

4º CONGRESSO BRASILEIRO DE UTILIZAÇÃO DE RECURSOS ESPACIAIS

DESENVOLVIMENTO DE SOFTWARE PARA O CONTROLE DE MISSÕES ESPACIAIS

Typing for all pages (with the exception of page II) should commence here

LAYOUT PARA A PRIMEIRA PAGINA

LAYOUT FOR THE FIRST PAGE

UMA ESTRUTURA GERAL PARA O "SOFTWARE" DE APLICAÇÃO PARA O CONTROLE DE MISSÕES ESPACIAIS

Décio Castilho Ceballos
Instituto de Pesquisas Espaciais - INPE
Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq
Caixa Postal 515
12200 - São José dos Campos - SP - Brasil

Resumo

Descreve-se uma estrutura de "software" de aplicação com características de modularidade e flexibilidade para o controle de missões espaciais que envolvem satélites artificiais. A flexibilidade resulta do fato de que os sub sistemas podem ser invocados quase independentemente de qualquer sequência. A estrutura modular facilita o desenvolvimento, as adaptações para novas missões e a implementação do "software" acima, com recursos computacionais mínimos. O "software" é classificado quanto à carga computacional em dois grupos. O primeiro, indicado para computador de grande porte, caracteriza-se por utilizar razoável quantidade de memória, precisão de cálculo e requerer processamento esparso. O segundo sugere a utilização de processamento distribuído em mini-computadores, por requerer processamento intensivo e repetitivo, pouca memória, pouca precisão e baixos tempos de resposta. Requerimentos de desempenho e de qualidade e aspectos de implementação são abordados.

A General Structure for the Applications Software in the Control of Space Missions

Abstract

A software structure with modularity and flexibility is presented for artificial satellite space missions control applications. System flexibility results from the fact that subsystems can be invoked almost independently of any sequence. The modular structure facilitates the development, adaptations to other missions as well as minimizes the computational resources needed for implementations. In terms of computational load that software is classified in two groups. The first one has the features of requiring large computers reasonable memory and accuracy capabilities and of being used sparcely. The second suggests the use of distributed processing in minicomputers due to the characteristic of intensive and repetitive processings, short time-response, small memory and low precision. Performance and quality requeriments as well as implementations aspects are described and discussed.

1. INTRODUÇÃO

Para missões espaciais, as operações de lançamento e manutenção em uma dada configuração de atitude e órbita requerem uma quantidade de tal de cálculos e de tomadas de decisões (veja-se por exemplo McCaskill e outros, 1972) que seria impossível realizá-las sem o auxílio de computador. Na literatura técnica disponível não se encontram trabalhos com bom nível de detalhes sobre recursos computacionais, suportes a missões espaciais. Alguns trabalhos (Domb, 1975; McCaskill e outros, 1972; Velez, 1981; Wertz, 1978) fazem considerações pouco detalhadas sobre estruturas computacionais para controle de satélites. Outros (Wertz, 1978; McCaskill e outros, 1972; Wagner and Velez, 1972) descrevem processos gerais envolvidos

em uma missão espacial, ou ainda (Rennels, 1981; Hansen, 1977), aspectos gerais sobre sistemas computacionais.

O objetivo deste trabalho é caracterizar os recursos computacionais para o controle de satélites.

Para isso, é descrita a estrutura geral do "software" de aplicação e feitas considerações sobre a implementação desse "software".

Os recursos computacionais adequados para o controle de satélites podem ser caracterizados por requerimentos. Neste trabalho, adotou-se a seguinte subdivisão desses requerimentos: funcionais, que indicam os objetivos do processamento; de desempenho, que indicam quão bem os requerimentos funcionais devem ser realizados; qualitativos, que indicam caracterís

Não ultrapasse esta linha
Redução de 10 para 8

No text below this line
Reduction 10 to 8

A distribuição de
tudo as páginas
(exceto a 1ª, 2ª e 3ª),
deve começar aqui

Typing for all pages
(with the exception
of page 1) should
commence here

LAYOUT PARA A
PRIMEIRA PÁGINA

LAYOUT FOR
THE FIRST PAGE

ticas de qualidade do sistema computacional; e de contorno à solução que, face aos vínculos existentes, impõem filosofias e orientam o desenvolvimento do projeto. Um requerimento funcional bem definido é designado modo funcional, ou simplesmente modo. Um agrupamento de modos é designado subsistema.

