Tal entrega é descrita pelo processo MO\_EVENT\_REPORT\_cf\_ind (quando em modo confirmado) ou pelo processo MO\_EVENT\_REPORT\_ncf\_ind (quando em modo não confirmado). Quando em modo não confirmado o comportamento posterior é descrito pela chamada recursiva ao processo (MO\_NOTIFICATION) para possibilitar o recebimento de outras notificações. Quando em modo confirmado, o comportamento posterior é descrito pelo processo MO\_NOTIFICATION1.

```
process MO_NOTIFICATION[...]: noexit :=
    MO_EVENT_REPORT_ncf_ind;MO_NOTIFICATION[...]
    [] MO_EVENT_REPORT_cf_ind;MO_NOTIFICATION1[...] where ...
endproc
```

Quando em modo confirmado, o provedor aguarda uma resposta do usuário gerente (positiva ou negativa). Através do processo `MO_NOTIFICATION1`, o provedor `CMISE` pode enviar outra notificação (de modo confirmado ou não) ao usuário gerente ou então receber uma resposta do usuário gerente de uma notificação enviada em modo confirmado.

```
process MO_NOTIFICATION1[...]:noexit:=
    MO_EVENT_REPORT_cf_ind;MO_NOTIFICATION1[...]
    [] MO_EVENT_REPORT_ncf_ind;MO_NOTIFICATION1[...]
    [] MO_EVENT_REPORT_cf_rsp;MO_NOTIFICATION1[...]
endproc
```

## b) Restrições locais do provedor `CMISE` no sítio do agente

O processo `LOC_AO` (que descreve o comportamento do provedor `CMISE` no sítio do agente) apresenta um comportamento similar ao processo `LOC_MO` (que descreve o comportamento do provedor `CMISE` no sítio do gerente).

O provedor `CMISE` no sítio do agente recebe requisições (`_rqt`) de **notificações** do usuário agente que as envia ao usuário gerente. Quando em modo confirmado, as notificações devem ser respondidas pelo usuário gerente para que o provedor `CMISE` no sítio do agente entregue confirmações (`_cnf`) ao usuário agente.

O provedor `CMISE` no sítio do agente entrega (`_ind`) ao usuário agente as **operações** recebidas do usuário gerente. Quando em modo confirmado, o provedor `CMISE` no sítio do agente aguarda resposta (`_rsp`) do usuário agente para entregar confirmação ao usuário gerente.

O processo que descreve o recebimento de notificações do usuário agente pelo provedor `CMISE` no sítio do agente é o processo `AO_NOTIFICATION`.

```
process AO_NOTIFICATION[...]: noexit :=
    AO_EVENT_REPORT_ncf_rqt;AO_NOTIFICATION[...]
    [] AO_EVENT_REPORT_cf_rqt;AO_NOTIFICATION1[...] where ...
endproc
```

Quando uma notificação é recebida pelo provedor `CMISE` no sítio do agente em modo confirmado (descrita pelo evento `AO_EVENT_REPORT_cf_rqt`) o comportamento seguinte é descrito pelo processo `AO_NOTIFICATION1`, em que são possíveis outros recebimentos de notificações ou o envio de confirmações de notificações ao usuário agente.

```
process AO_NOTIFICATION1[...]:noexit:=
    AO_EVENT_REPORT_cf_rqt;AO_NOTIFICATION1[...]
    [] AO_EVENT_REPORT_ncf_rqt;AO_NOTIFICATION1[...]
    [] AO_EVENT_REPORT_cf_cnf;AO_NOTIFICATION1[...]
endproc
```

## 2.2.2 Restrições fim-a-fim do `CMISE`

As restrições `FIM_A_FIM` representam as restrições de ligação entre o sítio local ao gerente do provedor `CMISE` e o sítio local ao agente do provedor `CMISE`. As restrições `FIM_A_FIM` são estruturadas segundo o fluxo de informações **do gerente para o agente** (representado pelo processo `MO_AO`) e o fluxo de informações **do agente para o gerente** (representado pelo processo `AO_MO`). Tais fluxos são combinados independentemente pois não compartilham portas.

```
process FIM_A_FIM O[...]:noexit:=
    MO_AO[...] ||| AO_MO[...] where ...
endproc
```



O processo `MO_AO` corresponde às informações que fluem **do sítio local ao gerente** do provedor `CMISE` **para o sítio local ao agente** do provedor `CMISE`. Esse processo é estruturado segundo a ligação das requisições recebidas do usuário gerente com suas respectivas entregas ao usuário agente (descritas pelo processo `MO_AO_RQT_IND`) e segundo a ligação do recebimento das respostas do usuário gerente às confirmações entregues ao usuário agente (descritas pelo processo `MO_AO_RSP_CNF`).

```
process MO_AO[...]:noexit:=
    MO_AO_RQT_IND[...] ||| MO_AO_RSP_CNF[...]
endproc
```

As **requisições** recebidas do usuário gerente pelo provedor `CMISE` e as respectivas **entregas** ao usuário agente são descritas no processo `MO_AO_RQT_IND`.

```
process MO_AO_RQT_IND[...]: noexit :=
    MO_ASSOCIATE_cf_rqt;      AO_ASSOCIATE_cf_ind;      MO_AO[...]
[] MO_RELEASE_cf_rqt;      AO_RELEASE_cf_ind;      MO_AO[...]
[] MO_ABORT_ncf_rqt;      AO_ABORT_ncf_ind;      MO_AO[...]
[] MO_GET_cf_rqt;      AO_GET_cf_ind;      MO_AO[...]
    ...
[] MO_ACTION_ncf_rqt;      AO_ACTION_ncf_ind;      MO_AO[...]
endproc
```

As **respostas** recebidas do usuário gerente pelo provedor `CMISE` e as respectivas **entregas** das **confirmações** ao usuário agente são descritas no processo `MO_AO_RSP_CNF`.

```
process MO_AO_RSP_CNF[...]: noexit :=
    MO_ASSOCIATE_cf_rsp;      AO_ASSOCIATE_cf_cnf;      MO_AO[...]
[] MO_RELEASE_cf_rsp;      AO_RELEASE_cf_cnf;      MO_AO[...]
[] MO_EVENT_REPORT_cf_rsp;  AO_EVENT_REPORT_cf_cnf;  MO_AO[...]
endproc
```

O processo `AO_MO` corresponde às informações que fluem **do sítio local ao agente** do provedor `CMISE` **para o sítio local ao gerente** do provedor `CMISE`. Esse processo é estruturado similarmente ao processo `MO_AO`.

```
process AO_MO[...]:noexit:=
    AO_MO_RQT_IND[...] ||| AO_MO_RSP_CNF[...]
endproc
```

