

**SCOM-X25: uma experiência com metodologias de desenvolvimento de software
na implementação de um protocolo de comunicação**

Ana Maria Ambrosio
Flávio de Freitas Barbosa

Instituto Nacional de Pesquisas Espaciais - INPE
CxP 515 - 12201 São José dos Campos - SP
E-mail: GAN%CCS@fosp.fapesp.br

RESUMO

Este artigo tem como objetivo apresentar a experiência com a aplicação de uma metodologia de desenvolvimento de **software** aplicada ao projeto (**design**) para implementação de um protocolo, no caso, o X.25. A metodologia aplicada envolveu conceitos de análise orientada a objetos adaptados a uma metodologia para desenvolvimento estruturado de sistemas de tempo real.

O protocolo X.25 foi implementado como parte do projeto SCOM-X.25, que consta de uma placa de comunicação de dados para conexão a uma rede de longa distância através de uma solução não proprietária.

O uso de uma metodologia de desenvolvimento para implementação do protocolo teve como objetivo a qualidade do produto. Um modelo da arquitetura de implementação do protocolo facilitou o cumprimento dos requisitos funcionais e principalmente dos não funcionais, como manutenibilidade, gerenciamento e controle nas fases de codificação e validação.

1- INTRODUÇÃO

O sistema SCOM-X25 foi idealizado para suprir necessidades de comunicação entre equipamentos das Estações Terrenas de Rastreo e o Centro de Controle de Satélites desenvolvidos e operados pelo INPE. Como primeira aplicação, a Missão Espacial Completa Brasileira (MECB) prevê a utilização deste projeto com o Conjunto de Medida de Velocidade (CMV) da Estação Terrena localizada em Cuiabá. Este Conjunto mede a velocidade do satélite através do efeito doppler. As solicitações de medidas partem do Centro de Controle localizado em São José dos Campos, onde são utilizadas para cálculo de previsão de órbita. A transmissão e recepção destes dados utiliza a Rede de Comunicação de Dados particular do INPE - RECDAS (REde de Comunicação de DAdoS).

Dado que o projeto de um sistema de comunicação funciona como um meio e não como um fim, uma série de generalizações foram feitas para tornar o projeto o mais aberto possível e não específico a esta aplicação.

Assim, o SCOM-X25 possibilita o acesso, não só ao CMV, mas a quaisquer equipamentos tipo IBM-PC a redes públicas que operem segundo o padrão X.25.

Esta solução de comunicação adotada através do SCOM-X25, ao contrário das soluções comerciais da época, conta com a execução do protocolo dentro da própria placa.

O **software** que implementa o protocolo X.25, denominado SX25 é executado diretamente sobre o processador da placa do SCOM-X25, permitindo ao equipamento que se ligar a ele ficar livre para seu processamento final, sem concorrer com a execução do protocolo e seus afins. Esta característica é importante para aplicações específicas como é o caso do **CMV**.

Para implementação deste protocolo, foram estabelecidos os pontos livres deixados pela norma [1] através de autômatos de estados finitos [2].

As normas e os autômatos especificam o protocolo. Com estes elementos pode-se considerar o protocolo a ser desenvolvido como um sistema de tempo real, com requisitos de alta performance e neste caso, deve rodar sobre uma placa desprovida de sistema operacional ou qualquer outro suporte básico de **software**.

Com os princípios de tempo real e uma especificação através de autômato de estados finito, partiu-se para a aplicação da metodologia de desenvolvimento de sistemas de tempo real de Ward/Mellor [4], a qual orienta a decomposição funcional do sistema.

Porém, para o cumprimento dos requisitos não funcionais, principalmente o de manutenibilidade e o de extensibilidade do sistema, introduziu-se à metodologia anteriormente citada, os conceitos de análise [5] e projeto [6] orientados a objetos.

2- O PROJETO SCOM-X25

O projeto SCOM-X25 é constituído de duas partes: o **hardware** - a placa PX25 (Placa de Comunicação X.25) e o **software** - SX25 (Software de Comunicação de dados para Protocolo X.25). Uma área de memória, Memória Compartilhada (MC), faz a interface entre o SCOM-X25 e o equipamento usuário, como mostra a figura 1.