Os requerimentos computacionais variam consideravelmente de uma missão espacial para outra, pois dependem dos objetivos da missão, dos equipamentos no satélite e da filosofia de projeto da trajetória nominal. Ainda assim, por questões de qualificação e custo, o sistema computacional deve possuir elevado grau de generalidade, de modo que, com mínimas adaptações, possa ser utilizado em missões sucessivas, ou mesmo, dar suporte simultâneo a múltiplas missões (Wertz, 1978; Domb, 1975).

Os requerimentos impõem um sistema computacional, que pode ser dividido em sistema de aplicação e sistema de apoio. Considerando-se uma escala de níveis de máquinas virtuais, o sistema de aplicação seria o último nível. Os dados processados por esse sistema provêm ou são utilizados diretamente na aplicação. O sistema de apoio consiste em "hardware", compiladores e sistemas operacionais, os quais simulam máquinas virtuais que dão suporte ao processamento de aplicação. O sistema de aplicação é normalmente desenvolvido para fins exclusivos, enquanto o sistema de apoio o é para múltiplos propósitos, sendo geralmente disponível comercialmente.

O "software" suporte para controle de missões espaciais, quanto às características de interação com o homem, pode ser: totalmente automático, sem nenhuma interação com operadores; totalmente conversacional, operando totalmente dentro da orientação dos operadores; parcialmente automático e conversacional. A melhor opção (Wertz, 1978) é utilizar um sistema que possa operar em qualquer uma dessas alternativas, conforme o estágio da missão.

2. REQUERIMENTOS FUNCIONAIS E ESTRUTURA FUNCIONAL

A Figura 1 apresenta uma estrutura de "software", baseada em aspectos comuns ao sistema de aplicação, com excelentes características de modularidade e flexibilidade. Uma estrutura modular (Wertz, 1978) implica que cada subsistema tem um número mínimo de interfaces. Isto facilita o desenvolvimento, a adaptação para outras missões e a implementação desse "software". A flexibilidade permite que o controle do processamento chame os subsistemas, de forma quase independente da sequência. Por exemplo, um programa de determinação de estado pode ser repetido várias vezes, utilizando-se os dados pré-processados uma única vez. A seguir são descritos, de forma resumida, os vários subsistemas que compõem essa estrutura de "software".

O subsistema de apoio conversacional deve habilitar o operador para o controle da sequência de processamentos, o manuseio de dados, a

correção de erros e a realização de serviços gráficos. Os seus modos funcionais são: modo suporte para o controle de processamento na forma conversacional, modo suporte para a plotagem de gráficos, e modo manuseio de dados.

O objetivo do subsistema de gerenciamento automático é processar dados de saída dos demais subsistemas e dados nominais correspondentes ao planejamento da missão, visando definir as tarefas a serem executadas proxima mente, e sua sequência de prioridades; interagir com o subsistema de apoio conversacional, de modo a compatibilizar o processamento automático e conversacional. Os seus principais modos funcionais são: modo observador do estado da missão; modo rastreador de falhas; modo de definição de prioridades.

A função do banco de dados é armazenar dados dos sujeitos de telemetria, dados de rastreamento, dados pré-processados, dados nominais gerados em simulação, resultados de processamento, constantes físicas, etc, em uma forma independente do programa que utiliza o dado, sem redundâncias, com acesso simples, e otimização dos recursos. Os principais modos funcionais do banco de dados são: modo preparador da operação lógica; modo transformador de parâmetros lógicos em físicos; modo realizador das operações de criação, atualização e leitura de um registro.