As **requisições** recebidas do usuário agente pelo provedor `CMISE` e as respectivas **entregas** ao usuário gerente são descritas no processo `AO_MO_RQT_IND`.

```
process AO_MO_RQT_IND[...]: noexit :=
    AO_ASSOCIATE_cf_rqt;      MO_ASSOCIATE_cf_ind;      AO_MO[...]
[] AO_RELEASE_cf_rqt;      MO_RELEASE_cf_ind;      AO_MO[...]
[] AO_ABORT_ncf_rqt;      MO_ABORT_ncf_ind;      AO_MO[...]
[] AO_EVENT_REPORT_cf_rqt;  MO_EVENT_REPORT_cf_ind;  AO_MO[...]
[] AO_EVENT_REPORT_ncf_rqt;  MO_EVENT_REPORT_ncf_ind;  AO_MO[...]
endproc
```

O recebimento de **respostas** do usuário agente pelo provedor `CMISE` no sítio do agente e as respectivas **entregas** de confirmações ao usuário gerente são descritas pelo processo `AO_MO_RSP_CNF`.

```
process AO_MO_RSP_CNF[...]: noexit :=
    AO_ASSOCIATE_cf_rsp;      MO_RELEASE_cf_cnf;      AO_MO[...]
[] AO_RELEASE_cf_rsp;      MO_RELEASE_cf_cnf;      AO_MO[...]
[] AO_GET_cf_rsp;      MO_GET_cf_cnf;      AO_MO[...]
[] AO_CANCEL_GET_cf_rsp;    MO_CANCEL_GET_cf_cnf;MO_GET_cf_rsp;  AO_MO[...]
[] AO_CREATE_cf_rsp;      MO_CREATE_cf_cnf;      AO_MO[...]
[] AO_DELETE_cf_rsp;      MO_DELETE_cf_cnf;      AO_MO[...]
[] AO_SET_cf_rsp;      MO_SET_cf_cnf;      AO_MO[...]
[] AO_ACTION_cf_rsp;      MO_ACTION_cf_cnf;      AO_MO[...]
endproc
```

### 3 Especificação do serviço da plataforma PLAT

*"O nível de abstração é uma das mais importantes características para o desenvolvimento de sistemas. LOTOS apresenta, além do elevado nível de abstração, outras características inerentes à TDF como o poder de expressão, a estruturação de especificações e a definição formal"[QuCu 94].*

A plataforma PLAT pode ser vista como uma caixa preta que dispõe de interfaces de comunicação com os seus usuários (gerente e agente que seguem o modelo de gerência OSI). Veja a Figura 3.1.

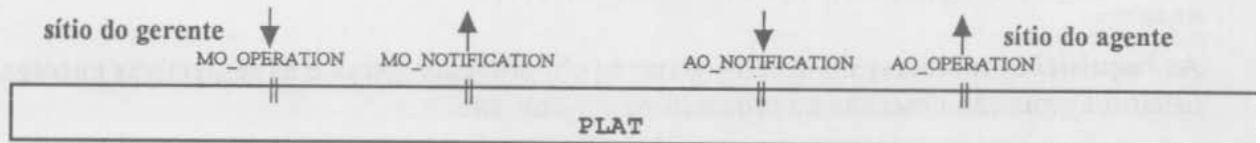


Figura 3.1 - Interface de PLAT com as aplicações genéricas.

A interface de PLAT com o usuário gerente é formada pelas portas:

- MO\_OPERATION para receber requisição de operação;
- MO\_NOTIFICATION para entregar notificação.

A interface de PLAT com o usuário agente é formada pelas portas:

- AO\_OPERATION para entregar requisição de operação;
- AO\_NOTIFICATION para receber notificação.

Utilizando o estilo orientado para restrições, o serviço de comunicação oferecido por PLAT pode ser especificado a partir dos aspectos locais (de cada sítio de PLAT) e dos aspectos fim-a-fim (ligando os sítios). Veja a Figura 3.2.

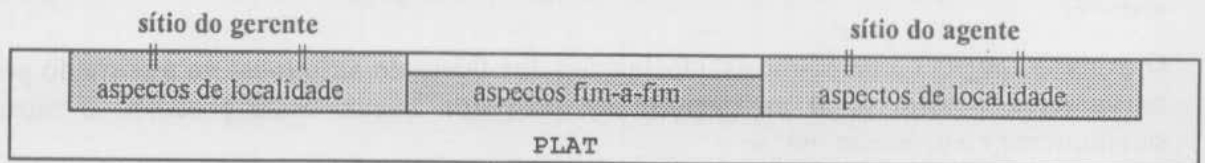


Figura 3.2 - Restrições de PLAT.

As restrições de localidade, descritas pelo processo LOCAL\_P, compartilham eventos com as restrições fim-a-fim, descritas pelo processo FIM\_A\_FIM\_P. Por isso, o operador || (composição paralela dependente) pode ser usado na especificação do processo PLAT.

```
process PLAT[...]:noexit:=
    LOCAL_P[MO_OPERATION,MO_NOTIFICATION,AO_NOTIFICATION,AO_OPERATION]
    || FIM_A_FIM_P[MO_OPERATION,MO_NOTIFICATION,AO_NOTIFICATION,AO_OPERATION]
    where ...
endproc
```

As restrições locais, descritas pelo processo LOCAL\_P, compreendem as restrições locais ao sítio do gerente, descritas pelo processo LOC\_M, e as restrições locais ao sítio do agente, descritas pelo processo LOC\_A. Como LOC\_M e LOC\_A constituem restrições locais independentes, elas devem ser combinadas com o operador |||. Veja a Figura 3.3.

