SOBRE O USO DE NÚMEROS DIFUSOS NA AVALIAÇÃO DA CONFIABILIDADE DE SOFTWARE

José A. La Terza Ferraiuolo[1]

Victor R. R. Celestino[2]

Karl H. Kienitz[3]

Sumário:

A contribuição deste trabalho está concentrada na aplicação da Teoria de Conjuntos Difusos para o cálculo da confiabilidade do SOAB, com o objetivo de possibilitar a maximização da probabilidade de sucesso do lançamento do VLS.

É dada uma ênfase inicial à crescente atenção que o assunto confiabilidade de software vem recebendo desde os anos 60, revisando inclusive classes de modelos de confiabilidade de softwares propostos.

Sabe-se, contudo que o modelamento probabilístico depende da disponibilidade de dados e medidas adequadas, o que é uma característica bastante difícil no caso de softwares complexos ou inovadores. Em situações nas quais a obtenção de dados a partir de medidas pode ser dificultada ou impossibilitada, será necessário avaliar a confiabilidade baseado na opinião de analistas, programadores e grupos revisores. O processamento dessas informações vai além da capacidade do modelo de confiabilidade convencional. Utilizando conceitos da Teoria de Conjuntos Difusos os valores probabilísticos das freqüências relativas de ocorrência dos eventos são substituídos por valores possibilísticos, ou seja, por números difusos. É apresentada ainda uma metodologia para cálculo da confiabilidade do SOAB a partir da análise por árvore de falhas, merecendo destaque os passos da metodologia desenvolvida, que processa as informações dos especialistas, transformando-as em números difusos.

---

[1] Aluno de Graduação em Engenharia Mecânica, ITA
[2] Engenheiro Aeronáutico, Bolsista do RHAE (CNPq), Instituto de Aeronáutica e Espaço
[3] Ph.D. em Engenharia Elétrica, Professor Adjunto, Instituto Tecnológico de Aeronáutica

Abreviaturas e Símbolos:

VLS	Veículo Lançador de Satélites
CDB	Computador de bordo
SOAB	Software Operacional e Aplicativo de Bordo
INPE	Instituto Nacional de Pesquisas Espaciais
MCS	Modelo de Confiabilidade de Software
FMEA	Análise dos Modos de Falha e seus Efeitos
PHA	Processo de Hierarquia Analítica
REC	Rede Elétrica de Controle

1- Introdução:

1.1- O VLS:

Com a missão básica de injetar em órbita sátelites do INPE de coleta de dados, o VLS é um foguete que possui quatro estágios que serão controlados por um CDB. Na fase final de pré-lançamento e durante toda a fase de vôo, o VLS será comandado pelo software residente no CDB.

1.2- Confiabilidade de Software:

De uma forma geral, a confiabilidade de software é definida no contexto de probabilidade; ou seja, ela é definida como a probabilidade do software cumprir sua tarefa estabelecida, em um dado ambiente, para um número pré-definido de casos de entrada, admitindo que o hardware e a entrada estão livre de erros.

Interpretações de confiabilidade:

-capacidade do software alcançar a confiabilidade desejada (projeto robusto).

-garantia da obtenção da confiabilidade, o que requer testes e/ou avaliações.

É interessante ressaltar que o assunto **confiabilidade de software** tem recebido atenção crescente desde a década de 60. Já na década de 70, foram desenvolvidos diversos modelos de confiabilidade, baseados em testes ou processos de "debugging". Podemos citar alguns modelos bastante conhecidos: o modelo de Jelinski-Moranda(1972), o modelo de Shooman(1972), o modelo NHPP de Goel-Okumoto(1978), entre outros.

Pesquisas em confiabilidade de software são motivadas: por sua crescente importância nos sistemas modernos; pela ocorrência frequente de falhas em software e a suas conseqüências muitas

vezes catastróficas (caso do VLS); pelo crescimento da participação do custo do software no custo total, sendo necessário um maior controle com relação a sua qualidade.

2- Revisão da Literatura:

Prado(1990) apresenta uma ferramenta de análise desenvolvida por H. A. Watson(1962) que é a Árvore de Falhas. No início da década de 70, esta ferramenta foi bastante desenvolvida e seu uso tornou-se muito difundido. Os métodos baseados em Árvore de Falhas são hoje os mais largamente usados.

Os modelos de confiabilidadede software existentes foram classificados por Cai et alli (1991) em quatro classes:

-Macro: utilizam dados de "debugging" para predizer a confiabilidade de software empregando metodologias não-bayseianas e ignorando a estrutura interna do software. Podem ser chamados de modelos de caixa-preta. Todos os modelos baseados em contagem de erros são incluidos nesta classe. No entanto, o MCS Macro pode ser dividida em duas sub-classes: modelos de tempo de calendário e modelos de tempo de execução. A maioria dos modelos Macro está na classe dos que utilizam o tempo de calendário como índice de tempo. Exemplos típicos desta classe são os modelos proposto por Moranda(1975) e Musa(1975 e 1979);

-Micro: incorporam algumas métricas de software, tais como a complexidade do software e o número de linhas de código, para predizer o comportamento da confiabilidade de software. Podem ser considerados modelos de caixa-branca, apesar de dados de "debugging" poderem ser empregados também. Exemplo típico desta classe é o modelo proposto por Shooman(1972).