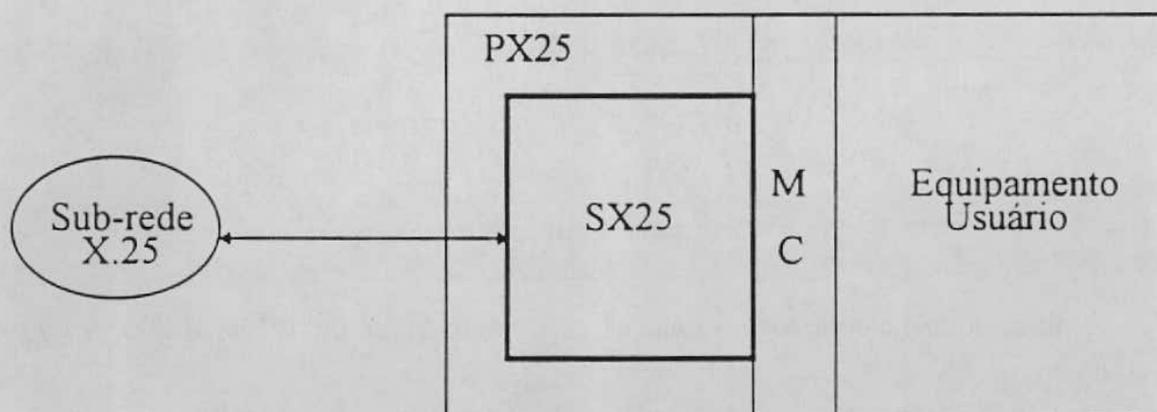


Fig 1: O Projeto SCOM-X25

2.1 - A PLACA PX25

A placa PX25 tem por objetivo possibilitar o acesso de equipamentos que utilizam barramento padrão IBM-PC a redes tipo WAN (**Wide Area Network**). O padrão IBM-PC para barramento (ISA) permite a compatibilidade com um elevado número de aplicações.

A figura 2 apresenta o diagrama de blocos da PX25, cujo objetivo é mostrar sua versatilidade para diversas aplicações em comunicação de dados.

