

# Um Modelo de Avaliação de Especificações Semi-Formais de Requisitos de Software Baseado na Teoria dos Conjuntos Nebulosos

Luiz Eduardo Galvão Martins, Ana Estela Antunes da Silva  
UNIMEP - Universidade Metodista de Piracicaba  
{lgmartin, aeasilva}@unimep.br

**Resumo.** O objetivo deste trabalho é apresentar um modelo de avaliação de especificações semi-formais de requisitos de software baseado na Teoria dos Conjuntos Nebulosos. Especificações semi-formais normalmente são apresentadas como um conjunto de diagramas onde os requisitos do software são declarados. Especialistas do domínio do problema e engenheiros de requisitos devem conjuntamente avaliar se a documentação produzida reflete de forma adequada os requisitos do software a ser implementado. Vários atributos de qualidade têm sido propostos para avaliar se uma especificação de requisitos é de fácil entendimento, completa, não ambígua, consistente internamente, possível de ser implementada, entre outros. A avaliação destes atributos de qualidade carrega um alto grau de subjetividade e incerteza, para os quais uma avaliação mais flexível, utilizando valores lingüísticos como baixo, médio e alto, torna-se adequada. Apresentamos ao longo do artigo os passos para a realização de tal avaliação, utilizando como base para o processo a Teoria dos Conjuntos Nebulosos.

**Palavras-Chave:** Atributos de Qualidade, Especificação Semi-Formal de Requisitos, Função de Pertinência, Teoria dos Conjuntos Nebulosos.

## 1. Introdução

A necessidade de um documento que expresse de forma clara e objetiva os requisitos de um software a ser implementado tem se tornado uma questão estratégica para as empresas e indústrias que produzem ou fazem uso intensivo de software. Na maioria dos casos, o documento que expressa os requisitos é uma especificação semi-formal, baseada em diagramas como diagrama de fluxo de dados, diagrama entidade-relacionamento, diagrama de casos de uso, diagrama de classes etc. [1][2]. Embora não sejam de difícil leitura, especificações semi-formais podem carregar um alto grau de subjetividade na sua interpretação (tanto da parte dos clientes como dos desenvolvedores do software), o que sugere um cuidado especial no processo de avaliação de tais especificações.

A Teoria dos Conjuntos Nebulosos oferece recursos que permitem uma representação matemática de conceitos imprecisos que cotidianamente utilizamos para avaliar entidades do mundo à nossa volta. Por exemplo, é comum utilizarmos os termos muito, pouco, baixo, médio, alto, perto, longe etc. para avaliarmos grandezas que temos dificuldades de precisá-las. Esses termos são chamados de valores

lingüísticos, e expressam uma faixa de valores. Com a Teoria dos Conjuntos Nebulosos, conseguimos uma representação formal para expressar fatos como um conjunto de pessoas altas, um conjunto de temperaturas baixas, um conjunto de carros velhos, um conjunto de bons alunos etc. A característica fundamental de todos esses conjuntos, quando avaliados por “instrumental humano” (sentidos como visão, tato etc.), é que as fronteiras de tais conjuntos não são muito rígidas, do tipo pertence ou não pertence (avaliação bivalente), mas ao contrário, costumam ser flexíveis, embutindo a idéia de graus de pertinência ao conjunto (avaliação multivalente).

Nas seções seguintes, apresentamos um modelo de avaliação de especificações semi-formais de requisitos de software que adota, como base de sustentação, a Teoria dos Conjuntos Nebulosos, com o intuito de tratar de forma mais adequada a incerteza e subjetividade inerentes ao processo de definição dos requisitos e de sua conseqüente documentação. Para tanto, definimos conjuntos nebulosos baixo, médio e alto para alguns atributos de qualidade, os quais permitem a realização da avaliação de uma especificação semi-formal de requisitos. Acreditamos que tal forma de avaliação leva em consideração o alto grau de imprecisão que existe quando avaliamos um documento semi-formal de especificação de requisitos, permitindo, portanto, um resultado de avaliação de maior confiabilidade.

## 2. Definição dos Atributos de Qualidade

Vários trabalhos têm sido desenvolvidos com o intuito de identificar e definir atributos de qualidade para a avaliação de especificação de requisitos de software, dentre os quais podemos destacar os esforços apresentados em [3][4]. O trabalho desenvolvido por Davis *et al.* [3] amplia a lista de atributos de qualidade definidos em [4].

Davis *et al.* identificam uma lista de 24 atributos de qualidade que podem ser utilizados como parâmetros em processos de avaliação de especificações de requisitos. Os 24 atributos de qualidade são: não ambigüidade, completitude, corretitude, facilidade de entendimento, facilidade de verificação, consistência interna, consistência externa, possibilidade de implementação, concisão, independência de projeto, rastreabilidade, modificabilidade, armazenamento eletrônico, execução/interpretação/prototipação, anotação por importância relativa, anotação por estabilidade relativa, anotação por versão, não redundância, abstração/detalhamento, precisão, reusabilidade, base de referência, organização e referência cruzada.