-Bayesiano: também pertence aos modelos chamados caixa-preta. Empregam dados de "debugging" para reportar o comportamento da confiabilidade de software, porém consideram alguns dos parâmetros do modelo como variáveis aleatórias. Exemplos típicos desta classe são os modelos propostos por Littlewood(1973 e 1979) e Jewell (1985).

-Validação: são usados na fase de validação, na qual os erros são detetados, mas não são eliminados. Com um relatório de dados de testes, procura-se estimar a confiabilidade. Nenhum esforço é realizado no sentido de melhorar a confiabilidade. Exemplo típico desta classe é o modelo proposto por Nelson(1978).

Adicionalmente, é importante ressaltar que alguns modelos não se encaixam em nenhuma destas classes. São modelos que não focalizam no comportamento da confiabilidade do software como um todo, mas são direcionados para uma métrica específica, tais como o número de erros remanescentes. Exemplos típicos são os modelos propostos por Islam & Lombardi(1982), Huang(1984 e 1985) e Roca(1987).

Uma inspeção dos modelos de confiabilidade de software existentes, revela de imediato uma característica comum, que é a hipótese da probabilidade, incorporada nesses modelos, com pouquíssimas exceções. A hipótese de probabilidade é considerada razoável em decorrência de duas observações:

-Incerteza na escolha das entradas;

-Características do software. Não se pode precisar exatamente quando uma rotina específica é executada, nem qual rotina é executada em um tempo específico.

No entanto, Cai et Alii(1991) apresentam várias assertivas baseadas na experiência de vários autores na verificação da hipótese da probabilidade, das quais podemos concluir que os modelos existentes nelas baseados não tiveram sucesso.

O problema essencial é a questão quanto à adequabilidade da hipótese de probabilidade para explicar o comportamento da confiabilidade de software. Em geral, a hipótese de probabilidade não é adequada em decorrência da violação de uma ou mais das seguintes hipóteses:

1. Os tempos entre falhas são independentes. Esta hipótese é usada em praticamente todos os MCS de tempo de calendário. Em geral, esta hipótese não é válida pelo fato dos testes não serem selecionados aleatoriamente, mas sim baseados na experiência anterior com o software. Isto causa uma aparente relação de dependência entre uma e outra correção de erro.

2. Um erro detetado é imediatamente corrigido. Isto significa que o tempo requerido para remover um erro é desprezível, o que é tão mais incorreto, quanto maior for a complexidade do software. A experiência mostra que em média 10% do tempo entre duas falhas consecutivas é consumido na correção da falha anterior.

3. Nenhum erro adicional é introduzido durante o processo de correção de uma falha. Esta é uma hipótese obviamente muito forte, já que a experiência mostra que a correção de uma falha pode significar alterações profundas no código. Além disto, os processos de "debugging" são imperfeitos.

4. A taxa de falhas decresce com o tempo de teste, o que implica que a confiabilidade do software cresce monotonicamente. Esta hipótese deixa de ser válida se a hipótese 3 não for obedecida. Além disso, ela pode não ser verdadeira em função das entradas utilizadas ao longo do processo de teste, já que a confiabilidade do software tem uma grande dependência das entradas.

5. A taxa de falhas é proporcional ao número de erros remanescentes no software. Esta hipótese implica que cada erro remanescente tem a mesma chance de ser detetado num dado intervalo entre falhas. No entanto os módulos de software não são executados com a mesma freqüência. A experiência mostra que aproximadamente 90% é consumido na execução de apenas 10% do código.

6. A confiabilidade é uma função do número de erros remanescentes. Esta hipótese é só parcialmente correta, pois a confiabilidade de software depende pesadamente das entradas e os dados de "debugging" são afetados pelos casos de testes.

7. O tempo é usado como uma base para as taxas de falhas. No entanto, a experiência mostra que existem outras métricas, tais como número de linhas de código testadas, número de funções testadas e número de casos de teste executados, podem ser mais adequados para obter resultados consistentes.

8. A taxa de falhas aumenta entre falhas. Um grande esforço de testes tende a aumentar a taxa de falhas, porém, esta hipótese pode não ser verdadeira dependendo dos casos testados e dos esforço de testes.

9. Os testes são representativos do uso operacional. Esta hipótese é necessária para prever a confiabilidade na fase operacional a partir de dados de "debugging". Infelizmente, este nem sempre é o caso pelo fato das entradas a qual o software é submetido nem sempre são esperadas.

10. As falhas de software são independentes das características do software. Esta hipótese nem sempre é válida já que existem modelos Micro. O correto seria incorporar tanto as caracrerísticas do software como os dados de "debugging" para determinar a confiabilidade do software.