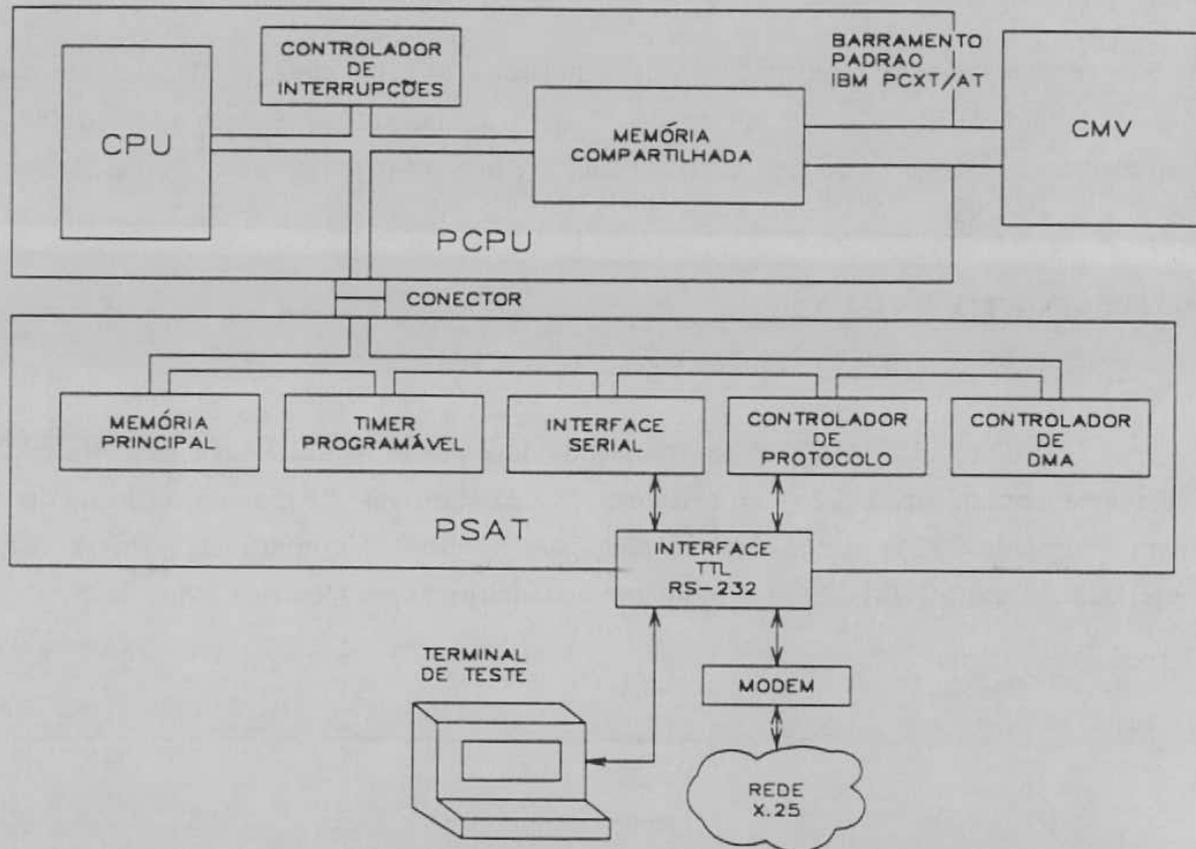


Fig 2: Diagrama de Blocos da PX25

A placa suporta conexões a enlaces com velocidades de 1.200, 4.800, 9.600 e 19.200 bps.

A comunicação da PX25 com o equipamento usuário, ao qual deve ser conectada, é feita através de memória compartilhada, sendo parte reservada a transmissão e parte reservada à recepção de dados do usuário, acionadas através de um mecanismo de interrupção.

Interfaces programáveis realizam funções de temporização básica, de comunicação serial com um console (para a fase de configuração e de teste do sistema), de controle de interrupções, de controle de acesso direto à memória (DMA) e principalmente de comunicação serial com a linha síncrona.

A memória consiste de 256 Kbytes de RAM e de 256 KBytes de ROM, mais 64 KBytes para troca de mensagens com o usuário, sendo que cada byte equivale a um octeto.

2.2 - O SOFTWARE SX25

O **software** de comunicação para protocolo X.25 (SX25) tem como função implementar o protocolo X.25 da CCITT (**International Telegraph and Telephone Consultative Committee**). Ele permite a aplicativos residentes em equipamentos ligados à PX25 se comunicarem com aplicativos remotos através de uma rede de longa distância.

Os principais requisitos não funcionais a serem cumpridos são a portabilidade, a confiabilidade e principalmente a facilidade para manutenção e adaptação a novas aplicações (extensibilidade). Além dos requisitos específicos de se cumprir as normas CCITT do protocolo X.25, existe também o requisito de desempenho de atender até dez canais lógicos simultaneamente.

Para facilitar o cumprimento de seus objetivos, o sistema foi dividido em quatro componentes. A figura 3 ilustra os componentes dispostos hierarquicamente e suas interfaces.

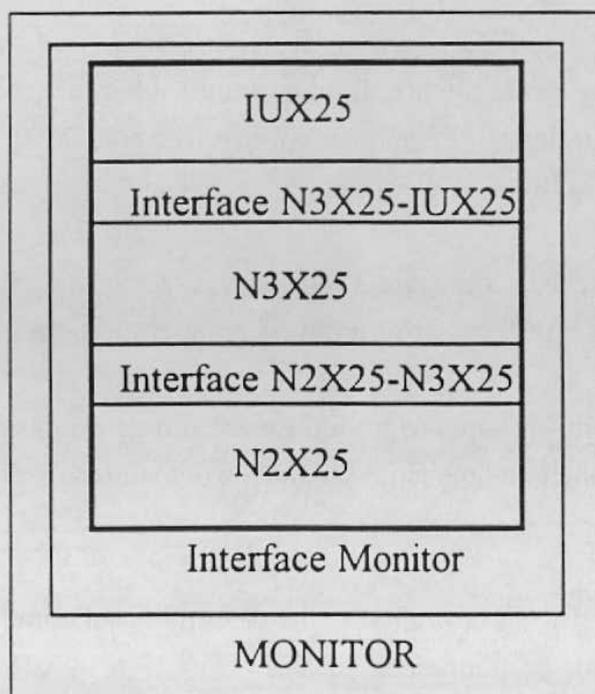


Fig 3: Componentes do SX25 e suas interfaces

Os componentes são descritos a seguir:

MONITOR: este componente é composto por um conjunto de primitivas destinadas ao tratamento básico do ambiente físico de **hardware**. Ele permite aos demais componentes tornarem-se independentes da plataforma sobre a qual o protocolo X.25 for executado;

N2X25: implementa as funções do nível de enlace do protocolo X.25. Utiliza primitivas do MONITOR e fornece um conjunto de primitivas ao nível 3;

N3X25: implementa as funções do nível de rede do protocolo X.25. Utiliza primitivas do MONITOR, do N2X25 e fornece um conjunto delas ao componente IUX25;

IUX25: possui a responsabilidade de implementar a comunicação com o equipamento usuário. Utiliza as primitivas fornecidas pelo N3X25. Gera e interpreta mensagens trocadas com o equipamento usuário, realizando algumas funções de um protocolo de transporte.