O subsistema de pré e pós-processamento realiza funções terminais relativas ao centro de controle, e caracterizam-se por operações simples, intensas e repetitivas. Os seus principais modos funcionais são: modo pré-processador de dados de rastreamento e telemetria dos sensores; modo sincronizador de telecomando; modo de controle e acompanhamento das antenas e dispositivos; modo de envio de telecomandos; modo de transmissão de dados.

O objetivo do subsistema de acompanhamento é fazer atualizações e previsões das características do estado dinâmico, acompanhar as condições dos equipamentos, analisar e prever condições de visibilidade. O subsistema de acompanhamento opera em tempo quase real e corresponde a uma grande carga de processamento; seus modos devem possuir características de flexibilidade, para serem suficientemente rápidos na fase de lançamento e suficientemente precisos na fase operacional. Os principais modos desse subsistema são: modo propagador de efemérides e matriz de covariância correspondente; modo propagador de atitude e matriz de covariância correspondente; modo de análises de visibilidade; modo preditor de pontos espaciais da órbita.

O objetivo do subsistema de determinação de órbita é determinar a posição, a velocidade e, eventualmente, os parâmetros do modelo, através da utilização de dados de observação coletados durante a missão. Os seus modos funcionais são: modo de determinação preliminar de órbita e modo de determinação de órbita.

Não ultrapasse esta linha

Redução de 10 para 8

No text below this line

Reduction 10 to 8

COMANDO PARA O CONTROLE DE SATÉLITES

... (exceto o ...)
...
... (with the exception of page 1) should commence here

LAYOUT PARA A PRIMEIRA PÁGINA
LAYOUT FOR THE FIRST PAGE

quase-real, dependendo do processo. A maior parte do processamento é do tipo quase-real, o que permite o processamento do tipo "batch", podendo ser utilizado eventualmente um sistema de usuários, dadas as prioridades especiais nos dias de lançamento. Os processos em tempo real, normalmente, também são pouco res- tritivos (Wertz, 1978), admitem conversaciona- lidade e podem ser executados em minicomputa- dores dedicados.

Tolerância à falha, deve ser objeto de prioritária atenção dentro do projeto de um sistema computacional, como suporte à missão espacial. Em situações críticas dentro da missão, uma falha, ou um conjunto de falhas, in- tolerável ao sistema computacional pode cor- responder à perda da missão. A probabilidade de ocorrência de uma situação intolerável ao sistema computacional deve ser imposta. Um sis- tema com essa característica é dito tolerante a falhas.

4. REQUERIMENTOS QUALITATIVOS

Simplicidade, confiabilidade, adaptabili- dade, generalidade, eficiência e portabili- dade são características de qualidades desejá- veis a qualquer elemento de um sistema compu- tacional e, inclusive, ao sistema computacio- nal como um todo. Hansen (1977) apresenta es- sas características de qualidade como atribu- tos de um bom programa.

A característica de simplicidade está bas- tante associada à clareza. Um elemento do sis- tema computacional pode ser grande demais pa- ra ser entendido e manuseado como um todo. Uma primeira decisão, no sentido de resolver es- se problema, é construí-lo subdividido em ele- mentos menores, módulos, de modo que um des- ses módulos possa ser entendido facilmente por um especialista. Uma segunda decisão é fazer esses módulos transparentes para a utilização, isto é, para utilização desses módulos é ne- cessário entender somente os aspectos de en- trada e saída. Uma terceira providência é uti- lizar interfaceamento minimal entre os módu- los e, finalmente, utilizar programação estru- turada; segundo Donaldson (1973), programação estruturada é a maneira mais razoável de orga- nizar e codificar programas, tornando-os de fácil entendimento e flexíveis quanto a modi- ficações.

