

PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 11/95

**Modelo da Implementação (Design) de um
Sistema de Controle para uma Área de
Linhas de Engarrafamento**

Ellens Barbosa
Bruno Maffeo

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900

RIO DE JANEIRO - BRASIL

PUC RIO - DEPARTAMENTO DE INFORMÁTICA

ISSN 0103-9741

Monografias em Ciência da Computação, Nº 11/95

Editor: Carlos J. P. Lucena

Maio, 1995

**Modelo da Implementação (Design) de um
Sistema de Controle para uma Área de
Linhas de Engarrafamento***

Ellens Barbosa

Bruno Maffeo

* Trabalho patrocinado pelo Ministério de Ciência e Tecnologia da
Presidência da República Federativa do Brasil.

In charge of publications:

Rosane Teles Lins Castilho

Assessoria de Biblioteca, Documentação e Informação

PUC Rio — Departamento de Informática

Rua Marquês de São Vicente, 225 — Gávea

22453-900 — Rio de Janeiro, RJ

Brasil

Tel. +55-21-529 9386

Telex +55-21-31048

Fax +55-21-511 5645

E-mail: rosane@inf.puc-rio.br

Modelo da Implementação (Design)
de um
Sistema de Controle para uma Área de Linhas de Engarrafamento
de um
Composto Químico

Ellens Barbosa
Bruno Maffeo
(e-mail : maffeo@inf.puc-rio.br)

PUCRioInf - MCC 11/95

Abstract - This work constitutes an application of conceptual tools and modelling techniques for the construction of the Implementation Model (Design) for a real-time system which monitors and controls a chemical product bottling lines area. It employs extensions, proposed for the treatment of real-time systems, of tools and techniques used for the modelling of conventional socio-technical systems. Using an activity-oriented approach, it addresses aspects related to the representation of system dynamics and the choice of a viable implementation alternative.

Keywords - systems modelling, real-time systems design, process-control systems design, Implementation Model (Design), Processor Model, Task Model, Module Model.

Resumo - Este trabalho constitui uma aplicação de ferramentas conceituais e técnicas de modelagem para a construção do Modelo da Implementação (Design) de um sistema de tempo-real para monitorar e controlar uma área de linhas de engarrafamento de um composto químico. São utilizadas extensões, propostas para o tratamento sistemas de tempo-real, de ferramentas e técnicas empregadas na modelagem de sistemas sócio-técnicos convencionais. Usando uma abordagem orientada para atividades, são visados os aspectos relacionados à representação da dinâmica do sistema e à escolha de uma alternativa de implementação viável.

Palavras-chave - modelagem de sistemas, design de sistemas de tempo-real, design de sistemas para controle de processos, modelagem conceitual, Modelo da Implementação (Design), Modelo da Configuração de Processadores, Modelo da Configuração de Processos, Modelo da Configuração de Módulos.

ÍNDICE

1. INTRODUÇÃO	1
2. O MODELO DA IMPLEMENTAÇÃO (DESIGN)	2
3. O MODELO DA IMPLEMENTAÇÃO (DESIGN) DO SICLEN.....	6
4. ALTERNATIVAS DE MCPR DO SICLEN.....	11
5. A ALTERNATIVA ESCOLHIDA - MCP, MCM E DICIONÁRIO DE DADOS.....	68
6. CONCLUSÃO	84
REFERÊNCIAS BIBLIOGRÁFICAS.....	87
APÊNDICE.....	90

1. Introdução

Em um sentido amplo, especificar um sistema pode ser considerado como o processo de traduzir necessidades representadas na mente (concepção) de uma ou mais pessoas para um modelo de sistema que satisfaça essas necessidades (implementação). Assim como a linguagem natural incorpora o conjunto de símbolos e regras que usamos para comunicar informalmente fatos sobre a realidade que nos rodeia, o processo de desenvolvimento de sistemas requer uma linguagem adequada não só ao propósito de especificar rigorosamente requisitos funcionais, mas também ao de analisá-los visando propor alternativas de implementação.

No caso dos *sistemas de tempo-real (STR)*, tal perspectiva reveste-se de importância crucial pois, com a crescente disponibilidade de processadores decorrente do avanço tecnológico, tendemos cada vez mais a confiar-lhes até nossas próprias vidas. Automóveis, aviões, navios, usinas nucleares e muitas outras realizações humanas dependem de controle em tempo-real para operar com segurança. Falhas que venham a ocorrer nesses sistemas podem causar sérios prejuízos: na melhor das hipóteses, apenas perda de tempo e recursos financeiros e materiais; na pior, perdas de vidas humanas [1,2,6,11]. Assim sendo, considera-se fundamental para o processo de desenvolvimento de sistemas dessa classe a existência de um instrumento (ou conjunto de instrumentos) que permita percorrer, com mais segurança e qualidade [17], a distância entre a concepção e a implementação.

Como foi visto em [5,12,15,23], o objetivo básico a ser atingido através da construção do Modelo da Essência é a descrição completa, concisa e precisa dos requisitos essenciais do sistema. Para isso, busca-se formulação do problema de automação a ser resolvido - o Modelo do Contexto - e de uma solução abstrata em relação a quaisquer alternativas de implementação - o Modelo do Comportamento. Entretanto, exceto nos casos de sistemas muito simples, esse modelo não pode ser usado como uma planta para a criação da configuração de hardware e software que efetivamente executará o trabalho do sistema. Da proposta de solução abstrata à efetiva implantação do sistema, é necessário proceder a uma escolha criteriosa da alternativa de implementação, na qual esteja incorporada a consideração de aspectos ligados às imposições e/ou limitações ambientais que porventura possam existir, tais como, entre outras, as associadas a custo de software/hardware, distribuição de recursos operacionais, exigências de desempenho e confiabilidade e qualidade das interfaces com usuários.

A complexidade dessa nova etapa do processo de desenvolvimento pode ser melhor administrada através da construção de um segundo modelo - o Modelo da Implementação [9,23] -, sucessor lógico do Modelo da Essência. O Modelo da Implementação deve possibilitar o estudo e a aferição das diversas possibilidades de implementação, permitindo eleger a solução física mais adequada que incorpore os requisitos funcionais elicitados e representados no Modelo da Essência. Em síntese, o Modelo da Implementação destina-se a tornar visível o design do sistema a ser construído, bem como a registrar os critérios de decisão que nortearam a escolha de suas diferentes facetas. Esse modelo agrega várias das vantagens analíticas já constatadas no Modelo da Essência e, em particular, a estrutura adotada neste trabalho - Modelo da Configuração de Processadores, Modelo da Configuração de Processos e Modelo da Configuração de Módulos - parece ser analogamente conveniente para a fase exploratória dos aspectos de implementação intrínsecos à solução física. Além disso, esse modelo permite especificar e comunicar rigorosamente o conjunto de requisitos operacionais não-essenciais necessários à efetiva construção e implantação do sistema. A estrutura adotada caracteriza uma abordagem orientada a atividades, privilegiando a visualização dos elementos funcionais ativos (processadores, processos e módulos) que darão

forma à implementação do sistema. Um design orientado a objetos poderia também ter sido visado, tendo como base conceitual o Modelo da Essência. Assim como na modelagem da essência, o processo de modelagem da alternativa de implementação traduz-se na aplicação cuidadosa de um conjunto de critérios de segmentação e/ou abstração (esta última em grau inferior àquele adotado na modelagem da essência) visando dividir um problema complexo em domínios de menor porte e/ou densidade, compatíveis com a capacidade de apreensão do ser humano.

Este trabalho visa demonstrar construtivamente a pertinência de aplicar-se as ferramentas conceituais e técnicas de modelagem da implementação de STR propostas em [9] e parcialmente utilizadas em [25]. A aplicação refere-se ao **SICLEN - Sistema de Controle de Linhas de ENgarramento** -, cuja modelagem da essência foi apresentada em [26]. Conforme descrito nesta última referência, o SICLEN é uma versão aperfeiçoada do sistema modelado no apêndice B de [8] e tem por objetivo monitorar e controlar um conjunto de linhas de engarramento de um composto químico que deve ter seu pH estabilizado. Em alto nível de abstração, serão analisadas, através de seus respectivos Modelos de Configuração de Processadores, duas possíveis alternativas de implementação. Através dessa análise, será evidenciada a importância dessa modelagem de alto nível no sentido de proporcionar critérios objetivos que permitam selecionar a alternativa mais adequada para a implementação do sistema.

O tratamento adotado para a elaboração desses modelos apresenta uma ampliação das ferramentas e técnicas de modelagem da implementação empregadas em [9,25], cabendo mencionar, todavia, que não pertence ao escopo deste trabalho qualquer análise comparativa a propostas de outros autores.

2. O Modelo da Implementação (Design)

Conforme mencionado no item anterior, em termos da necessária abstração de características físicas relativas ao uso da tecnologia de implementação a ser empregada na construção de um sistema sócio-técnico, há que se lidar, normalmente, com um considerável desnível entre o conteúdo do Modelo da Essência (requisitos essenciais) e a efetiva construção do sistema (procedimentos manuais, código inteligível por máquinas digitais, computadores e outros dispositivos de hardware). Para que esse desnível seja ultrapassado com segurança, é imprescindível a construção de um outro modelo, de nível intermediário de abstração, que sirva de planta para a elaboração dos aspectos físicos do sistema. Esse é o Modelo da Implementação (MI), obtido a partir do Modelo da Essência (ME) através de um processo que, geralmente, envolve reorganização e expansão.

A reorganização do ME consistirá na atribuição:

- ◆ das ações programadas para as diversas atividades essenciais a unidades de execução;
- ◆ de partes da memória essencial a unidades de armazenamento.

Essas unidades (processadores, processos, módulos, bancos de dados, tabelas, arquivos etc.) são os elementos físicos básicos do MI, univocamente identificáveis no ambiente no ambiente operacional de que farão parte.

A expansão do ME, por sua vez, implicará a adição de outras atividades e/ou informações não-essenciais, visando superar a hipótese nada realista usada na sua construção referente ao emprego de tecnologia ideal na implementação do sistema. Conseqüentemente, o

processo de obtenção do MI irá, não raro, requerer também alguma distorção em relação à modelagem da essência (por exemplo, alocar partes de uma atividade primitiva a processadores distintos) decorrente dessa necessidade de compensar as limitações da tecnologia disponível.

Em nível decrescente de abstração em relação às características físicas do sistema a ser construído, o MI é composto pelos seguintes submodelos [9,23] :

A) Modelo da Configuração de Processadores (MCPr)

O objetivo do MCPr é retratar, em nível elevado de abstração e dentro do contexto de uma opção de design específica, a alocação dos aspectos funcionais/informacionais contidos no Modelo da Essência, bem como daqueles não-essenciais que se opte por incorporar ao sistema.

O componente básico do MCPr é o **processador**, unidade de execução característica desse submodelo. Um processador pode ser pessoa ou máquina capaz de executar instruções e processar dados. Portanto, nesse modelo, pretende-se representar a arquitetura de processadores que servirá de base para a implementação do sistema. Certamente, a escolha correta da alternativa de implementação mais adequada às necessidades do cliente/usuário requer, garantida a viabilidade técnica e financeira do sistema, uma análise comparativa baseada em cálculos de custo/benefício; os MCPr associados às diferentes alternativas que venham a ser propostas constituem instrumentos apropriados a essa análise.

Eventualmente, restrições impostas pelo ambiente externo e/ou limitações de recursos financeiros implicarão uma implementação constituída de um único computador. Entretanto, numa fase exploratória, pode-se imaginar que diversos processadores digitais, atuando de modo concorrente, realizarão as atividades contidas no ME utilizando as informações nele especificadas. No MCPr são também descritas as interfaces entre os processadores - isto é, as comunicações existentes entre eles e suas eventuais necessidades de sincronização - e as interfaces entre os processadores e o ambiente externo.

B) Modelo da Configuração de Processos (MCP)

Neste segundo nível, entende-se que cada processador pode possuir diversas unidades de execução - **processos**. Genericamente, consideramos um processo qualquer conjunto de ações/informações que possa ser referenciado globalmente no ambiente operacional de um dado processador. Em outras palavras, trata-se de uma unidade de execução característica do MCP composta por um lote de instruções de aplicação que seja tratado univocamente na arquitetura de hardware e software básico do processador que o contém. Um processo também é designado por outros nomes tais como "programa", "co-rotina", "task" etc.

O MCP destina-se a representar a alocação da parte do ME que coube a um dado processador às unidades de execução e armazenamento típicas da arquitetura desse processador, incluindo também as atividades que garantirão a comunicação e coordenação eficiente das ações entre os processadores.

Contrariamente ao que ocorre no MCPr, onde supõe-se a possibilidade de real paralelismo na execução de ações alocadas a processadores distintos, admite-se no MCP apenas uma concorrência simulada na execução de processos alocados ao mesmo processador (multi-programação). Além disso, quando houver necessidade de troca de mensagens entre processos visando estabelecer sincronismo de ações, considera-se que essa troca deverá

ocorrer através do ambiente operacional do processador (habitualmente, por meio do sistema operacional).

C) Modelo da Configuração de Módulos (MCM)

Este estágio do processo de modelagem envolve a reorganização das ações e informações alocadas a cada processo estabelecido no MCP, objetivando facilitar a construção do código interpretável por processadores digitais. O termo **módulo** [29] será usado para referenciar a unidade de execução deste modelo, compatível com o ambiente operacional de um processo. Outros termos, tais como “sub-rotina”, “procedimento” etc., são também comumente empregados.

O módulo, diferentemente do processador e do processo, associa-se estritamente a processamento seqüencial (não concorrente). Módulos relacionam-se através de chamadas síncronas, onde o módulo chamado assume o controle durante o processamento de sua tarefa e o devolve ao módulo chamador ao concluir sua execução. Não se pressupõe, portanto, a existência de processamento concorrente no MCM.

O MCM, que tradicionalmente é expresso através da linguagem de representação denominada Diagrama Hierárquico de Módulos, visa primordialmente definir uma estrutura de controle hierarquizada que favoreça uma geração de código correto e fácil de manter.

Por constituir basicamente o formato usado neste trabalho, vale descrever sucintamente a forma de apresentação do Modelo da Implementação baseada nas propostas existentes em [9,23] :

1. MODELO DA CONFIGURAÇÃO DE PROCESSADORES

1.1 DEFINIÇÃO DE ASPECTOS DE IMPLEMENTAÇÃO

Esta seção é constituída por uma descrição textual a qual, eventualmente ilustrada por figuras e gráficos, apresenta os elementos básicos da alternativa de implementação em relação aos processadores. São definidas as características próprias de cada processador (configuração de hardware, softwares básicos, funcionalidade no contexto do sistema, interfaces com o ambiente externo e entre os processadores etc.) bem como os critérios de alocação que orientaram a escolha da arquitetura do sistema e a alocação das atividades essenciais e não-essenciais.

1.2 SEÇÃO ESQUEMÁTICA

É a parte do MCPr composta pelos diagramas representativos da arquitetura do sistema e da alocação das atividades essenciais e não-essenciais, empregando basicamente o Diagrama de Fluxos de Dados e Controle (DFDC), extensão do DFD que permite modelar aspectos relacionados a tempo-real, e o Diagrama de Estados e Transições (DET), que permite especificar atividades de controle.

Esta seção subdivide-se em :

1.2.1 Esquema dos Processadores

Representação gráfica, em alto nível de abstração, dos processadores que compõem o sistema. É análogo ao Esquema das Atividades Essenciais do ME.

1.2.2 Organização Hierárquica dos Processadores

Similar à Organização Hierárquica das Atividades do ME, é uma representação gráfica que visa, através de uma apresentação “*top-down*”, dominar a complexidade do Esquema dos Processadores e facilitar a visão crítica do processo de construção do MCP_r bem como o processo de revisão do mesmo.

1.2.3 Alocação da Essência e de Requisitos Não-Essenciais a cada Processador

É a representação gráfica das atividades/informações essenciais e não-essenciais alocadas a cada processador separadamente. Opcionalmente, pode subdividir-se em:

1.2.3.1 Esquema das Atividades e Esquema da Informação;

1.2.3.2 Organização Hierárquica dos Esquemas das Atividades e da Informação.

2. MODELO DA CONFIGURAÇÃO DE PROCESSOS

2.1 DEFINIÇÃO DOS ASPECTOS DE IMPLEMENTAÇÃO DO FUNCIONAMENTO DE CADA PROCESSADOR EM TERMOS DOS PROCESSOS ALOCADOS

De modo análogo à seção Definição de Aspectos de Implementação do Sistema do MCP_r, constitui uma descrição textual, eventualmente apoiada por figuras e gráficos, apresentando as características genéricas do gerenciamento da multiplexação dos processos (por exemplo, tabelas de alocação dos processos de cada processador).

2.2 SEÇÃO ESQUEMÁTICA

É composta pelos diagramas representativos (DFDC e DET) dos processos alocados a cada processador, bem como das atividades/informações essenciais e não-essenciais alocados a cada processo. Subdivide-se em :

2.2.1 Esquema dos Processos

Representação gráfica (DFDC e DET) dos processos alocados a um dado processador. Considerando-se um processador como um sistema, é análogo ao Esquema das Atividades Essenciais do ME.

2.2.2 Organização Hierárquica dos Processos

Similar à Organização Hierárquica dos Processadores, é uma representação gráfica que visa, através de uma apresentação “*top-down*”, dominar a complexidade do Esquema dos Processos e facilitar a visão crítica do processo de construção do MCP assim como o processo de revisão do mesmo.

2.2.3 Alocação da Essência e de Requisitos Não-Essenciais a cada Processo

É a representação gráfica das atividades/informações essenciais e não-essenciais alocadas a cada processo separadamente. Opcionalmente, pode subdividir-se em :

1.2.3.1 Esquema das Atividades e Esquema da Informação; e

1.2.3.2 Organização Hierárquica dos Esquemas das Atividades e da Informação.

3. MODELO DA CONFIGURAÇÃO DE MÓDULOS

3.1 DIAGRAMAS HIERÁRQUICOS DE MÓDULOS ASSOCIADOS A CADA PROCESSO ALOCADO A CADA PROCESSADOR

Nesta seção deverão constar os DHM obtidos a partir dos DFDC de cada processo. Essa conversão é amplamente discutida em [29].

3.2 DESCRIÇÃO DETALHADA DE CADA MÓDULO ASSOCIADO A CADA PROCESSO ALOCADO A CADA PROCESSADOR

Constitui a representação do funcionamento de cada módulo de cada hierarquia. Esta representação pode ser textual (por exemplo, empregando pseudocódigo) ou não (por exemplo, empregando fluxogramas, diagramas de Nassi-Shneiderman, diagramas de estado-transição, árvores e tabelas de decisão etc.).

4. DICIONÁRIO DE DADOS

Uma vez que o processo de construção do MI realiza-se por reorganização e expansão do ME, torna-se necessário definir precisamente os novos elementos elicitados na modelagem da implementação. O Dicionário de Dados do MI é o repositório dessa definição, englobando e estendendo o Dicionário de Dados do ME. Desse modo, ficam asseguradas a completeza e a globalidade do processo de modelagem.

3. O Modelo da Implementação (Design) do SICLEN

De um modo geral, o Modelo da Implementação do SICLEN apresenta-se estruturado na forma acima descrita. Em seu detalhamento, algumas adaptações foram feitas de modo que o modelo retratasse todos os aspectos relevantes do sistema. Essas adaptações incluem uma ampliação do conjunto de ferramentas e técnicas de modelagem da implementação estabelecido

em [9,23] e constituem uma ampliação da proposta de apresentação contida em [23], necessária tendo em vista o objetivo de prover uma modelagem suficientemente flexível para permitir a especificação de todos os requisitos do sistema sem prejuízo do rigor semântico que o modelo deve possuir.

Conforme mencionado no texto introdutório, serão apresentadas duas alternativas de arquitetura de processadores (MCP_r) para o SICLEN, das quais uma será selecionada. Em seguida, serão apresentados o MCP e o MCM relativos à arquitetura escolhida. Assim sendo, preliminarmente, são apresentadas, contextualmente, as adaptações efetuadas.

3.1 Notação gráfica

A) Modelo da Configuração de Processadores

Emprega basicamente a notação gráfica adotada no Modelo da Essência com as seguintes extensões :

- Processadores

Os processadores são representados por retângulos fechados por linha cheia com estrutura interna tripla que incorpora :

a) um campo identificador que contém as letras "PR" seguidas da estrutura que identifica o processador - $m\{.n\}$, onde m e n são naturais seqüenciais - e, opcionalmente, os seguintes caracteres :

- "{i}", indicando a existência de mais de uma instância de um mesmo tipo de processador; no caso do SICLEN, "i" é um número natural seqüencial que identifica uma linha de engarrafamento;

- "*", indicando que trata-se de um processador primitivo (um único processador e não uma agregação de processadores) que, por isto, não possui detalhamento gráfico.

b) um campo descritor que descreve sucintamente a ação do processador;

c) um campo rastreador que permite o mapeamento do Esquema dos Processadores para a Organização Hierárquica dos Processadores; adicionalmente, quando for conveniente, poderá ser usado na Organização Hierárquica dos Processadores para indicar a área do ambiente externo à qual o processador esteja alocado.

- Atividades alocadas a um processador

As atividades essenciais e não-essenciais alocadas a um dado processador são representadas por retângulos fechados - por linha cheia no caso de uma atividade operacional e por linha vazada no caso de atividade de controle - com estrutura interna tripla que incorpora:

a) um campo identificador que contém a letras "G" no caso de atividade operacional ou "GC" no caso de atividade de controle, seguida da estrutura que identifica a atividade - m[{.n}], onde m e n são naturais seqüenciais - e opcionalmente os seguintes caracteres :

- "{i}", indicando a existência de mais de uma instância de um mesmo tipo de atividade; no caso do SICLEN, "i" é um número natural seqüencial que identifica uma linha de engarrafamento;

- "*" , indicando que a atividade não possui detalhamento gráfico, isto é, que trata-se de uma atividade primitiva.

b) um campo descritor que descreve sucintamente a ação da atividade;

c) um campo rastreador que permite o mapeamento das atividades essenciais estabelecidas no ME para os elementos estabelecidos nas seções do MCPPr correspondentes à alocação da essência e de requisitos não-essenciais a cada processador.

No processo de alocação da essência e de requisitos não-essenciais a elementos da tecnologia de implementação, pode ocorrer: 1º) alocação total de atividade essencial sem inclusão de requisitos não-essenciais; 2º) alocação parcial (segmentação da atividade) de atividade essencial sem inclusão de requisitos não-essenciais; 3º) alocação total de atividade essencial com inclusão de requisitos não-essenciais; 4º) alocação parcial com inclusão de requisitos não-essenciais; e 5º) inclusão de atividade composta exclusivamente de requisitos não-essenciais. Assim sendo, o campo rastreador deverá espelhar essas situações incluindo, respectivamente :

1º) somente o identificador da atividade empregado no ME, significando que trata-se exatamente da atividade estabelecida naquele modelo. Exemplo : F4.1;

2º) o identificador da atividade empregado no ME seguido de uma letra minúscula, significando que trata-se de uma determinada porção da atividade estabelecida naquele modelo. Neste caso, emprega-se uma letra distinta para cada porção (a partir da letra "a"). Exemplo : F4.1a e F4.1b;

3º) o identificador da atividade empregado no ME seguido do caracter "+", significando que trata-se da atividade estabelecida naquele modelo acrescida de requisitos não-essenciais. Exemplo : F4.1+;

4º) o identificador da atividade empregado no ME seguido de uma letra minúscula e do caracter "+", significando que trata-se de uma determinada porção da atividade estabelecida naquele modelo acrescida de requisitos não-essenciais. Exemplo : F4.1a+; e

5º) campo totalmente em branco, significando que, por ser composta exclusivamente de requisitos não essenciais, não possui correspondência com o ME.

B) Modelo da Configuração de Processos

De modo similar ao MCPPr, emprega basicamente a notação gráfica adotada no Modelo da Essência com as seguintes extensões :

- Processos

Os processos são representados por retângulos fechados por linha cheia com estrutura interna tripla a qual incorpora :

a) um campo identificador que contém a letra "P" seguida da estrutura que identifica o processo - $m[\{.n\}]$, onde m e n são naturais seqüenciais - e, opcionalmente, os seguintes caracteres :

- " $\{i\}$ ", indicando a existência de mais de uma instância de um mesmo tipo de processo; no caso do SICLEN, "i" é um número natural seqüencial que identifica uma linha de engarrafamento; e

- "*", indicando que trata-se de um processo primitivo (um único processo e não uma agregação de processos) que, por isto, não possui detalhamento gráfico.

b) um campo descritor que descreve sucintamente a ação do processo; e

c) um campo rastreador que permite o mapeamento do Esquema dos Processos para a Organização Hierárquica dos Processos; adicionalmente, quando for conveniente no sentido de eliminar possíveis ambigüidades, poderá ser usado na Organização Hierárquica dos Processos para indicar o processador ao qual o processo esteja alocado.

- Atividades alocadas a um processo

As atividades essenciais e não-essenciais alocadas a um dado processo são representadas por retângulos fechados - por linha cheia no caso de uma atividade operacional e por linha vazada no caso de atividade de controle - com estrutura interna tripla que incorpora:

a) um campo identificador que contém a letra "H" no caso de atividade operacional ou "HC" no caso de atividade de controle, seguida da estrutura que identifica a atividade - $m[\{.n\}]$, onde m e n são naturais seqüenciais - e, opcionalmente, os seguintes caracteres :

- " $\{i\}$ ", indicando a existência de mais de uma instância de um mesmo tipo de atividade; no caso do SICLEN, "i" é um número natural seqüencial que identifica uma linha de engarrafamento; e

- "*", indicando que trata-se de uma atividade primitiva (uma única atividade e não uma agregação de atividades) que, por isto, não possui detalhamento gráfico.

b) um campo descritor que descreve sucintamente a ação da atividade;

c) um campo rastreador que permite o mapeamento das atividades estabelecidas no MCPPr para as seções do MCP correspondentes à alocação da essência e de requisistos não-essenciais a cada processo.

De modo análogo ao MCPPr, o campo rastreador poderá apresentar :

- somente o identificador da atividade empregado no MCPPr, significando que trata-se exatamente da atividade estabelecida naquele modelo, ou seja, não sofreu modificações. Exemplo : G4.1;

- o identificador da atividade empregado no MCPr seguido de uma letra minúscula, significando que trata-se de uma determinada porção da atividade estabelecida naquele modelo. Neste caso, emprega-se uma letra distinta para cada porção (a partir da letra "a"). Exemplo : G4.1a e G4.1b;
- o identificador da atividade empregado no MCPr seguido do caracter "+", significando que trata-se da atividade estabelecida naquele modelo acrescida de requisitos não-essenciais. Exemplo : G4.1+;
- o identificador da atividade empregado no MCPr seguido de uma letra minúscula e do caracter "+", significando que trata-se de uma determinada porção da atividade estabelecida naquele modelo acrescida de requisitos não-essenciais. Exemplo : G4.1a+; e
- campo totalmente em branco, significando que, por ser composta exclusivamente de requisitos não essenciais, não possui correspondência com o MCPr.