O modelo de avaliação de especificação de requisitos que propomos não considera os 24 atributos de qualidade citados. Foram escolhidos seis atributos de qualidade: não ambigüidade, corretitude, consistência interna, anotação por importância relativa, anotação por estabilidade relativa e anotação por versão. A redução do número de atributos de qualidade foi uma decisão para efeitos de simplificação do modelo, uma vez que o método de inferência adotado no processo de avaliação utiliza regras de associação, conectando os atributos de qualidade. Considerando que cada atributo de qualidade poderá assumir três possíveis valores lingüísticos (baixo, médio e alto) o número de regras necessário para combinar os 24 atributos seria muito elevado.

Mesmo considerando seis atributos de qualidade, o número de regras para tratar todas as combinações entre os valores lingüísticos envolvidos ainda é alto, 729

regras<sup>1</sup>. Este número de regras pôde ser reduzido, dividindo-se os atributos de qualidade em dois grupos, conforme apresentado no capítulo 3.

Na escolha dos seis atributos foi levada em conta a importância destes no processo de avaliação de uma especificação de requisitos, e os pesos atribuídos em Davis *et al.* para cada um dos atributos (todos eles possuem um peso elevado segundo a classificação de Davis).

Os valores discretos de cada um dos atributos de qualidade, que servirão de entrada para o processo de avaliação proposto, são obtidos conforme apresentados nas definições e expressões a seguir [3]:

1. Não ambigüidade ( $AQ_1$ ): uma especificação de requisitos de software é considerada não ambígua se e somente se todos os requisitos nela declarados têm somente uma possível interpretação por seus avaliadores.

$$AQ_1 = 10(n_{ui} / n_r)$$

onde,  $n_{ui}$  é o número de interpretações idênticas obtidas entre os avaliadores<sup>2</sup> e  $n_r$  é o número de requisitos identificados na especificação.

2. Consistência Interna ( $AQ_2$ ): uma especificação de requisitos de software possui consistência interna se nela não for encontrado não determinismos, ou seja, que não ocorrem “respostas” diferentes do software para situações idênticas de uso do mesmo.

$$AQ_2 = 10(n_u - n_n / n_u)$$

onde,  $n_u$  é o número de funções únicas especificadas e  $n_n$  é o número de funções únicas não determinísticas.

3. Corretitude ( $AQ_3$ ): é o grau de requisitos corretos identificados na especificação de requisitos de software. Um requisito é considerado correto se ele representa algo necessário para o software a ser construído.

$$AQ_3 = 10(n_c / n_r)$$

onde,  $n_c$  é o número de requisitos corretos obtido pelos avaliadores e  $n_r$  é o número total de requisitos (validados e não validados).

4. Anotação por Importância Relativa ( $AQ_4$ ): uma especificação de requisitos de software possui anotação por importância relativa se e somente se qualquer leitor puder facilmente determinar quais requisitos são de maior importância para o usuário/cliente do software.

$$AQ_4 = 10(n_{rair} / n_r)$$

---

<sup>1</sup>  $V^A$  regras, onde  $V$  é o número de valores lingüísticos que um atributo de qualidade pode assumir, e  $A$  é a quantidade de atributos de qualidade ( $3^6 = 729$  regras).

<sup>2</sup> Os avaliadores são os especialistas do domínio do problema e os engenheiros de requisitos.

onde,  $n_{rair}$  é o número de requisitos anotados por importância relativa e  $n_r$  é o número total de requisitos.

5. Anotação por Estabilidade Relativa ( $AQ_5$ ): uma especificação de requisitos de software possui anotação por estabilidade relativa se qualquer leitor puder facilmente determinar quais requisitos estão muito propensos a mudanças, quais estão razoavelmente propensos a mudanças e quais estão pouco propensos a mudanças.

$$AQ_5 = 10(n_{raer} / n_r)$$

onde,  $n_{raer}$  é o número de requisitos anotados por estabilidade relativa e  $n_r$  é o número total de requisitos.

6. Anotação por Versão ( $AQ_6$ ): uma especificação de requisitos de software possui anotação por versão se qualquer leitor puder facilmente determinar quais requisitos estarão satisfeitos em quais versões do produto (software).

$$AQ_6 = 10(n_v / n_r)$$

onde,  $n_v$  é o número de requisitos anotados por versão e  $n_r$  é o número total de requisitos.

### 3. O Modelo de Avaliação Proposto

O modelo de avaliação de uma especificação semi-formal de requisitos que apresentamos neste trabalho está dividido em duas camadas, conforme mostrado na figura 1.