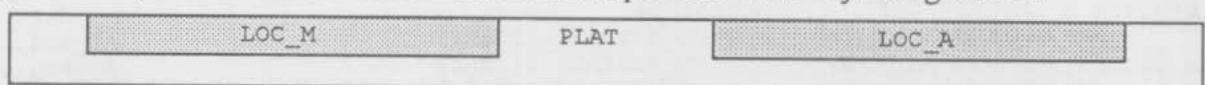


Figura 3.3 - Restrições locais de PLAT.

```
process
LOCAL_P[MO_OPERATION,MO_NOTIFICATION,AO_NOTIFICATION,AO_OPERATION]:noexit:=
    LOC_M[MO_OPERATION,MO_NOTIFICATION]
    ||| LOC_A[AO_NOTIFICATION,AO_OPERATION] where ...
endproc
```

As restrições locais ao sítio do gerente, descritas pelo processo `LOC_M`, envolvem uma escolha indeterminística entre o recebimento de uma requisição de serviço de operação de gerência (descrito pelo evento `MO_OPERATION`), e a entrega de uma requisição de serviço de notificação de gerência (descrito pelo evento `MO_NOTIFICATION`). Tal escolha é expressa com o operador `[]`.

```
process LOC_M[MO_OPERATION,MO_NOTIFICATION]: noexit :=
    MO_OPERATION;LOC_M[...] [] MO_NOTIFICATION;LOC_M[...]
endproc
```

As restrições locais ao sítio do agente, descritas pelo processo recursivo `LOC_A`, envolvem uma escolha indeterminística entre a entrega de uma requisição de serviço de operação de gerência (descrito pelo evento `AO_OPERATION`) e o recebimento de uma requisição de serviço de notificação (descrito pelo evento `AO_NOTIFICATION`). Tal escolha é expressa com o operador `[]`.

```
process LOC_A[AO_OPERATION,AO_NOTIFICATION]: noexit :=
    AO_OPERATION;LOC_A[...] [] AO_NOTIFICATION;LOC_A[...]
endproc
```

As restrições fim-a-fim (descritas pelo processo `FIM_A_FIM`) referem-se aos aspectos de comunicação que ligam o sítio do gerente ao sítio do agente. Elas podem ser estruturadas segundo o fluxo de mensagens do gerente para o agente (descritas pelo processo `MA`) e segundo o fluxo de mensagens do agente para o gerente (descritas pelo processo `AM`). Veja a Figura 3.4.

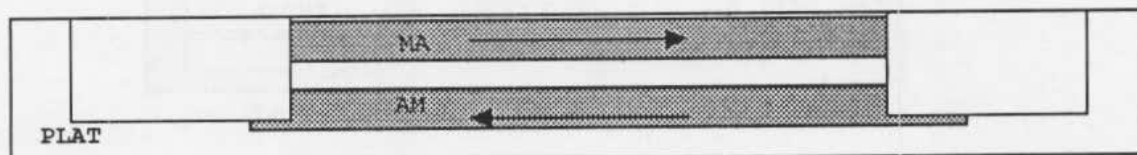


Figura 3.4 - Restrições fim-a-fim de PLAT.

As restrições descritas pelos processos `MA` e `AM` são independentes. Assim, a combinação desses processos é realizada com o uso do operador `|||`.

```
process FIM_A_FIM_P[MO_OPERATION,MO_NOTIFICATION,AO_NOTIFICATION,AO_OPERATION]:noexit:=
    MA[MO_OPERATION,MO_NOTIFICATION,AO_NOTIFICATION,AO_OPERATION]
    ||| AM[MO_OPERATION,MO_NOTIFICATION,AO_NOTIFICATION,AO_OPERATION]
    where ...
endproc
```

As restrições descritas pelo processo `MA` correspondem ao recebimento de uma operação no sítio do gerente e à sua entrega no sítio do agente.

```
process MA[MO_NOTIFICATION,AO_NOTIFICATION]:noexit:=
    MO_OPERATION;AO_OPERATION;MA[...]
endproc
```

As restrições descritas pelo processo `AM` correspondem ao recebimento de uma notificação no sítio do agente e à sua entrega no sítio do gerente.

```
process AM[MO_OPERATION,AO_OPERATION]:noexit:=
    AO_NOTIFICATION;MO_NOTIFICATION;AM[...]
endproc
```

### 3.1 Especificação do protocolo da plataforma PLAT

Para refinar a especificação do serviço oferecido pela plataforma, apresentam-se aspectos internos da caixa preta PLAT, que passa a ser observada em termos dos recursos que a compõem. Segundo [FrIs 93] [FaIm 93], uma plataforma de apoio a aplicações pode prover à camada de aplicação, os recursos de:

- comunicação (que suportam os paradigmas cliente-servidor e gerente-agente);
- interface homem-máquina (que permitem acesso através de interfaces GUI e ASCII);
- base de dados (que oferecem transparência quanto ao SGBD que está sendo acessado); e
- ferramentas de desenvolvimento (que visam facilitar e agilizar a implementação).

Veja a Figura 3.5.



Figura 3.5 - Recursos da plataforma PLAT.

Pode-se representar por PLATCOM o recurso de comunicação do provedor PLAT. Tal recurso é uma combinação de três recursos mais simples: uma entidade de protocolo ENT\_M, uma entidade de protocolo ENT\_A e um provedor de comunicação subjacente, CMISE. Veja a Figura 3.6.

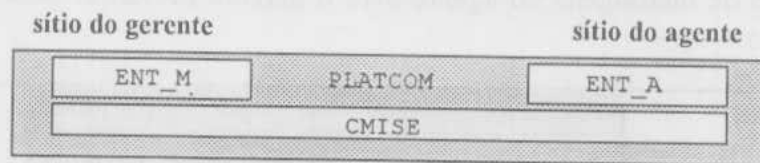


Figura 3.6 - Recurso de comunicação do provedor PLAT.

A comunicação entre PLATCOM (recurso de comunicação da plataforma PLAT) e o CMISE se dá através da ocorrência de primitivas de comunicação [IS 9596]. Veja a Figura 3.7.

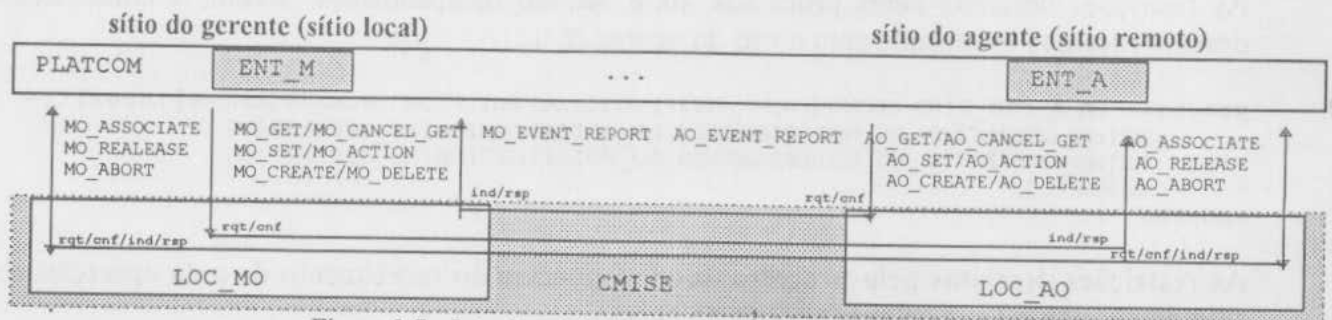


Figura 3.7 - Interface inferior de PLATCOM com o CMISE.