C) Modelo da Configuração de Módulos

Os módulos são representados por retângulos fechados por linha cheia com estrutura interna tripla a qual incorpora :

- a) um campo identificador que contém a letra "M" seguida da estrutura que identifica o módulo - m[.n}], onde m e n são naturais seqüenciais;
- b) um campo descritor que descreve sucintamente a ação do módulo; e
- c) um campo rastreador que permite o mapeamento das atividades estabelecidas no MCP para a seção Diagramas Hierárquicos de Módulos Associados a cada Processo Alocado a cada Processador do MCM. Quando for conveniente no sentido de eliminar possíveis ambigüidades, poderá ser usado para indicar a qual processador/processo o módulo esteja alocado.

3.2 Identificação e comentários em diagramas

Os diagramas são identificados por um cabeçalho que conterà informações relativas aos aspectos de modelagem por ele abordado tais como :

- o nome da seção;
- indicação de detalhamento gráfico coerente com a apresentação "top-down" do conjunto de elementos de modelagem primitivos que compõem um elemento agregado;
- identificador e nome do processador/processo/módulo visando identificar o elemento gráfico que está sendo detalhado; e
- número da folha (natural seqüencial a partir de "1", empregado sempre que um diagrama se estender por mais de uma página).

Opcionalmente, os diagramas poderão conter comentários explicativos apresentados textualmente entre asteriscos.

3.3 Mapeamento de fluxos de dados/controlé entre diagramas

A notação de identificadores de processadores/atividades entre asteriscos posicionados sob um fluxo de dados/controlé tem por objetivo facilitar o rastreamento de fluxos e significa

que a origem (ou destino) do fluxo se encontra em outra página do mesmo diagrama ou em outro diagrama.

3.4 Inclusão de novos DET e de uma Seção Detalhada

Conforme exposto no subitem 3.1, a reorganização e expansão do ME pode causar modificações nas atividades primitivas. Nestes casos, deverão ser alteradas as especificações das atividades operacionais (Listas de Pré/Pós-Condições) e de controle (DET). Apesar da notação gráfica apresentada permitir comunicar tais ocorrências, essas indicações por si só não são suficientes para garantir não-ambigüidade à modelagem, e por esse fato foram adicionados ao MCP_r e MCP :

- novos DET para especificar as atividades de controle modificadas, incluídos nas seções de Organização Hierárquica dos Esquemas de Atividades; e

- uma Seção Detalhada contendo as Listas de Pré/Pós-Condições que especificam as atividades operacionais modificadas.

4. Alternativas de MCP_r do SICLEN

4.1 Primeira alternativa

A primeira alternativa de arquitetura do SICLEN possui, além dos acionadores, sensores e conversor analógico-digital, dois processadores digitais programáveis que realizarão as tarefas de monitoração e controle da área de engarrafamento. Um deles é responsável pelo controle da interface da aplicação com os usuários (supervisor da área e operadores de linha), e o outro controla os dispositivos do processo de engarrafamento propriamente dito.

Cabe mencionar que, de modo a facilitar o entendimento deste trabalho, empregamos o tipo de letra "Courier New" para diferenciar as seções relativas à modelagem propriamente dita do SICLEN do texto explicativo.

1. Definição de Aspectos de Implementação do Sistema

1.1 Processadores

O Sistema de Controle de Linhas de Engarrafamento - SICLEN será constituído por um conjunto de equipamentos que incluem processadores digitais programáveis e processadores não-programáveis, todos alimentados por uma fonte elétrica de 110 volts. O sistema controlará 3 linhas de engarrafamento (FIGURA 1) e seus processadores deverão possuir as seguintes características:

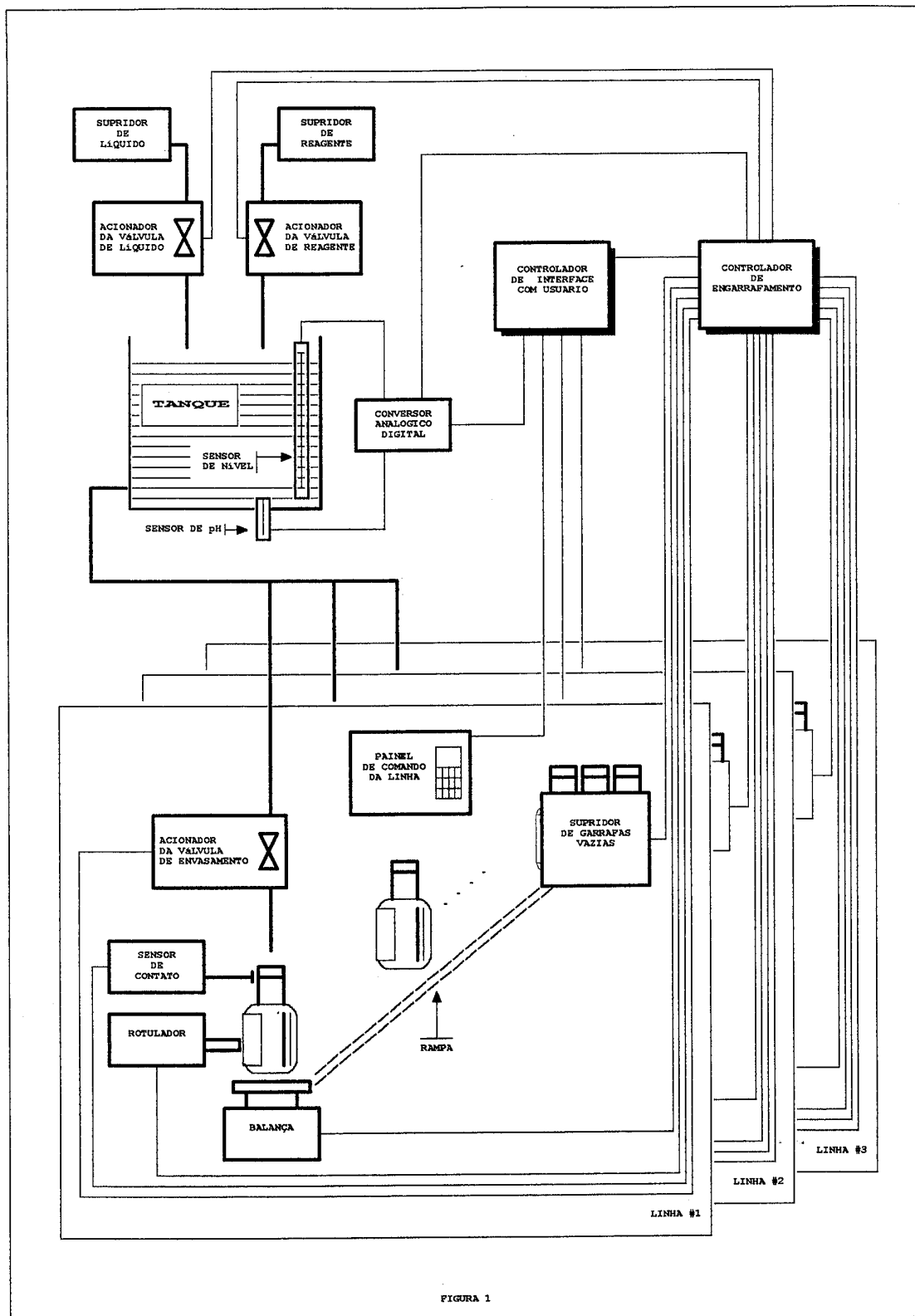


FIGURA 1

1) PR1.1 (PR'1) - controlador de interface com usuário

Microcomputador que executará as funções de :

- painel de comando da área (através do qual o supervisor controlará o funcionamento do processo de engarrafamento); e

- controlador de interface com supervisor/operadores de linha.

Tendo em vista considerações de desempenho e custo, foi selecionada a seguinte configuração :

- CPU Intel 386 DX de 40 MHz;
- 8 Mb de memória RAM;
- memória cache de 256 Kb;
- barramento ISA;
- disco rígido padrão IDE de 320 Mb com tempo de acesso de 11 ms e taxa de transferência de 1,2 Kb/s;
- drive de 3 1/2";
- teclado alfanumérico de 101 teclas;
- monitor de vídeo SVGA monocromático de 14";
- interface de vídeo SVGA com 512 Kb;
- cartão adaptador de rede Ethernet com porta 10baseT;
- sistema operacional de suporte MS-DOS versão 6.2;
- software gerenciador de rede local Novell Netware Lite.

2) PR1.2 (PR'2) - controlador de engarrafamento

Microcomputador que executará as funções de monitoração e controle do tanque e dos equipamentos de cada linha de engarrafamento.

Tendo em vista considerações de desempenho e custo, foi selecionada a seguinte configuração :

- CPU Intel 486 DX2 de 66 MHz;
- 8 Mb de memória RAM;
- memória cache de 256 Kb;
- barramento ISA/EISA;
- disco rígido padrão IDE de 270 Mb com tempo de acesso de 11 ms e taxa de transferência de 1,2 Kb/s;
- drive de 3 1/2";
- teclado alfanumérico de 101 teclas;
- monitor de vídeo VGA monocromático de 14";
- interface de vídeo VGA com 512 Kb;
- cartão adaptador de rede Ethernet com porta 10baseT;
- sistema operacional MS-DOS versão 6.2.

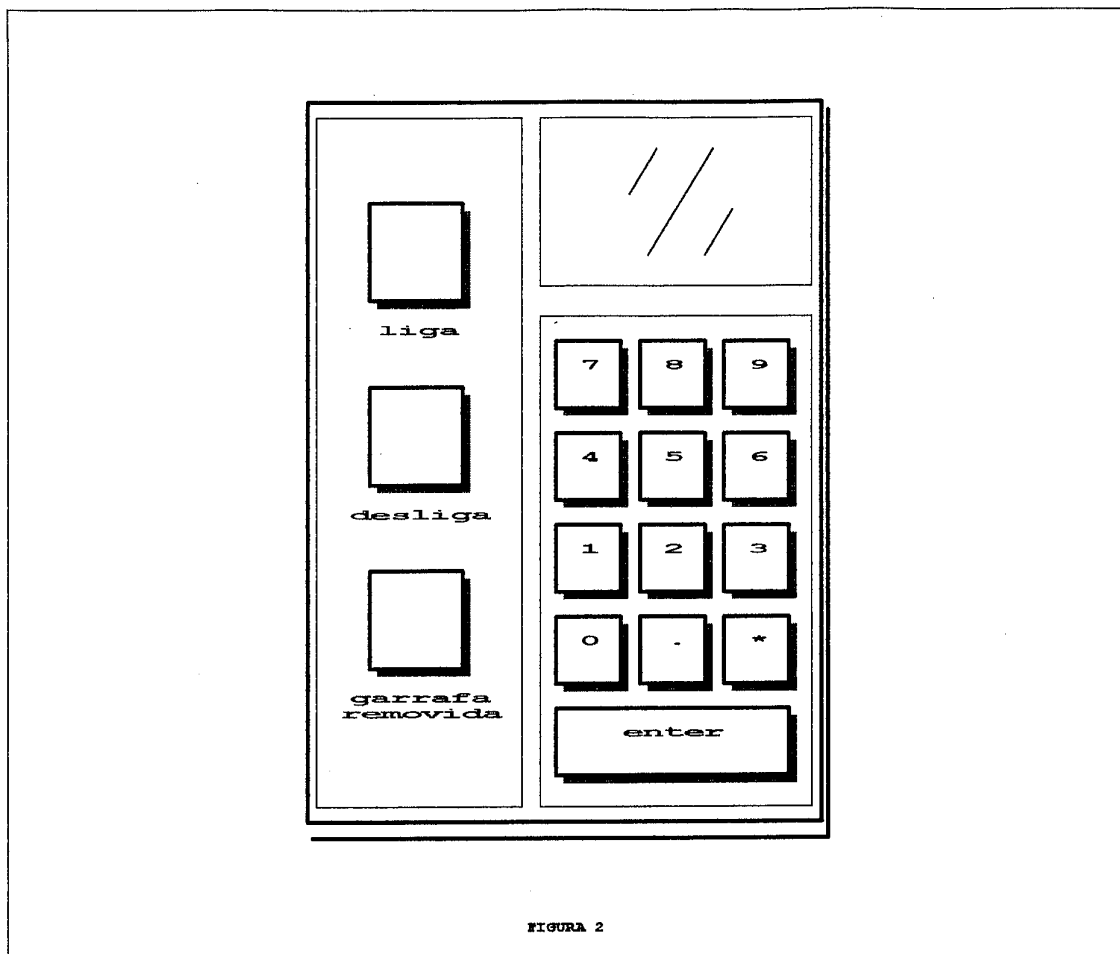
3) PR2{i} (PR'3{i}) - painel de comando de linha

Cada linha de engarrafamento possuirá um dispositivo desta natureza, o qual permitirá ao operador controlar o funcionamento da sua linha. Será dotado de (FIGURA 2):

- botões de ativação/desativação da linha e sinalização de remoção de garrafa da posição de envasamento;
- teclado numérico de 12 teclas;
- display de cristal líquido capaz de receber dados do PR1.1 e exibi-los no formato gerado por esse processador e entrada de efetuada pelo operador.

4) PR3.1 (PR'4) - sensor de pH

Dispositivo a ser instalado no tanque que deverá ser capaz de fornecer continuamente o valor corrente do pH do líquido ao conversor analógico-digital (PR3.3).



5) PR3.2 (PR'5) - sensor de nível

Dispositivo a ser instalado no tanque que deverá ser capaz de fornecer continuamente o valor corrente do nível do líquido ao conversor analógico-digital (PR3.3).

6) PR3.3 (PR'6) - conversor analógico/digital

Dispositivo capaz de converter informações da forma analógica para a forma digital, possibilitando à aplicação obter dados interpretáveis do pH e nível correntes do líquido no tanque.

7) PR4.1 (PR'7) - acionador da válvula de líquido

Dispositivo eletro-mecânico capaz de regular a abertura da válvula de admissão de líquido do tanque recebendo do controlador de engarrafamento (PR1.2) o valor digital dessa abertura.

8) PR4.2 (PR'8) - acionador da válvula de reagente

Dispositivo eletro-mecânico capaz de regular a abertura da válvula de admissão de reagente do tanque recebendo do controlador de engarrafamento (PR1.2) o valor digital dessa abertura.

9) PR4.3 (PR'9) - acionador da válvula de envasamento

Dispositivo eletro-mecânico a ser alocado a cada linha de engarrafamento capaz de regular a abertura da válvula de envasamento da linha, recebendo do controlador de engarrafamento (PR1.2) o valor digital dessa abertura.

1.2 Interfaces

1.2.1 Inter-processadores

A troca de dados e comandos entre os processadores programáveis do SICLEN será gerenciada pelo software gerenciador de rede local residente no PR1.1.

1.2.2 Interface Usuário-máquina

a) Supervisor

Como mencionado anteriormente, o PR1.1 incorporará o painel de controle da área. Através do teclado o supervisor poderá habilitar/inabilitar a área de engarrafamento (modelado pelos sinais "habilitar_área" e "inabilitar_área") e ajustar o ponto de pH do líquido a ser engarrafado (modelado pelo fluxo de dados discreto "ajuste_pH"). A aplicação exibirá no monitor de vídeo uma tela (FORMATO DE TELA 1) contendo o pH e nível correntes do líquido (modelado pelo fluxo de dados contínuo "condições_tanque"), o estado de cada linha (modelado pelo fluxo de dados contínuo "estado_linhas"), caracteres resultantes da digitação do ajuste de pH e, quando for o caso, mensagens acusando a rejeição de valores inadequados de ajuste de pH (modelado pelo fluxo de dados discreto "rejeição_ajuste_pH"). Através do alto-falante, PR1.1 será capaz de emitir alarmes sonoros alertando o supervisor da ocorrência de anomalia do pH ou do nível do líquido (modelado pelos sinais "anomalia_pH" e "anomalia_nível_tanque").

b) Operador de linha

Pelo acionamento dos botões e do teclado numérico do painel de controle da linha - PR2{i} -, o operador poderá ligar/desligar a linha de engarrafamento sob sua responsabilidade e sinalizar ao sistema que já removeu a garrafa da posição de envasamento (modelado pelos sinais "ligar_linha(i)", "desligar_linha(i)" e "garrafa_remover(i)") bem como alterar o tamanho da garrafa utilizada (modelado pelo fluxo de dados discreto "tamanho_garrafa(i)"). O controlador de interface com usuário será responsável pela geração da tela a ser exibida no display de cristal líquido dos painéis de comando de linha (FORMATO DE TELA 2). Serão exibidos nessa tela o estado (ligada/ desligada) e volume da garrafa empregada na linha (modelado pelo fluxo de dados discreto "estado_da_linha(i)"), caracteres resultantes da digitação de novos tamanhos de garrafa e mensagens acusando possíveis rejeições de valores inadequados de tamanho de garrafa (modelado pelo fluxo de dados discreto "rejeição_tamanho_garrafa(i)").

SICLEN			
Condições do tanque			
pH-corrente = 6.35		nível (%) = 0.85	
Condições das linhas			
linha	1	2	3
estado	ligada	ligada	desligada
volume da garrafa	2.0 L	3.5 L	*****
Linha de comandos			
-> valor de ajuste de pH inválido !			

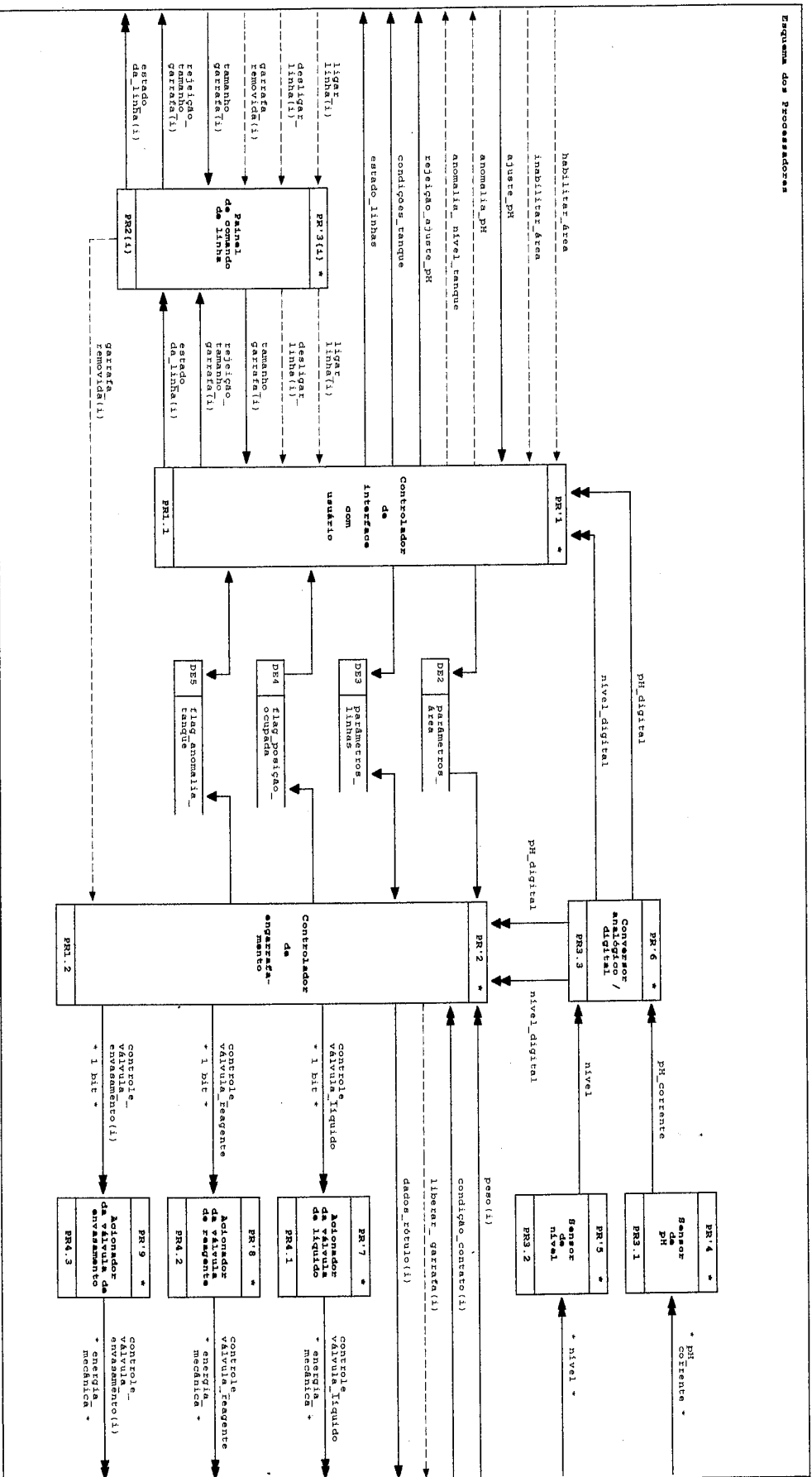
FORMATO DE TELA 1

SICLEN - Linha # 1	
Condição da linha	
estado =	ligada
volume-garrafa =	2.0 L
Linha de comandos	
-> Volume da garrafa inválido !	