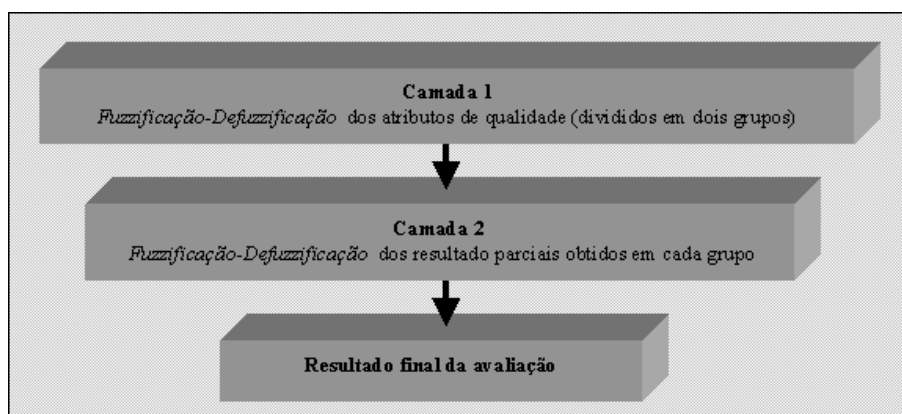


Figura 1. Visão geral do modelo de avaliação proposto

Na camada 1, os atributos de qualidade são tratados em dois grupos separados, cada grupo contendo três atributos. O grupo 1 é formado pelos atributos não ambigüidade, corretitude e consistência interna, e o grupo 2 é formado pelos atributos anotação por importância relativa, anotação por estabilidade relativa e anotação por versão. A divisão dos atributos em dois grupos foi uma alternativa encontrada para a redução do número de regras. Para cada grupo foram criadas 27 regras, combinando todas as associações possíveis entre os três atributos de qualidade. No grupo 1 foram colocados os atributos de maior importância para a avaliação, e no grupo 2 os de importância menor. Assim, cada grupo de atributos passa pelo processo de *fuzzificação*, inferência e *defuzzificação*, gerando um resultado numérico para cada grupo (resultados parciais do processo).

Os resultados numéricos gerados a partir dos grupos são as entradas necessárias para a camada 2. Assim, a camada 2 junta os resultados parciais gerados pela camada 1, os quais passam também pelo processo de *fuzzificação*, inferência e *defuzzificação*. O resultado gerado pela camada 2 é o resultado final do modelo de avaliação. Os procedimentos executados em cada camada serão explicados na seções seguintes.

Para que o modelo de avaliação proposto possa ser melhor entendido, vamos exemplificar os procedimentos de cada camada considerando uma avaliação numérica (usando valores discretos) feita para uma especificação de requisitos hipotética. A *Tabela 1* fornece os valores discretos para os seis atributos de qualidade considerados no modelo de avaliação.

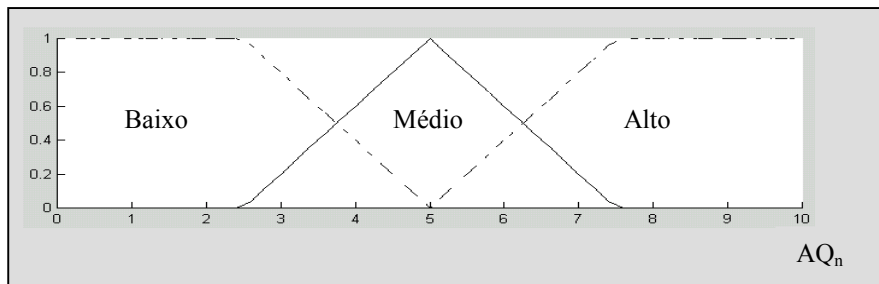
<i>Atributos de Qualidade</i>	<i>Identificadores</i>	<i>Valores Discretos</i>
<i>Não Ambigüidade</i>	<i>AQ<sub>1</sub></i>	7
<i>Consistência Interna</i>	<i>AQ<sub>2</sub></i>	5
<i>Corretitude</i>	<i>AQ<sub>3</sub></i>	3,5
<i>Anotação por Importância Relativa</i>	<i>AQ<sub>4</sub></i>	9
<i>Anotação por Estabilidade Relativa</i>	<i>AQ<sub>5</sub></i>	6,5
<i>Anotação por Versão</i>	<i>AQ<sub>6</sub></i>	2

**Tabela 1.** Identificadores e valores discretos dos atributos de qualidade

### 3.1 O Processo de *Fuzzificação* da Primeira Camada

O processo de *fuzzificação* consiste na transformação de valores discretos para valores nebulosos. Nesse processo calcula-se o grau de pertinência que um valor discreto possui em relação a um conjunto nebuloso (um valor discreto pode pertencer a vários conjuntos nebulosos, com graus de pertinência distintos em cada um deles). O cálculo do grau de pertinência é feito com o uso de funções de pertinência (vide *Figura 2*).

Para cada atributo de qualidade foram definidas três funções de pertinência que representam os valores lingüísticos baixo, médio e alto, dessa forma os valores lingüísticos passam a ser tratados como conjuntos nebulosos. O domínio de valores para os conjuntos nebulosos baixo, médio e alto são os mesmos para os seis atributos de qualidade. A *Figura 2* apresenta as funções de pertinência adotadas.