#### 3.1.1 As entidades de protocolo do recurso de comunicação

O comportamento do processo ENT\_M pode ser estruturado segundo as restrições das interfaces superior e inferior (descritas pelos processos ENT\_M\_SUP e ENT\_M\_INF, respectivamente) e as restrições de ligação das interfaces (descritas pelo processo ENT\_M\_LIG). Veja a Figura 3.8.

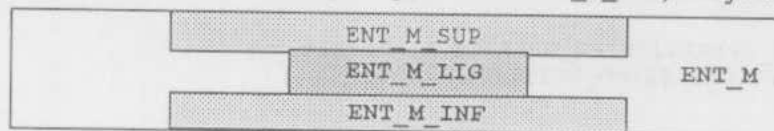


Figura 3.8 - Restrições da entidade ENT\_M.

Utilizando o estilo orientado para restrições, o comportamento da entidade ENT\_M pode ser especificado com o uso do operador |||, para combinar as restrições ENT\_M\_SUP e ENT\_M\_INF, e estas, por sua vez, combinadas através do operador |[...]| (composição geral) com a restrição ENT\_M\_LIG.

```
process ENT_M[ <portas das interfaces superiores e inferiores>]:noexit:=
  ( ENT_M_SUP[ <portas da interface superior> ]
  ||| ENT_M_INF[ <portas da interface inferior> ] )
  |[ <portas das interfaces superiores e inferiores> ]|
  ENT_M_LIG[ <portas das interfaces superiores e inferiores>] where...
endproc
```

A interface **superior** da entidade ENT\_M, que realiza a comunicação com o processo LOC\_MO, é formada pelas portas:

- MO\_OPERATION para requerer operação;
- MO\_NOTIFICATION para receber notificação.

A interface **inferior** da entidade ENT\_M, para comunicação com o CMISE, é formada pelas portas:

- MO\_ASSOCIATE\_cf\_rqt, cnf, ind e \_rsp para estabelecimento de associação;
- MO\_RELEASE\_cf\_rqt, cnf, ind e \_rsp para encerramento de associação de modo negociado;
- MO\_ABORT\_ncf\_rqt e \_ind para encerramento de associação de modo abrupto;
- MO\_GET\_cf\_rqt e \_cnf para obtenção de valores de atributos;
- MO\_CANCEL\_GET\_cf\_rqt e \_cnf para cancelamento de operação GET;
- MO\_SET\_cf\_rqt e \_cnf para modificação de valores de atributos;
- MO\_ACTION\_cf\_rqt e \_cnf para execução de ação sobre os objetos;
- MO\_CREATE\_rqt e \_cnf para criação de instância de objeto;
- MO\_DELETE\_rqt e \_cnf para apagar objetos gerenciados;
- MO\_EVENT\_REPORT\_ind e \_rsp para recebimento de notificação.

Veja a figura 3.9.

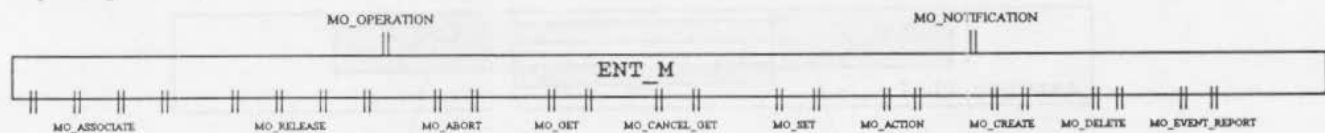


Figura 3.9 - Portas superiores e inferiores da entidade ENT\_M.

O processo ENT\_A pode ser estruturado similarmente ao processo ENT\_M, considerando as portas nas suas interfaces. A interface **superior** da entidade ENT\_A, que realiza a comunicação com o processo LOC\_AO, é formada pelas portas:

- AO\_OPERATION para receber operação;
- AO\_NOTIFICATION para requerer notificação.

A interface **inferior** da entidade ENT\_A, que se comunica com o CMISE, é formada pelas portas:

- AO\_ASSOCIATE\_rqt, cnf, ind e \_rsp para estabelecimento de associação;
- AO\_RELEASE\_rqt, cnf, ind e \_rsp para encerramento de associação de modo negociado;
- AO\_ABORT\_rqt e \_ind para encerramento de associação de modo abrupto;
- AO\_GET\_ind e \_rsp para obtenção de valores de atributos;
- AO\_CANCEL\_GET\_ind e \_rsp para cancelamento de operação GET;
- AO\_SET\_ind e \_rsp para modificação de valores de atributos;
- AO\_ACTION\_ind e \_rsp para execução de ação sobre os objetos;
- AO\_CREATE\_ind e \_rsp para criação de uma nova instância de objeto;
- AO\_DELETE\_ind e \_rsp para apagar objetos gerenciados;
- AO\_EVENT\_REPORT\_rqt e \_cnf para envio de notificação;

A partir do serviço PLAT e do serviço subjacente (CA) é possível derivar as entidades de protocolo ENT\_M (gerente) e ENT\_A (agente) [NoRi 94b] [BoGo 86] [KaHi 90] [HiBo 94] [Pire 94]. Tais derivações são realizadas através das transformações  $t_1$  até  $t_{12}$ . Veja a Figura 3.10.