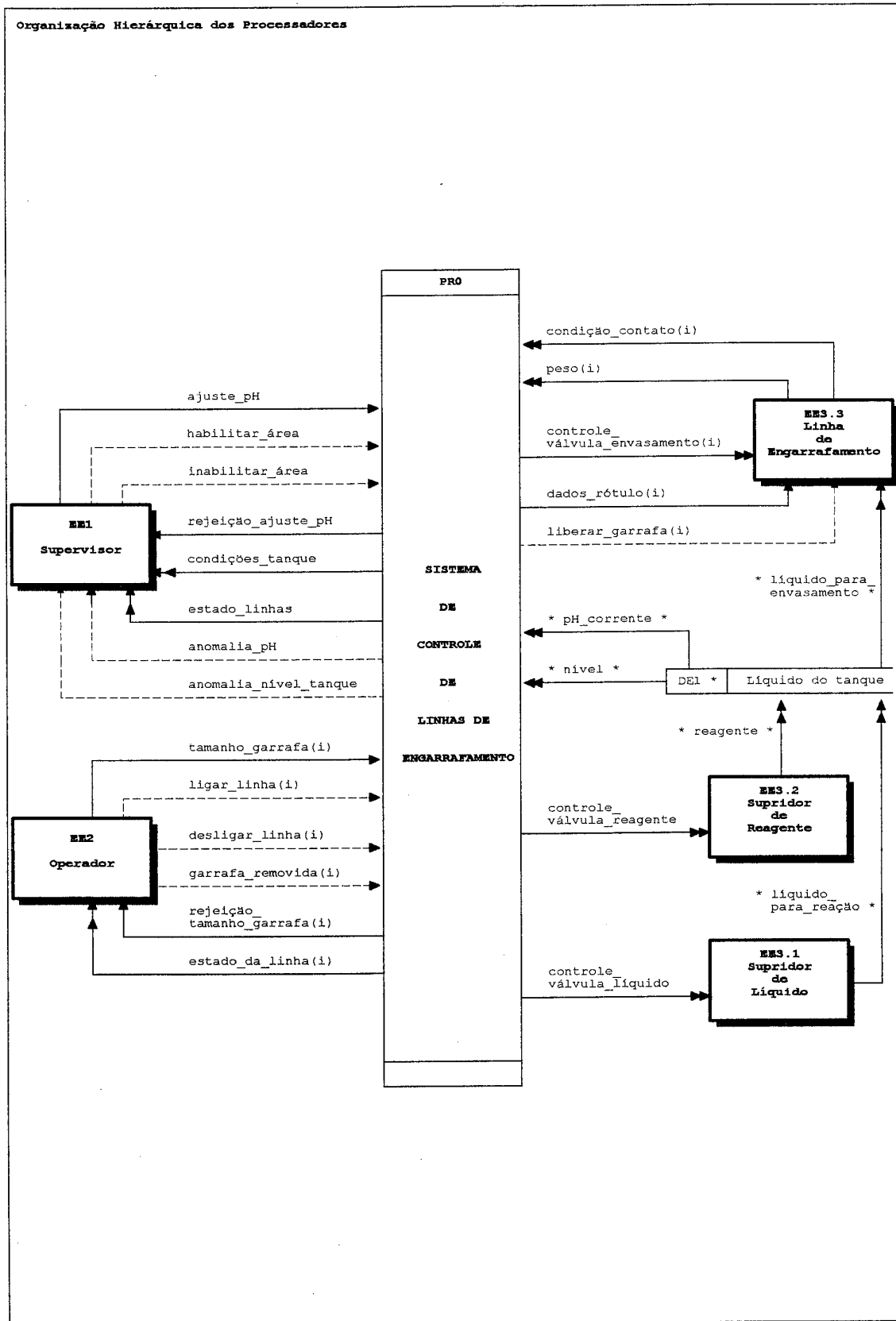
FORMATO DE TELA 2

2. Seção Esquemática

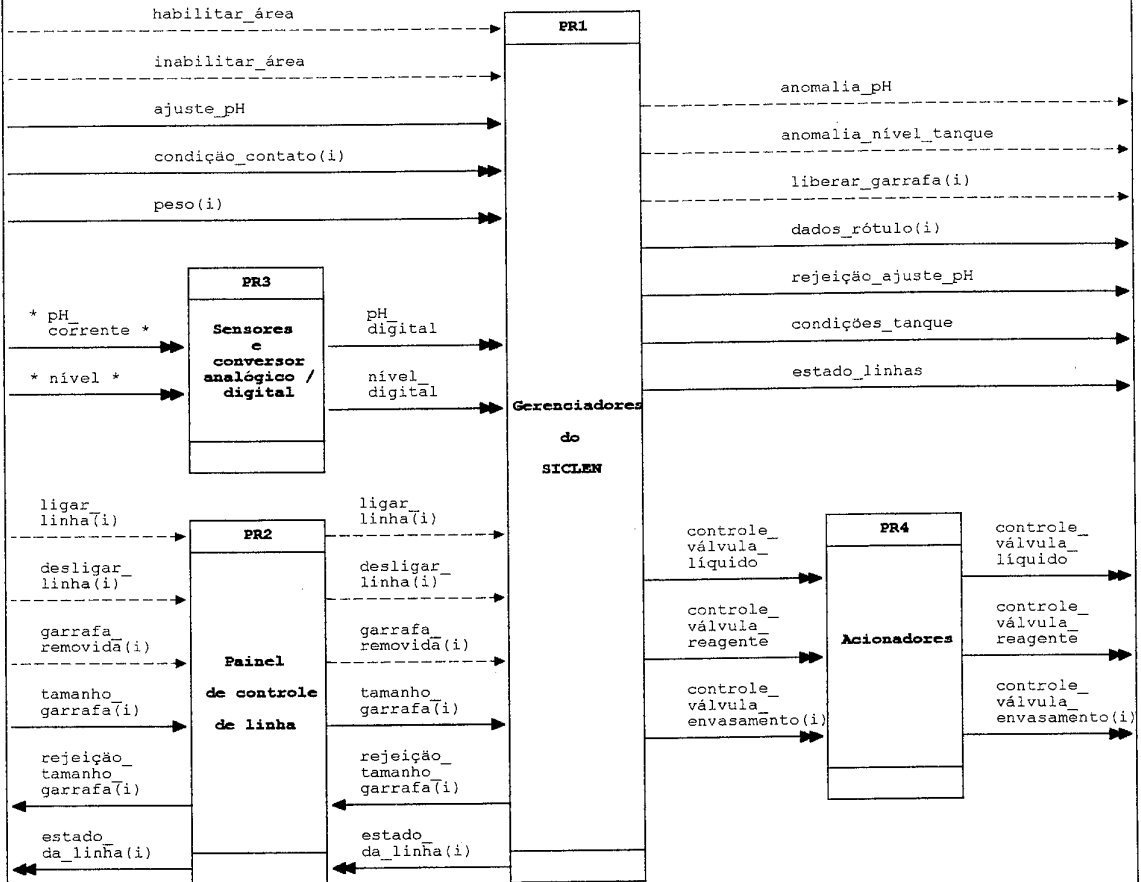
2.1 Esquema dos Processadores



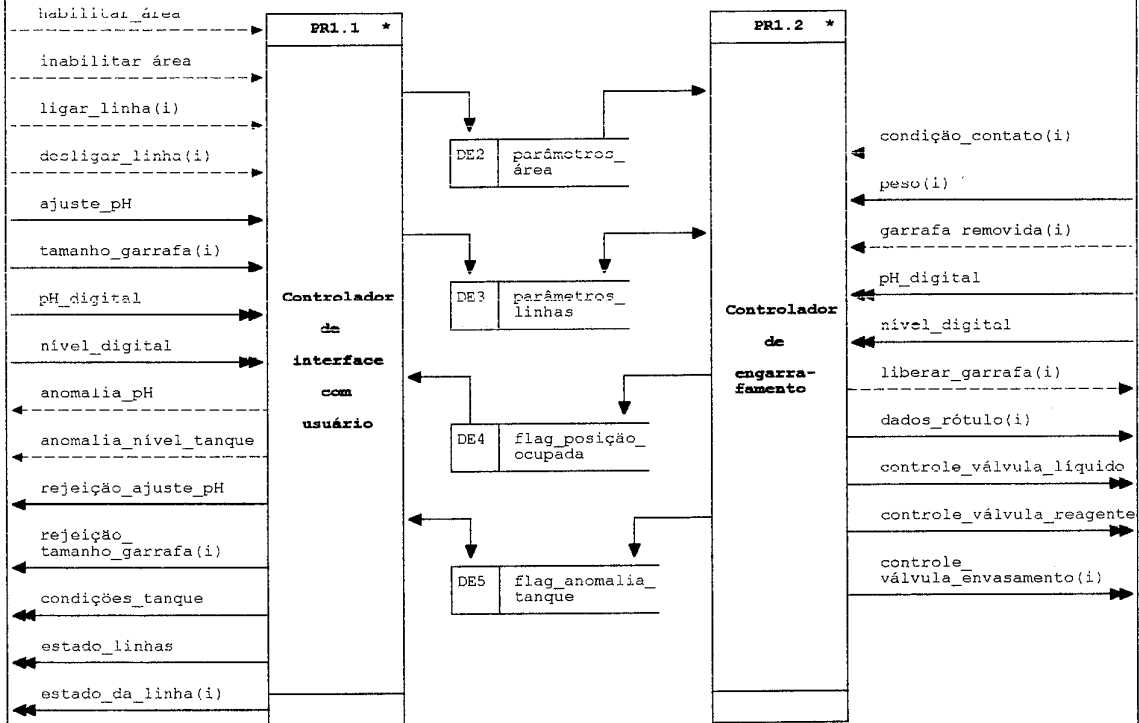
2.2 Organização Hierárquica dos Processadores



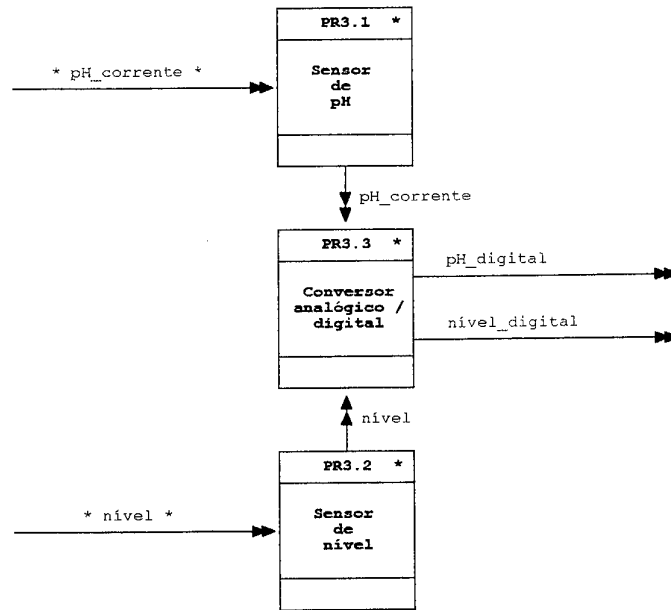
Organização Hierárquica dos Processadores - detalhamento de PRO



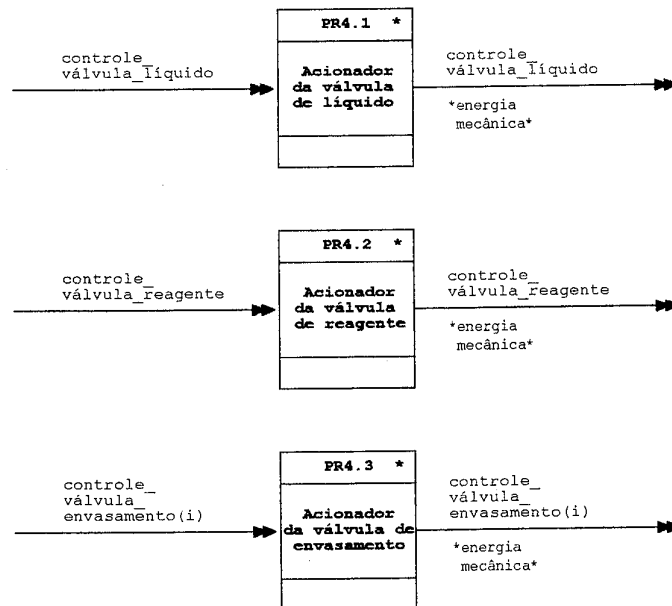
Organização Hierárquica dos Processadores - detalhamento de PR1 - Gerenciadores do SICLEN



Organização Hierárquica dos Processadores - detalhamento de PR3 - Sensores e conversor analógico/digital

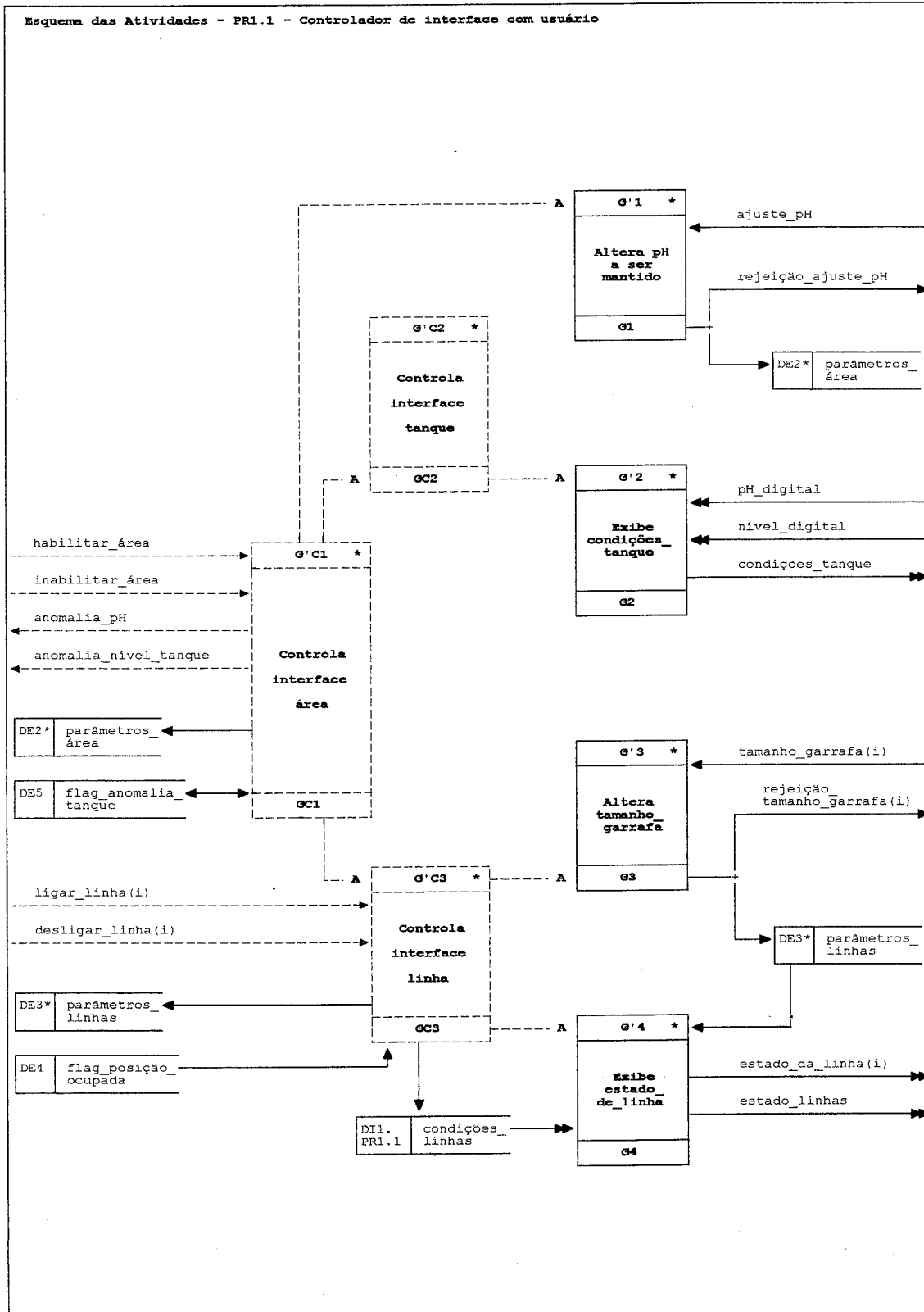


Organização Hierárquica dos Processadores - detalhamento de PR4 - Acionadores

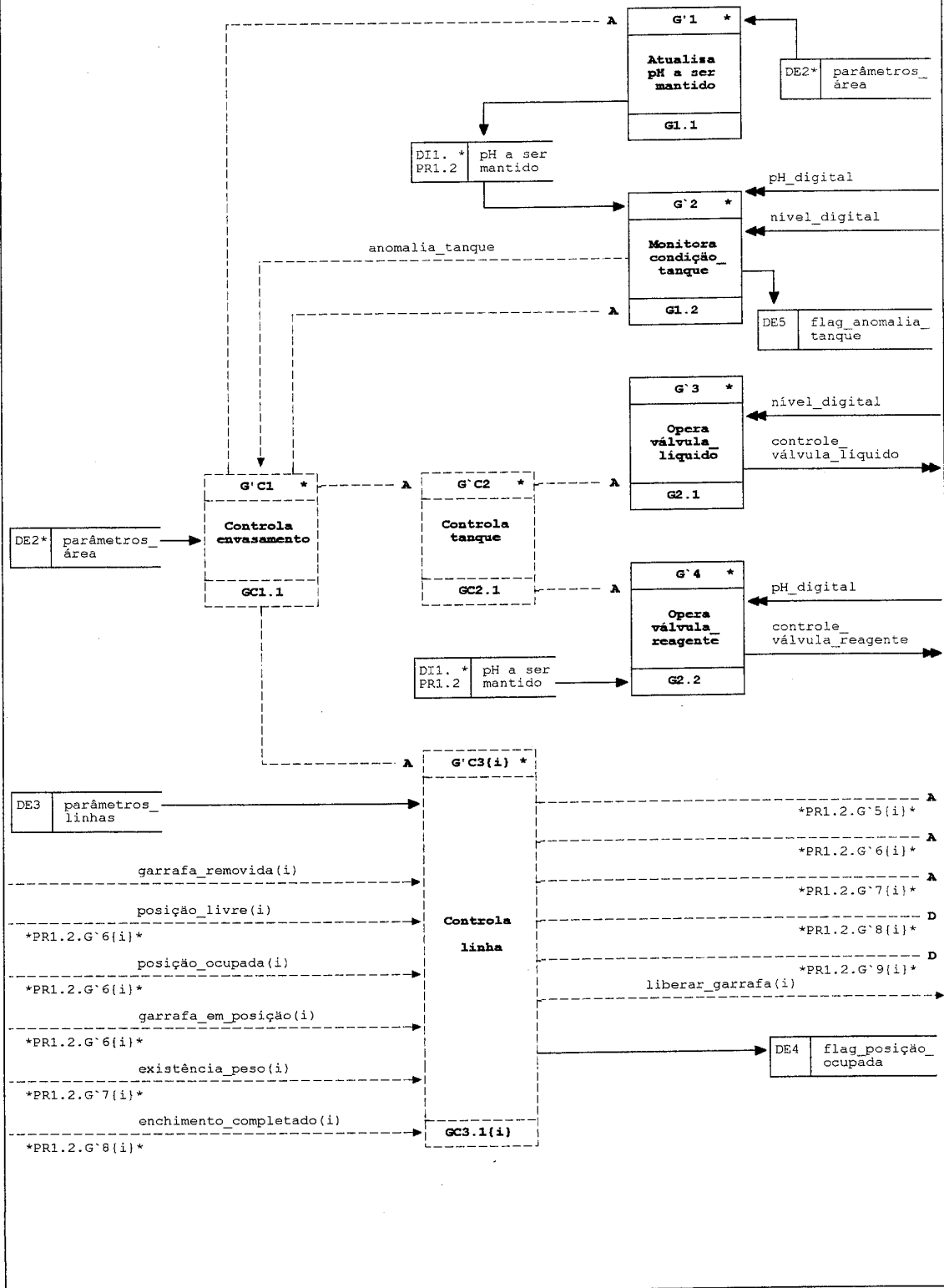


2.3 Alocação da Essência e de Requisitos Não-Essenciais a cada Processador

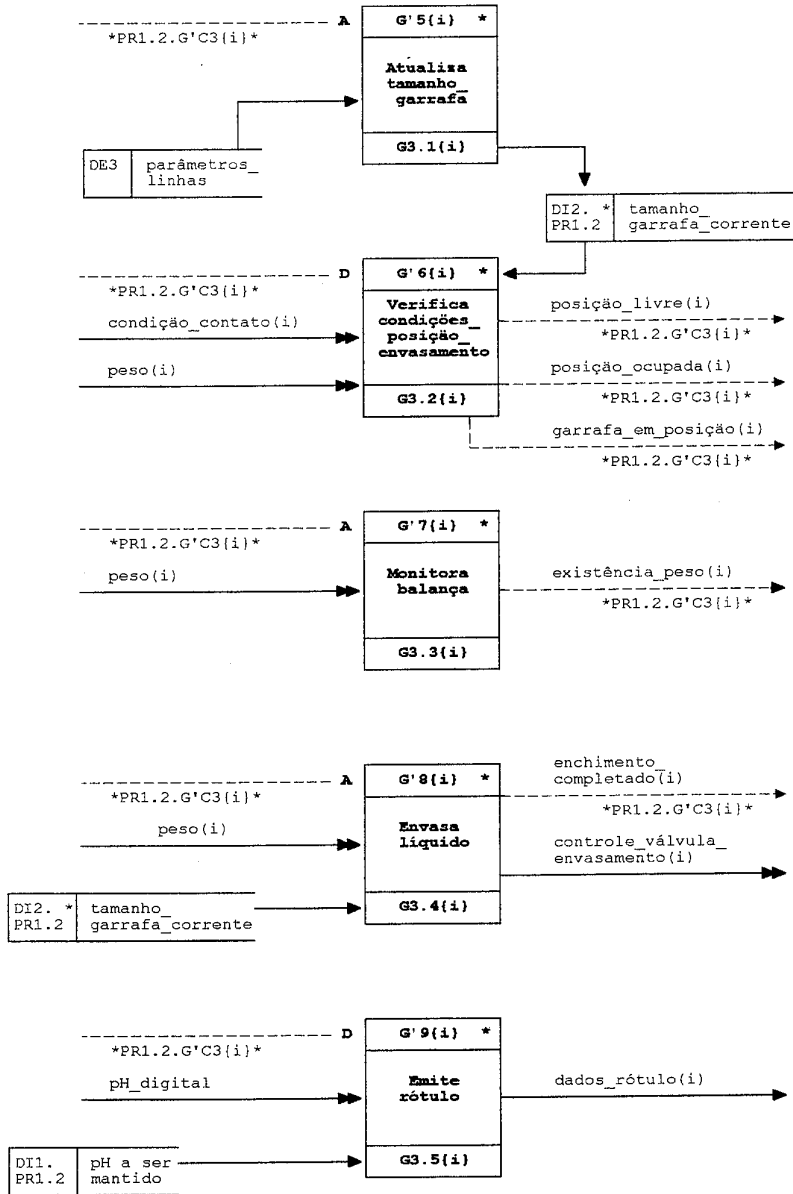
2.3.1 Esquema das Atividades



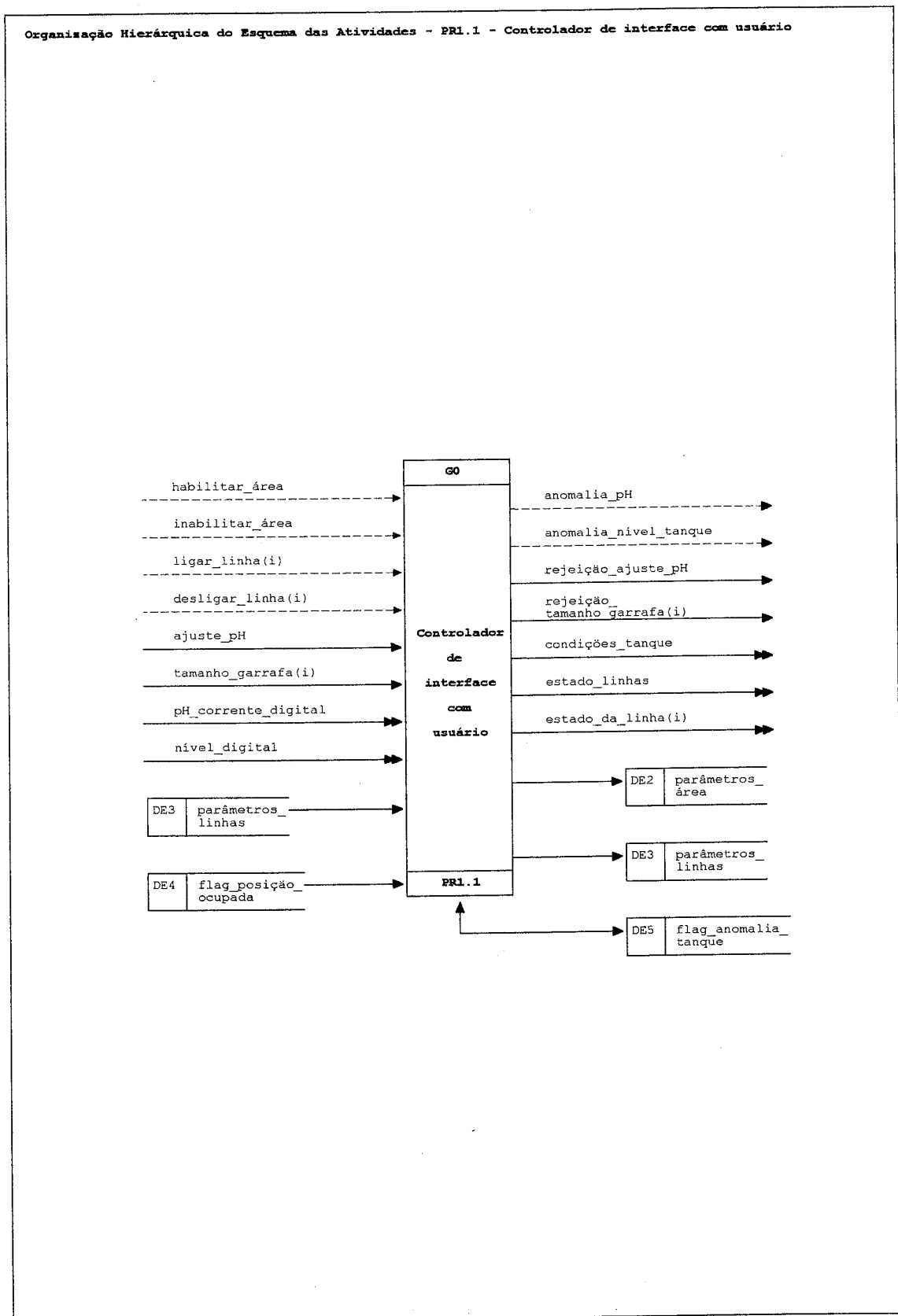
Esquema das Atividades - PR1.2 - Controlador de engarrafamento (folha 1)



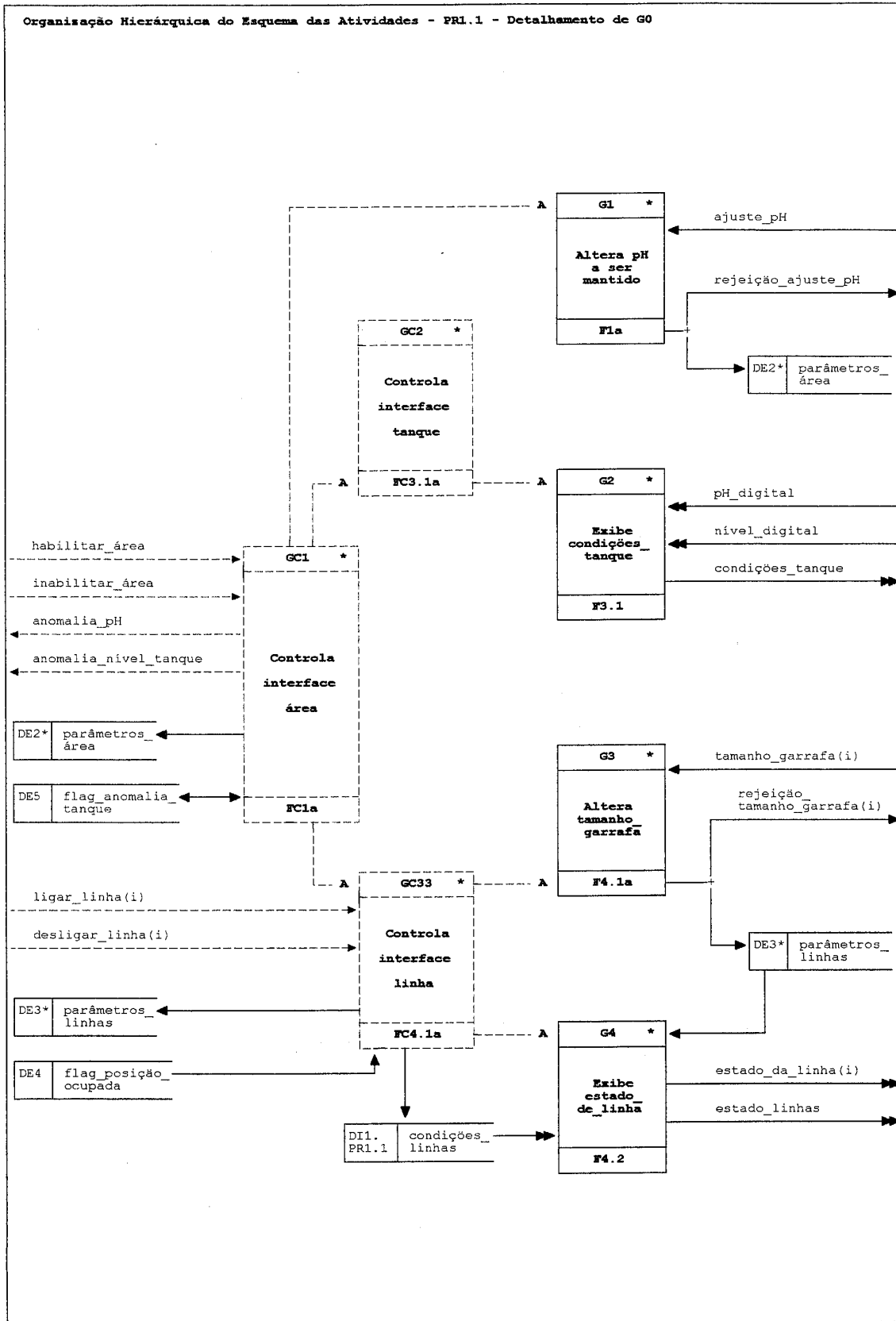
Esquema das Atividades (continuação) - PR1.2 - Controlador de engarrafamento (folha 2)



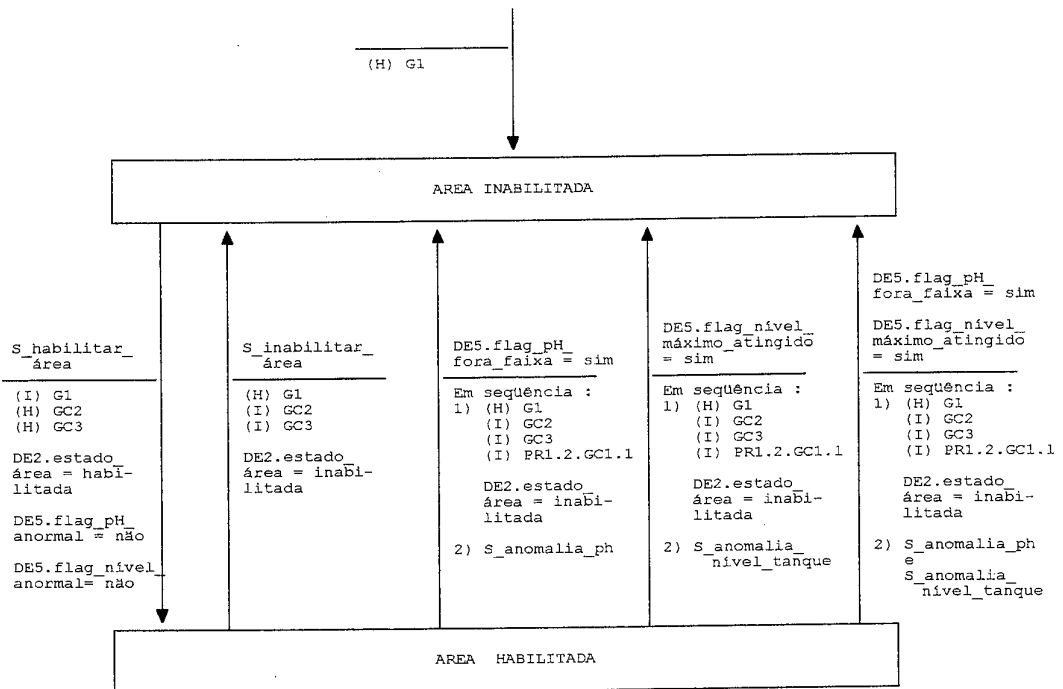
2.3.2 Organização Hierárquica dos Esquemas das Atividades