As interfaces entre os vários componentes, representadas na figura 3, são definidas por conjuntos de primitivas. A comunicação entre um componente e outro é feita exclusivamente através destes conjuntos de primitivas.

O componente MONITOR, como descrito acima, pode ser visto como uma interface entre o software e o hardware. O objetivo de sua concepção é o de satisfazer o requisito de portabilidade desta implementação do protocolo X.25. Ele é composto por dois sub-conjuntos de primitivas:

- 1) primitivas para realização das tarefas primárias para utilização dos recursos de hardware, como as interfaces programáveis e a memória compartilhada;
- 2) primitivas destinadas ao tratamento lógico de estruturas de dados das interfaces e da memória compartilhada e que serão chamadas pelos demais componentes.

A tabela 1 mostra as primitivas do segundo subconjunto do componente MONITOR oferecidas aos componentes N2X25, N3X25 e IUX25. Estas primitivas são classificadas de acordo com sua aplicação.

Tab 1: Primitivas do componente MONITOR
(Interface Monitor)

Comunicação com a rede	
	Frame_Tx Frame_Rx Frame_Size
Controle de temporizações	
	Set_Timer_Num Timer_Set Stop_Timer Start_Timer Status_TO
Controle de velocidade de transmissão	
	Set_Speed_N1
Gerenciamento de memória	
	Set_Buffer Buffer_Alloc Status_Heap Free_Buffer
Comunicação com a memória compartilhada	
	Put_Block Get_Block Block_Size

Os componentes N2X25 e N3X25 controlam todas as questões referentes ao estabelecimento e gerenciamento de enlaces do nível 2 e de circuitos virtuais do nível 3, possibilitando a transferência de informações entre a placa PX25 e algum DTE ou DCE remoto.

As tabelas 2 e 3, a seguir, apresentam o conjunto de primitivas oferecidas pelo nível 2 e pelo nível 3, respectivamente.

Tab 2: Primitivas do componente N2X25
(Interface N2X25-N3X25)

Transferência de Dados	
	L-Info-Req L-Info-Ind
Estado Geral do Enlace	
	L_Reset_Stat L_Link_Stat

Tab 3: Primitivas do componente N3X25
(Interface N3X25-IUX25)

Pacotes INTERRUPT	
	N-Interrupt-Req N-Interrupt-Ind N-Interrupt-Conf
Pacotes CLEAR	
	N-Clear-Req N-Clear-Ind N-Clear-Conf N-Clear-Stat
Pacotes DATA	
	N-Data-Req N-Data-Ind
Pacotes CALL	
	N-Call-Req N-Inc-Call N-Call-Conf N-Bit-D
...	
Estado do Canal Lógico	
	N_LC_Stat
Estado do Enlace	
	N_Link_Stat

Como funções relativas à implementação do protocolo em si, foram requisitados dois subcomponentes, a saber:

CFGX25 - este subcomponente deve configurar parâmetros sujeitos a alterações de acordo com a aplicação, tais como, velocidade de transmissão do meio físico, valores limites dos temporizadores, janelas de transmissão, entre outros;

NUC-X25 - este subcomponente deve executar serviços básicos de manipulação de PDUs (**Protocol Data Units**), tais como, montagem e desmontagem de quadros e pacotes.

Para cada um destes subcomponentes um conjunto de primitivas foi estabelecido. Estas primitivas são apresentadas na tabela 4, a seguir.

Tab 4: Primitivas de serviços básicos do protocolo.

<p>Serviços de Quadros</p> <p>K_Frm_Asm_Info K_Frm_Asm_Sup K_Frm_Asm_Unum K_Frm_Dism</p>
<p>Serviços de Pacotes</p> <p>K_Pac_Asm_Con K_Pac_Asm_Req K_Pac_Asm_Conf K_Pac_Asm_Data K_Pac_Asm_Flow K_Pac_Dism</p>
<p>Configuração</p> <p>Set_Net_Speed Set_N2_Window Set_N2_Max_Timer_Counter Set_T1_Limit Set_N3_Window Set_Max_Packed_Size Set_T20_Limit Set_T21_Limit Set_T2n_Limit Set_LC_Numbers</p>

O componente IUX25 tem como uma de suas principais funções, servir de interface entre os componentes que implementam o protocolo (N2X25 e N3X25) e o **software** do equipamento usuário.