Um elemento computacional deve possuir um alto grau de confiabilidade, dadas as dificul- dades que uma interrupção de um sistema, que opera em tempo real ou mesmo em tempo quase- real, pode provocar. As falhas devem ser evi- tadas, quando não toleradas. As falhas podem ser evitadas através do desenvolvimento crite- rioso, de testes sistemáticos para detecção de erros, de treinamento adequado de operadores, etc. As falhas podem ser toleradas, veja Ren- nels (1981), através de uma suficiente capaci- dade de confinamento, detecção, diagnóstico e recomposição do sistema..

Adaptabilidade é uma característica de qua- lidade importante sob dois aspectos: um siste- ma computacional para controle de satélites envolve custos e tempo de desenvolvimento, por isso, e por questões de qualificação, deve

Redução de 10 para 8

ser utilizado em missões espaciais sucessivas. Um segundo aspecto da adaptabilidade visa emer- gência. Embora, adaptações de emergência de- vam ser evitadas, se o sistema for suficiente- mente flexível a essas adaptações, elas podem ser extremamente úteis e, eventualmente, sal- var uma missão.

O grau de generalidade de um sistema com- putacional para missões espaciais está rela- cionado com o grau de generalidade de seus ele- mentos, e inversamente relacionado ao número de elementos de aplicação específica a cada missão. Quanto maior o grau de generalidade, menores serão os custos de adaptação e maio- res as possibilidades de envolvimento desse sistema em missões espaciais diversas.

A eficiência de um sistema computacional para controle de satélites pode ser definida por uma relação de serviços e de custos. Ca- racterísticas do serviço prestado, por exem- plo, carga e velocidade de processamento, in- dice de confiabilidade, são estabelecidas atra- vés dos requerimentos de desempenho. Um siste- ma eficiente deve considerar outras utiliza- ções nos períodos de baixa demanda, e a utili- zação de recursos não exclusivos ao centro de controle nos picos de demanda.

5. REQUERIMENTOS DE CONTORNO

Os requerimentos de contorno impõem dire- trizes ao projeto, a partir de critérios não envolvidos diretamente com o seu objetivo. Prá- ticas padrões, limitações de recursos, filoso- fias de investimentos, estrutura de organiza- ção, etc, impõem requerimentos ao projeto, de forma a adequá-lo aos interesses da organiza- ção usuária e de desenvolvimento. Os requeri- mentos de contorno limitam o espaço de solu- ções do problema, podendo eventualmente limi- tar desvantajosamente as opções do projetista.

6. IMPLEMENTAÇÃO

Os subsistemas representados na Figura 1 podem fazer parte de um mesmo bloco de progra- mas, a ser processado em um grande computa- dor, ou formar blocos para serem processados de forma distribuída em uma rede de minicompu- tadores. Os requerimentos computacionais não requerem as facilidades dos médios e grandes computadores para todas as funções. Vem sendo demonstrado que as funções intensivas e repe- titivas, como por exemplo o pré-processamen- to, são mais eficientemente realizadas em mi- nicomputadores.

A utilização de computadores segundo a fi- losofia do sistema canadense, apresentado em Domb (1975), parece ser uma excelente opção. Esse sistema utiliza um rede exclusiva, rela- tivamente pequena, de minicomputadores e com- putadores não-exclusivos de grande porte. Em uma eventual implementação nessa forma, um mi- ni-processor encarregar-se-ia de executar os subsistemas de pré e pós processamento e de banco de dados; um outro, os subsistemas de gerenciamento automático e conversacional; e um outro ainda dedicado a eventuais outros pro- cessos em tempo real; e finalmente todo pró- cessamento em tempo quase-real, dos demais sub- sistemas, é feito em "batch", em um grande com- putador.