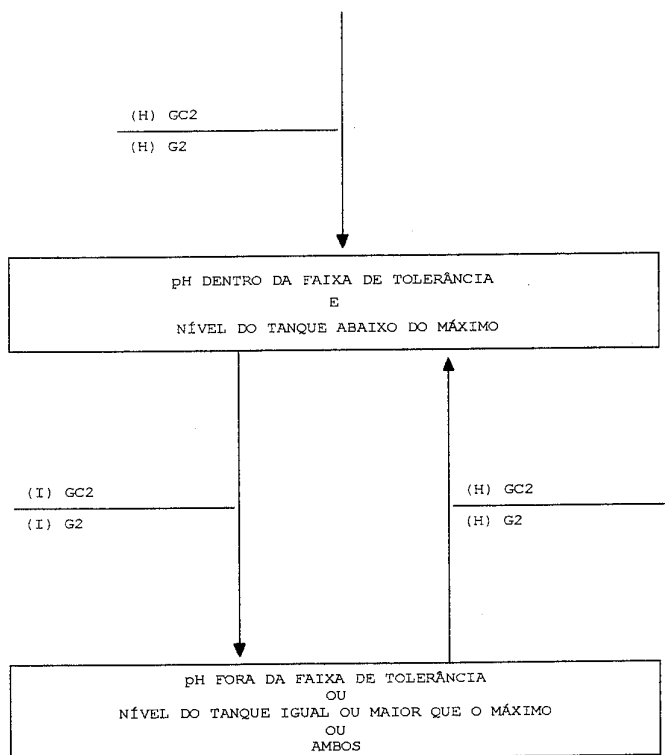
Organização Hierárquica do Esquema das Atividades - PR1.1 - Detalhamento de G0



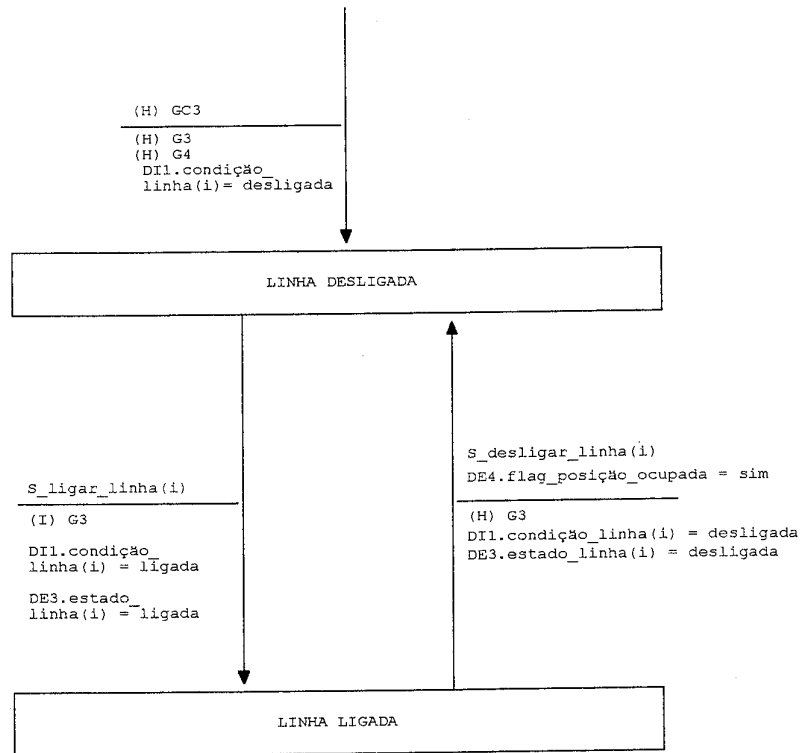
Especificação de PR1.1.GC1 - Controla interface área



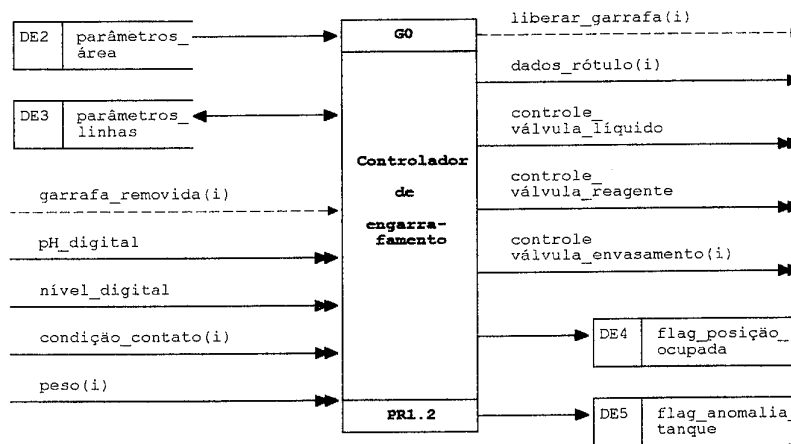
Especificação de PR1.1.GC2 - Controla interface tanque



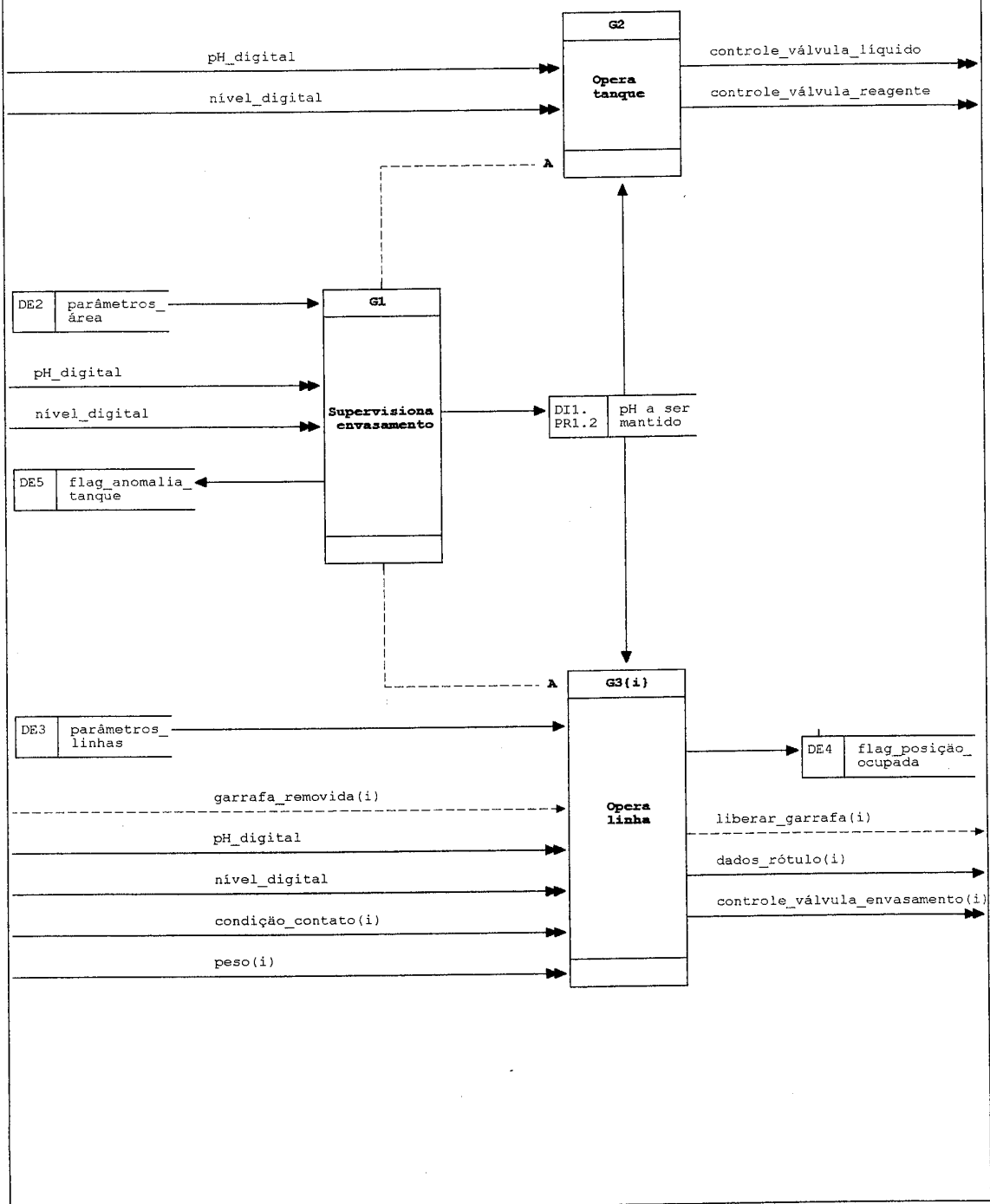
Especificação de PR1.1.GC3 - Controla interface linha



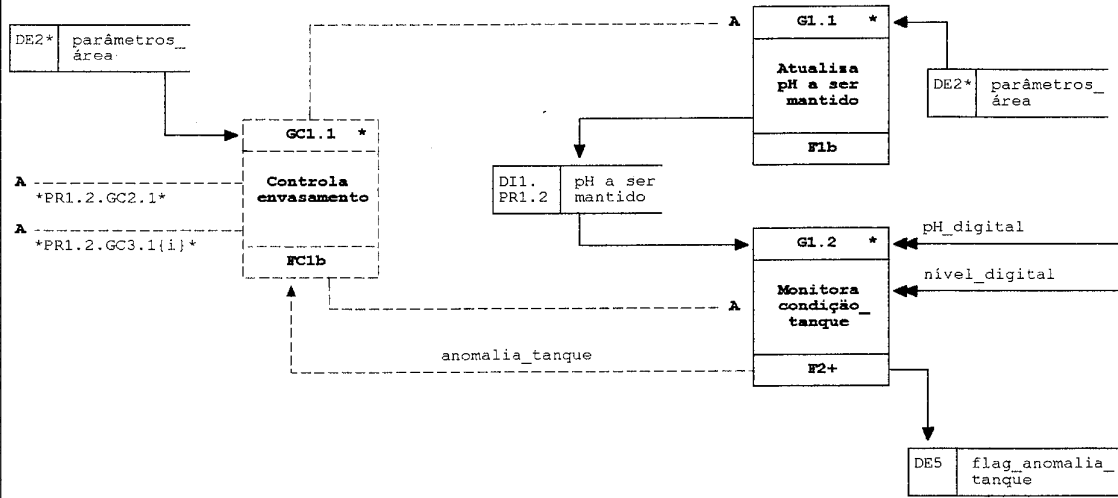
Organização Hierárquica do Esquema das Atividades - PR1.2 - Controlador de engarrafamento



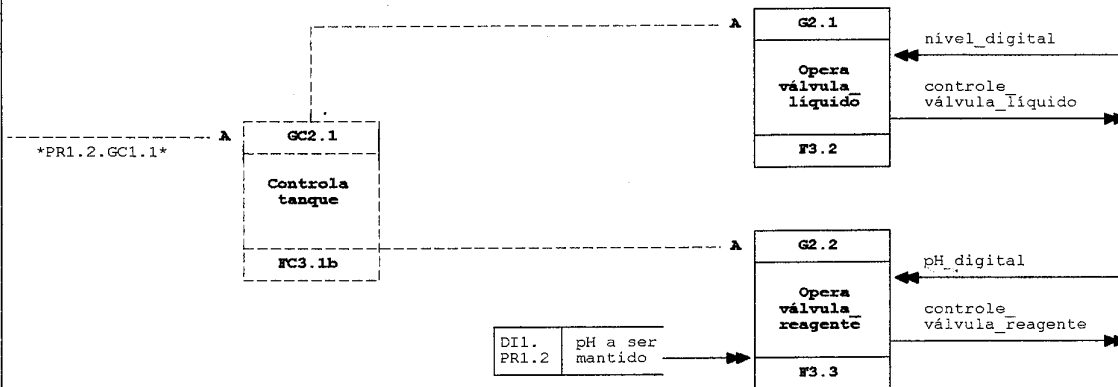
Organização Hierárquica do Esquema das Atividades - PR1.2 - Controlador de engarrafamento - detalhamento de G0



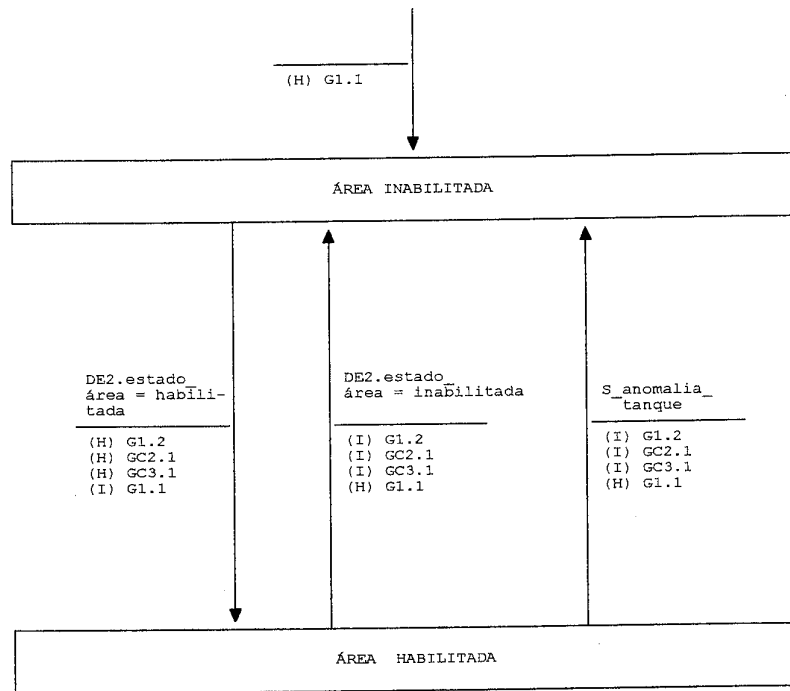
Organização Hierárquica do Esquema das Atividades - PR1.2 - Controlador de engarrafamento - detalhamento de G1



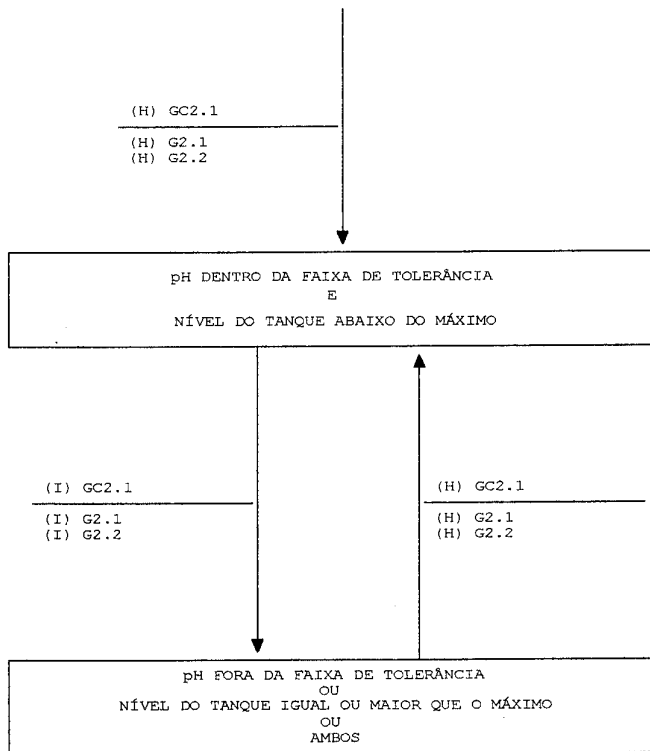
Organização Hierárquica do Esquema das Atividades - PR1.2 - Controlador de engarrafamento - detalhamento de G2



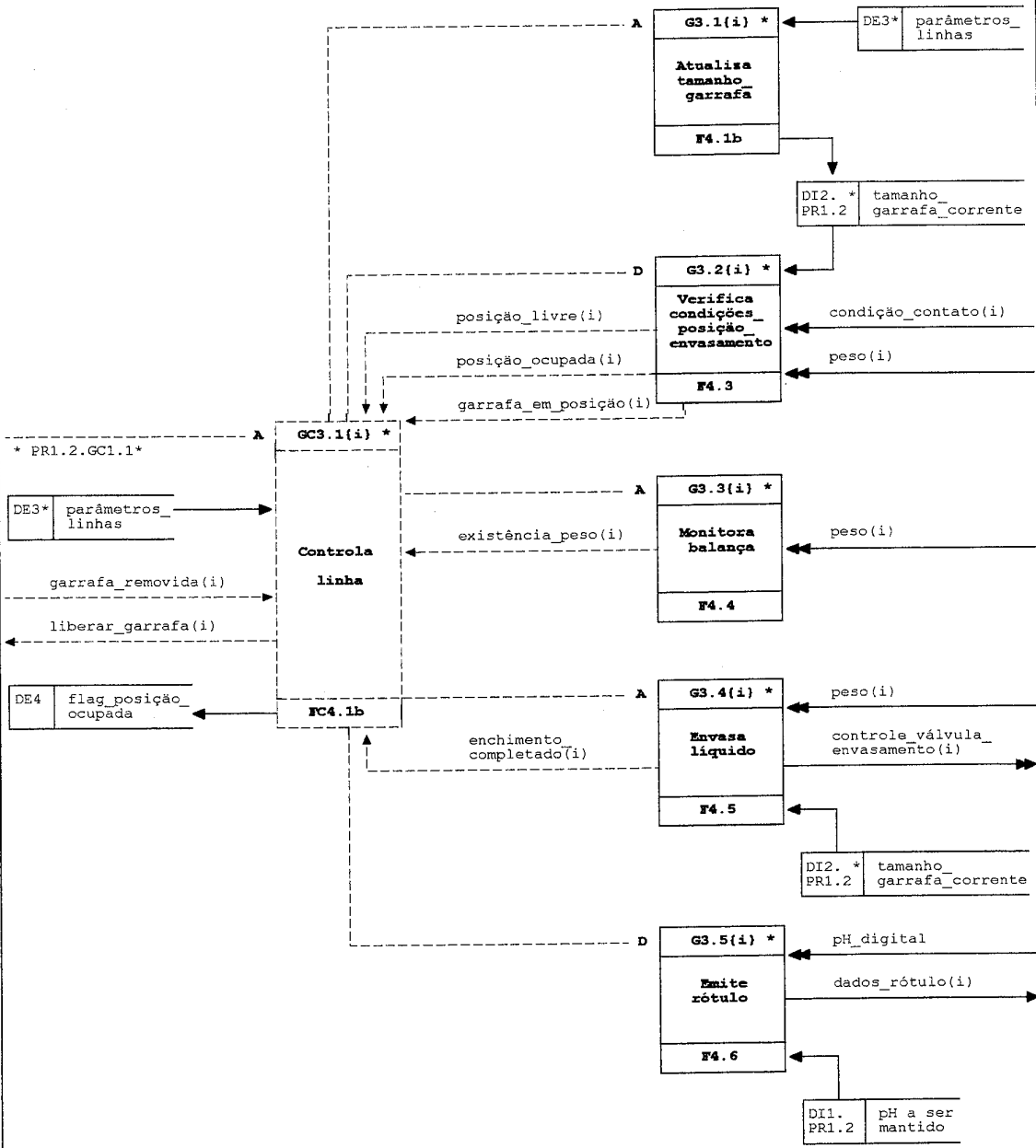
Especificação de PR1.2.GC1.1 - Controla área



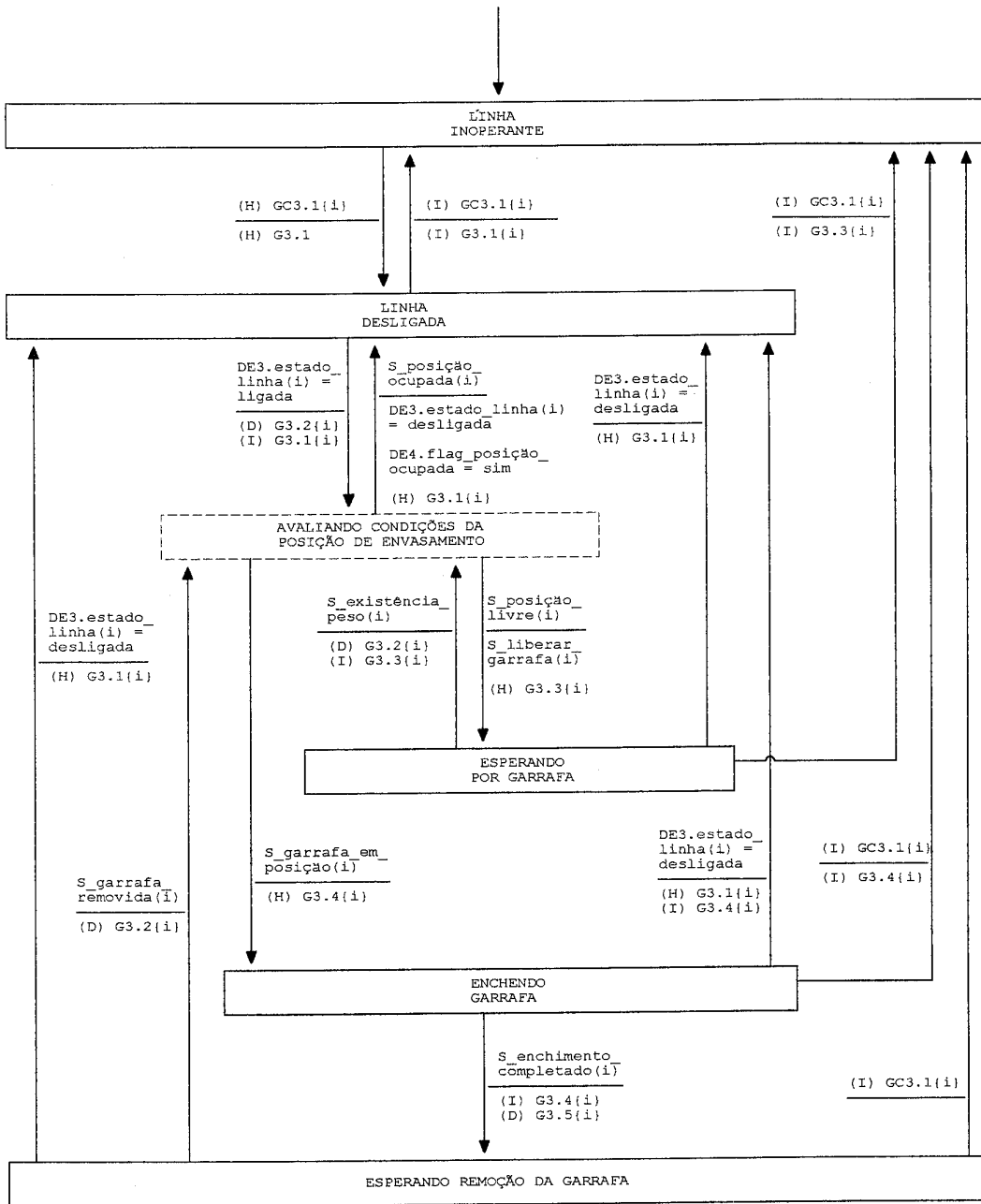
Especificação de PR1.2.GC2.1 - Controla tanque



Organização Hierárquica do Esquema das Atividades - PR1.2 - Controlador de engarrafamento - detalhamento de G3



Especificação de FRI.2.GC3.1(i) - Controla linha



3. Seção Detalhada

3.1 Seção de Definições Globais

1) De um modo geral, a apresentação de uma pré/pós-condição procura seguir, aproximadamente, a sintaxe de uma linguagem de primeira ordem.

2) O termo extra-lógico **SUBSTITUÍDO PELO** significa a modificação do valor de um elemento de dado de um depósito interno, o qual passa a possuir o valor do atributo do elemento de modelagem que segue o termo.

3.2 Descrição Detalhada de Atividades Primitivas -

Lista de Pré/Pós-Condições

PRI.1.G1 - Altera pH a ser mantido

Seção de Definições Locais

Cláusulas :

C1_novo_pH_válido

"FDD_ajuste_pH" TALQUE <0

<1 valor de "FDD_ajuste_pH.ED_novo_pH" > = valor correspondente a pH_mínimo de "ED_pH_nominal" 1>

E

<1 valor de "FDD_ajuste_pH.ED_novo_pH" < = valor correspondente a pH_máximo de "ED_pH_nominal" 1> 0>

C2_pH_atualizado

valor de "DE2.parâmetros_área.pH_a_ser_mantido" SUBSTITUÍDO PELO valor de "FDD_ajuste_pH.ED_novo_pH"

Lista de pré/pós-condições

=> pré-condição 1 : -

HABILITADA

E

C1_novo_pH_válido

=> pós-condição 1 : -

C2_pH_atualizado

E

NÃO "FDD_rejeição_ajuste_pH"

=> pré-condição 2 : -

HABILITADA

E

NÃO C1_novo_pH_válido

=> pós-condição 2 : -

"FDD_rejeição_ajuste_pH"

E

NÃO C2_pH_atualizado

=> pré-condição 3 : -

INABILITADA

=> pós-condição 3 : -

NÃO <0 C2_pH_atualizado

E

FDD_rejeição_ajuste_pH" 0>

PR1.1.G3 - Altera tamanho garrafa

Seção de Definições Locais

Cláusulas :

C1_novo_tamanho_válido

"FDD_tamanho_garrafa(i)" TALQUE <0.