**Figura 2.** Funções de pertinência para os atributos de qualidade (AQ<sub>n</sub>)

Aplicando as funções de pertinência aos valores discretos dos atributos de qualidade apresentados na tabela 1 obtemos os seguintes valores *fuzzificados*:

$AQ_1$ é <i>médio</i> com grau de pertinência 0,2	$AQ_4$ é <i>alto</i> com grau de pertinência 1
$AQ_1$ é <i>alto</i> com grau de pertinência 0,8	$AQ_5$ é <i>médio</i> com grau de pertinência 0,4
$AQ_2$ é <i>médio</i> com grau de pertinência 1	$AQ_6$ é <i>alto</i> com grau de pertinência 0,6
$AQ_3$ é <i>médio</i> com grau de pertinência 0,4	$AQ_6$ é <i>baixo</i> com grau de pertinência 1
$AQ_3$ é <i>baixo</i> com grau de pertinência 0,6	

### 3.2 O Método de Inferência da Primeira Camada

O método de inferência é o processo de tratamento dos valores *fuzzificados* na entrada do modelo, e que irá “manipular” diversos conjuntos nebulosos até chegar num único conjunto nebuloso de saída (resultado de várias operações sobre os conjuntos nebulosos envolvidos no modelo proposto).

O método de inferência adotado está baseado em regras de associação. Para a primeira camada foram criadas 54 regras de associação, 27 para cada grupo de atributos de qualidade, conforme mostrado na *Tabela 2*.

Cada regra definida na tabela 2 possui três atributos de qualidade na parte antecedente, conectados pelo operador lógico *e*. A simbologia utilizada para representação das regras possui o seguinte significado:

*B* – Baixo                      *M* – Médio                      *A* – Alto  
*MB* – Medianamente Baixo    *MA* – Medianamente Alto  
*QEG*<sub>1</sub> – Identificador para Qualidade da Especificação no Grupo 1  
*QEG*<sub>2</sub> – Identificador para Qualidade da Especificação no Grupo 2