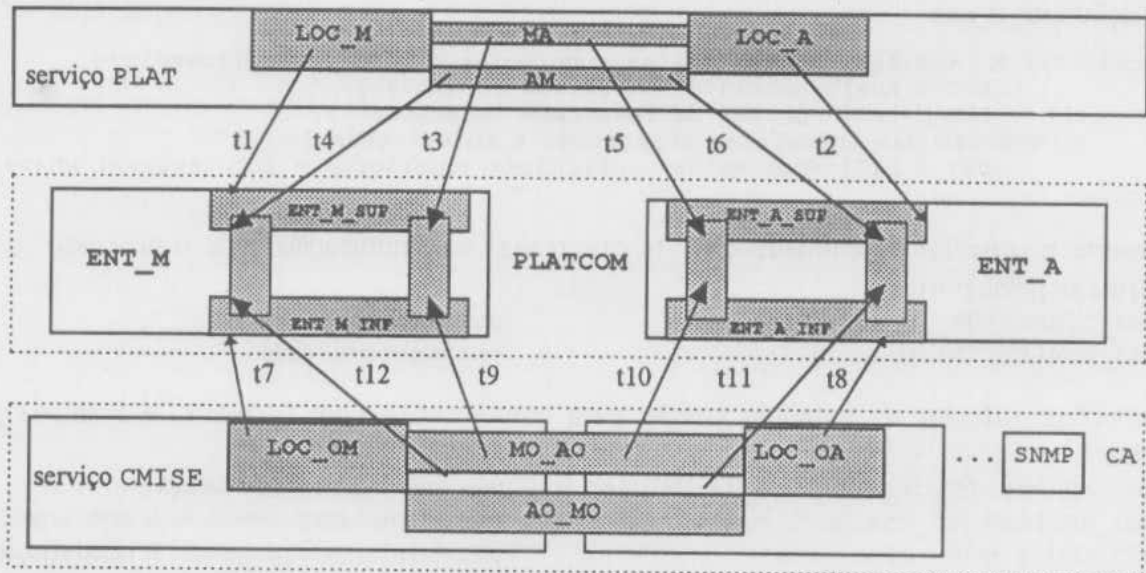


Figura 3.10 - Derivação das restrições do protocolo PLAT.

### a) Derivação das interfaces superiores das entidades ENT\_M e ENT\_A

A derivação do processo ENT\_M\_SUP é realizada com a aplicação da transformação  $t_1$  sobre o processo LOC\_M e a derivação do processo ENT\_A\_SUP é realizada com a aplicação da transformação  $t_2$  sobre o processo LOC\_A. Veja a Figura 3.11.

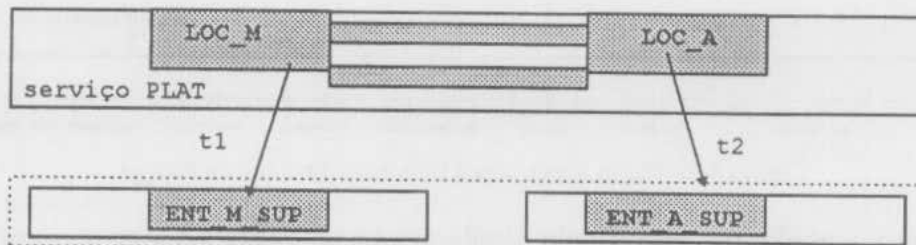


Figura 3.11 - Derivação das restrições ENT\_M\_SUP e ENT\_A\_SUP.

$$\text{ENT\_M\_SUP} = t_1(\text{LOC\_M})$$

$$\text{ENT\_A\_SUP} = t_2(\text{LOC\_A})$$

A interface superior da entidade ENT\_M espelha o comportamento do processo LOC\_M, que é receber operações e entregar notificações. A transformação  $t_1$  é considerada uma transformação identidade do processo LOC\_M para o processo ENT\_M\_SUP.

```
process ENT_M_SUP[MO_OPERATION,MO_NOTIFICATION]: noexit :=
  MO_OPERATION;ENT_M_SUP[...] [] MO_NOTIFICATION;ENT_M_SUP[...]
endproc
```

Similarmente à transformação  $t_1$ , que espelha o comportamento do processo LOC\_M, a transformação  $t_2$  espelha o comportamento do processo LOC\_A para o processo ENT\_A\_SUP.

```
process ENT_A_SUP[MO_OPERATION,MO_NOTIFICATION]: noexit :=
  AO_OPERATION;ENT_A_SUP[...] [] AO_NOTIFICATION;ENT_A_SUP[...]
endproc
```

### b) Derivação das interfaces inferiores das entidades ENT\_M e ENT\_A

A derivação do processo ENT\_M\_INF é realizada com a aplicação da transformação  $t_7$  sobre o processo LOC\_MO. A derivação do processo ENT\_A\_INF é realizada com a aplicação da transformação  $t_8$  sobre o processo LOC\_AO. Veja a Figura 3.12.

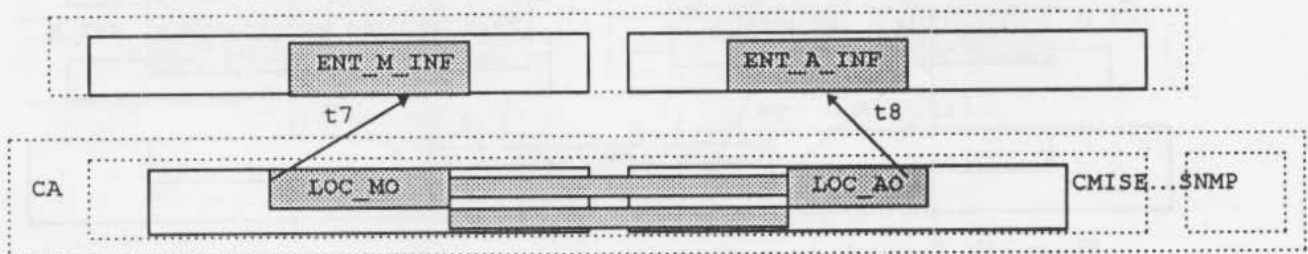


Figura 3.12 - Derivação das restrições ENT\_M\_INF e ENT\_A\_INF.

$$\text{ENT\_M\_INF} = t_7(\text{LOC\_MO})$$

$$\text{ENT\_A\_INF} = t_8(\text{LOC\_AO})$$

```
process ENT_M_INF[ <portas inferiores> ];noexit:=
  ((EM_ASSOCIATION[...] [> EM_TERM1[...]])
  >>((EM_OPERATION[...]||EM_NOTIFICATION[...]) [>EM_TERM2[...]) [>EM_TERM1[...]))
  >> ENT_M_INF[...] where ...
endproc
```

Os processos EM\_ASSOCIATION, EM\_TERM1, EM\_TERM2, EM\_OPERATION e EM\_NOTIFICATION apresentam comportamentos similares aos processos MO\_ASSOCIATION, MO\_TERM1, MO\_TERM2, MO\_OPERATION e MO\_NOTIFICATION, respectivamente.