<1 valor de "FDD_tamanho_garrafa(i).ED_volume(i)" >= valor correspondente a volume_mínimo de "ED_volume(i)" 1>

E

<1 valor de "FDD_tamanho_garrafa(i).ED_volume(i)" <= valor correspondente a volume_máximo de "ED_volume(i)" 1>

E

<1 valor de "FDD_tamanho_garrafa(i).ED_peso_garrafa_vazia(i)" >= valor correspondente a peso_mínimo de "ED_volume(i)" 1>

E

<1 valor de "FDD_tamanho_garrafa(i).ED_peso_garrafa_vazia(i)" <= valor correspondente a peso_mínimo de "ED_volume(i)" 1> 0>

C2_tamanho_garrafa_corrente_atualizado

<1 valor de "DE3.parâmetros_linhas.ED_volume(i)" SUBSTITUÍDO PELO valor de "FDD_tamanho_garrafa(i).ED_volume(i)" 1>

E

<1 valor de "DE3.parâmetros_linhas.ED_peso_garrafa_vazia(i)" SUBSTITUÍDO PELO valor de "FDD_tamanho_garrafa(i).ED_peso_garrafa_vazia(i)" 1>

Lista de pré/pós-condições

=> pré-condição 1 : -

HABILITADA

E

C1_novo_tamanho_válido

=> pós-condição 1 : -

C2_tamanho_garrafa_corrente_atualizado

E

NÃO "FDC_rejeição_tamanho_garrafa(i)"

=> pré-condição 2 : -

HABILITADA

E

NÃO C1_novo_tamanho_válido

=> pós-condição 2 : -

"FDD_rejeição_tamanho_garrafa(i)"

E

NÃO C2_tamanho_garrafa_atualizado 0>

=> pré-condição 3 : -

INABILITADA

=> pós-condição 3 : -

NÃO <0 "FDD_rejeição_tamanho_garrafa(i)"

E

C2_tamanho_garrafa_atualizado 0>

PR1.2.G1.1 - Atualiza pH a ser mantido

Seção de Definições Locais

Cláusula :

C1_pH_a_ser_mantido_atualizado

valor de "DI1.PR1.2_pH a ser mantido.ED_pH nominal" SUBSTITUÍDO PELO
valor de "DE2.parâmetros_área.pH_a_ser_mantido"

Lista de pré/pós-condições

=> pré-condição 1 : -

HABILITADA

=> pós_ condição 1 : -

C1_pH_a_ser_mantido_atualizado

=> pré-condição 2 : -

INABILITADA

=> pós-condição 2 : -

NÃO C1_pH_a_ser_mantido_atualizado

PR1.2.G1.2 - Monitora condições tanque

Seção de Definições Locais

1) "erro_pH" designa o valor máximo aceitável de afastamento do pH do líquido do tanque em relação ao ponto de ajuste.

2) "nível_máximo" designa o valor do nível do líquido no tanque que sinaliza início de transbordamento.

3) Cláusulas :

C1_pH_anormal

pH_corrente TALQUE <0

<1 *pH_corrente* > (valor de "DI1_pH a ser mantido_ED_pH_nominal" + erro_pH) 1>

OU

<1 *pH_corrente* < (valor de "DI1_pH a ser mantido_ED_pH_nominal" - erro_pH) 1> 0>

C2_nível_anormal

nível >= nível_máximo

C3_ocorrência_anomalia_pH

valor de "DE5.flag_anomalia_tanque.flag_pH_anormal" = "sim"

C3_ocorrência_anomalia_nível

valor de "DE5.flag_anomalia_tanque.flag_nível_anormal" = "sim"

Lista de pré/pós-condições

=> pré-condição 1 : -

HABILITADA

E

C1_pH_anormal

E

NÃO C2_nível_anormal

=> pós-condição 1 : -

C3_ocorrência_anomalia_pH

E

NÃO C4_ocorrência_anomalia_nível


```
=> pré-condição 2 : -
    HABILITADA
    E
    NÃO C1_pH_anormal
    E
    C2_nível_anormal
=> pós-condição 2 : -
    C4_ocorrência_anomalia_nível
    E
    NÃO C3_ocorrência_anomalia_pH

=> pré-condição 3 : -
    HABILITADA
    E
    C1_pH_anormal
    E
    C2_nível_anormal
=> pós-condição 3 : -
    C3_ocorrência_anomalia_pH
    E
    C4_ocorrência_anomalia_nível

=> pré-condição 4 : -
    INABILITADA
=> pós-condição 4 : -
    NÃO <0 C3_ocorrência_anomalia_pH
        E
        C4_ocorrência_anomalia_nível 0>
```

PR1.2.G3.1 - Atualiza tamanho garrafa

Seção de Definições Locais

Cláusula :

C1_tamanho_garrafa_corrente_atualizado

<1 valor de "DI2.PR1.2_tamanho_garrafa_corrente.ED_volume(i)" SUBSTITUÍDO PELO valor de "DE3.parâmetros_linhas.ED_volume(i)" 1>

E

<1 valor de "DI2.PR1.2_tamanho_garrafa_corrente.ED_peso_garrafa_vazia(i) SUBSTITUÍDO PELO valor de "FDD_tamanho_garrafa(i).ED_peso_garrafa_vazia(i)" 1>

Lista de pré/pós-condições

=> pré-condição 1 : -

HABILITADA

=> pós-condição 1 : -

C1_tamanho_garrafa_corrente_atualizado

=> pré-condição 2 : -

INABILITADA

=> pós-condição 2 : -

NÃO C1_tamanho_garrafa_corrente_atualizado

4.2 Segunda alternativa

Em termos de processadores digitais programáveis, a segunda alternativa de arquitetura de processadores do SICLEN difere da primeira por ser composta de um controlador geral da área de engarrafamento ao qual estão subordinados controladores individuais de linha, isto é, cada linha possui seu próprio processador controlador. Interagindo com os demais processadores (acionadores, sensores e conversor analógico-digital), o primeiro detém o comando de toda a área de engarrafamento, “enxergando” as linhas como subsistemas independentes, e os segundos são responsáveis apenas por suas respectivas linhas.

1. Definição de Aspectos de Implementação do Sistema

1.1 Processadores

O Sistema de Controle de Linhas de Engarrafamento - SICLEN será constituído por um conjunto de equipamentos que incluem processadores programáveis e processadores não-programáveis, todos alimentados por uma fonte elétrica de 110 volts. O sistema controlará 3 linhas de engarrafamento (FIGURA 1) e seus processadores deverão possuir as seguintes características :

1) PR1.1 (PR'1) - controlador de área

Microcomputador que executará as funções de :

- controlador principal do processo de engarrafamento;
- painel de controle do supervisor;
- controlador das linhas de comunicação inter-processadores.

Configuração :

- CPU Intel 486 DX2 de 66 MHz;
- barramento ISA;
- 8 Mb de memória RAM;
- memória cache de 256 Kb;
- disco rígido padrão IDE de 340 Mb com tempo de acesso de 11 ms e taxa de transferência de 1,2 Kb/s;
- drive de 3 1/2";
- teclado alfanumérico de 101 teclas;
- monitor de vídeo monocromático de 14" padrão SVGA;
- interface de vídeo SVGA com 512 Kb;
- cartão adaptador de rede Ethernet com porta 10baseT; e
- sistema operacional de rede local de dados Novell Netware Lite, acompanhado de um módulo host (hub) Ethernet 10baseT com no mínimo 4 portas.

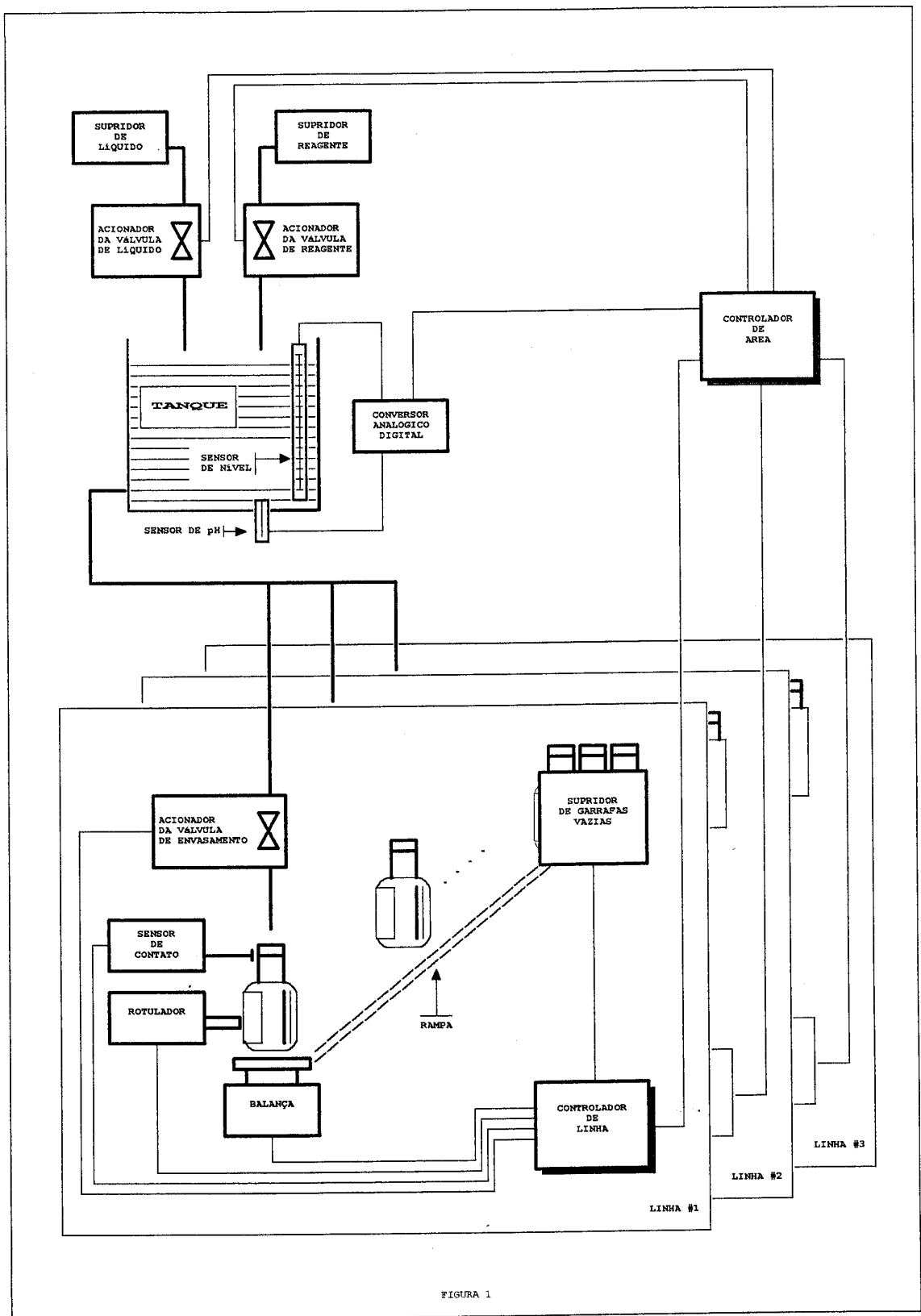


FIGURA 1

2) PR1.2{i} (PR'2{i}) - controlador da linha i

Microcomputador alocado a cada linha de engarrafamento, executando as funções de :

- controle e monitoração dos equipamentos da linha; e
- painel de controle do operador da linha.

Configuração :

- CPU Intel 486 DX2 de 66 MHz;
- barramento ISA;
- 8 Mb de memória RAM;
- memória cache de 256 Kb;
- disco rígido padrão IDE de 340 Mb com tempo de acesso de 11 ms e taxa de transferência de 1,2 Kb/s;
- drive de 3 1/2";
- teclado alfanumérico de 101 teclas;
- monitor de vídeo monocromático de 14" padrão SVGA;
- interface de vídeo SVGA com 512 Kb; e
- cartão adaptador de rede Ethernet com porta 10baseT.

3) PR2.1 (PR'3) - sensor de pH

Dispositivo a ser instalado no tanque que deverá ser capaz de fornecer continuamente o valor corrente do pH do líquido ao conversor analógico-digital (PR2.3).

4) PR2.2 (PR'4) - sensor de nível

Dispositivo a ser instalado no tanque que deverá ser capaz de fornecer continuamente o valor corrente do nível do líquido ao conversor analógico-digital (PR2.3).

5) PR2.3 (PR'5) - conversor analógico/digital

Dispositivo capaz de converter informações da forma analógica para a forma digital, possibilitando à aplicação obter dados interpretáveis do pH e nível correntes do líquido no tanque.

6) PR3.1 (PR'6) - acionador da válvula de líquido

Dispositivo eletro-mecânico capaz de regular a abertura da válvula de admissão de líquido do tanque recebendo do controlador de área (PR1.1) o valor digital dessa abertura.

7) PR3.2 (PR'7) - acionador da válvula de reagente

Dispositivo eletro-mecânico capaz de regular a abertura da válvula de admissão de reagente do tanque recebendo do controlador de área (PR1.1) o valor digital dessa abertura.

8) PR3.3{i} (PR'8{i}) - acionador da válvula de envasamento

Dispositivo eletro-mecânico a ser alocado a cada linha de engarrafamento capaz de regular a abertura da válvula de envasamento da

linha, recebendo do respectivo controlador de linha (PR1.2{i}) o valor digital dessa abertura.

1.2 Interfaces

1.2.1 Inter-processadores

A troca de dados entre os processadores programáveis do SICLEN será gerenciada pelo sistema operacional de rede local de dados residente no PR1.1, o qual atuará como servidor não-dedicado da rede.

1.2.2 Interface Usuário-máquina

a) Supervisor

Através do teclado do processador controlador de área - PR1.1 -, o supervisor poderá habilitar/inabilitar a área de engarrafamento (modelado pelos sinais "habilitar_área" e "inabilitar_área") e ajustar o ponto de pH do líquido a ser engarrafado (modelado pelo fluxo de dados discreto "ajuste_pH"). A aplicação exibirá no monitor de vídeo uma tela (FORMATO DE TELA 1) contendo os valores de pH e nível correntes do líquido (modelado pelo fluxo de dados contínuo "condições_tanque"), o estado de cada linha (modelado pelo fluxo de dados contínuo "estado_linhas"), caracteres resultantes da digitação do ajuste de pH e, quando for o caso, mensagens acusando a rejeição de valores inadequados de ajuste de pH (modelado pelo fluxo de dados discreto "rejeição_ajuste_pH"). Este processador deverá ser capaz ainda de emitir alarmes sonoros alertando o supervisor da ocorrência de anomalia em valores do pH ou do nível do líquido (modelado pelos sinais "anomalia_pH" e "anomalia_nível_tanque").

b) Operador de linha

Pelo acionamento do teclado do processador controlador de linha - PR1.2{i} -, o operador poderá ligar/desligar a linha de engarrafamento sob sua responsabilidade e sinalizar ao sistema que já removeu a garrafa da posição de envasamento (modelado pelos sinais "ligar_linha(i)", "desligar_linha(i)" e "garrafa_removida(i)") bem como alterar o tamanho da garrafa utilizada (modelado pelo fluxo de dados discreto "tamanho_garrafa(i)"). A aplicação exibirá no monitor de vídeo uma tela (FORMATO DE TELA 2) contendo o estado (ligada/desligada) e volume da garrafa empregada na linha (modelado pelo fluxo de dados discreto "estado_da_linha(i)"), caracteres resultantes da digitação de novos tamanhos de garrafa e, quando for o caso, mensagens acusando possíveis rejeições de valores inadequados de tamanho de garrafa (modelado pelo fluxo de dados discreto "rejeição_tamanho_garrafa(i)").

SICLEN			
Condições do tanque			
pH-corrente =		6.35	nível (%) = 0.85
Condições das linhas			
linha	1	2	3
estado	ligada	ligada	desligada
volume da garrafa	2.0 L	3.5 L	*****
Linha de comandos			
-> valor de ajuste de pH inválido !			

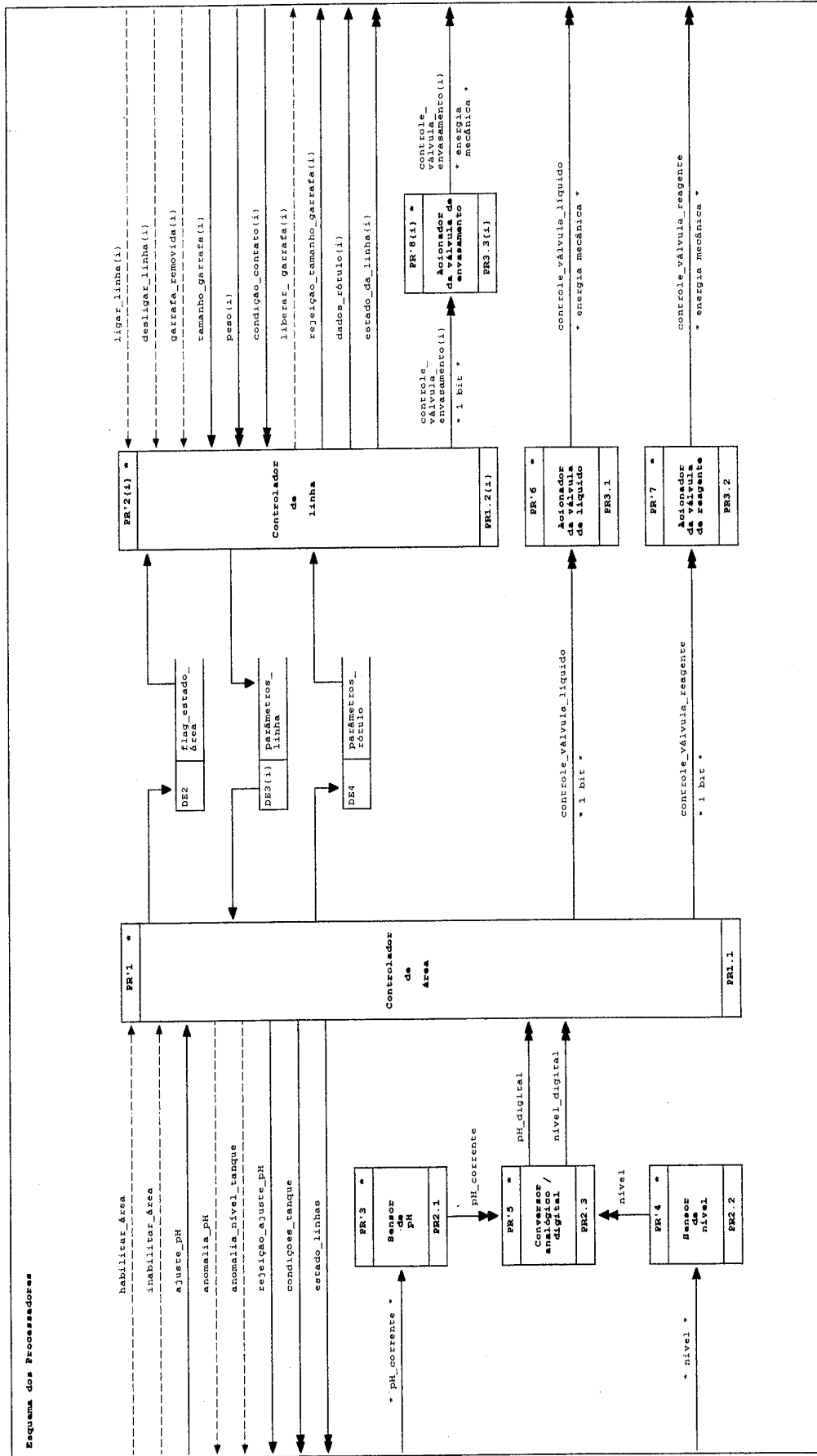
FORMATO DE TELA 1

SICLEN - Linha # 1	
Condição da linha	
estado =	ligada
volume-garrafa =	2.0 L
Linha de comandos	
-> Volume da garrafa inválido !	

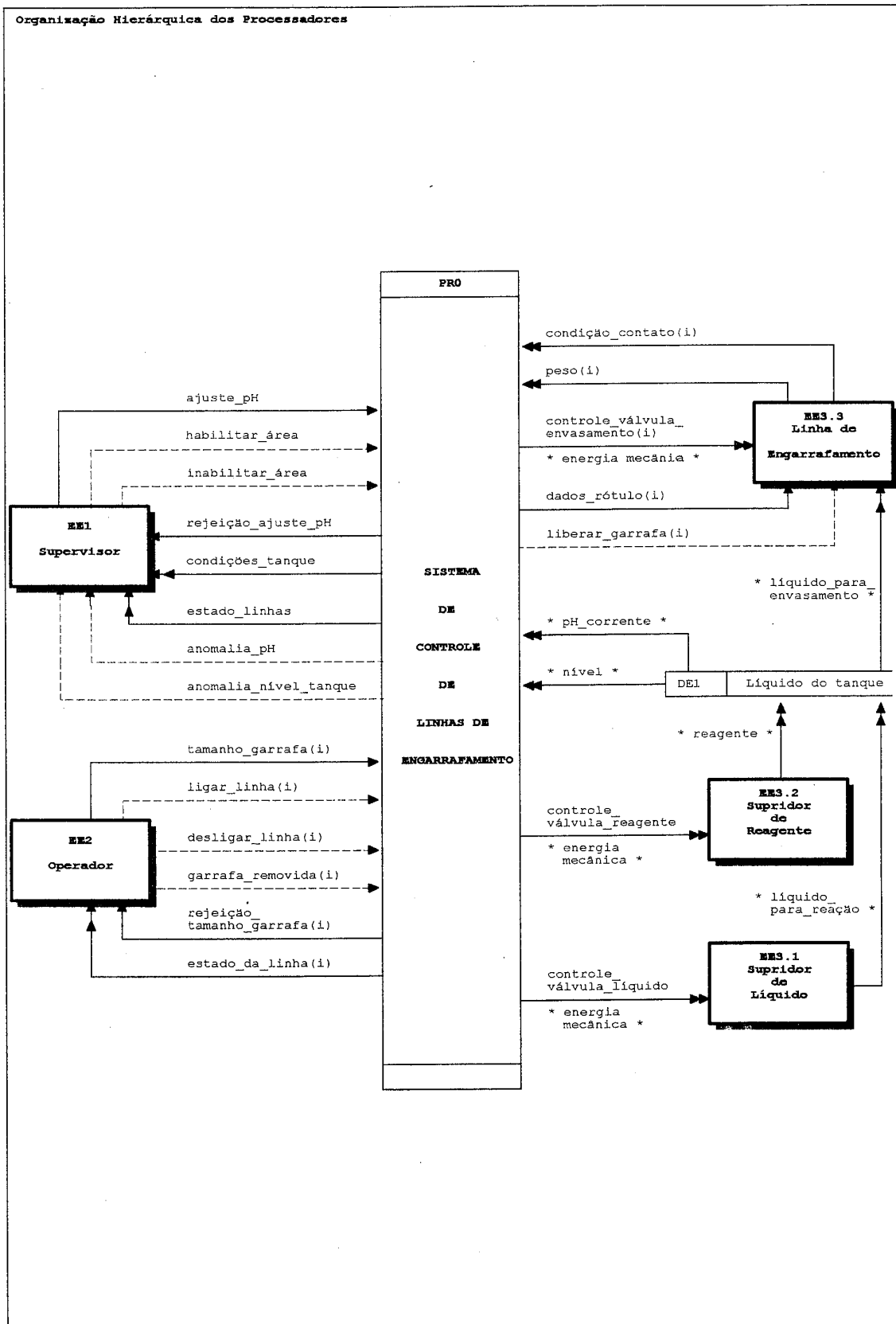
FORMATO DE TELA 2

2. Seção Esquemática

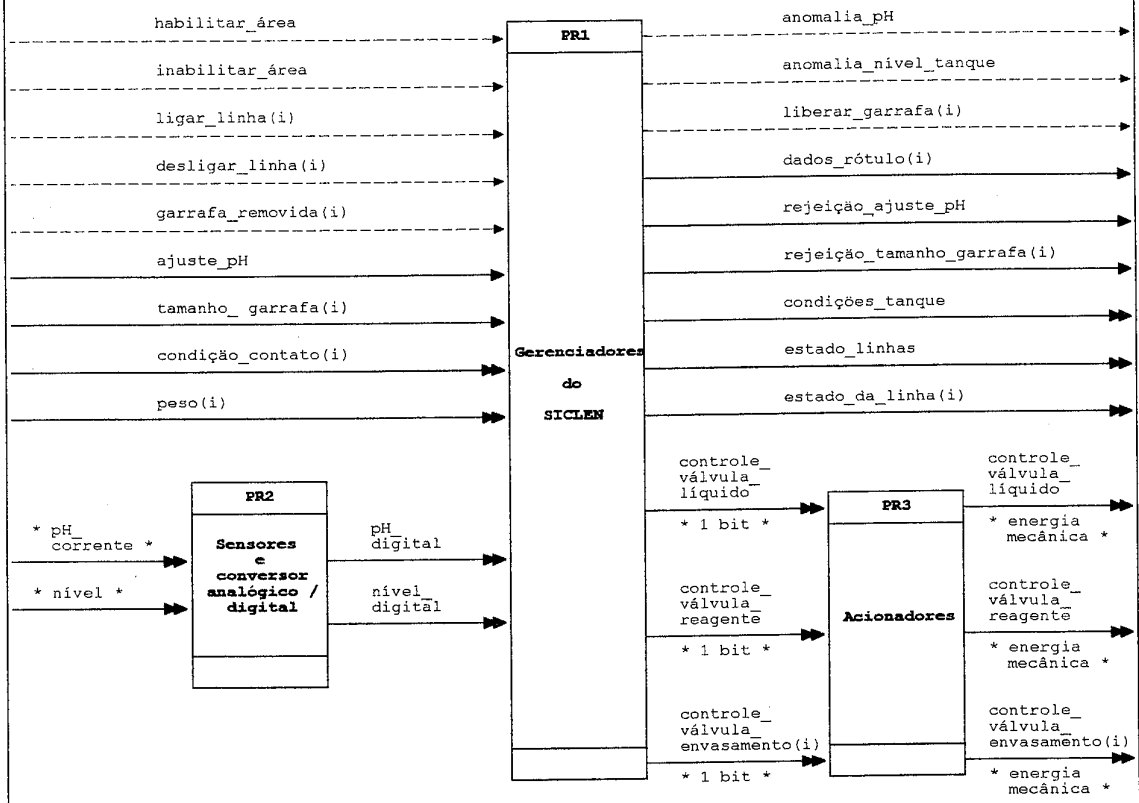
2.1 Esquema dos Processadores



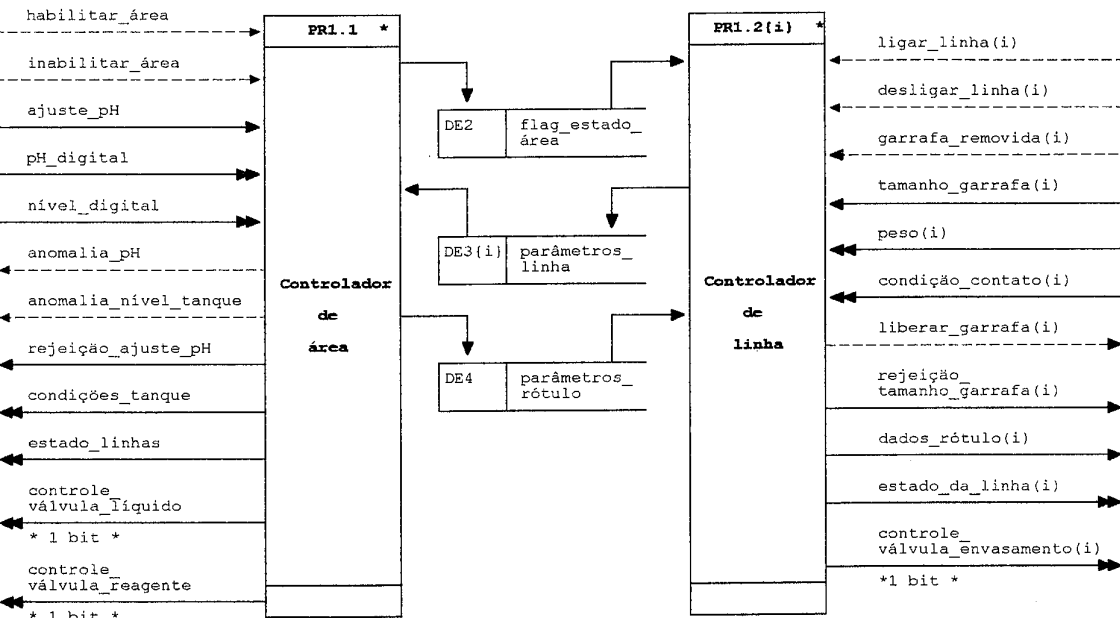
2.2 Organização Hierárquica dos Processadores