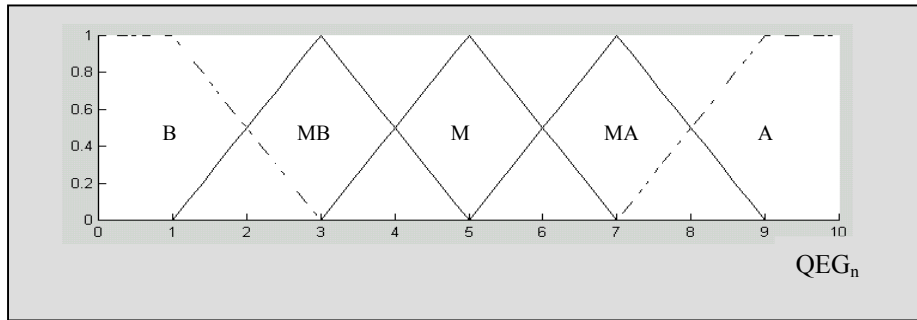
<b>Regras de Associação do Grupo 1</b>	<b>Regras de Associação do Grupo 2</b>
$R_1: \text{Se } AQ_1 = B \text{ e } AQ_2 = B \text{ e } AQ_3 = B \text{ então } QEG_1 = B$	$R_{28}: \text{Se } AQ_4 = B \text{ e } AQ_5 = B \text{ e } AQ_6 = B \text{ então } QEG_2 = B$
$R_2: \text{Se } AQ_1 = B \text{ e } AQ_2 = B \text{ e } AQ_3 = M \text{ então } QEG_1 = B$	$R_{29}: \text{Se } AQ_4 = B \text{ e } AQ_5 = B \text{ e } AQ_6 = M \text{ então } QEG_2 = B$
$R_3: \text{Se } AQ_1 = B \text{ e } AQ_2 = B \text{ e } AQ_3 = A \text{ então } QEG_1 = MB$	$R_{30}: \text{Se } AQ_4 = B \text{ e } AQ_5 = B \text{ e } AQ_6 = A \text{ então } QEG_2 = MB$
$R_4: \text{Se } AQ_1 = B \text{ e } AQ_2 = M \text{ e } AQ_3 = B \text{ então } QEG_1 = B$	$R_{31}: \text{Se } AQ_4 = B \text{ e } AQ_5 = M \text{ e } AQ_6 = B \text{ então } QEG_2 = B$
$R_5: \text{Se } AQ_1 = B \text{ e } AQ_2 = M \text{ e } AQ_3 = M \text{ então } QEG_1 = M$	$R_{32}: \text{Se } AQ_4 = B \text{ e } AQ_5 = M \text{ e } AQ_6 = M \text{ então } QEG_2 = M$
$R_6: \text{Se } AQ_1 = B \text{ e } AQ_2 = M \text{ e } AQ_3 = A \text{ então } QEG_1 = M$	$R_{33}: \text{Se } AQ_4 = B \text{ e } AQ_5 = M \text{ e } AQ_6 = A \text{ então } QEG_2 = M$
$R_7: \text{Se } AQ_1 = B \text{ e } AQ_2 = A \text{ e } AQ_3 = B \text{ então } QEG_1 = MB$	$R_{34}: \text{Se } AQ_4 = B \text{ e } AQ_5 = A \text{ e } AQ_6 = B \text{ então } QEG_2 = MB$
$R_8: \text{Se } AQ_1 = B \text{ e } AQ_2 = A \text{ e } AQ_3 = M \text{ então } QEG_1 = M$	$R_{35}: \text{Se } AQ_4 = B \text{ e } AQ_5 = A \text{ e } AQ_6 = M \text{ então } QEG_2 = M$
$R_9: \text{Se } AQ_1 = B \text{ e } AQ_2 = A \text{ e } AQ_3 = A \text{ então } QEG_1 = MA$	$R_{36}: \text{Se } AQ_4 = B \text{ e } AQ_5 = A \text{ e } AQ_6 = A \text{ então } QEG_2 = MA$
$R_{10}: \text{Se } AQ_1 = M \text{ e } AQ_2 = B \text{ e } AQ_3 = B \text{ então } QEG_1 = B$	$R_{37}: \text{Se } AQ_4 = M \text{ e } AQ_5 = B \text{ e } AQ_6 = B \text{ então } QEG_2 = B$
$R_{11}: \text{Se } AQ_1 = M \text{ e } AQ_2 = B \text{ e } AQ_3 = M \text{ então } QEG_1 = MB$	$R_{38}: \text{Se } AQ_4 = M \text{ e } AQ_5 = B \text{ e } AQ_6 = M \text{ então } QEG_2 = MB$
$R_{12}: \text{Se } AQ_1 = M \text{ e } AQ_2 = B \text{ e } AQ_3 = A \text{ então } QEG_1 = M$	$R_{39}: \text{Se } AQ_4 = M \text{ e } AQ_5 = B \text{ e } AQ_6 = A \text{ então } QEG_2 = M$
$R_{13}: \text{Se } AQ_1 = M \text{ e } AQ_2 = M \text{ e } AQ_3 = B \text{ então } QEG_1 = M$	$R_{40}: \text{Se } AQ_4 = M \text{ e } AQ_5 = M \text{ e } AQ_6 = B \text{ então } QEG_2 = M$
$R_{14}: \text{Se } AQ_1 = M \text{ e } AQ_2 = M \text{ e } AQ_3 = M \text{ então } QEG_1 = M$	$R_{41}: \text{Se } AQ_4 = M \text{ e } AQ_5 = M \text{ e } AQ_6 = M \text{ então } QEG_2 = M$
$R_{15}: \text{Se } AQ_1 = M \text{ e } AQ_2 = M \text{ e } AQ_3 = A \text{ então } QEG_1 = MA$	$R_{42}: \text{Se } AQ_4 = M \text{ e } AQ_5 = M \text{ e } AQ_6 = A \text{ então } QEG_2 = MA$
$R_{16}: \text{Se } AQ_1 = M \text{ e } AQ_2 = A \text{ e } AQ_3 = B \text{ então } QEG_1 = MA$	$R_{43}: \text{Se } AQ_4 = M \text{ e } AQ_5 = A \text{ e } AQ_6 = B \text{ então } QEG_2 = MA$
$R_{17}: \text{Se } AQ_1 = M \text{ e } AQ_2 = A \text{ e } AQ_3 = M \text{ então } QEG_1 = M$	$R_{44}: \text{Se } AQ_4 = M \text{ e } AQ_5 = A \text{ e } AQ_6 = M \text{ então } QEG_2 = M$
$R_{18}: \text{Se } AQ_1 = M \text{ e } AQ_2 = A \text{ e } AQ_3 = A \text{ então } QEG_1 = A$	$R_{45}: \text{Se } AQ_4 = M \text{ e } AQ_5 = A \text{ e } AQ_6 = A \text{ então } QEG_2 = A$
$R_{19}: \text{Se } AQ_1 = A \text{ e } AQ_2 = B \text{ e } AQ_3 = B \text{ então } QEG_1 = MB$	$R_{46}: \text{Se } AQ_4 = A \text{ e } AQ_5 = B \text{ e } AQ_6 = B \text{ então } QEG_2 = MB$
$R_{20}: \text{Se } AQ_1 = A \text{ e } AQ_2 = B \text{ e } AQ_3 = M \text{ então } QEG_1 = M$	$R_{47}: \text{Se } AQ_4 = A \text{ e } AQ_5 = B \text{ e } AQ_6 = M \text{ então } QEG_2 = M$
$R_{21}: \text{Se } AQ_1 = A \text{ e } AQ_2 = B \text{ e } AQ_3 = A \text{ então } QEG_1 = MA$	$R_{48}: \text{Se } AQ_4 = A \text{ e } AQ_5 = B \text{ e } AQ_6 = A \text{ então } QEG_2 = MA$
$R_{22}: \text{Se } AQ_1 = A \text{ e } AQ_2 = M \text{ e } AQ_3 = B \text{ então } QEG_1 = M$	$R_{49}: \text{Se } AQ_4 = A \text{ e } AQ_5 = M \text{ e } AQ_6 = B \text{ então } QEG_2 = M$
$R_{23}: \text{Se } AQ_1 = A \text{ e } AQ_2 = M \text{ e } AQ_3 = M \text{ então } QEG_1 = MA$	$R_{50}: \text{Se } AQ_4 = A \text{ e } AQ_5 = M \text{ e } AQ_6 = M \text{ então } QEG_2 = MA$
$R_{24}: \text{Se } AQ_1 = A \text{ e } AQ_2 = M \text{ e } AQ_3 = A \text{ então } QEG_1 = A$	$R_{51}: \text{Se } AQ_4 = A \text{ e } AQ_5 = M \text{ e } AQ_6 = A \text{ então } QEG_2 = A$
$R_{25}: \text{Se } AQ_1 = A \text{ e } AQ_2 = A \text{ e } AQ_3 = B \text{ então } QEG_1 = MA$	$R_{52}: \text{Se } AQ_4 = A \text{ e } AQ_5 = A \text{ e } AQ_6 = B \text{ então } QEG_2 = MA$
$R_{26}: \text{Se } AQ_1 = A \text{ e } AQ_2 = A \text{ e } AQ_3 = M \text{ então } QEG_1 = A$	$R_{53}: \text{Se } AQ_4 = A \text{ e } AQ_5 = A \text{ e } AQ_6 = M \text{ então } QEG_2 = A$
$R_{27}: \text{Se } AQ_1 = A \text{ e } AQ_2 = A \text{ e } AQ_3 = A \text{ então } QEG_1 = A$	$R_{54}: \text{Se } AQ_4 = A \text{ e } AQ_5 = A \text{ e } AQ_6 = A \text{ então } QEG_2 = A$