11. As estimativas de confiabilidade de software são independentes dos procedimentos de inferência. Esta hipótese é normalmente aceita de forma implícita, contudo, a fim de se obter uma boa predição não é suficiente ter um bom modelo, pois um procedimento de inferência do parâmetro também está envolvido.

12. Existem dados suficientes e de alta qualidade. Cada software é único e o processo de "debugging" nunca é repetido, logo, usualmente não se obtém dados suficientes para sustentar a hipótese de probabilidade e os dados disponíveis nem sempre tem a qualidade desejada.

Pelas razões acima descritas, será adotado um método que não inclui a hipótese de probabilidade.

Para fazer com que um modelo de confiabilidade de software seja ao mesmo tempo aplicável a qualquer software e tratável do ponto de vista matemático, Singer(1990) propõe uma metodologia onde se substitui o modelo probabilístico pelo possibilístico, aplicando-se números difusos, já que o comportamento do software é difuso por natureza.

3- Teoria de Números Difusos:

3.1- Conjuntos Difusos:

A Teoria de Conjuntos Difusos fornece meios para se lidar com conceitos e medidas de natureza inexata e pode ser usada para estabelecer uma formulação matemática para informação imprecisa expressa através de frases linguísticas. Ela foi enunciada por Zadeh (1965) e já existem diversos livros texto disponíveis, entre eles Dubois & Prade (1980).

Essa Teoria tem como objetivo principal desenvolver uma metodologia para a formulação e solução de problemas muito complexos ou mal definidos para serem analisados por técnicas convencionais. Devido à sua natureza não-ortodoxa, essa teoria tem sido, e ainda é, muito controvertida. Mesmo assim, acredita-se que ela será reconhecida como uma evolução natural do pensamento científico e o ceticismo diante de sua utilidade será superado.

3.1.1- Definição:

Na teoria clássica de conjuntos, um elemento pertence ou não a um conjunto. Dado um universo de elementos U e um elemento particular $x \in U$, o grau de pertinência $\mu_A(x)$ com respeito a um conjunto ($A \subseteq U$) é dado por:

$$\mu_A(x) = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{se } x \notin A \end{cases}$$

A função $\mu_A(x) : U \rightarrow \{ 0, 1 \}$ é chamada função característica na teoria clássica de conjuntos.

Em 1965, Zadeh propôs uma caracterização mais ampla, na medida em que ela permite a existência de graus de pertinência diferentes de 0 e 1. O fator de pertinência pode assumir qualquer valor entre 0 e 1, onde 0 indica completa exclusão e o valor 1, completa pertinência. Com essa generalização, aumenta-se o poder de expressão da função característica.

Seja U um universo de discurso, podendo ser contínuo ou discreto. Um conjunto difuso A, neste universo de discurso é definido por uma função de pertinência μ_A que assume valores em um intervalo [0, 1]:

$$\mu_A : U \rightarrow [0, 1]$$

O conjunto suporte de um conjunto difuso A é o conjunto dos pontos x de U tal que $\mu_A(x) > 0$. Um conjunto difuso cujo suporte é um único ponto de U com $\mu_A = 1$ é chamado de um conjunto difuso unitário.

3.1.2- Operações de União e Intersecção:

Segundo Zadeh (1965), as operações clássicas de união ($\cup$) e intersecção ($\cap$) de conjuntos de U podem ser extendidas pelas seguintes fórmulas:

$$\forall x \in U , \mu_{A \cup B}(x) = \max (\mu_A(x), \mu_B(x));$$

$$\forall x \in U , \mu_{A \cap B}(x) = \min (\mu_A(x), \mu_B(x)),$$

onde $\mu_{A \cup B}$ e $\mu_{A \cap B}$ são, respectivamente, as funções de pertinência de $A \cup B$ e $A \cap B$. Estas fórmulas resultam na união e intersecção usuais quando a função de pertinência do conjunto toma valores apenas em {0, 1}.

3.1.4- Conjuntos difusos convexos e normalizados:

3.1.4.1- Conjuntos difusos convexos:

Um conjunto difuso é dito convexo se, e somente se, $\forall x_1 \in U, \forall x_2 \in U, \forall \delta \in [0, 1]$,

$$\mu_A (\delta x_1 + (1-\delta)x_2) \geq \min (\mu_A (x1), \mu_A (x2)).$$

Podemos notar que essa definição não implica que μ_A seja uma função convexa de x, conforme a figura 1.

3.1.4.2- Conjunto difuso normal:

Um conjunto difuso é normal quando sua função de pertinência é tal que $\exists x \ / \ \mu(x) = 1$.

3.2- Números Difusos:

3.2.1- Intervalos difusos:

A um conjunto difuso convexo e normalizado da reta real, quando a função de pertinência para um intervalo difuso é:

. contínua em $\Re$ no intervalo fechado [0,1];

. constante e igual a zero em $(-\infty, x_1]$;

. estritamente crescente em $[x_1, x_2]$;

. constante e com valor unitário em $[x_2, x_3]$;

. estritamentedecrescente em $[x_3, x_4]$;

. constante e igual a zero em $[x_4, +\infty)$.