Desta maneira, cabe a este componente regulamentar o uso, por parte do equipamento usuário, dos recursos de protocolo que lhe são oferecidos. Ele permite ao equipamento usuário manipular os dados na forma de mensagens; para tanto fornece um serviço de segmentação e concatenação. Controla a comunicação na memória compartilhada, efetuando consistência de valores e controle de fluxo das mensagens.

3 - METODOLOGIA DE DESENVOLVIMENTO NA FASE DE PROJETO

Toda definição de componentes e primitivas (de interface), faz parte da especificação de requisitos do sistema a ser desenvolvido, onde os componentes são dispostos hierarquicamente, como ilustrado na figura 3.

O maior desafio para a fase de projeto foi então, atender a uma especificação que já determinava componentes e primitivas, mas por outro lado, requisitava um **design** com encapsulamento de dados, processos independentes e alto desempenho. Além disso, como fator de qualidade, a obrigação de se obedecer a uma metodologia nesta fase do desenvolvimento do software foi mais um requisito.

Duas visões foram aplicadas na concepção do **design** do SX25. Uma visão algorítmica (ou procedural), estável, uma vez que implementa as regras do protocolo (já especificadas).

Outra visão de objetos, menos estável do ponto de vista de implementação, pois encapsula uma estrutura de dados implementada de acordo com os recursos de memória disponível e a linguagem de programação escolhida.

Adotou-se, inicialmente, a metodologia para desenvolvimento estruturado para sistema de tempo real de Ward/Mellor [4]. Trata-se de uma metodologia não específica para o desenvolvimento de um protocolo de comunicação.

Esta metodologia permite modelar o sistema a ser desenvolvido como uma rede de atividades que aceitam e produzem mensagens de dado e de controle, denominada

ESQUEMA DE TRANSFORMAÇÃO. Este esquema é baseado na notação para Diagramas de Fluxo de Dados proposta por DeMarco [8] com extensões que aprimoram a representação dos aspectos de tempo real. As extensões incluem fluxos de dados de tempo-contínuo, fluxos de eventos, depósitos de eventos, transformações de controle, entre outros.

Cabe observar que uma *transformação de controle* tem seu comportamento definido através de um diagrama de transição de estado. Como a especificação do X.25 encontra-se sob a forma de um diagrama de transição de estado, muitas características desta metodologia foram aplicadas.

Dela, foram adotados os esquemas de transformação que representam, através de níveis, visões mais detalhadas do sistema, com o propósito de se obter uma decomposição funcional em módulos ou entidades executáveis.

Do aspecto de análise orientada a objetos, um dos princípios essenciais para a administração da complexidade é o encapsulamento da informação. Então, como iniciativa particular, adotou-se para organização dos dados um tratamento que entrelaça à metodologia de Ward/Mellor os conceitos de ATRIBUTOS e SERVIÇOS definidos por Coad e Yourdon [5].

Um *atributo* é definido em [5] como sendo um dado para o qual cada objeto tem seu próprio valor, já, *serviço* é um comportamento específico que um objeto deve exibir.

Com estes conceitos um outro tipo de entidade foi criada e introduzida nos diagramas de transformação, o COMPONENTE DE INTERFACE.

O componente de interface é composto por um conjunto de dados, chamados ATRIBUTOS mais um conjunto de funções exercidas sobre os atributos, denominadas SERVIÇOS. Todo atributo deste conjunto pode ser acessado única e exclusivamente pelos serviços pertencentes ao Componente de Interface a que ele pertence.

O *componente de interface* pode ser visto, por um lado, como uma transformação no sentido em que pode ser representado através de níveis de detalhamento. Desta forma, deu-se para os dados (encapsulados no componente de interface) a mesma forma de tratamento dado às transformações (representação das funções). Para cada conjunto de dados coesos pode haver uma decomposição em níveis de detalhamento.

Por outro lado, o componente de interface pode ser visto como um depósito (STORE - definido por Ward/Mellor), pois guarda e oferece as informações necessárias às transformações do sistema.