**Tabela 2.** Base de regras utilizada no processo de inferência da primeira camada

Para os valores *fuzzificados* do exemplo dado, as seguintes regras de associação são ativadas:  $R_{13}$ ,  $R_{14}$ ,  $R_{22}$  e  $R_{23}$  do grupo 1;  $R_{49}$  e  $R_{52}$  do grupo 2. Cada regra ativada deverá sofrer um processo de conjunção dos antecedentes, onde utilizamos o operador *mínimo* (*min*) para a execução deste procedimento [5]. Os operandos envolvidos são os graus de pertinência correspondentes dos atributos de qualidade em relação aos conjuntos nebulosos baixo, médio e alto. Assim, temos os seguintes vetores de pertinência para as regras envolvidas:

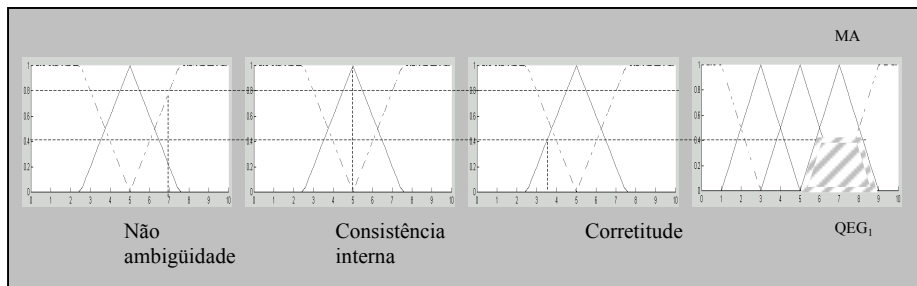
$$\begin{aligned}
 R_{13} &= \{0,2 \ 1 \ 0,6\}, \text{ onde } \min(0,2 \ 1 \ 0,6) = 0,2 & R_{22} &= \{0,8 \ 1 \ 0,6\}, \text{ onde } \min(0,8 \ 1 \ 0,6) = 0,6 \\
 R_{14} &= \{0,2 \ 1 \ 0,4\}, \text{ onde } \min(0,2 \ 1 \ 0,4) = 0,2 & R_{23} &= \{0,8 \ 1 \ 0,4\}, \text{ onde } \min(0,8 \ 1 \ 0,4) = 0,4 & \textbf{Grupo 1} \\
 R_{49} &= \{1 \ 0,4 \ 1\}, \text{ onde } \min(1 \ 0,4 \ 1) = 0,4 & R_{52} &= \{1 \ 0,6 \ 1\}, \text{ onde } \min(1 \ 0,6 \ 1) = 0,6 & \textbf{Grupo 2}
 \end{aligned}$$

Após a conjunção dos antecedentes deve ser realizada a inferência de cada regra ativada. A realização da inferência envolverá o resultado da conjunção dos antecedentes e o conjunto nebuloso correspondente ao valor lingüístico definido na parte conseqüente da regra [5]. As funções de pertinência que representam os conjuntos nebulosos baixo (B), medianamente baixo (MB), médio (M), medianamente alto (MA) e alto (A) são apresentadas na *Figura 3*.