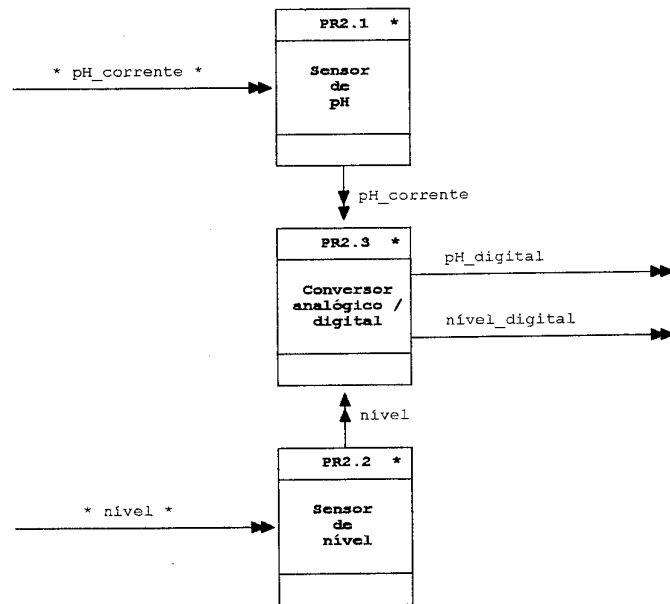
Organização Hierárquica dos Processadores - detalhamento de PRO



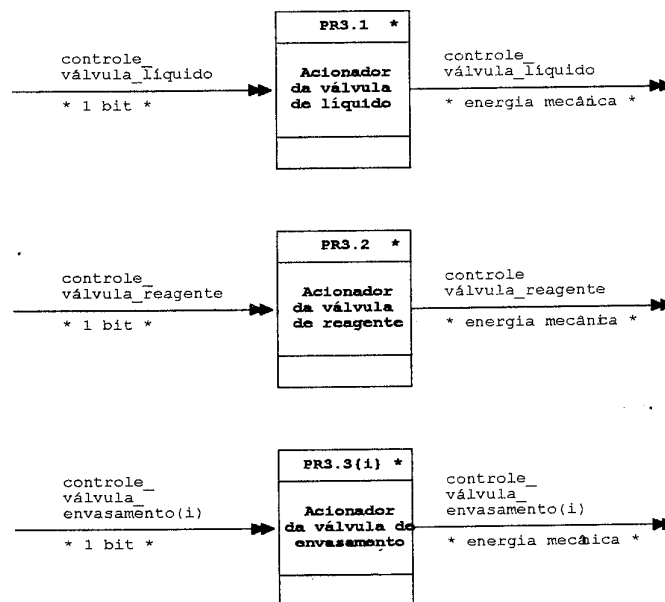
Organização Hierárquica dos Processadores - detalhamento de PR1 - Gerenciadores do SICLEN



Organização Hierárquica dos Processadores - detalhamento de PR2 - Sensores e conversor analógico/digital

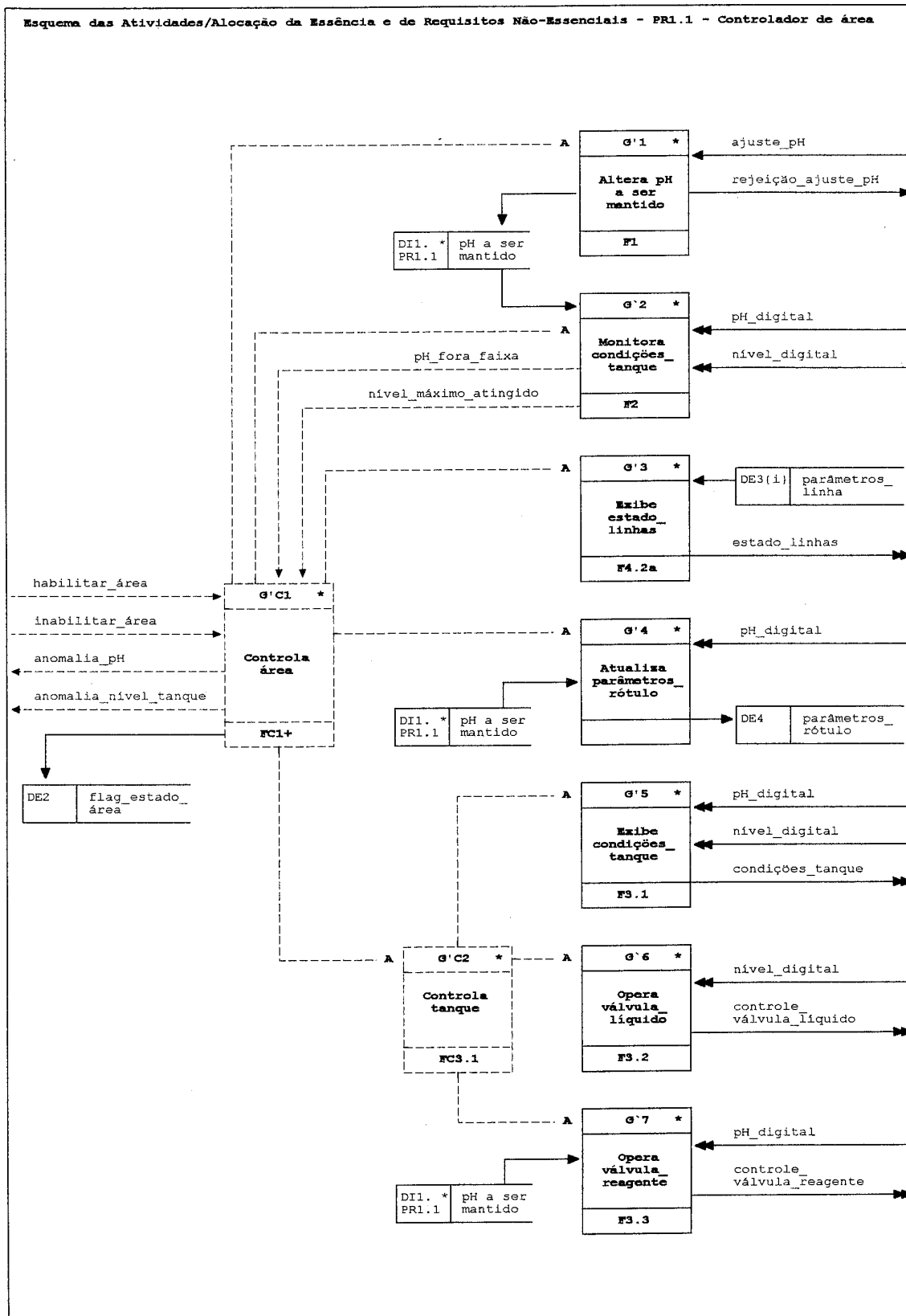


Organização Hierárquica dos Processadores - detalhamento de PR3 - Acionadores

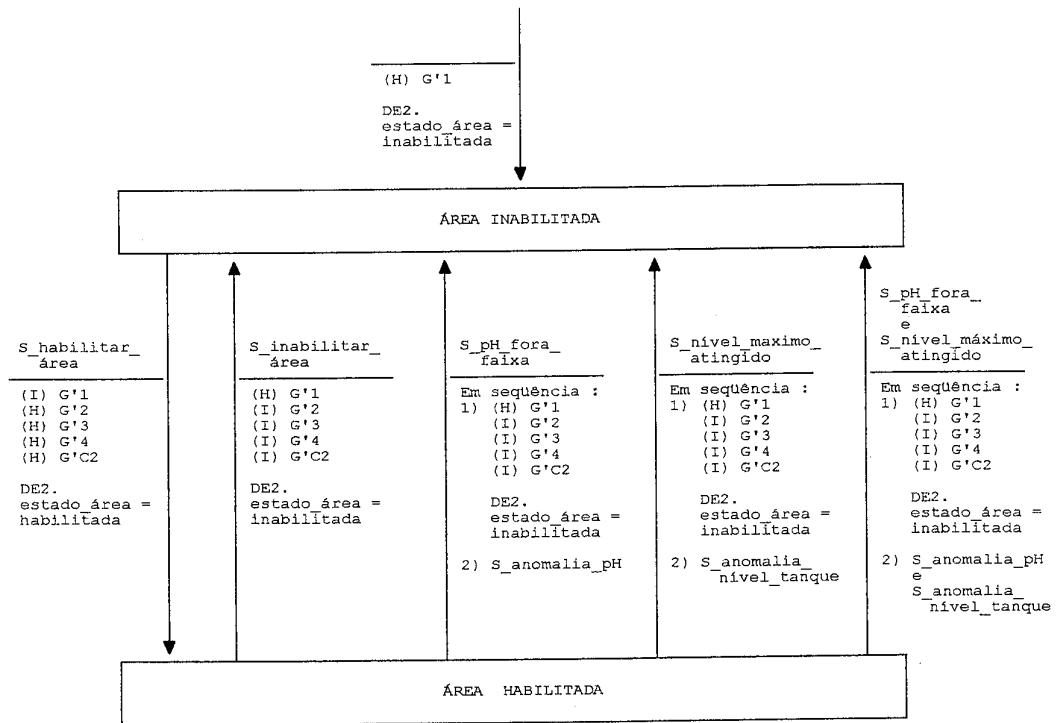


2.3 Esquemas das Atividades/

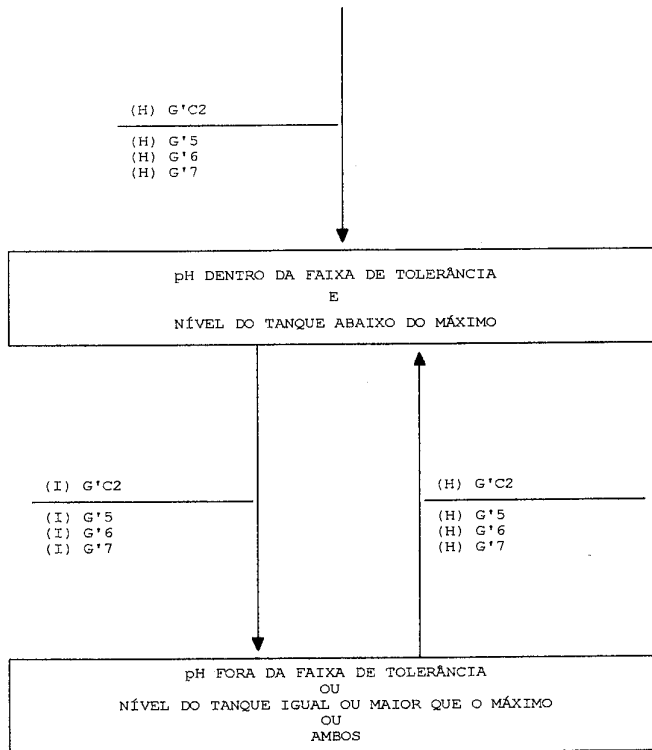
Alocação da Essência e de Requisitos Não-Essenciais a cada Processador



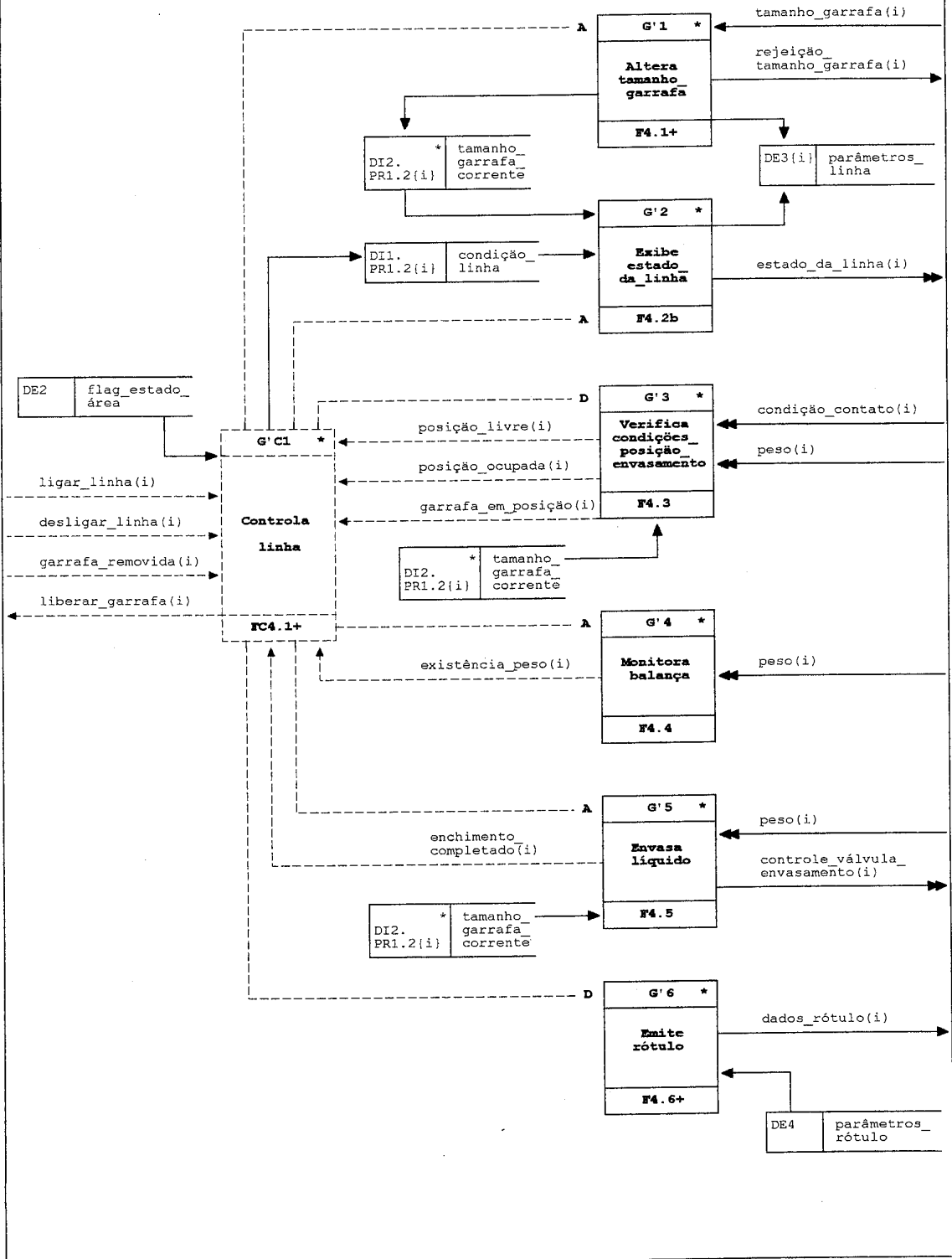
Especificação de PR1.1.G'C1 - Controla área



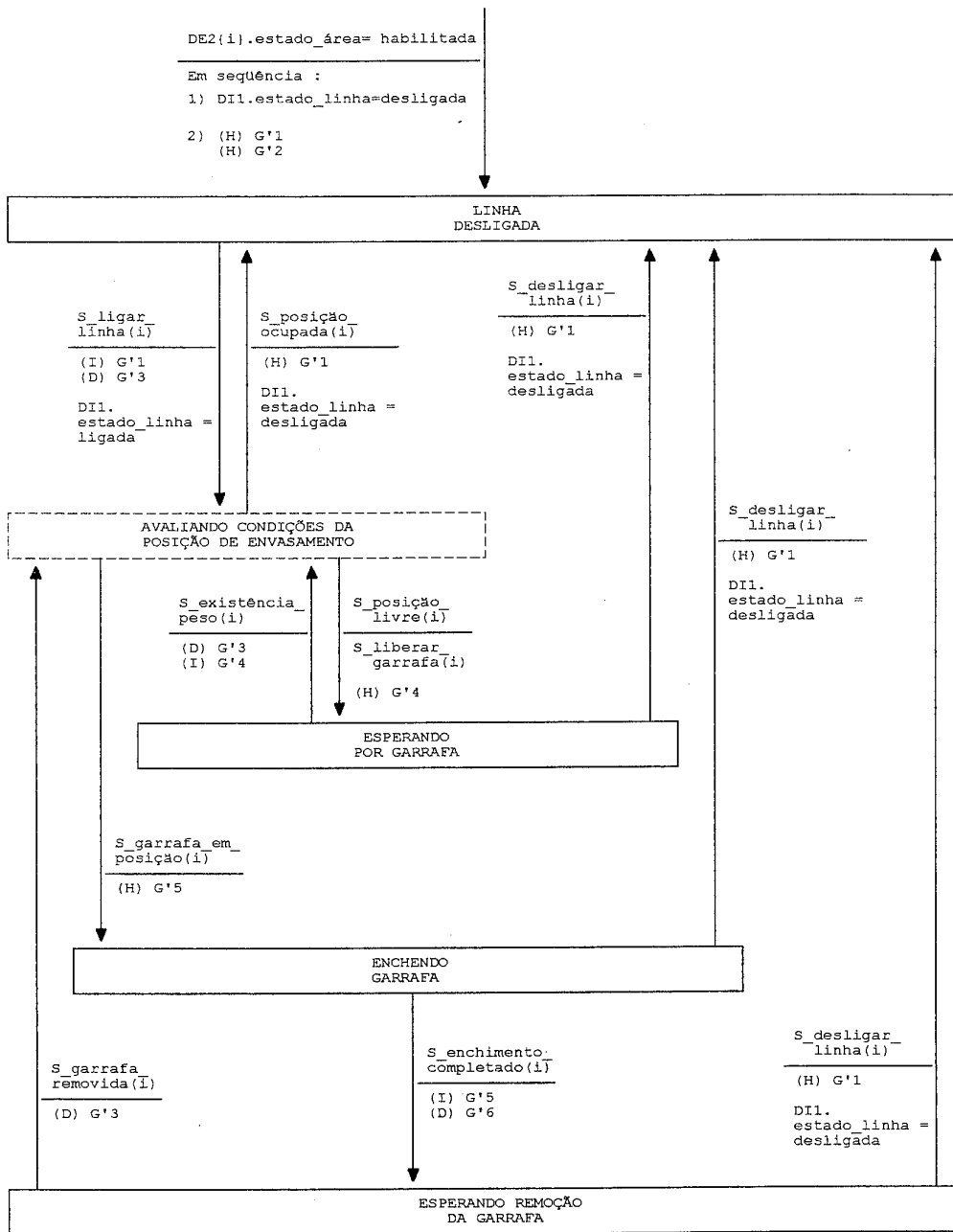
Especificação de PRI.1.G'C2 - Controla tanque



Esquema das Atividades/Alocação da Essência e de Requisitos Não-Essenciais - PR1.2(i) - Controlador de linha



Especificação de PR1.2(i).G'Cl - Controla linha



3. Seção Detalhada

3.1 Seção de Definições Globais

- 1) De um modo geral, a apresentação de uma pré/pós-condição procura seguir, aproximadamente, a sintaxe de uma linguagem de primeira ordem.
- 2) O termo extra-lógico SUBSTITUÍDO PELO significa a modificação do valor de um elemento de dado de um depósito interno, o qual passa a possuir o valor do atributo do elemento de modelagem que segue o termo.

3.2 Descrição Detalhada de Atividades Primitivas -

Lista de Pré/Pós-Condições

PR1.1.G'3 - Exibe estado linhas

Lista de pré/pós-condições

=> pré-condição 1 : -

HABILITADA

=> pós-condição 1 : -

"FDC_estado_linhas" TAL QUE PARA TODO i <0

<1 valor de "FDC_estado_linhas.ED_i" = valor de "ED_i" 1>

E

<1 valor de "FDC_estado_linhas.ED_condição_linha(i)" = valor de

"DE3{i}_parâmetros_linha.ED_condição_linha(i)" 1>

E

<1 valor de "FDC_estado_linhas.ED_volume(i)" = valor de

"DE3{i}_parâmetros_linha.ED_volume(i)" 1> 0>

=> pré-condição 2 : -

INABILITADA

=> pós-condição 2 : -

NÃO "FDC_estado_linhas"

PR1.1.G'4 - Atualiza parâmetros rótulo

Seção de Definições Locais

Cláusula :

C1_parâmetros_rótulo_atualizados

<0 valor de "DE3_parâmetros_rótulo.ED_pH_corrente" SUBSTITUÍDO PELO
valor de "FDC_pH_digital.ED_pH_corrente" 0>

E

<0 valor de "DE3_parâmetros_rótulo.ED_pH_nominal" SUBSTITUÍDO PELO
valor de "DI1_pH a ser mantido.ED_pH_nominal" 0>

Lista de pré/pós-condições

=> pré-condição 1 : -

HABILITADA

E

"FDC_pH_digital

=> pós-condição 1 : -

C1_parâmetros_rótulo_atualizados

=> pré-condição 2 : -

INABILITADA

=> pós-condição 2 : -

NÃO C1_parâmetros_rótulo_atualizados

PR1.2{i}.G'1 - Altera tamanho garrafa

Seção de Definições Locais

Cláusulas :

C1_novo_tamanho_válido

"FDD_tamanho_garrafa(i)" TAL QUE <0

<1 valor de "FDD_tamanho_garrafa(i).ED_volume(i)" > = valor correspondente a volume_mínimo de "ED_volume(i)" 1>

E

<1 valor de "FDD_tamanho_garrafa(i).ED_volume(i)" <= valor correspondente a volume_máximo de "ED_volume(i)" 1>

E

<1 valor de "FDD_tamanho_garrafa(i).ED_peso_garrafa_vazia(i)" > = valor correspondente a peso_mínimo de "ED_peso_garrafa_vazia(i)" 1>

E

<1 valor de "FDD_tamanho_garrafa(i).ED_peso_garrafa_vazia(i)" < = valor correspondente a peso_mínimo de "ED_peso_garrafa_vazia(i)" 1> 0>

C2_tamanho_garrafa_corrente_atualizado

<0 valor de "DI2_tamanho_garrafa_corrente.ED_volume(i)" SUBSTITUÍDO PELO valor de "FDD_tamanho_garrafa(i).ED_volume(i)" 0>

E

<0 valor de "DI2_tamanho_garrafa_corrente.ED_peso_garrafa_vazia(i)" SUBSTITUÍDO PELO valor de "FDD_tamanho_garrafa(i).ED_peso_garrafa_vazia(i)" 0>

E

<0 valor de "DE3{i}_parâmetros_linha.ED_volume(i)" SUBSTITUÍDO PELO valor de "FDD_tamanho_garrafa(i).ED_volume(i)" 0>

Lista de pré/pós-condições

=> pré-condição 1 : -

HABILITADA

E

C1_novo_tamanho_válido

```
=> pós-condição 1 : -
    C2_tamanho_garrafa_corrente_atualizado
    E
    NÃO "FDC_rejeição_tamanho_garrafa(i)"

=> pré-condição 2 : -
    HABILITADA
    E
    NÃO C1_novo_tamanho_válido

=> pós-condição 2 : -
    "FDD_rejeição_tamanho_garrafa(i)"
    E
    NÃO C2_tamanho_garrafa_atualizado

=> pré-condição 3 : -
    INABILITADA

=> pós-condição 3 : -
    NÃO <0 "FDD_rejeição_tamanho_garrafa(i)"
    E
    C2_tamanho_garrafa_atualizado 0>
```

PR1.2{i}.G'2 - Exibe estado da linha

Seção de Definições Locais

Cláusula :

C1_condição_linha_atualizada

<1 valor de "DE4_parâmetros_linha.ED_condição_linha" SUBSTITUÍDO PELO valor de "DI1_condição_linha.ED_condição_linha" 1>

Lista de pré/pós-condições

=> pré-condição 1 : -

HABILITADA

=> pós-condição 1 : -

"FDC_estado_da_linha" TAL QUE <0

<1 valor de "FDC_estado_da_linha.ED_condição_linha" = valor de
"DI1_condição_linha.ED_condição_linha" 1>

E

<1 valor de "FDC_estado_da_linha.ED_volume" = valor de

"DI2_tamanho_garrafa_corrente.ED_volume" 1> 0

E

C1_condição_linha_atualizada

=> pré-condição 2 : -

INABILITADA

=> pós-condição 2 : -

NÃO <0 "FDC_estado_linhas" E C1_condição_linha_atualizada 0>

PR1.2(i).G'6 - Emite rótulo

Lista de pré/pós-condições

=> pré-condição 1 : -

DISPARADA

=> pós-condição 1 : -

"FDD_dados_rótulo(i)" TAL QUE <0

<1 valor de "FDD_dados_rótulo(i).ED_pH_corrente" = valor de

"DE4_parâmetros_rótulo.ED_pH_corrente" 1>

E

<1 valor de "FDD_dados_rótulo(i).ED_pH_nominal" = valor de

"DE4_parâmetros_rótulo.ED_pH_nominal" 1> 0>

5. A alternativa escolhida - MCP, MCM e Dicionário de Dados

Entre as duas alternativas apresentadas, foi escolhida a segunda tendo em vista os seguintes critérios :

a) confiabilidade

Pela alocação de um processador de controle a cada linha, uma possível pane em algum(ns) dele(s) não compromete o funcionamento da área como um todo, podendo esta continuar operando as demais linhas enquanto a linha defeituosa é reparada.

b) acoplamento inter-processadores

O reduzido grau de acoplamento entre o processador de controle da área e os processadores de controle de linha decorrente do compartilhamento de dados favorece o desempenho e a manutenção do sistema.

c) flexibilidade

A arquitetura escolhida possibilita um aumento razoável do número de linhas de engarrafamento com reduzido impacto de alteração na implementação do sistema.

d) menor distorção em relação ao Modelo da Essência

Das duas arquiteturas estudadas, esta oferece a menor distorção entre a essência e a implementação, conferindo ao sistema melhor manutenibilidade e portatibilidade. Do ponto de vista do processo de desenvolvimento, oferece um elevado grau de domínio de complexidade em função do mapeamento quase direto entre o Modelo da Essência e o Modelo da Configuração dos Processadores.