Os componentes de interface foram identificados de forma a manterem um comportamento bem definido, poderem ser unicamente identificados e a qualquer momento pode-se identificar um estado, o que segundo Booch [7] define uma classe.

No entanto, independente de sua classificação, o componente de interface, na visão de objetos, teve sua aplicação no mapeamento dos conjuntos de primitivas definidas na fase de especificação do protocolo em SERVIÇOS pertencentes a uma entidade de implementação.

Os componentes que enfatizam uma sequência de ações ou procedimentos e temporizações regidos pelo protocolo de comunicação são chamados Componentes Funcionais, os quais representam a visão algorítmica.

Na representação gráfica denominam-se Componentes Funcionais, as transformações (retângulos com bordas arredondadas) e Componentes de Interfaces, os depósitos (STORES) envolvidos por um retângulo tracejado, como pode ser observado na figura 4 da próxima seção.

Os arcos orientados que aparecem no Esquema de Transformação do SX25, figura 4, denominados F1, F2, F3, F4, etc. representam SERVIÇOS pertencentes a um Componente de Interface. Os serviços que são utilizados pelos Componentes Funcionais ligados ao Componente de Interface fazem parte da descrição dos arcos (ou fluxos), detalhados no Documento de Projeto Preliminar [9].

4 - APLICAÇÃO DA METODOLOGIA AO PROJETO DO SX25

Como resultado dos conceitos e da metodologia adotada, conforme descrita na seção anterior, é apresentado na figura 4 o ESQUEMA DE TRANSFORMAÇÃO resultante da decomposição do SX25.

Os componentes funcionais ETN2 e ETN3 implementam funcionalmente os componentes N2X25 e N3X25 descritos no item 2.2. Cada um deles esconde atrás de si um diagrama de transição de estados que determina a sequência de ações nos níveis 2 e 3 para cada evento ocorrido de acordo com a especificação [2]. O diagrama de transição de estado é representado através de uma transformação de controle no esquema de transformação resultante da decomposição de cada um dos componentes em questão. (O esquema dos componentes que foram decompostos não é apresentado no escopo deste artigo).

A implementação da interface especificada entre os níveis 2 e 3 (Interface N2X25-N3X25) restringe-se a implementação dos serviços do componente de interface PCT, o qual incorpora todas as primitivas especificadas na tabela 2.

A implementação da interface entre os componentes N3X25 e o componente IUX25 (Interface N3X25-IUX25), definida através do conjunto de primitivas mostrado na tabela 3, é realizada no componente de interface denominado ECV.

As primitivas do MONITOR, apresentadas na tabela 1 (Interface Monitor), foram distribuídas pelos componentes de interface que oferecem serviços que necessitam de acesso ao hardware. Um destes componentes é o TMP que incorporou todos os serviços de controle de temporizações. No TMP varios temporizadores lógicos utilizados pelo protocolo são implementados fisicamente através do **timer** programável residente na placa PX25.

A figura 5 ilustra, de forma esquemática e parcial, o mapeamento das primitivas especificadas nas interfaces para serviços dos *componentes de interface*.

O componente de interface denominado BFR absorveu as primitivas para execução dos serviços básicos do X.25 como montagem e manipulação de quadros e pacotes, apresentados na tabela 4, e incorporou também a manipulação de segmentos (partes de uma mensagem do usuário).