**Figura 3.** Funções de pertinência para a qualidade da especificação em cada grupo (QEG<sub>n</sub>)

A inferência na regra é realizada através de uma *t\_norma*<sup>3</sup>. Adotamos como *t\_norma* para o processo de inferência novamente o operador *mínimo (min)* [5]. Assim, para cada regra ativada, será aplicada a *t\_norma min* entre o resultado da conjunção dos antecedentes da regra e o conjunto nebuloso correspondente ao valor lingüístico do conseqüente da regra. A *Figura 4* ilustra este processo para a *R<sub>23</sub>*.



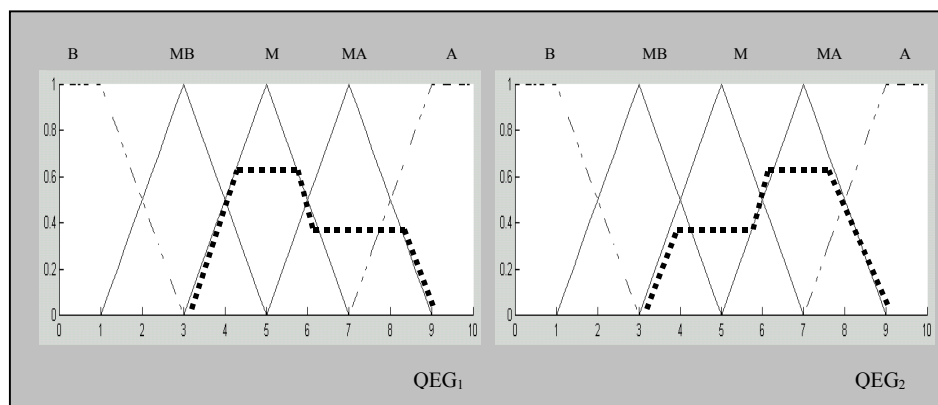
**Figura 4.** Processo de inferência utilizando a *t\_norma min* entre o grau de pertinência resultante da conjunção dos antecedentes (0,4 neste exemplo) e o conjunto nebuloso do conseqüente da regra (MA)

A área hachurada que aparece no gráfico da *figura 4* é o conjunto nebuloso resultante da inferência aplicando a *t\_norma min*. Este processo de inferência ocorrerá em cada regra ativada.

O procedimento final do método de inferência adotado é a agregação de todos os conjuntos nebulosos resultantes da inferência realizada em cada regra ativada. Utilizamos o operador *máximo (max)* para realizar a agregação, que resultará na união de todos os conjuntos nebulosos resultantes do processo de inferência das regras ativadas. A *figura 5* ilustra o resultado do método de inferência da primeira camada aplicado ao exemplo adotado.

<sup>3</sup> Uma *t\_norma* permite a realização de uma operação entre conjuntos nebulosos, como por exemplo a intersecção e união entre conjuntos nebulosos.





**Figura 5.** Resultado do método de inferência da primeira camada

### 3.3 O Processo de *Defuzzificação* da Primeira Camada

O processo de *defuzzificação* consiste em transformar o conjunto nebuloso resultante do processo de inferência em um valor discreto. O processo de defuzzificação adotado na primeira camada é pelo *Centro do Máximo (C\_o\_M)* [6]. A equação abaixo expressa como obter o valor *defuzzificado* pelo método do *C-o-M* [5].

$$F^{-1}(x) = \frac{\sum_{i=1}^n A_i(x) a_i}{\sum_{i=1}^n A_i(x)}$$

onde,  $n$  é o número de conjuntos nebulosos envolvidos na agregação,  $A_i(x)$  é a altura de cada conjunto nebuloso e  $a_i$  é o valor do domínio onde se encontra o pico da função de pertinência que representa o valor lingüístico no conseqüente da regra. Assim, aplicamos este método de *defuzzificação* para cada grupo de atributos de qualidade. Para o exemplo apresentado obtemos os seguintes resultados:

$$F^{-1}(qeg_1) = \frac{0,2.5 + 0,2.5 + 0,6.5 + 0,4.7}{0,2 + 0,2 + 0,4 + 0,6} = 5,57$$

$$F^{-1}(qeg_2) = \frac{0,4.5 + 0,6.7}{0,4 + 0,6} = 6,2$$

Portanto, como resultado *defuzzificado* da primeira camada temos que a qualidade da especificação de requisitos no grupo 1 ( $qeg_1$ ) é 5,57 e a qualidade da especificação de requisitos no grupo 2 ( $qeg_2$ ) é 6,2.