A definição acima engloba qualquer intervalo, inclusive o próprio conjunto dos números reais.

3.2.2-Números difusos:

Um número difuso ou difuso é um conjunto difuso contínuo do universo U com função de pertinência $\mu(x)$ satisfazendo a seguinte condição:

$$\max_{x \in \Re} \mu(x) = 1 \qquad (1)$$

De acordo com essa definição, a função de pertinência de um número difuso pode ter diversas formas. A figura 2 mostra possíveis valores para a função de pertinência do número difuso M. A largura de $\mu(x)$ é uma medida do espalhamento do número difuso M.

A definição de número difuso de acordo com (1) evidencia também as diferenças entre as noções de números difusos e números estocásticos.

De acordo com Singer (1990), as operações algébricas podem ser extendidas aos números difusos. Existem algoritmos para obtenção de qualquer função de pertinência contínua, porém eles são relativamente complicados e consomem bastante tempo para calcular funções difusas com um grande número de variáveis. Pode-se, entretanto, reduzir os esforços e o tempo de processamento

aproximando a verdadeira função de pertinência (que em muitos casos é desconhecida) por outras mais simples.

Em muitos casos práticos, a função de pertinência $\mu(x)$ pode ser aproximada por duas funções L(x) e R(x) com um ponto de intersecção max $\mu(x) \equiv 1$, conforme Dubois & Prade. Números com este tipo de função de pertinência são denominados números difusos L-R.

3.2.3- Números difusos L-R:

Um número difuso $\tilde{A}$ é do tipo L-R quando a sua função de pertinência for do tipo:

$$\mu_{\tilde{A}}(x) = L((m-x)/\alpha) \text{ para } x \leq a, \ \alpha > 0,$$
$$R((x-m)/\beta) \text{ para } x \geq a, \ \beta > 0.$$

Num contexto de confiabilidade, o escalar a torna-se o valor médio da freqüência relativa de ocorrência de um determinado evento. α e β são chamados espalhamento à esquerda e à direita, respectivamente, e L e R são funções apropriadas de $(a-x)/\alpha$ e $(x-a)/\beta$, respectivamente.

3.2.4- Operações com Números Difusos:

Podemos representar simbolicamente um número difuso L-R por (a, α, β). A partir dessa notação e de vários lemas e teoremas demonstrados em Dubois & Prade, as operações algébricas extendidas aos números difusos podem ser formuladas como a seguir:

Troca de sinal:

$$-(a, \alpha, \beta)LR = (-a, \beta, \alpha)RL$$

Adição $\oplus$:

$$(a, \alpha, \beta)LR \oplus (b, \gamma, \delta)LR = (a + b, \alpha + \gamma, \beta + \delta)LR$$

Multiplicação $\otimes$:

$$(a, \alpha, \beta)LR \otimes (b, \gamma, \delta)LR \cong (ab, a\gamma + b\alpha, a\delta + b\beta)LR \text{ caso } a > 0 \text{ e } b > 0.$$

4- Engenharia de Confiabilidade:

4.1- Objetivos:

Para o VLS estabeleceu-se um determinado valor desejado para sua confiabilidade. A metodologia a ser proposta para a confiabilidade de software visa possibilitar que a confiabilidade do SOAB se encontre dentro de um limiar aceitável para aquele objetivo.

Para atingir essa meta, devemos identificar, após uma primeira depuração dos dados obtidos, qual ou quais funções do SOAB têm um comportamento crítico; realizando, assim, algum trabalho adicional. Caso se faça necessário, deve-se utilizar a redundância.

4.2- Fontes de Dados de Confiabilidade:

Nesta contribuição somente a confiabilidade de software é considerada. Assume-se que tanto o hardware como os algoritmos implementados estarão livre de erros.

A principal fonte de dados será a opinião dos especialistas envolvidos no projeto VLS, uma vez que medidas exaustivas, que validam a hipótese probabilística, inexistem. De posse desses dados, seá utilizado um enfoque que usa números difusos, ideal para situações onde não há dados disponíveis. Essa é a única maneira sistemática de avaliarmos a confiabilidade, ainda que de forma preliminar.

Quando o produto entrar numa fase mais madura, no sentido de haver uma massa de dados disponiveis, após realização de diversos testes com todos os sub-sistemas, e preferencialmente um número suficiente de lançamentos, podere-se-á passar a utilizar uma abordagem probabilística.

4.3- Modelamento:

4.3.1- Árvore de Falhas:

O desenvolvimento da ferramenta **árvore de falhas** ocorreu nos laborátorios da Bell Telephone, por volta de 1962. Publicaram-se os primeiros artigos em 1965, num Simpósio de Segurança patrocinado pela Universidade de Washington e pela Boeing Co., onde a técnica foi aplicada extensivamente.