### c) Derivação das ligações entre as interfaces das entidades ENT\_M e ENT\_A

A derivação do processo ENT\_M\_LIG\_IS\_O, que corresponde ao fluxo da interface inferior da ENT\_M para a interface superior da ENT\_M, é realizada a partir da transformação  $t_4$  e da transformação  $t_{12}$ . A derivação do processo ENT\_M\_LIG\_SI\_O, que corresponde ao fluxo da interface superior da ENT\_M para a interface inferior da ENT\_M, é realizada a partir da transformação  $t_3$  e da transformação  $t_9$ .

$$\text{ENT\_M\_LIG\_IS\_O} = t_4(\text{MO\_AM}) + t_{12}(\text{AO\_MO})$$

$$\text{ENT\_M\_LIG\_SI\_O} = t_3(\text{MO\_MA}) + t_9(\text{MO\_AO})$$

A derivação do processo ENT\_A\_LIG\_IS\_O que corresponde ao fluxo da interface inferior da ENT\_A para a interface superior da ENT\_A, é realizada a partir da transformação  $t_5$  e da transformação  $t_{10}$ . A derivação do processo ENT\_A\_LIG\_SI\_O que corresponde ao fluxo da interface superior da ENT\_A para a interface inferior da ENT\_A, é realizada a partir da transformação  $t_6$  e da transformação  $t_{11}$ .

$$\text{ENT\_A\_LIG\_IS\_O} = t_5(\text{MA}) + t_{10}(\text{MO\_AO})$$

$$\text{ENT\_A\_LIG\_SI\_O} = t_6(\text{AM}) + t_{11}(\text{AO\_MO})$$

Veja a Figura 3.13.

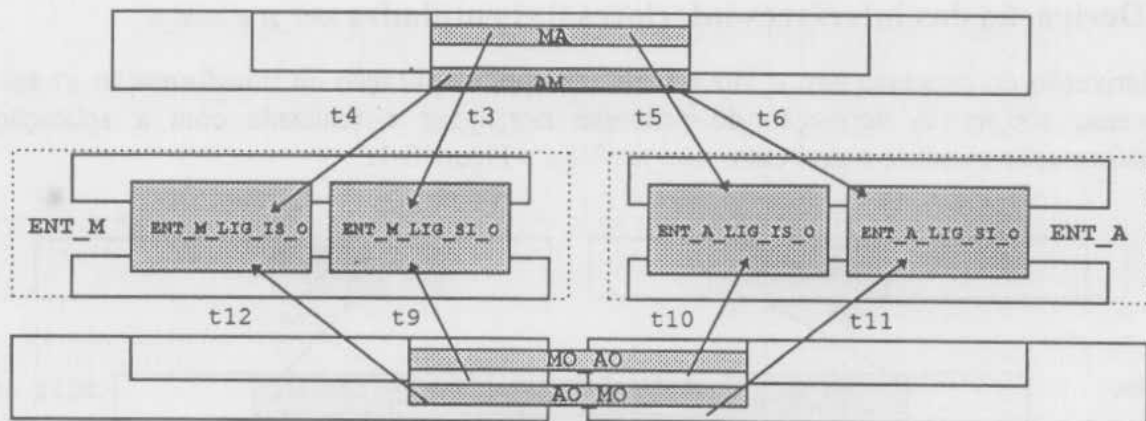


Figura 3.13 - Derivação das restrições de ligação entre as interfaces de ENT\_M e ENT\_A.

### 3.1.2 Protocolos de outros recursos da plataforma

O protocolo do recurso de **Interface Homem-Máquina** pode seguir um padrão gráfico baseado em janelas (*GUI - Graphical User Interface*). Nesse caso, um dos padrões gráficos que podem ser utilizados é o *OSF Motif Toolkit*, fornecido pela *Open Software Foundation*, que é promovido por companhias como *DEC*, *Hewlett-Packard* e *Microsoft*.

O protocolo do recurso de **Sistema Gerenciador de Banco de Dados (SGBD)**, pode ter independência de gerenciador de Base de Dados, com servidor *SQL (Structured Query Language)*. A linguagem *SQL* está presente em documentos *ANSI X2.135-1986* e *ISO/TC97/SC21/WG3N117*. A *SQL* é composta por um grupo de operações para definição, manipulação e controle de dados em um BD relacional sendo independente de hardware. *SELECT*, *INSERT*, *UPDATE* e *DELETE* são algumas operações *SQL* sobre uma Base de Dados.

As **Ferramentas para o Desenvolvimento de Aplicações** podem contribuir, por exemplo, para a criação de novos provedores de serviços para otimizar um serviço específico. Outra possibilidade é a inclusão de novas primitivas em um serviço específico.

## 4 Conclusões

*"Obviously, standards development is an important application area for formal description techniques. This activity is substantially broader than definition of programming language syntax and semantics. (...)*

*The ultimate goal of a standard is to provide good specifications.*

*If good specifications are not provided, compliance with such a standard will not guarantee interoperability.*

*The reason in simple: incomplete or unclear definition of semantic" [Kilo 94].*

Ultimamente têm sido propostas plataformas para integrar aplicações de gerência de redes. Veja, por exemplo, [FrIs 93] e [FaIm 93]. O presente trabalho propõe a especificação formal de uma plataforma para dar apoio a aplicações genéricas de gerência no caso de redes heterogêneas. Ao contrário das soluções de gerenciamento proprietárias, que são de difícil interoperabilidade, a plataforma proposta neste trabalho foi concebida para integrar aplicações genéricas de gerência.

A plataforma proposta possui duas interfaces. O primeiro tipo de interface (interface superior da plataforma) refere-se à interação da plataforma com as aplicações genéricas de gerenciamento, permitindo a comunicação da plataforma com os agentes e gerentes das aplicações genéricas de gerência. O segundo tipo de interface (interface inferior da plataforma)



refere-se à interação da plataforma com as aplicações de uso comum da camada de aplicação, permitindo à plataforma o uso do protocolo SNMP e do CMIS/CMIP, por exemplo.

A especificação da plataforma é realizada com a utilização da TDF LOTOS e da metodologia proposta em [NoRi 94a,b,c,d,e] baseada em estilos de especificação, refinamentos sucessivos e transformações. Através do estilo orientado para restrições e do estilo orientado para recursos são descritos o serviço e o protocolo de comunicação da aplicação de gerência LOG. Para a descrição do protocolo são utilizadas transformações sobre a especificação de serviço de PLAT e do serviço subjacente à plataforma. Metodologias baseadas em transformações são também apresentadas em [KaHi 90] [BoGo 86] e [Pire 94]. Nestes trabalhos são relatadas as dificuldades de desenvolvimento de trabalhos nessa área, principalmente relacionadas às regras de transformações que não conseguem manipular mais do que 'toy examples'.