Assim, são apresentados a seguir os respectivos MCP e MCM.

5.1 Modelo da Configuração de Processos

1. Definição dos Aspectos de Implementação de cada Processador em termos dos Processos Alocados

1.1 Processos

1.1.1 Processador PR1.1 - Controlador de área

Possui um único processo da aplicação (PR1.1.P'1 - Gerencia área), o qual gerencia as atividades de monitoração e controle das condições do líquido no tanque e a interface com o supervisor da área. O acesso aos dados compartilhados é realizado por meio do software gerenciador da rede residente neste processador, que é ativado por interrupção no momento que a aplicação necessita efetuar uma operação de acesso a esses dados.

1.1.2 Processadores PR1.2{i} - Controladores de linha

Possui um único processo (PR1.2{i}.P'1 - Gerencia linha) que gerencia as atividades de monitoração e controle dos equipamentos da linha e a interface com o operador.

1.2 Sistema Operacional

O sistema operacional de suporte a ser empregado nos processadores programáveis será o MS-DOS versão 6.0.

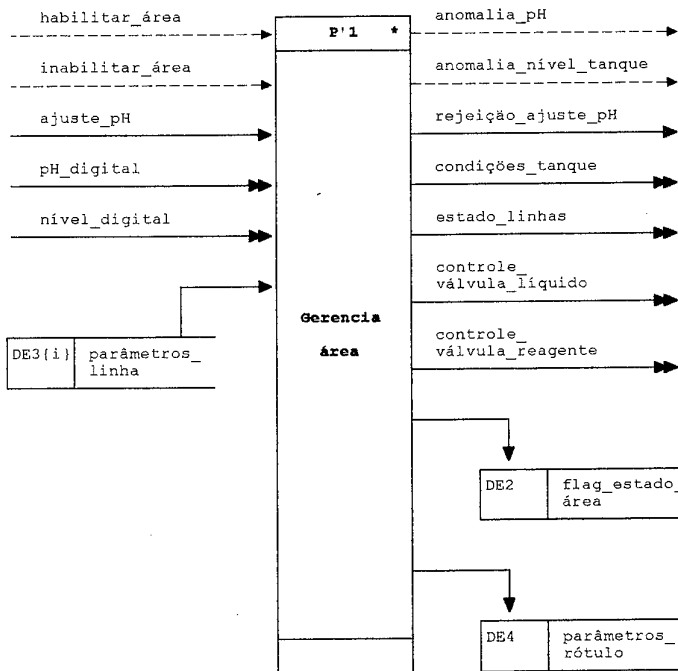
2. Definição de Aspectos da Modelagem

Dadas as características de desempenho dos processadores que suportam a aplicação, tornou-se viável a alocação das atividades por processos garantindo a menor distorção possível entre a essência e a implementação. Isto oferece um elevado grau de domínio da complexidade intrínseca do sistema em função do mapeamento quase direto entre o Modelo da Configuração de Processadores e o Modelo da Configuração de Processos.

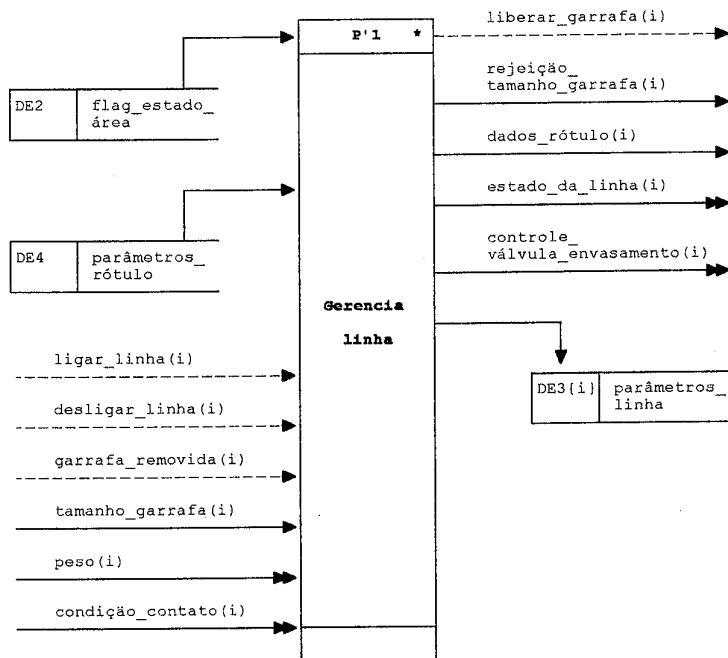
3. Seção Esquemática

3.1 Esquema dos Processos

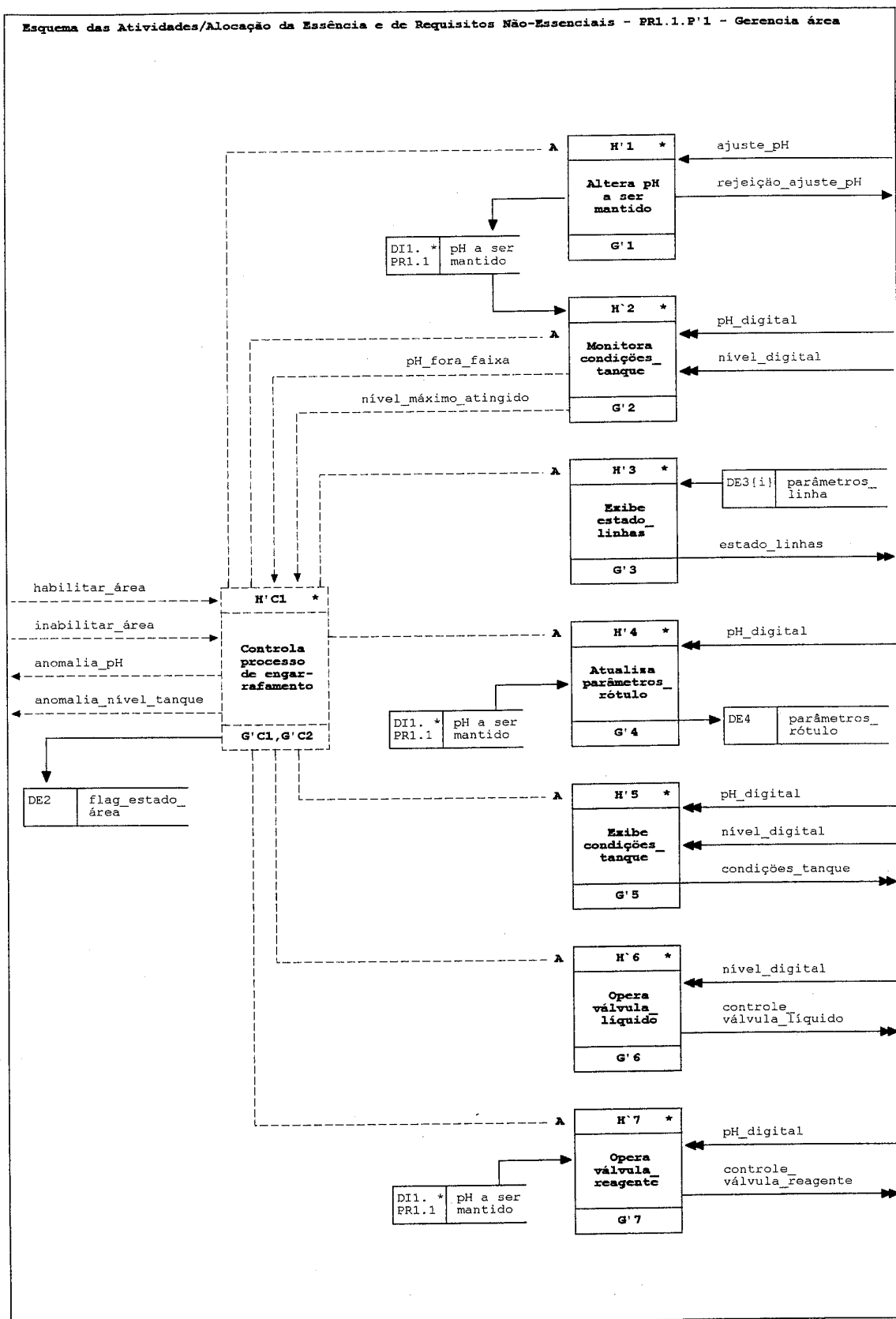
Esquema dos Processos - PR1.1 - Controlador de área

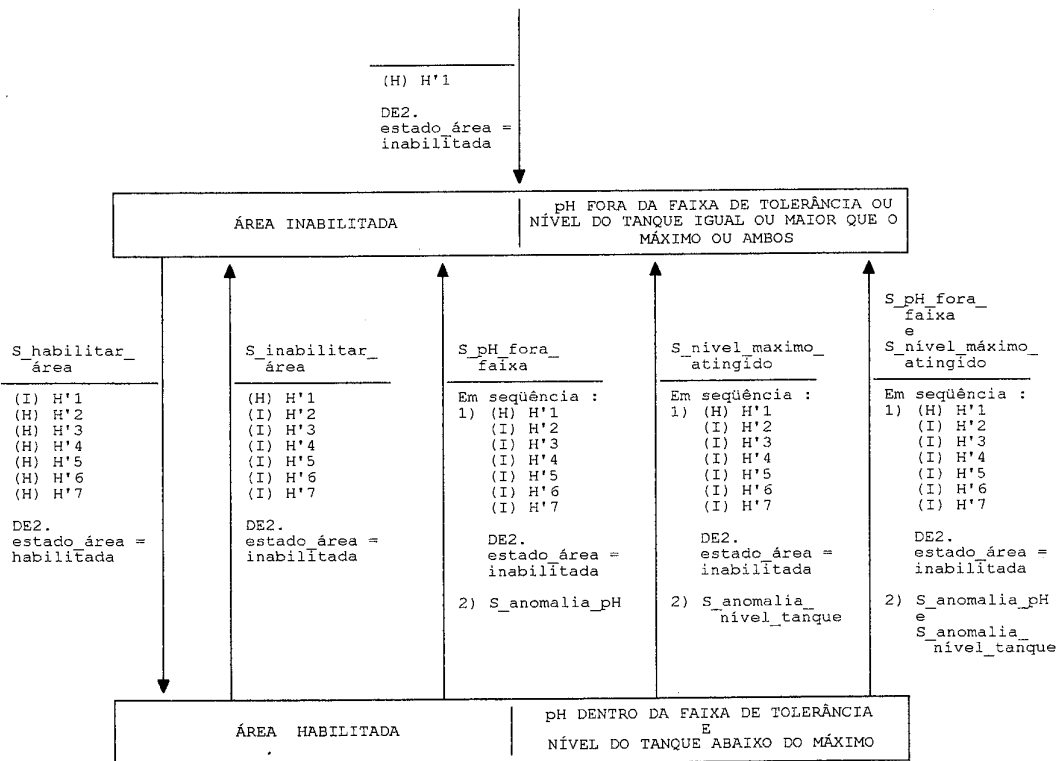


Esquema dos Processos - PR1.2(i) - Controlador de linha

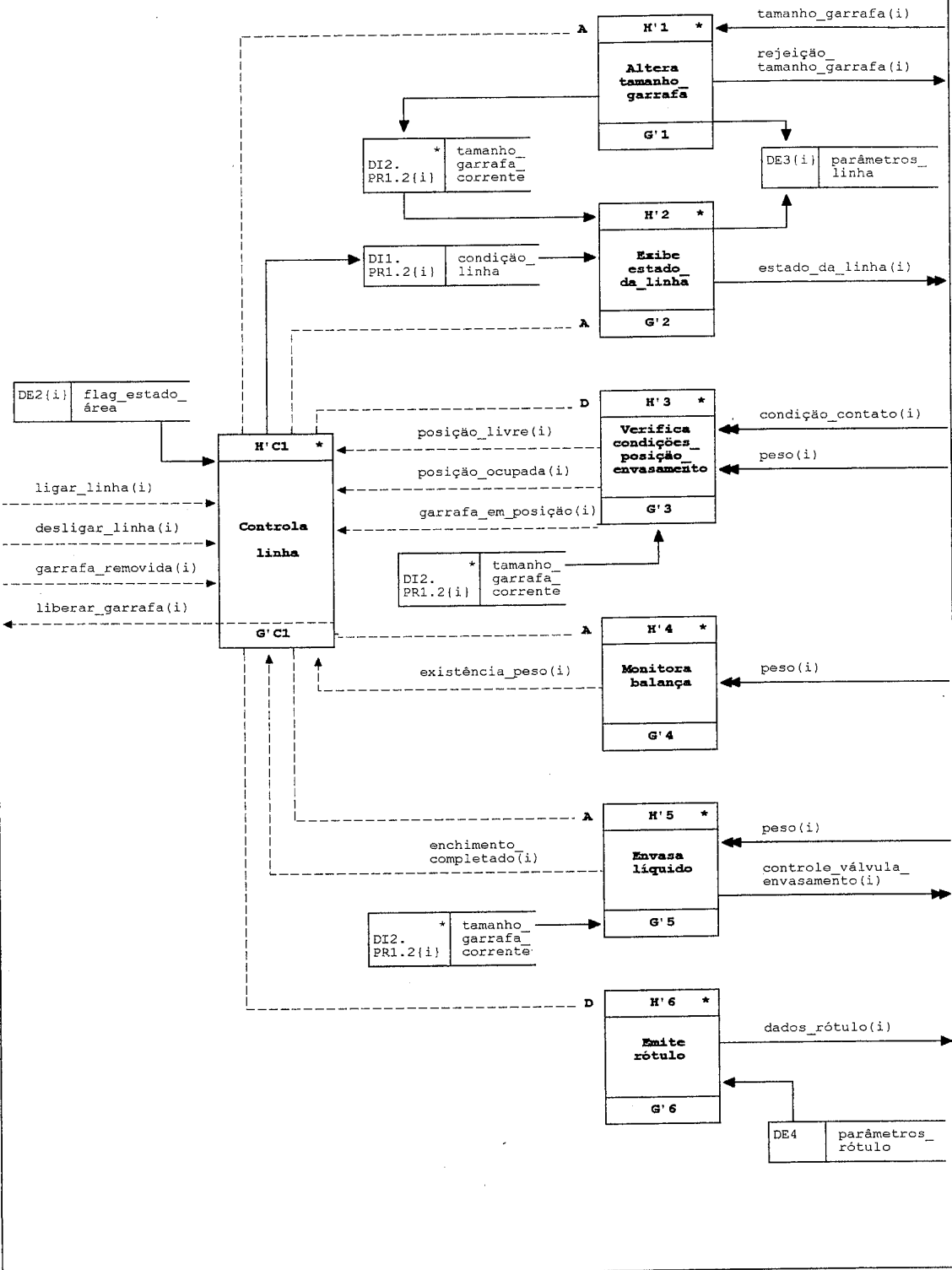


3.2 Esquemas das Atividades/Alocação da Essência e de Requisitos Não-Essenciais a cada Processo



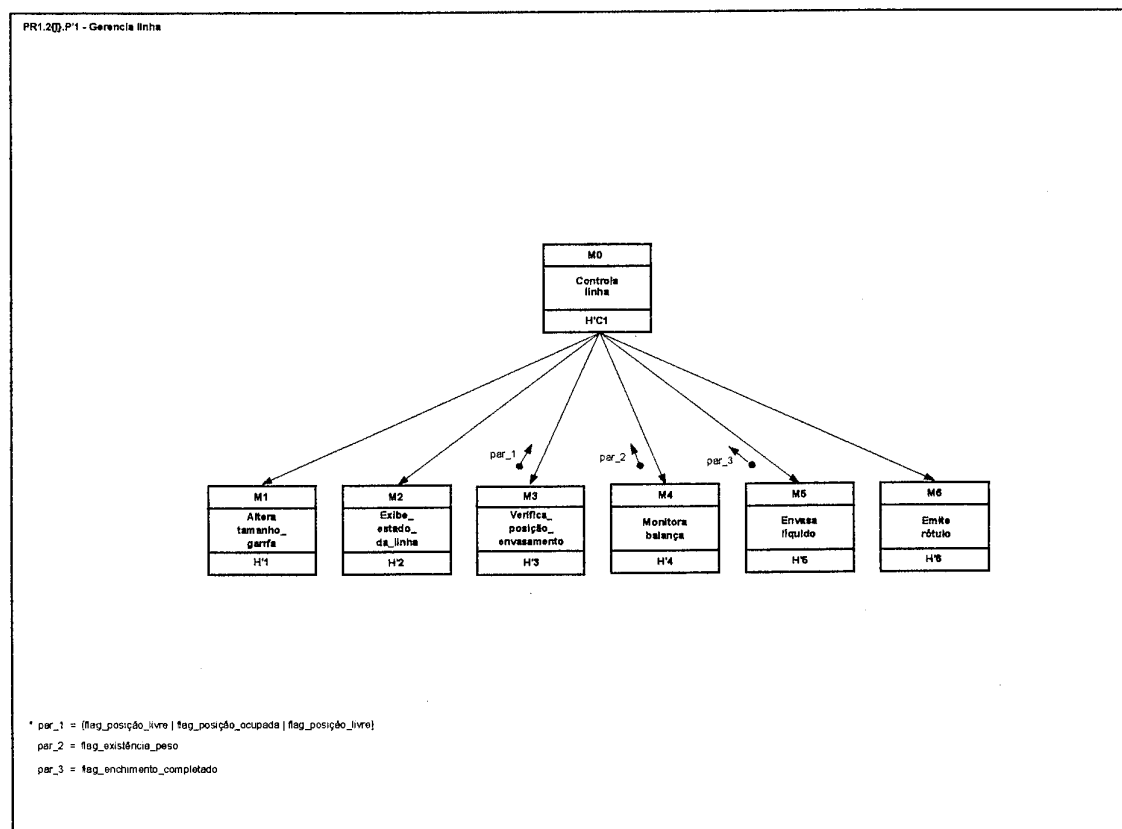
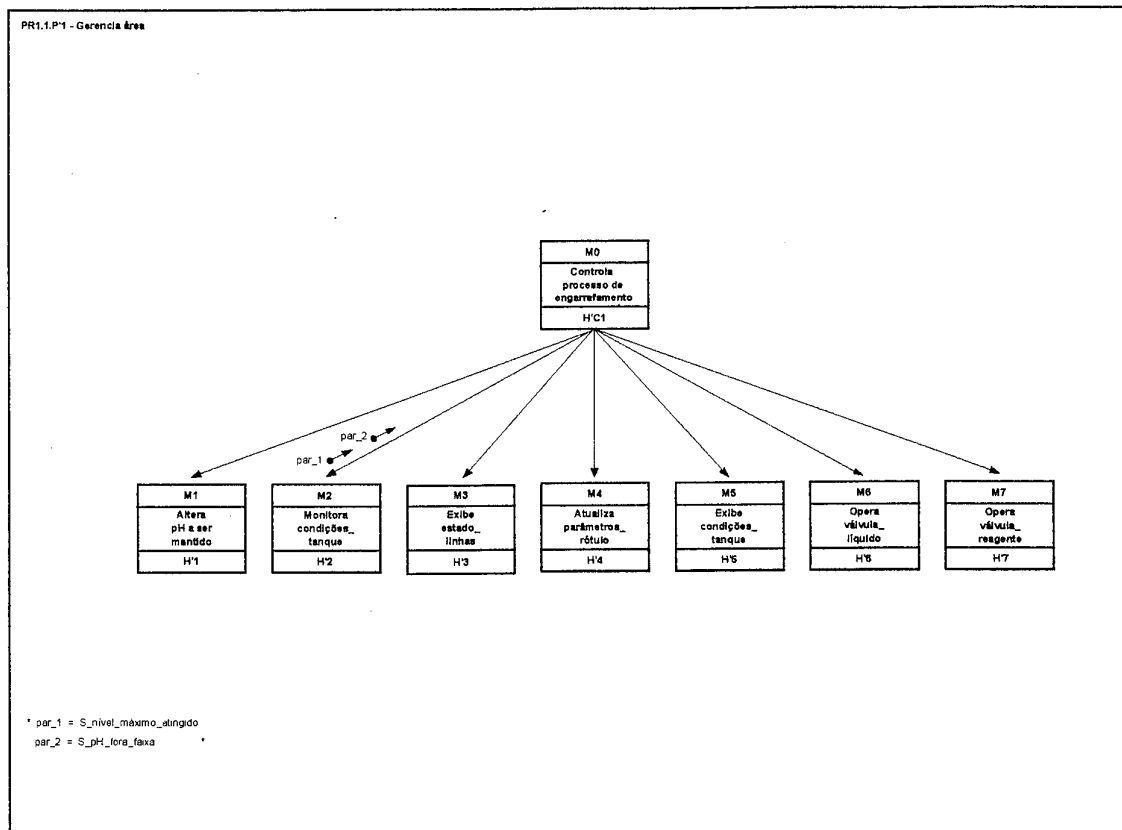


Esquema das Atividades/Alocação da Essência e de Requisitos Não-Essenciais - PR1.2(i).P'1 - Gerencia linha



5.2 Modelo da Configuração de Módulos

1. Diagramas Hierárquicos de Módulos associados a cada Processo



2. Descrição Detalhada dos Módulos associados a cada Processo

2.1 PR1.1.P'1 - Gerencia área

M0_Controla processo de engarrafamento;

procedimento inabilitação_padrão;

```
início
  estado_área <-- inabilitada;
  DE2(i).estado_área <-- inabilitada;
  executa M6;
  executa M7;
fim-procedimento;
```

procedimento trata_estado_1;

```
início
  enquanto estado_área = inabilitada faça
    executa M1;
    se S_inabilitar_área ocorreu
      então estado_área <-- habilitada;
  fim-enquanto;
fim-procedimento;
```

procedimento trata_estado_2;

```
início
  enquanto estado_área = habilitada faça
    executa M2;
    se S_inabilitar_área ou S_pH_fora_faixa ou S_nível_máximo_attingido
      ocorreu
        então estado_área <-- inabilitada
        senão executa M3;
        executa M4;
        executa M5;
        executa M6;
        executa M7;
    fim-senão;
  fim-se;
  fim-enquanto;
  inabilitação_padrão;
  se S_pH_fora_faixa ocorreu
    então emite S_anomalia_pH;
  se S_nível_máximo_attingido ocorreu
    então emite S_anomalia_nível_tanque;
fim-procedimento;
```

```
início {procedimento principal}
  inabilitação_padrão;
  loop
    trata_estado_1;
    trata_estado_2;
  fim-loop;
fim-M0.
```


M1_Altera pH a ser mantido;

```
início
  se ajuste_pH ocorreu
    então obtém novo_pH;
      se (novo_pH >= 0) e (novo_pH <= 14)
        então DI1.pH_nominal <-- novo_pH
          senão exibe mensagem 'valor de pH inválido';
      fim-então;
fim-M1;
```

M2_Monitora condições_tanque;

```
início
  obtém pH_corrente;
  obtém nível_corrente;
  obtém DI1.pH_nominal;
  se (pH_corrente > (pH_nominal + erro_pH)) ou
    (pH_corrente < (pH_nominal - erro_pH))
    então emite S_pH_fora_faixa;
  se nível_corrente >= nível_máximo
    então emite S_nível_máximo_atingido;
fim-M2;
```

M3_Exibe estado_linhas;

```
início
  para i de 1 até nr_linhas faça
    obtém DE3(i).condição_linha;
    obtém DE3(i).volume;
  fim-para;
  para i de 1 até nr_linhas faça
    exibe condição_linha(i);
    exibe volume(i);
  fim-para;
fim-M3;
```

M4_Atualiza parâmetros_rótulo;

```
início
  obtém pH_corrente;
  obtém DI1.pH_nominal;
  DE4.pH_corrente <-- pH_corrente;
  DE4.pH_nominal <-- pH_nominal;
fim-M4;
```

M5_Exibe condições_tanque;

```
início
  obtém pH_corrente;
  obtém nível_corrente;
  exibe pH_corrente;
  exibe Nível_corrente;
fim-M5;
```

```

M6_Opera válvula_líquido;

início
  se estado_área = inabilitada
    então controle_válvula_líquido <-- fechada
    senão obtém nível_corrente;
      se nível_corrente < nível_padrão
        então controle_válvula_líquido <-- aberta
        senão controle_válvula_líquido <-- fechada;
      fim-senão;
  fim-M6;

```

```

M7_Opera válvula_reagente;

início
  se estado_área = inabilitada
    então controle_válvula_reagente <-- fechada
    senão obtém pH_corrente;
      obtém DI1.pH_nominal;
      se pH_corrente < pH_nominal
        então controle_válvula_reagente <-- aberta
        senão controle_válvula_reagente <-- fechada;
      fim-senão;
  fim-M7;

```

2.2 PR1.2{i}.P'1 - Gerencia linha

```

M0_Controla linha;

procedimento linha_desligada;

início
  estado_linha <-- desligada;
  DI1.estado_linha <-- desligada;
  fim-procedimento;

procedimento verifica_estado_linha;

início
  se S-desligar_linha ocorreu
    então linha_desligada;
  fim-procedimento;

procedimento verifica_estado_área;

início
  se DE2.estado_área = inabilitada
    então estado_área <-- inabilitada;
    executa M5;
  fim-então;
  fim-procedimento;

```

```

procedimento trata_estado_1;

início
  enquanto (estado_linha = desligada) e (estado_área = habilitada)
  faça
    executa M2;
    executa M1;
    se S_ligar_linha ocorreu
      então DI1.estado_linha <-- ligada;
      estado_linha <-- avaliando_pos_env;
      executa M2;
      executa M3;
      fim-então;
    fim-se;
    verifica_estado_área;
  fim-enquanto;
fim-procedimento;

procedimento trata_estado_2;

início
  se S_posição_livre ocorreu
    então emite S_liberar_garrafa;
    execute M4;
    estado_linha <-- esperando_gf
  fim-então;
  senão se S_posição_ocupada ocorreu
    então linha_desligada;
    senão se S_garrafa_em_posição ocorreu
      então executa M5;
      estado_linha <-- enchendo_gf;
    fim-então;
  fim-se;
fim-procedimento;

procedimento trata_estado_3;

início
  enquanto (estado_linha = esperando_gf) e (estado_área = habilitada)
  faça
    executa M4;
    se S_existência_peso ocorreu
      então executa M3;
      estado_linha <-- avaliando_pos_env;
    fim-se;
    verifica_estado_linha;
    verifica_estado_área;
  fim-enquanto;
fim-procedimento;

```

```

procedimento trata_estado_4;
início
  enquanto (estado_linha = enchendo_gf) e (estado_área = habilitada)
  faça
    executa M5;
    se S_enchimento_completado ocorreu
      então executa M6;
      estado_linha <-- esperando_remoção_gf;
    fim-se;
    verifica_estado_linha;
    verifica_estado_área;
  fim-enquanto;
fim-procedimento;

procedimento trata_estado_5;

início
  enquanto (estado_linha = esperando_remoção_gf) e (estado_área =
  habilitada) faça
    se S_garrafa_removerida ocorreu
      então executa M3;
      estado_linha <-- avaliando_pos_env;
    fim-se;
    verifica_estado_linha;
    verifica_estado_área;
  fim-enquanto;
fim-procedimento;

início {procedimento principal}
  estado_área <-- inabilitada;
  executa M5;
  loop
    se DE2.estado_área = habilitada faça
      estado_área <-- habilitada;
      linha_desligada;
      enquanto estado_área = habilitada faça
        conforme estado_linha =
          desligada           : trata_estado_1;
          avaliando_pos_env   : trata_estado_2;
          esperando_gf        : trata_estado_3;
          enchendo_gf         : trata_estado_4;
          esperando_remoção_gf : trata_estado_5;
        fim-conforme;
      fim-enquanto;
    fim-se;
  fim-loop;
fim-M0.