Uma árvore de falhas :

a) Assinala os aspectos de falha importantes e de interesse, no sistema;

b) Permite visualizar o comportamento do sistema, em relação às suas falhas;

c) Está dirigida para a descoberta de falhas;

d) Permite que o analista concentre seu esforço separadamente em cada falha;

e) Possui opções para análise do sistema sob o ponto de vista qualitativo e quantitativo;

f) Fornece uma visualização gráfica daquelas partes do sistema que podem ser removidas por mudanças de projeto.

Pode-se acrescentar, entretanto, que uma árvore de falhas, como qualquer outro relátorio de engenharia, não passa de uma ferramenta para comunicação e como tal, deve ser um registro claro e demonstrável.

A estrutura fundamental de uma árvore de falhas pode ser observada na figura 3. O **evento maior** é o evento indesejável e está ligado a outros eventos básicos de falha pela definição desses eventos e por portas lógicas

A vantagem principal de uma árvore de falhas em relação a outras técnicas tais como FMEA e PHA é que a análise está restrita somente na identificação dos elementos e eventos do sistema que levam a uma falha ou a um acidente particular.

4.3.2- O modelamento da árvore de falhas e a análise de confiabilidade:

O procedimento da análise da árvore de falhas e da confiabilidade pode ser dividido sob ponto de vista téorico nos passos a seguir:

- Investigando a descrição verbal causa-efeito do sistema incluindo os antecedentes e conseqüências relevantes com todas as relações causais entre eles.

- Reformulando os antecedentes e conseqüências, transformando-as na forma de efeitos Booleanos, tendo somente dois estados: existência e não existência.

- Formulando os eventos na forma de funções Booleanas. O conjunto de todas as funções Booleanas aplicadas nos eventos básico, intermediário e maiores formam um diagrama ordenado, que pode ser considerado o modelo de evento do sistema.

- O modelo qualitativo do evento pode ser ampliado, atribuindo-se valores probabilísticos (freqüência) aos eventos básicos e derivando, através de Álgebra Booleana e Teoria da Probabilidade, valores probabilísticos para os eventos induzidos e, finalmente, o evento maior.

- O último passo deverá ser o controle da confiabilidade do valor probabilístico deduzido utilizando a metodologia da Teoria da Probabildade.

A maneira mais simples de se formular um problema de confiabilidade na forma Booleana, dado inicialmente a partir de uma descrição verbal, é através da utilização dos operadores Booleanos padrão: AND, OR and NEG. Pode ser mostrado que todas as funções lógicas podem ser expressas usando-se esses operadores elementares.

Suponha-se que o modelo do evento contenha somente váriaveis mutuamente exclusivas; logo, a confiabilidade do sistema pode ser obtida substituindo-se todas as váriaveis independentes **Ai** pelas freqüências relativas de ocorrências (probabilidades) p_{Ai} e substituindo os operadores lógicos OR e AND pela adição e multiplicação algébrica, respectivamente. As váriaveis Booleanas negativas $\tilde{A}i$ são substituídas pelas probabilidades $p_{\tilde{A}i} = 1 - p_{Ai}$. Note que substituindo os operadores lógicos OR, AND e NEG (nós de ligação do modelo da figura 4) por operadores probabilísticos apropriados, obtém-se um modelo probabilístico de mesma estrutura.

A função probabilidade resultante de um operador n-ário AND é:

$P_y^{and} = \Pi p_i$ (2)

onde pi é a probabilidade da entrada i, e P_y *é a probabilidade do evento de saída.*

As funções de probabilidade resultantes de um operador n-ário OR e um NEG são:

$P_y^{or} = 1-\Pi(1-p_{Ai})$ (3)

$P_y^{neg} = 1-p_A$ (4)

Dadas essas funções, a função de confiabilidade probabilística do sistema da figura 4 será:

$Py = [1-(1-p_A)*(1-p_B)]*(1-p_C)$

4.3.2.1- Passagem do modelamento probabilístico para possibilístico:

As probabilidades dos eventos nada mais são do que freqüências relativas de ocorrência. Sabemos que em sistemas técnicos reais, como sátelites, mísseis, etc, as falhas são raras, logo as freqüências de ocorrência observadas mostram em geral espalhamentos largos. Uma outra causa da inexatidão dessas freqüências é o comportamento, geralmente não-estacionário e não-ergódico de fenômenos naturais, especialmente em sistemas feitos pelo Homem. Torna-se, portanto, natural manusear o problema de confiabilidade com a teoria de conjuntos difusos.

Segundo Singer (1990) , para se passar do modelo probabilístico para o possibilístico, simplesmente deve-se considerar as freqüências relativas dos eventos como números difusos, ou seja, são válidas as expressões (2), (3) e (4) no domínio difuso.