## 5 Agradecimentos

Os autores agradecem ao Grupo de Redes e Gerência da UFSC (especialmente à professora Elizabeth Sueli Specialski, ao professor Carlos Becker Westphall e ao professor José Eduardo De Lucca), ao colega Alexandre Moraes e ao Engenheiro Paulo Francisco de Vilhena Toledo (BRISA) pelas discussões sobre a concepção desta plataforma.

Agradecem também à BRISA e aos pesquisadores Paulo Roberto Freire Cunha (Universidade Federal de Pernambuco), Luís Ferreira Pires (Universiteit Twente - Holanda), Juan Jose Gil Rios (Centro para el Desarrollo Tecnológico Industrial -Espanha), Steve Van Trees (Federal Aviation Administration/George Mason University - USA), Ana Maria Moreira e Paul Gibson (University of Stirling - Escócia), Jose A. Mañas (Univ. Madrid - Espanha) e Laurent Mounier, Alain Kerbrat, Jean-Claude Fernandez e Hubert Garavel (IMAG - França), que colaboraram com o envio de normas, anais, especificações, artigos e ferramentas que auxiliaram a elaboração deste trabalho.

## Anexo

A verificação sintática, verificação semântica e simulação exaustiva do sistema MS estão sendo realizadas com a ferramenta Caesar (VERIMAG-França, hubert@imag.imag.fr). Com o uso do Open/Caesar, é possível a simulação interativa, a geração de teste e a detecção de deadlock.

O arquivo que contém a especificação lotos (ms.lotos) possui 1196 linhas.

```
-rw-r--r-- 1 mirella 109562 Jan 06 00:54 ms.lotos
```

Foi utilizado o seguinte comando de linha no sistema operacional UNIX em uma estação Sun:

```
caesar -aldebaran ms.lotos
```

Foi apresentado o seguinte resultado:

```
caesar : analyse syntaxique de ``ms``
caesar : analyse semantique de ``ms``
caesar :   - liaison des portes
caesar :   - liaison des processus
caesar :   - liaison des types
caesar :   - calcul des signatures
caesar :   - liaison des sortes
caesar :   - liaison des variables
caesar :   - liaison des operations
```

```

caesar :    - calcul des fonctionnalites
caesar : restriction de ``ms``
caesar : expansion de ``ms``
caesar : examen des types de ``ms``
caesar : generation de ``ms``
caesar : optimisation de ``ms``
caesar : simulation de ``ms``
caesar :    - production du simulateur
caesar :    - compilation du simulateur
caesar :    - execution du simulateur
caesar :    - impression du graphe de ``ms`` pour ``aldebaran``

```

Foram gerados os seguintes arquivos: `ms.aut` (produzido pelo uso da opção `-aldebaran`), `ms.gph` (grafos de estado) e `ms.net` (redes de Petri).

```

-rw-r--r--  1 mirella      1302 Oct 14 11:44 ms.aut
-rw-r--r--  1 mirella     11694 Oct 14 11:43 ms.gph
-rw-r--r--  1 mirella     11977 Oct 14 11:43 ms.net

```

No estágio atual do presente trabalho, estão sendo realizadas verificações da especificação com o uso da ferramenta Aldébaran (IMAG, fernand@imag.imag.fr), a partir do arquivo `ms.aut`, gerado com a ferramenta *Cæsar*.

Com a ferramenta *Aldébaran* são possíveis várias verificações (por redução ou por comparação), como por exemplo, *strong bisimulation equivalence*, *delay bisimulation*, *observational relation*, *branching bisimulation*, *safety relation*, *livelock states* e *deadlock states*.

De forma pioneira, a especificação está sendo testada na França com a ferramenta *Magel* (Laurent.Mounier@imag.fr). Tal ferramenta, que suporta grandes especificações, verificou a inexistência de deadlocks na especificação e também a quantidade de estados:  $6.85739e+15$ .

Com a ferramenta *TOPO* (Universidade de Madrid, topo@dit.upm.es) foi possível a tradução da especificação `ms.lot` para a linguagem de programação *C*. O arquivo gerado tem 7007 linhas.

Alguns comandos de linha da ferramenta *TOPO*, para a tradução da especificação *LOTOS* em código *C*, no sistema operacional *UNIX* de uma estação *Sun*:

```

setenv TOPO
toposet ms.lot -tname ms
topo ms.lot -behaviour

```

Alguns arquivos produzidos foram:

```

-rw-r--r--  1 mirella      13047 Oct 13 18:00 ms.lfe
-rw-r--r--  1 mirella       1499 Oct 13 18:02 ms.as
-rw-r--r--  1 mirella       2499 Oct 13 18:04 ms.asf
-rw-r--r--  1 mirella     21042 Oct 13 18:04 ms.lbm
-rwxrwxrwx  1 mirella       1105 Oct 13 11:06 ms.ctx
-rw-r--r--  1 mirella       3542 Oct 13 18:12 ms.agf
-rw-r--r--  1 mirella     23546 Oct 13 18:02 ms.ls
-rw-r--r--  1 mirella     13205 Oct 13 18:04 ms.lsf
-rw-r--r--  1 mirella     23546 Oct 13 18:02 ms.c
-rw-r--r--  1 mirella     13205 Oct 13 18:04 ms.hh