```

```

M1_Altera tamanho_garrafa;
início
  se tamanho_garrafa ocorreu
    então condição_volume <-- inválida;
    condição_peso <-- inválida;
    enquanto condição_volume = inválida faça
      obtém volume;
      se (volume > = volume_mínimo) e (volume < =
volume_máximo)
        então DI2.volume <-- volume;
        DE3(i).volume <-- volume;
        condição_volume <-- válida;
        senão exibe mensagem 'valor de volume inválido';
        fim-se;
      fim-enquanto;
    enquanto condição_peso = inválida faça
      obtém peso_garrafa_vazia;
      se (peso_garrafa_vazia > = peso_mínimo) e
        (peso_garrafa_vazia < = peso_máximo)
        então DI2.peso_garrafa_vazia <-- peso_garrafa_vazia;
        condição_peso <-- válida
        senão exibe mensagem 'valor de peso de garrafa vazia
        inválido';
        fim-se;
      fim-enquanto;
    fim-se;
fim-M1;

```

```

M2_Exibe estado_da_linha;

```

```

início
  obtém DI1.condição_linha;
  obtém DI2.volume;
  exibe condição_linha;
  exibe volume;
  DE3(i).condição_linha <-- condição_linha;
fim-M2;

```

```

M3_Verifica condições_posição_envasamento;

```

```

início
  obtém condição_contato;
  obtém leitura_balança;
  obtém DI2.peso_garrafa_vazia;
  conforme leitura_balança =
    peso_garrafa_vazia : se condição_contato = sim
      então emite S_garrafa_em_posição
      senão emite S_posição_ocupada;
    0 : se condição_contato = sim
      então emite S_posição_ocupada
      senão emite S_posição_livre;
  fim-conforme;
fim-M3;

```

```

M4_Monitora balança;

```

```

início
  obtém leitura_balança;
  se leitura_balança > 0 então emite S_existência_peso;
fim-M4;

```

M5_Envasa líquido;

início

se estado_área = inabilitada

então controle_válvula_ensvasamento <-- fechada

senão obtém leitura_balança;

obtém volume;

se (leitura_balança = densidade_líquido * volume)

então controle_válvula_líquido <-- fechada;

emite S_enchimento_completado

senão controle_válvula_líquido <-- aberta;

fim-se;

fim-se;

fim-M5;

M6_Emite rótulo;

início

obtém DE4.pH_nominal;

obtém DE4.pH_corrente;

emite pH_nominal;

emite pH_corrente;

fim-M6;

5.5 Dicionário de Dados

1. Definição de Aspectos da Modelagem

Esta parte do Modelo da Implementação visa complementar o dicionário de dados produzido no Modelo da Essência devido à necessidade de se definir os novos elementos de modelagem incluídos na alternativa de implementação escolhida ou mesmo redefinir/substituir os já existentes.

2. Elementos de modelagem complementares

Depósito Externo

NOME	DE2_flag_estado_área
DEFINIÇÃO	modela a área de dados compartilhados pelo processador de área e os processadores de linha através da qual estes últimos reconhecem o estado corrente de funcionamento da área de engarrafamento.
COMPOSIÇÃO	ED_flag_área
NOME	DE3{i}_parâmetros_linha
DEFINIÇÃO	modela a área de dados compartilhados pelo processador de área e o processador da linha "i" através da qual o processador de linha torna disponível ao processador de área as informações de estado corrente e volume da garrafa em uso da linha "i".
COMPOSIÇÃO	ED_condição_linha + ED_volume
NOME	DE4_parâmetros_rótulo
DEFINIÇÃO	modela a área de dados compartilhados pelo processador de área e os processadores de linha através da qual o processador de área torna disponível aos processadores de linha as informações que deverão constar do rótulo de uma garrafa já enchida.
COMPOSIÇÃO	ED_pH_nominal + ED_pH_corrente

Depósito Interno

NOME	DI1.PR1.2{i}_condição_linha
DEFINIÇÃO	modela a informação interna do processador da linha "i" sobre o estado de ativação da linha sob seu controle. Esta definição substitui a entrada "DI3_condições_linha" existente no dicionário de dados do Modelo da Essência.
COMPOSIÇÃO	ED_condição_linha

Elemento de Dado

NOME ED_flag_área

DEFINIÇÃO modela a condição de funcionamento da área de engarrafa-mento em um dado momento, podendo assumir valores correspondentes às situações de área habilitada ou inabilitada.

Fluxo de Dados Contínuo

NOME FDC_nível

DEFINIÇÃO modela a forma de obtenção do sistema do valor analógico do nível corrente do líquido no tanque.

NOME FCD_nível_digital

DEFINIÇÃO modela a forma de obtenção do processador de área do valor digital do nível corrente do líquido no tanque.

COMPOSIÇÃO ED_nível

NOME FDC_pH_corrente

DEFINIÇÃO modela a forma de obtenção do sistema do valor analógico do pH corrente do líquido no tanque.

NOME FDC_pH_digital

DEFINIÇÃO modela a forma de obtenção do processador de área do valor digital do pH corrente do líquido no tanque.

COMPOSIÇÃO ED_pH_corrente

6. Conclusão

Utilizando o SICLEN como ilustração, procurou-se neste trabalho evidenciar a adequação da estrutura de modelagem apresentada para o exame das possíveis alternativas de implementação de um sistema. Com essa estrutura, pretendeu-se gerar uma alternativa de implementação consistente com os requisitos especificados no Modelo da Essência. Essa abordagem visa conduzir o engenheiro de software por caminhos mais seguros que apresentem, gradativamente, as questões que naturalmente hão de surgir no esforço de construção de um sistema sem que sua atenção seja desviada do objetivo maior que é o atendimento às reais necessidades do cliente/usuário. No caso de sistemas de tempo-real para controle de processos, tal aspecto reveste-se de particular importância, dada a natureza crítica dessa classe de sistemas.

Nessas condições, o objetivo básico do trabalho consistiu em realizar um experimento controlado envolvendo o emprego das ferramentas e técnicas aqui apresentadas para a modelagem da implementação de sistemas de tempo-real para monitorar e controlar processos. Avaliando essa experiência, verificamos que :

- ♦ a existência prévia do Modelo da Essência (ME) completo do sistema, característica específica do experimento realizado e não exigência metodológica, revelou-se extremamente útil para o processo de design, pois esse modelo constituiu um excelente subsídio ao apresentar o enunciado e a solução abstrata do problema de modo claro e sem ambigüidades. Além disso, a geração do Modelo da Implementação (MI), através de reorganização e expansão do ME, foi bastante facilitada na medida que já se podia contar com uma representação conceitual do padrão organizacional a ser automatizado expresso em linguagens de representação utilizadas também no MI. A vantagem dessa uniformidade lingüística ficou bem clara durante o processo de construir os Esquemas de Atividades e os Diagramas de Estados e Transições específicos do design. Além desses aspectos basicamente instrumentais, o emprego da recursividade inerente ao conceito de sistema [23], agora direcionado para as técnicas de modelagem, facilitou a construção e a análise dos Esquemas dos Processadores e do Esquema dos Processos ao permitir analogias pertinentes com o Esquema Transacional do ME. Obviamente, a elaboração da organização hierárquica dos esquemas do MI também beneficiou-se desse tipo de analogia.

Em resumo, apesar de não constituir requisito para o processo de desenvolvimento, o experimento realizado indicou que a existência prévia de um Modelo da Essência completo não é só útil por fornecer uma especificação conceitual exata do sistema e subsidiar a geração do Modelo da Implementação, mas também por contribuir para o processo de aprendizado, por parte do engenheiro de software, referente à aplicação dos conceitos comuns aos modelos facilitando a reutilização dos mesmos.

- ♦ salvo nos casos de sistemas muito simples e/ou restrições que impliquem imponderavelmente uma determinada configuração de processadores e processos, o esforço de modelagem que conduz à geração de um Modelo da Implementação (Design) completo do sistema constitui um recurso indispensável à busca da alternativa de implementação mais adequada ao cliente/usuário. Estabelecendo uma “ponte” de nível intermediário de abstração entre o Modelo da Essência e o Modelo da Automação, favorece o estudo das diversas alternativas. O experimento realizado indicou que o emprego desse modelo confere ao processo de desenvolvimento um aumento em qualidade e confiabilidade pelos seguintes motivos :

a) a estrutura do MI (MCPr, MCP e MCM) favorece o exame dos aspectos de implementação em níveis gradativos de complexidade. Considerando-o, assim como o ME, uma aplicação cuidadosa de um conjunto de critérios de segmentação e/ou abstração destinados a dividir um problema complexo - a automação de um padrão organizacional - em domínios de menor porte/densidade, apresenta-se compatível com a capacidade de apreensão intelectual do ser humano. Cada aspecto básico de implementação (processadores, processos e módulos) é abordado em submodelos distintos, o que propicia a organização do estudo desses aspectos permitindo que o esforço intelectual do modelador seja concentrado no devido momento ao exame de um grupo reduzido de características. A própria estrutura de modelagem garante, sintaticamente, a integração desses submodelos e, com isso, o modelador é capaz de conservar sua atenção, em cada momento específico do processo de modelagem, nos aspectos semânticos que caracterizem o melhor atendimento às necessidades do cliente/usuário.

A estrutura do MI também favorece poderosamente o estudo comparativo entre alternativas, na medida que estabelece uma forma padronizada de apresentação das configurações idealizadas pelo modelador. Dispondo desse padrão, a tarefa de escolher a alternativa mais adequada simplifica-se, pois pode ser efetuada a nível de processadores, processos e módulos. Exemplificando, percebe-se que o MCPr já permite uma avaliação comparativa de custos e que uma estimativa nele baseada referir-se-á provavelmente aos elementos mais dispendiosos da implementação.

b) o MI permite investigar teoricamente o comportamento do sistema referente às diferentes alternativas de design, reforçando sua utilidade para efeito do estudo comparativo acima mencionado;

c) pelo fato de empregar vários conceitos, ferramentas e técnicas relacionadas ao ME, o MI apresenta-se, após a aquisição de alguma experiência por parte do modelador, fácil de construir e utilizar. Pode ser considerado uma aplicação ampliada, direcionada para aspectos concretos de implementação, do conhecimento necessário para a elaboração do ME;

d) o emprego de ferramentas conceituais gráficas no MI confere ao mesmo elevado poder semântico sem acréscimo de complexidade à sua compreensão e manuseio;

e) a notação empregada no MI, em particular a notação de rastreamento, confere ao modelo a capacidade de mapeamento necessária entre os submodelos que o compõe e também em relação ao ME;

f) a documentação decorrente do MI é suficientemente rigorosa para definir todos os aspectos de implementação sem ambigüidade (informação contínua, processamento contínuo, funções de controle, processamento paralelo etc.) e suficientemente flexível para registrar metodicamente as decisões que tenham direcionada a escolha de uma determinada alternativa de implementação. Isto facilita o processo de transmissão de mensagens entre os grupos de pessoas (grupo de análise / grupo de projeto (design) / grupo de implementação) [23] envolvidos no processo de construção e, principalmente, no processo de manutenção do sistema construído.

O experimento realizado indicou que o Modelo da Implementação constitui um recurso no mínimo desejável à condução de um processo de desenvolvimento mais seguro para o implementador, pois seu objetivo é justamente reduzir a dificuldade e a incerteza que podem ser encontrados no desnível de abstração que existe entre o Modelo da Essência e o Modelo da

Automação [23]. Tal aspecto é proporcionalmente relevante em função do porte e da complexidade intrínseca do problema a ser resolvido e representa uma preocupação importantíssima em automação industrial.

Contudo, poder-se-ia ainda questionar se a elaboração de um Modelo da Implementação completo, nos moldes apresentados neste trabalho, não aumentaria demasiadamente o custo e o tempo de construção do sistema. Não há dúvida de que qualidade possui um preço e este pode não estar ao alcance de um dado projeto de desenvolvimento. Nesse sentido, a conclusão que é possível extrair do experimento realizado indica que a omissão completa dessa etapa do processo de construção acarreta uma elevação de risco considerável para o projeto e que a gerência de desenvolvimento deveria, contextualmente, avaliar a possibilidade de sacrificar apenas parcialmente a elaboração do MI. Evidentemente, a decisão quanto a quais partes seriam omitidas releva de uma análise do tipo de sistema específico que o projeto incorpora. Possivelmente, este trabalho representa uma contribuição relevante para esse tipo de análise.

Finalizando, é possível concluir que o experimento conduzido no caso do SICLEN indicou que as ferramentas conceituais e técnicas estruturadas aqui apresentadas permitem :

⇒ elevado poder de análise sobre as alternativas de implementação que deverão incorporar os requisitos essenciais especificados no ME, reduzindo a possibilidade de ocorrência de erros na fase de implementação e, conseqüentemente, favorecendo a geração de produtos de qualidade que atendam às necessidades do cliente/usuário; e

⇒ a geração de uma “planta” que constitui um guia seguro para a construção do Modelo da Automação [23], que incorporará o código-fonte da porção do sistema alocada a computadores digitais e, se porventura existir uma porção alocada a processadores humanos, a descrição dos procedimentos manuais a serem executados.

Referências Bibliográficas

- [1] Boehm B. W., McLean R. L., e Urfig D. B., "Some experiences with automated aids to the design of reliable software", IEEE Transactions on Software Engineering, vol. SE-1, nr. 2, fevereiro 1975.
- [2] Endres A., "An analysis of errors and their causes in systems programs", IEEE Transactions on Software Engineering, vol. SE-1, nr. 6, pp.170-179, junho 1975.
- [3] Boehm B. W., "Software Engineering Economics", Englewood Cliffs, NJ, Prentice-Hall, 1981
- [4] Van Griethuysen J. J., (editor), "Concepts and Terminology for the Conceptual Schema and Information Base", International Standartization Organization (ISO), ANSI, Secretariat ISO/TC97/SC5, New York, 1982, Publication Number ISO/TC97/SC5-N695.
- [5] McMenamin S. M. e Palmer J. F., "Essential Systems Analysis", Yourdon Press, 1984.
- [6] Newman P. G., "Some computer-related disasters and other egregious horrors", ACM Software Engineering Notes, vol. 10, nr. 1, pp. 6-7, janeiro 1985.
- [7] Ward P. T. e Mellor S. J., "Structured Development for Real-Time Systems, vol. 1, Yourdon Press, 1985.
- [8] Ward P. T. e Mellor S. J., "Structured Development for Real-Time Systems, vol. 2, Yourdon Press, 1985
- [9] Ward P. T. e Mellor S. J., "Structured Development for Real-Time Systems, vol. 3, Yourdon Press, 1985
- [10] Fairley R., "Software Engineering Concepts", McGraw-Hill, 1985.
- [11] Levenson N. G., "Software safety : what, why , and how", ACM Computer Surveys, vol. 18, nr. 2, pp. 125-174, junho 1986.
- [12] Ward P. T., "Systems Development Without Pain", Prentice-Hall, 1986.
- [13] Boehm B. W., "Industrial Software Metrics Top Ten List", IEEE Software, setembro 1987.

- [14] Bruyn W., Jensen R., Keskar D. e Ward P. T., "ESML : An Extended Systems Modeling Language Based on the Data Flow Diagram", ACM SIGSOFT Software Engineering Notes, vol. 13, nr. 1, pg. 58, 1988.
- [15] Yourdon E., "Modern Structured Analysis", Prentice-Hall, 1989.
- [16] Maffeo B. e Ritto A. C. A., "O Esquema Semântico no Modelo do Contexto de um Sistema Computacional", XXIII Congresso de Informática da SUCEU, 1990.
- [17] Lott C. M., "Correctness is Congruent with Quality", ACM SIGSOFT Software Engineering Notes, vol. 15, nr. 5, outubro 1990.
- [18] Shaw M., "Prospects for an Engineering Discipline of Software", IEEE Software, pg. 15, novembro 1990.
- [19] Levenson N. G., "The Challenge of Building Process-Control Software", IEEE Software, pg. 55, novembro 1990.
- [20] Maffeo B., "ESML (Extended Systems Modeling language) : Uma revisão da Apresentação, Estrutura, Notação e Conteúdo", Monografias em Ciência da Computação do Departamento de Informática da PUC/Rio, 72 páginas, 1/1991.
- [21] Ritto A.C.A. e Maffeo B., "Definindo o Problema a ser tratado por um Sistema Computacional - O Modelo do Contexto", Monografias em Ciência da Computação do Departamento de Informática da PUC/Rio, 101 páginas, 11/1991.
- [22] Jaffe M. S., Levenson N. G., Heimdahl M. P. E. e Melhart B. E., "Software Requirements Analysis for Real-Time Process-Control Systems", IEEE Transactions on Software Engineering, vol. 17, nr. 3, pg. 241, março 1991.
- [23] Maffeo B., "Engenharia de Software e Especificação de Sistemas", Editora Campus, 1992.
- [24] France R. B., "Semantically Extended Data Flow Diagrams : A Formal Specification Tool", IEEE Transactions on Software Engineering, vol. 18, nr. 4, abril 1992.
- [25] Clemente K., "Modelagem de Sistemas Sócio-Técnicos : Estudo de Caso de um Piloto Automático para Automóvel", Dissertação de Mestrado, Departamento de Engenharia Elétrica, PUC/Rio, abril 1992.
- [26] Barbosa E. e Maffeo B., "Modelagem da Essência de um Sistema de Controle para uma Área de Linhas de Engarrafamento", Monografias em Ciência da Computação do Departamento de Informática da PUC/Rio, 104 páginas, 29/1992.

- [27] Richter G. e Maffeo B., "Towards a Rigorous Interpretation of ESML - Extended Systems Modelling Language", IEEE Transactions on Software Engineering, vol. 19, nr. 2, pg. 165, fevereiro 1993.
- [28] d'Almeida Sanchez M. L., Maffeo B. e do Prado Leite J. C. S., "Ferramentas e Técnicas para a Modelagem da Essência de Sistemas de Tempo-Real para Controle e Monitoramento de Processos", aceito para o X Congresso Brasileiro de Automática e VI Congresso Latino-Americano de Controle Automático.
- [29] Page-Jones, M., "The Practical Guide to Structured Systems Design", McGraw-Hill, 1988.

APÊNDICE

Método para Segmentação e Agregação de Atividades de Controle

O primeiro estágio na elaboração do Modelo da Implementação consiste em identificar o conjunto de processadores que poderão realizar o trabalho especificado no Modelo da Essência, sendo possível a partir desse ponto examinar a organização mais adequada e verificar como será efetuada a alocação das atividades a esses equipamentos. O modelo resultante (Modelo da Configuração de Processadores - MCPr) possivelmente possuirá um particionamento diferente, mas deverá alocar o mesmo conteúdo e prever o mesmo comportamento estabelecido no Modelo da Essência.

Assim, pela reorganização dos elementos essenciais e acréscimo de requisitos não-essenciais, é provável que ocorra :

- alocação de diferentes atividades a um mesmo processador;
- alocação de partes (segmentos) de atividades a processadores/processos diferentes;
- alocação de partes de atividades diferentes a um mesmo processador/processo.

Em se tratando de atividades de controle, o emprego de Diagramas de Estados e Transições (DET) visa gerar uma especificação que constitua a descrição detalhada de uma dinâmica complexa normalmente associada a esse tipo de atividade, diferentemente das atividades operacionais que são especificadas, neste trabalho, por Listas de Pré/Pós-Condições. Para atividades operacionais, não apresenta maiores dificuldades o processo de obtenção, a partir das especificações contidas no ME, das novas especificações correspondentes a atividades reorganizadas. Porém, no caso da reorganização de atividades de controle, essa tarefa requer maior cuidado e a modelagem realizada neste trabalho demonstrou que a dinâmica associada às atividades reorganizadas acrescenta, em geral, complexidade à tarefa de reespecificação.

Deve ser considerado, também, que enquanto no MCPr a descentralização do controle é possível por ser admitida concorrência estrita entre diversos processadores (conseqüentemente, o MCPr pode possuir um grau menor de distorção em relação à essência), no Modelo da Configuração de Processos não é possível admitir concorrência estrita, devendo o controle ser centralizado. A alocação de atividades de controle envolverá modificações para combinar, se for o caso, o comportamento controlador de duas ou mais atividade (ou partes delas) alocadas a um mesmo processo causando agregação e/ou segmentação dessas atividades.

Neste apêndice, serão apresentados os métodos propostos em [9] para segmentação/agregação de atividades de controle. Em que pese na modelagem da implementação do SICLEN não ter sido necessário empregá-los totalmente, demonstraremos onde foram utilizados e onde foi possível utilizar outro procedimento, sendo este último também abordado.

A1. Segmentação de Atividades de Controle

Ward e Mellor [9] propõem o seguinte método para segmentação do DET de uma atividade de controle (“*control transformation*”):

Passo 1. Reproduza o DET (uma cópia) da atividade que será segmentada para cada processador/processo que incorporará um segmento.

Passo 2. Identifique as precondições (“*condition*”) e poscondições (“*actions*”) para cada transição na cópia do DET de cada processador e selecione, para cada cópia, a situação apropriada dentre as que se seguem:

2.1 precondição reconhecida e poscondição executada pelo mesmo processador => adicione à respectiva cópia do DET uma poscondição para sinalizar que a precondição mencionada foi reconhecida;

2.2 precondição reconhecida pelo processador e poscondição executada por outro processador => substitua a poscondição pela indicação de um fluxo de controle que sinalize o reconhecimento da precondição;

2.3 precondição reconhecida por outro processador e poscondição executada pelo processador em exame => substitua a precondição pela indicação de um fluxo de controle que sinalize a ocorrência da precondição pelo outro processador;

2.4 precondição reconhecida e poscondição executada por outro processador => substitua a precondição pela indicação de um fluxo de controle que sinalize a ocorrência da precondição no outro processador e remova a poscondição.

Passo 3. Verifique cada estado em cada cópia de DET de acordo com os seguintes critérios :

3.1 todas as transições que saem do estado possuem precondições que são fluxos de controle de outro processador e não poscondições;

3.2 todas as transições que saem do estado estão direcionada a um único estado destino, cujas transições de saída por sua vez possuem todas precondições que são fluxos de controle de outro processador.

Caso o estado se enquadre em ambas as situações acima, faça :

3.3 remova o estado e suas transições de saída;

3.4 reorienta as transições de entrada para o estado destino mencionado em 3.2.

Repita este passo até que todos os estados naquelas condições sejam removidos.

Passo 4. Remova todas as poscondições que são fluxos de controle sem estado destino.

Passo 5. Se for o caso, renomeie os estados para dar maior clareza aos diagramas.

O método acima parece ser particularmente útil quando ocorre um significativo grau de distorção na modelagem da implementação. No caso do SICLEN, foi possível empregar uma

simplificação desse procedimento no MCP_r pois a alternativa escolhida apresentava pequena distorção. Foi bastante, nessas condições, efetuar a alocação das atividades de controle aos processadores e transferir, para cada DET associado, as precondições e poscondições relativas aos estímulos incidentes.

A2. Agregação de Atividades de Controle

Ward e Mellor [9] também propõem o método, descrito a seguir, para efetuar a combinação dos DET de duas atividades de controle relacionadas por subordinação - isto é, uma delas habilita/inabilita a outra. Para efeito de melhor compreensão, denominaremos de D1 o DET da atividade principal e D2 o DET da atividade subordinada.

Passo 1. Adicione um *estado indefinido* (provisório) a D2. Considere esse estado como um estado inicial e crie uma transição dele para o estado inicial original com uma precondição habilitadora (“(H) ...”). Adicione transições de todos os outros estados para o estado indefinido com precondições inabilitadoras (“(I) ...”) e poscondições que inabilitem todas as atividade operacionais presentes nesse DET.

Passo 2. Inicie um novo DET (que denominaremos a partir desse instante de D3) criando um estado inicial que combine o estado inicial original de D1 e o estado indefinido adicionado a D2. Esse estado inicial de D3 será referenciado por *estado combinado*.

Passo 3. Para cada precondição externa a D1 e D2 que cause uma transição do estado inicial original de D1 e/ou do estado indefinido adicionado a D2 que ainda não esteja presente em D3, construa uma transição do estado combinado e reproduza a precondição e a poscondição da transição original. Substitua todas as poscondições que emitem um sinal (*event flow*) para D1 ou D2 por poscondições disparadas (“(D) ...”) por esse sinal.

Passo 4. Se a precondição externa mencionada no passo anterior causar transições em D1 e D2, construa em D3 um novo estado combinado compondo os estados destinatários em D1 e D2 se esse estado for diferente de todos os estados já criados, e inclua a transição que contém a precondição externa para esse novo estado combinado. Se a precondição externa mencionada no passo anterior não causar transições em D1 e D2, construa em D3 um novo estado combinado a partir do diagrama original que contém a transição e o estado em exame corrente do outro diagrama se esse estado for diferente de todos os estados já criados, e inclua a transição para esse novo estado. Se o estado destinatário da transição que contém a precondição for um estado combinado criado em um passo anterior, conecte a ele a transição.

Passo 5. Repita os passos 3 e 4 para cada novo estado combinado no passo 4.

Este foi o método empregado na elaboração do Modelo da Configuração dos Processos do SICLEN, no qual efetuou-se a agregação de três atividades de controle alocadas a um mesmo processo. Nessa situação, em que uma atividade possuía duas outras subordinadas, realizou-se inicialmente a agregação da principal com uma subordinada, obtendo-se um primeiro resultado. Em seguida, foi feita a combinação desse resultado com a última atividade subordinada.