Reduction 10 to 8

4º CONGRESSO BRASILEIRO DE AUTOMÁTICA

DO NOT ESTABLISH OR REMOVE DOCUMENTS AS PAGES / LEAVE THIS SPACE BLANK ON ALL PAGES

A good profile, with a margin (except on the top), deve começar a

Typing for all pages (with the exception of page 1) should commence here

LAYOUT PARA A PRIMEIRA FOLHA

LAYOUT FOR THE FIRST PAGE

A linguagem de programação a ser adotada depende do programa ou do trecho de programa. O "assembler" pode ser indicado para um trecho de programa, onde a velocidade é fundamental. Na programação de processos concorrentes, que compartilham variáveis comuns, a utilização de linguagem de alto nível concorrente é indicada. Para processos sequenciais, uma linguagem estruturada de alto nível deve ser utilizada.

Os recursos para gerenciamento e vinculação de processamento ao relógio são permitidos pelo sistema operacional, ou através de linguagens especiais de alto nível. A utilização de linguagem de alto nível uniformiza o "software", facilita seu desenvolvimento e aumenta sua confiabilidade, dando os benefícios de um compilador.

7. CONCLUSÕES

O projeto e o desenvolvimento de uma missão espacial supõem uma série de suportes básicos, de modo a permiti-la dentro de prazos e custos toleráveis. Como exemplos desses suportes básicos tem-se: capacitação científica, tecnológica e industrial; laboratório de simulação e testes; veículos lançadores; sistemas de rastreamento e transmissão de dados; recursos computacionais para análise, simulação e controle de missão. O potencial das possíveis missões sob todos os aspectos, por exemplo: custo e prazo de desenvolvimento, está associado às características desses recursos. Os países com programas espaciais adiantados estão desenvolvendo lançadores recuperáveis, sistemas de comunicação e rastreamento baseado em satélites, sistemas computacionais para controle de missão com alto grau de generalidade (Velez, 1981), o que deverá aumentar o potencial das missões, a custos menores, em futuro próximo.

O desenvolvimento do "software" de aplicação é o caminho crítico no desenvolvimento dos recursos computacionais. O sistema de apoio, dentro do possível, deve ser projetado dentro dos recursos disponíveis comercialmente.

A literatura disponível é pobre em informações sobre sistemas computacionais, aplicados a um programa espacial, moderno e de pequeno porte. Os sistemas divulgados ou são obsoletos, ou de grandes dimensões. Mesmo assim, devido ao estágio atual da engenharia em "hardware" e "software", um projeto ótimo é viável, embora, alguma experiência deva ser adquirida, por tentativa e erro, ao longo do desenvolvimento.

8. REFERÊNCIAS

Domb, V., (1975). "Duplex Computer System for Control of Canada's ANIK Satellites", Canadian Aeronautic and Space Journal, Vol. 21, nº 2: 59-64.

Donaldson, J.R., (1973). "Structured Programming", Datamation, Vol. 19, nº 12: 52-54.

Hansen, E.B., (1977). The Architecture of Concurrent Programs, New Jersey, Prentice Hall, Inc. Englewood Cliffs, Capítulo 1.

Não ultrapasse esta linha

Redução de 10 para 8

Marelli, L. & Valentiny, G., (1976). "The Control Centre and Spacecraft Control", ESA Bulletin, nº 7: 14-18.

McCaskill, A.M. & Neill, D.V. & Satterlee, A.A., (1972). "Launch and Orbital Injection of INTELSAT IV Satellites", COMSAT Technical Review, Vol 2, nº 2, 391-435.

Rennels, D.A., (1981). "Distributed Fault-Tolerant Computer Systems", Computer, IEEE Computer Society, Vol. 13, nº 3, 55-65.

Velez, C.E., (1981). "The Next Generation in Satellite Flight Control System", Symposium and Workshop on Dynamics of Planets and Satellites, Embu, São Paulo.

Wagner, W.E. & Velez, C.E., (1972). Goddard Trajectory Determination Subsystem Mathematical Specifications, Greenbelt, Maryland, Goddard Space Flight Center.

Wertz, J.R., (1978). Spacecraft Attitude Determination and Control, Dordrecht Holland, D. Reidel Publishing Company.

No text below this line

Reduction 10 to 8