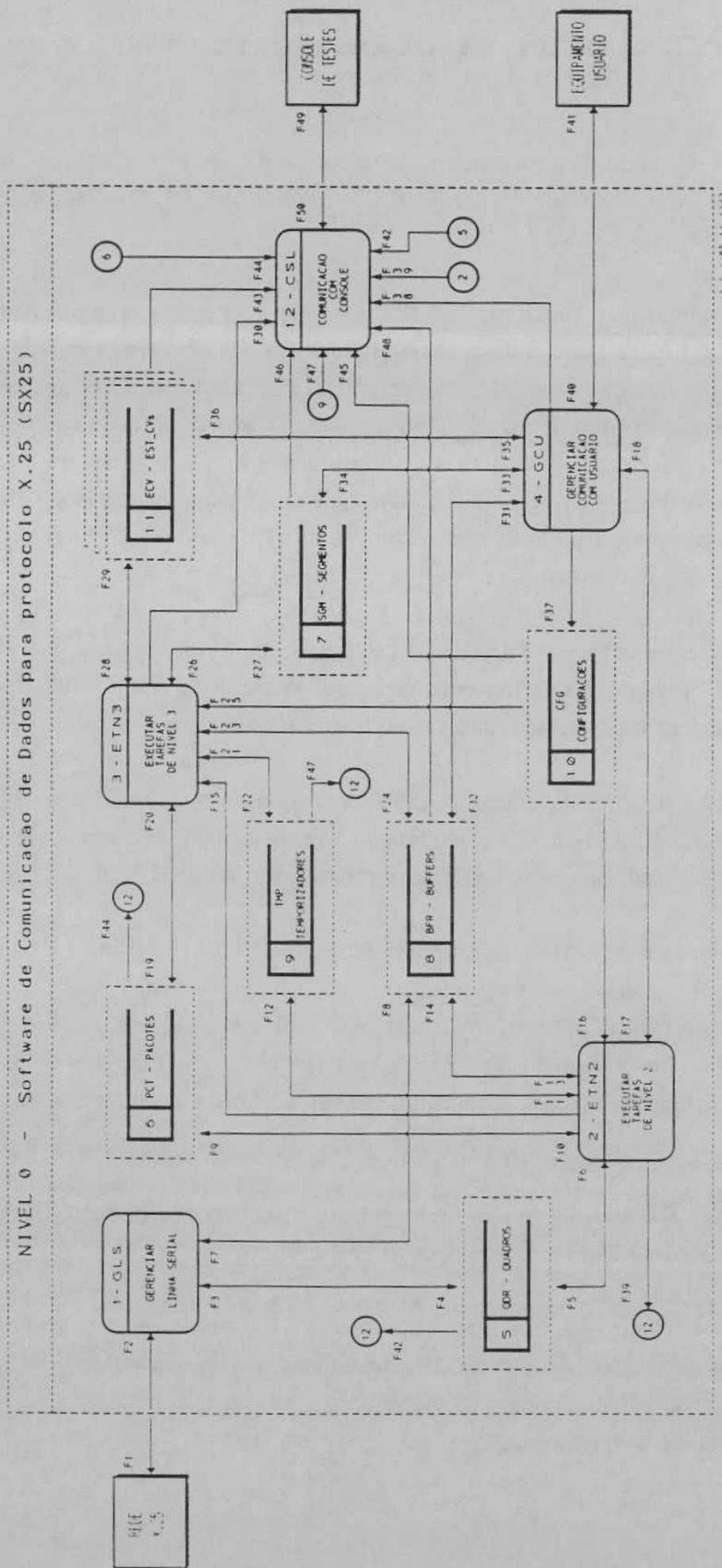


Figura 1: Esquema de Transformacao do SX25

Este componente também encapsula o gerenciamento de memória do sistema, a partir de um **pool de buffers** disponíveis na área de memória interna da PX25, bem como a estrutura de dados que implementa todos os PDUs utilizados pelo X.25 (LPDU = quadros, NPDU = pacotes e TPDU = segmentos).