4.3.2.2- Os operadores difusos:

Os operadores AND, OR e NEG serão obtidos, considerando-se a validade das expressões obtidas acima, substituindo-se, porém as operações algébricas pelas operações com números difusos, de acordo com o item 3.2.4.

4.4- Predição:

4.4.1- Predição:

A predição é usualmente baseada em conceitos de design e complexidade estimados. A complexidade é talvez a maior responsável pela pobreza de confiabilidade, especialmente para sistemas em que não são previstos ajustes imediatos ou manutenção corretiva, tais como satélites e mísseis.

Durante todo o desenvolvimento do projeto do SOAB, serão efetuados estimativas de confiabilidade não formais a diversos níveis. As estimativas serão feitas com critérios descritos conforme o item 4.2.

4.4.2- Metodologia:

A metodologia utilizada neste trabalho é a análise por árvore de falhas, que tem como objetivo, determinar a confiabilidade de um evento principal, no caso a operação com sucesso do SOAB.

Uma árvore de falhas simplificada para o SOAB encontra-se na figura 5. Deverão ser atribuídas possibilidades de falhas aos eventos básicos.

As possibilidades são obtidas de uma maneira bastante subjetiva, o que recomenda, conforme já citado, o uso da Teoria de Conjuntos Difusos. Ou seja, cada possibilidade considerada é caracterizada por um número difuso. Para permitir o uso de operações simples, são utilizados números difusos triangulares.

Para localizar os caminhos críticos da árvore de falhas, bem como determinar o valor da possibilidade de falha do evento principal, são utilizadas operações de União, Intersecção ou Complemento (dependendo do tipo de porta da árvore: OR, AND ou NEG).

4.4.2.1- Obtenção dos números difusos:

A coleta de dados é feita através da opinião dos especialistas na área. Cada especialista atribui um número difuso triangular, para cada um dos eventos básicos da árvore.

Primeiramente, é exposto ao especialista uma referência de utilização de função do SOAB em estados e modos da REC, incluindo o tempo estimado de cada fase de vôo.

Para evitar que o especialista diga, logo de início, que tem confiança plena sobre alguma função do SOAB em particular, devem ser lembradas as possíveis dificuldades:

1- O teste do módulo pode não ter sido executado conforme o previsto.

2- O teste pode ter sido especificado de maneira não apropriada, incompleto.

3- Houve um mau entendimento na hora de se assimilar o algoritmo.

Uma vez que essas informações estejam bem claras ao especialista, é feita a seguinte pergunta:

"Dado que a **Função X** do SOAB irá funcionar nos seguintes estados e modos da REC durante um tempo definido para a missão, qual a confiabilidade que você atribui para que esta função tenha o desempenho conforme o especificado e sem falhas? "

Para um perfeito entendimento do método que será descrito abaixo, deixaremos clara nossa linha de raciocínio. Escolhemos um número difuso triangular simétrico com uma freqüência relativa de ocorrência de 50% e já normalizado, conforme a figura 6. O objetivo é obter arbitrariamente (segundo a opinião de especialistas), a faixa de freqüência relativa de ocorrência com uma pertinência de 0.8. Conforme o cálculo efetuado, a largura dessa faixa será de 10%. Da mesma forma, para uma pertinência de 0.4, a largura da faixa será de 30%. Com isso, conseguir-se-á mapear e construir um número difuso triangular.

4.4.2.2.- Passos do método:

a) É dada uma folha de papel milimetrado ao especialista com diversos intervalos desenhados de 0 a 100% para cada função do SOAB. Pede-se que ele aloque um intervalo com 30% de largura em cada intervalo representativo da função cuja implementação seja de sua responsabilidade.

b) Repete-se a operação acima, substituindo o intervalo de 30% por um de 10%.

c) Para cada especialista entrevistado, plota-se os dados obtidos nos passos anteriores numa folha de papel milimetrada, lembrando que um intervalo de 10% ocorre com uma pertinência de 0.8 e um intervalo de 30% ocorre com uma pertinência de 0.4.

d) Como cada intervalo está associado a um valor de pertinência, obtém-se um valor m_3 (um provável valor para a confiabilidade com possibilidade de ocorrência igual a um), que será a média ponderada aos respectivos valores de pertinência. Ou seja, $m_3 = (0.8*m_1 + 0.4*m_2)/1.2$.

e) Tem-se, pois, dois conjuntos constituídos por três pontos: um formado pela extremidade esquerda de cada intervalo e o ponto m_3, e o outro pela extremidade direita de cada intervalo e pelo ponto m_3. Obtém-se, assim, através da regressão linear, as duas melhores retas para cada um dos dois conjuntos de pontos, respectivamente, conforme ilustrado na figura 7.

f) As retas obtidas se interceptarão, identificando qual o valor de confiabilidade que apresenta uma pertinência máxima. Porém, essa confiabilidade provavelmente não corresponde ao valor de pertinência igual a um.

g) Normaliza-se, portanto o número triangular obtido. Essa operação é feita simplesmente dividindo-se todos os valores de pertinência pela pertinência máxima obtida no item anterior.

h) Dessa forma, obtém-se diversos números difusos triangulares e normalizados que traduzem a confiabilidade de operar conforme o previsto e sem falhas para cada função da árvore de falhas que define o SOAB. Realizam-se, finalmente os cálculos para se obter a possibilidade do SOAB operar com sucesso, segundo o item 4.3.2.2.