```

## 6 Bibliografia

- [BaFi 94] Bauer, M.; Finnigan, P. F.; Hong, J. W.; Rolia, J. A.; Teorey, T. J.; Winters, G. A. "Reference architecture for distributed systems management". IBM S. Journal, 02/94.
- [BoGo 86] Bochmann, G. von; Gotzhein, R. "Deriving protocol specifications from service specifications". in: Communications, Architectures & Protocols, Proceedings of the ACM SIGCOMM'86 Symposium, Vemst, USA, 1986.
- [BoBr 87] Bolognesi, T.; Brinksma, E. "Introduction to the OSI specification language LOTOS". Computer Networks and ISDN Systems, V.14, pp. 25-29, 1987.
- [FaIm 93] Fayan, B. L.; Imai, C. T.; Faber, M. B.; Lorena, P. S. "Plataforma de suporte a aplicações de gerência". Revista Telebrás, 12/1993, pp. 110-121.
- [FeGa 93] Fernandez, J. C.; Garavel, H.; Mounier, L.; Rasse A.; Rodriguez, C.; Sifakis, J. "A toolbox for the verification of LOTOS programs". LGI-IMAG/VERILOG, França, 1993, p.14.
- [FrAu 89] Freestone, D.; Aujla, S. "Specifying ROSE in LOTOS". Elsevier Science Publishers, North-Holland, 1989, pp. 231-245.
- [FrIs 93] Freitas, J. P.; Ishibashi, W. W. "Princípios de construção de software para gerência integrada de redes e serviços". Revista Telebrás, 12/1993, pp. 103-109.
- [FrNo 94] Franceschi, A. S. M. de; Notare, M. S. M. A.; Riso, B. G. "Combinando ferramentas de auxílio a utilização de LOTOS para o desenvolvimento de protocolos", INFONOR'94. Universidad Catolica del Norte, Antofogasta - Chile, 11/94.
- [Gara 94] Garavel, H. "Cæsar 3.7 refernce manual". Grenoble - France, 1994, p. 32.
- [HiBo 94] Higashino, T.; Bochmann, G. von. "Automatic analysis and test case derivation for a restricted class of LOTOS expressions with data parameters". IEEE Transactions on Software Engineering. Vol.20, No.1, 01/94, pp. 29-42.
- [IS 8649] ISO: "Information processing systems - Open systems interconnection - Service definition for the Association Control Service Element", 1988.
- [IS 8650] ISO: "Information processing systems - Open systems interconnection - Protocol specification for the Association Control Service Element", 1990.
- [IS 8807] ISO: "Information processing systems - Open systems interconnection - LOTOS - A formal description technique based on the temporal ordering of observational behaviour", 1988.
- [IS 9595] ISO: "International Organization for Standardization - Information technology - Open Systems Interconnection - Common management information service definition", 1991.
- [IS 9596] ISO: "International Organization for Standardization - Information technology - Open Systems Interconnection - Common management information protocol", 1991.
- [KaHi 90] Kant, C.; Higashino, T.; Bochmann, G. von "Deriving protocol specifications from service specifications written in Basic LOTOS". Montreal, Canada, 1990.
- [KaSu 94] Kazama, K.; Suzuki, S.; Hatafuku, M.; Takahashi, K. "Development of interconnectability testing system". SBT/IEEE International Telecommunications Symposium. Rio de Janeiro RJ, 22-26/08/94, pp. 410-414.
- [Kilo 94] Kilov, Haim. "Formal methods and standards". Software Engineering Notes, vol. 19 no. 3, 07/94, pp. 40.
- [KuMe 94] Kurshan, R. P; Merritt, M. "Formal verification of fault-tolerant distributed algorithms". SBT/IEEE International Telecommunications Symposium. Rio de

- Janeiro RJ, 22-26/08/94, pp. 403-409.
- [Lang 90] Langerak, R. "Decomposition of functionality: a correctness preserving LOTOS transformation". X International Symposium on Protocol Specification, Testing and Verification, Ottawa - Canadá, pp 203-218, IFIP, 1990.
- [LePe 94] Lehmann Jr, E. O; Pedroza, A. C. P. "Especificação e verificação do protocolo CMIP para gerenciamento de rede". XII SBRC, 16-20/05/94, Curitiba PR, pp 639-658.
- [NoRi 94a] Notare, M. S. M. A.; Riso, B. G. "Utilização de restrições e recursos predefinidos na especificação de protocolos com a TDF LOTOS". XII SBRC, Curitiba PR, 16-20/05/1994, pp. 225-243.
- [NoRi 94b] Notare, M. S. M. A.; Riso, B. G. "Transformação de especificações com a técnica de descrição formal LOTOS". 2 Jornada USP - SUCESU-SP de Informática e Telecomunicações, São Paulo SP, 30/05-01/06/94, pp. 263-272.
- [NoRi 94c] Notare, M. S. M. A.; Riso, B. G. "Definição e especificação LOTOS de uma plataforma para dar suporte a aplicações de gerência de redes". TELEMÁTICA'94, Porto Alegre RS, 23-25/08/1994.
- [NoRi 94d] Notare, M. S. M. A.; Riso, B. G. "Uma plataforma para integrar aplicações de gerência em redes heterogêneas". Resumo - SUCESU'94 - XXVII Congresso Nacional de Informática e Telecomunicações. Salvador BA, 21-24/11/94.
- [NoRi 94e] Notare, M. S. M. A.; Riso, B. G. "Uma plataforma para dar suporte a aplicações de gerência em redes heterogêneas". Trabalho submetido a publicação.
- [Nota 93] Notare, M. S. M. A. "Transformação de especificações com a técnica de descrição formal LOTOS", Trabalho Individual, UFSC, 10/1993, p. 56.
- [PiLo 90] Pires, L. F.; Lopes de Souza, W. "Step-wise refinements design using LOTOS". III International Conference on Formal Description Techniques, Madrid, 1990.
- [Pire 94] Pires, L. F. "Architectural Notes: a framework for distributed systems development". CTIT Ph.D-thesis series No. 94-01, The Netherlands, p. 255.
- [PiSi 92] Pires, L. F.; Sinderen, M. van; Vissers, C. "On the use of predefined implementation constructs in distributed systems design". Universiteit Twente, 1992.
- [QuCu 94] Queiroz, J. A. M. de; Cunha, P. R. F. "Sistemas distribuídos: de especificações LOTOS a implementações". IX Escola de Computação. Recife, 24-31/07/94, p.208.
- [ReFr 93a] Rebelles, P.; Freitas, J. P. "Introdução aos modelos genéricos de arquitetura para rede de gerência de telecomunicações (TMN)". Revista Telebrás, 12/1993, pp. 12-23.
- [ReFr 93b] Rebelles, P.; Freitas, J. P. "Modelagem de informação aplicada à gerência integrada de redes de telecomunicações". Revista Telebrás, 12/1993, pp. 134-146.
- [ToNo 94] Tonin, N. A.; Notare, M. S. M. A.; Riso, B. G. "Algoritmos para a transformação de especificações LOTOS". Resumo. SUCESU'94 - XXVII Congresso Nacional de Informática e Telecomunicações. Salvador BA, 21-24/11/94.
- [ToNo 95a] Tonin, N. A.; Notare, M. S. M. A.; Riso, B. G. "Transformação de especificações de serviço em especificações de protocolo". Trabalho submetido a publicação.
- [ToNo 95b] Tonin, N. A.; Notare, M. S. M. A.; Riso, B. G. "Especificação LOTOS de uma célula flexível de montagem". Trabalho submetido a publicação.
- [Viho 94] Viho, C. G. "Errors suppression in protocols and service preservation". SBT/IEEE ITS 94. Rio de Janeiro RJ, 22-26/08/94, pp. 398-402.
- [ViSc 88] Vissers, C. A.; Scallo, G.; Sinderen M. van. "Architecture and specification style in formal descriptions of distributed systems", Universiteit Twente - Holanda, 1988.