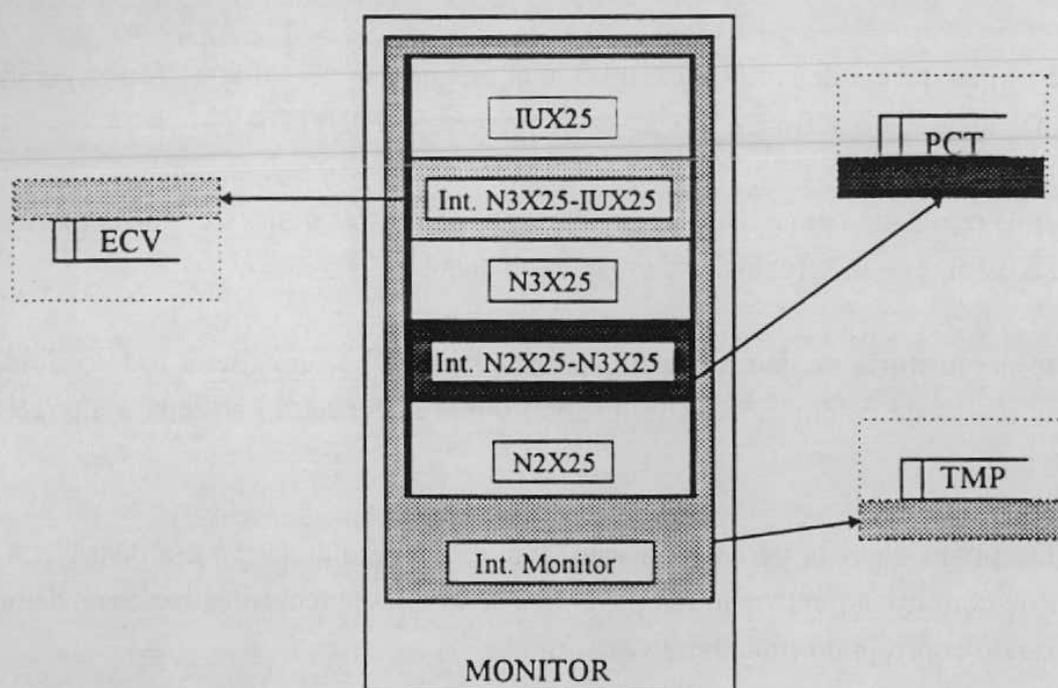


Fig 5: Mapeamento das primitivas especificadas para os componentes de interface

Cabe observar que o conjunto de primitivas especificadas entre os níveis do protocolo não previam as estruturas de dados nem o acesso a elas no sentido de atualização para utilização pela camada superior. Estes detalhes foram definidos e incluídos para a formação de um componente de interface.

Além disso, observa-se que todos os serviços especificados nos documentos de especificação dos requisitos de software [3], foram mapeados nos serviços de um dado componente de interface durante a fase de projeto do sistema. Este mapeamento permitiu uma significativa redução das diferenças entre as fases de especificação, projeto e codificação de um protocolo.

5 - CONCLUSÃO

A aplicação destas diferentes visões (ou metodologias de desenvolvimento de software) na decomposição da complexidade durante a fase de projeto (antes da implementação do protocolo) levou a obtenção das seguintes facilidades:

- 1) alterar a plataforma de hardware com o mínimo impacto no software (conseguindo-se portabilidade do software graças ao componente MONITOR);
- 2) alterar as regras de comunicação sem alterar as estruturas básicas de manipulação dos PDUs (obteve-se o requisito de manutenibilidade);
- 3) alterar as estruturas de dados que implementam os PDUs sem alterar toda codificação das regras de comunicação (designando uma resistência do sistema à alterações e melhorias).

Do ponto de vista gerencial a vantagem desta decomposição é a distribuição do trabalho de codificação entre muitos programadores que podem trabalhar paralelamente, sem perda do controle do trabalho de cada um.

A generalização em cada fase do SCOM-X25 permitiu uma versatilidade de aplicações. Com um suporte de serviços a ser executado em um microcomputador (equipamento usuário), esta placa pode suportar qualquer outro tipo de protocolo de transporte, até mesmo o TCP/IP.

Uma outra aplicação que já está sendo prevista, além do projeto CMV, é a de ter no equipamento usuário um simulador dos equipamentos da estação terrena de rastreamento de satélites para validação e teste do Sistema de Controle de Satélites.

AGRADECIMENTOS

Agradecemos a Osvaldo N. Ito pela contribuição na descrição da PX25 e à Eliane Martins pela revisão e contribuições para a versão final deste artigo. Agradecemos também à Keila Silveira Correia pela contribuição durante seu estágio e ao Marcos de Castro e Silva pelo empenho no trabalho de codificação deste sistema.

REFERÊNCIAS

- [1] CCITT Data Communications Networks Interfaces, Vol VIII.3, (Red Book), Rec X.25, 1984
- [2] MECB/INPE - A-DRS-0013 Documento de Especificação de Requisitos de Software do Componente CX25, 1994
- [3] MECB/INPE - A-DRS-0015 Documento de Especificação de Requisitos do Software de comunicação de dados para protocolo X.25 - SX25, 1994
- [4] Ward, P.T.; Mellor, S.J. Structured Development for Real-Time Systems, Volumes I, II e III, Yourdon Inc, NY, 1985
- [5] Coad, P. ; Yourdon, E. Análise baseada em objetos Ed Campos, tradução da 2a. Ed., 1992
- [6] Coad, P. ; Yourdon, E. Projeto baseado em objetos Ed Campos, 1993
- [7] Booch, G. Object Oriented Design with applications Benjamin/Cummings Publishing Company, Inc. Ca. USA, 1991
- [8] DeMarco, T. Structured Analysis and System Specification. (New York: Yourdon Press, 1978
- [9] MECB/INPE - A-DPS-0038 Documento de Projeto Preliminar do Software de Comunicação de Dados para Protocolo X.25, 1994