Observações importantes:

1- As entrevistas são feitas com cada especialista separadamente.

2- As possíveis inconsistências do entrevistado são resolvidas utilizando-se procedimentos específicos; caso se torne necessário, é feita nova entrevista.

5- Conclusão:

No trabalho, foram evidenciadas as dificuldades de obtenção de dados a partir de medidas, para softwares inovadores. Com isso, avaliou-se a confiabilidade do software através da opinião dos especialistas envolvidos no projeto, o que leva a consideração de conceitos além daqueles da probabilidade clássica.

Foi proposto o cálculo da confiabilidade usando-se a Teoria de Conjuntos Difusos, única forma sistemática de se obter os valores desejado a partir das informações disponíveis.

Referências:

1. CAI, K.Y., WEN, C.Y. & ZHANG, M.L. , "A critical review on software reliability modeling" , *Reliability Engineering and Systems Safety*, vol.32, pp.357-371, 1991.

2. CAI, K.Y., WEN, C.Y. & ZHANG, M.L. , "Fuzzy reliability modeling of gracefully degradable computing systems" , *Reliability Engineering and Systems Safety*, vol.33, pp.141-157, 1991.

3. DUBOIS, D. & PRADE, H. , *Fuzzy sets and systems: theory and applications*, Academic Press, New York, 1980.

4. LAPLANTE, *Real-Time Systems Design and Analysis: an Engineer's Handbook*, IEEE Press, New York, 1993.

5. MORANDA, P.B. , "Prediction of software reliability during 'debugging'", *Proc. Annual R&M Symposium*, pp.327-332, 1975.

6. MUSA, J.D. , "A theory of software reliability and its applications", *IEEE Trans Software Engineering*, vol. SE-1, n.3, pp.312-327, 1975.

7. PRADO, M. S. , *Segurança sistematizada: confiabilidade, segurança, perigo e avaliação de risco*, Apostila, Instituto de Fomento e Coordenação Industrial, S. José dos Campos, agosto de 1990.

8. SINGER, D. , "A fuzzy set approach to fault tree and reliability analysis" , *Fuzzy Sets and Systems*, vol.34, pp.145-155, 1990.

9. SHOOMAN, M.L. , "Probabilistic methods for software reliability prediction", *Proc. FTCS-2*, pp.211-215, 1972.

10. TURKSEN, I.B. , "Measurement of membership functions and their acquisition" , *Fuzzy Sets and Systems*, vol.40, pp.5-38, 1991.

11. ZADEH, L.A. , "Fuzzy sets", *Information and Control*, vol.8, pp.338-353, 1965.

Figuras:

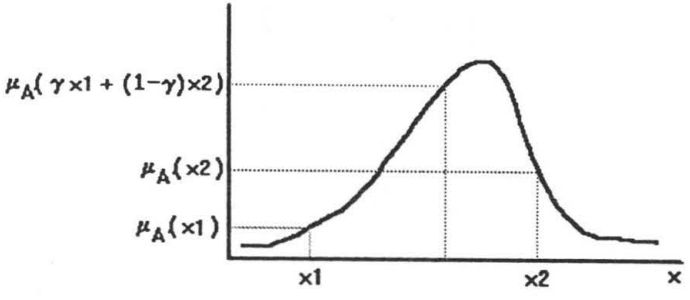

Fig. 1 - Conjunto Difuso Convexo

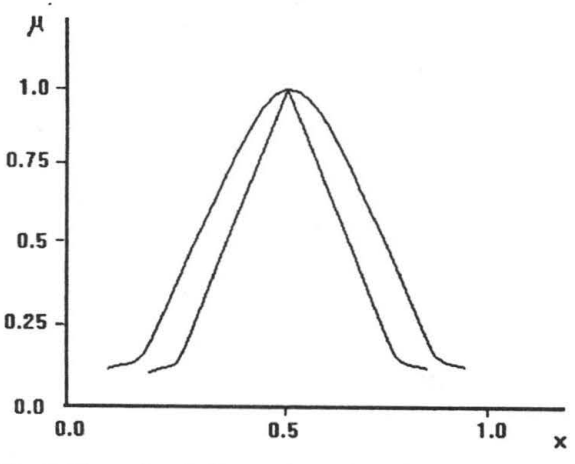

Fig. 2- Exemplos de função de pertinência do número **M**

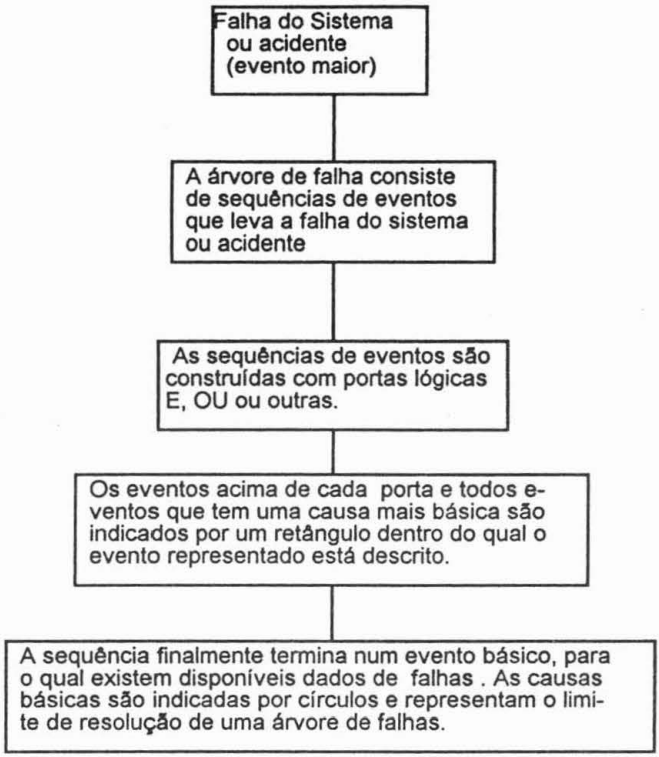

Fig.3- Estrutura fundamental de Árvore de Falhas.

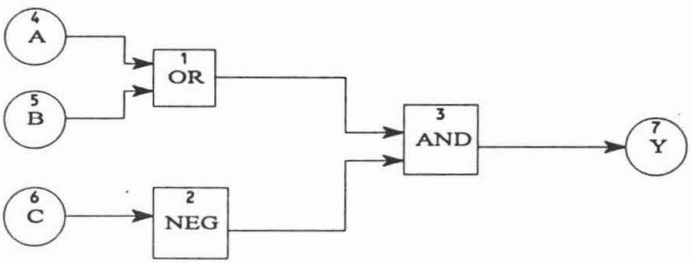

Fig.4- Modelo Probabilístico.

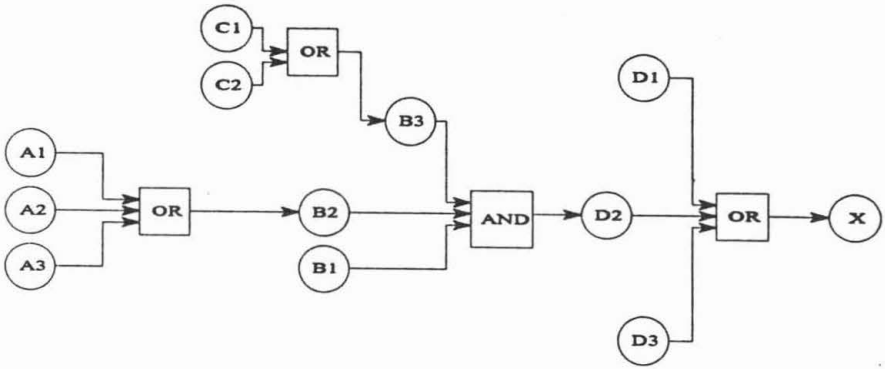

Fig.5- Possível exemplo de árvore para o SOAB.

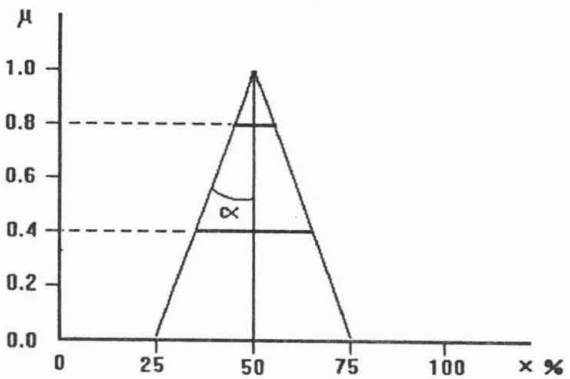

Fig.6- Base do método proposto.

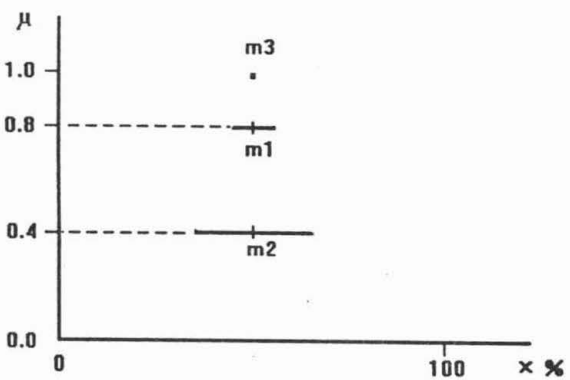

Fig.7 - Passos iniciais do método