### 3.4 O Processo de *Fuzzificação* da Segunda Camada

Os valores *defuzzificados* de cada grupo obtido na primeira camada (qualidade da especificação em cada grupo) são utilizados como entrada para os procedimentos da segunda camada. Assim, as funções de pertinência adotadas para a *fuzzificação* dos valores de entrada da segunda camada são as mesmas adotadas como saída no método de inferência da primeira camada (*Figura 3*). Continuando o exemplo trabalhado na primeira camada, temos agora os seguintes valores *fuzzificados*:

- $QEG_1$  é *média* com grau de pertinência 0,715
- $QEG_1$  é *medianamente alta* com grau de pertinência 0,285
- $QEG_2$  é *média* com grau de pertinência 0,4
- $QEG_2$  é *medianamente alta* com grau de pertinência 0,6

### 3.5 O Método de Inferência da Segunda Camada

O método de inferência da segunda camada também está baseado em regras de associação. A base de regras utilizada é apresentada na *Tabela 3*.

$R_{55}$ : Se $QEG_1 = B$ e $QEG_2 = B$ então $QFE = B$	$R_{68}$ : Se $QEG_1 = M$ e $QEG_2 = MA$ então $QFE = M$
$R_{56}$ : Se $QEG_1 = B$ e $QEG_2 = MB$ então $QFE = B$	$R_{69}$ : Se $QEG_1 = M$ e $QEG_2 = A$ então $QFE = MA$
$R_{57}$ : Se $QEG_1 = B$ e $QEG_2 = M$ então $QFE = MB$	$R_{70}$ : Se $QEG_1 = MA$ e $QEG_2 = B$ então $QFE = M$
$R_{58}$ : Se $QEG_1 = B$ e $QEG_2 = MA$ então $QFE = MB$	$R_{71}$ : Se $QEG_1 = MA$ e $QEG_2 = MB$ então $QFE = M$
$R_{59}$ : Se $QEG_1 = B$ e $QEG_2 = A$ então $QFE = M$	$R_{72}$ : Se $QEG_1 = MA$ e $QEG_2 = M$ então $QFE = MA$
$R_{60}$ : Se $QEG_1 = MB$ e $QEG_2 = B$ então $QFE = MB$	$R_{73}$ : Se $QEG_1 = MA$ e $QEG_2 = MA$ então $QFE = MA$
$R_{61}$ : Se $QEG_1 = MB$ e $QEG_2 = MB$ então $QFE = MB$	$R_{74}$ : Se $QEG_1 = MA$ e $QEG_2 = A$ então $QFE = MA$
$R_{62}$ : Se $QEG_1 = MB$ e $QEG_2 = M$ então $QFE = MB$	$R_{75}$ : Se $QEG_1 = A$ e $QEG_2 = B$ então $QFE = M$
$R_{63}$ : Se $QEG_1 = MB$ e $QEG_2 = MA$ então $QFE = M$	$R_{76}$ : Se $QEG_1 = A$ e $QEG_2 = MB$ então $QFE = MA$
$R_{64}$ : Se $QEG_1 = MB$ e $QEG_2 = A$ então $QFE = M$	$R_{77}$ : Se $QEG_1 = A$ e $QEG_2 = M$ então $QFE = MA$
$R_{65}$ : Se $QEG_1 = M$ e $QEG_2 = B$ então $QFE = MB$	$R_{78}$ : Se $QEG_1 = A$ e $QEG_2 = MA$ então $QFE = A$
$R_{66}$ : Se $QEG_1 = M$ e $QEG_2 = MB$ então $QFE = M$	$R_{79}$ : Se $QEG_1 = A$ e $QEG_2 = A$ então $QFE = A$
$R_{67}$ : Se $QEG_1 = M$ e $QEG_2 = M$ então $QFE = M$	

**Tabela 3.** Base de regras utilizada no processo de inferência da segunda camada

A tabela 3 combina todas as possibilidades de valores lingüísticos entre as qualidades de especificação dos dois grupos. No exemplo dado, os valores *fuzzificados* ativam as seguintes regras:  $R_{67}$ ,  $R_{68}$ ,  $R_{72}$  e  $R_{73}$ . A conjunção dos antecedentes da regra também é feita utilizando o operador *min*. Assim, temos os seguintes vetores de pertinência para as regras ativadas:

$$R_{67} = \{0,715 \ 0,4\}, \text{ onde } \min(0,715 \ 0,4) = 0,4$$

$$R_{72} = \{0,285 \ 0,4\}, \text{ onde } \min(0,285 \ 0,4) = 0,285$$

$$R_{68} = \{0,715 \ 0,6\}, \text{ onde } \min(0,715 \ 0,6) = 0,6$$

$$R_{73} = \{0,285 \ 0,6\}, \text{ onde } \min(0,285 \ 0,6) = 0,285$$

Após a conjunção dos antecedentes das regras, deve ser realizada a inferência em cada regra. Assim como na primeira camada, a *t norma min* também foi adotada na segunda camada do modelo. Os conjuntos nebulosos inferidos de cada regra ativada serão agregados pelo operador *max*, fazendo a união dos conjuntos nebulosos. A

Figura 6 apresenta o resultado após a agregação na segunda camada do processo de avaliação.

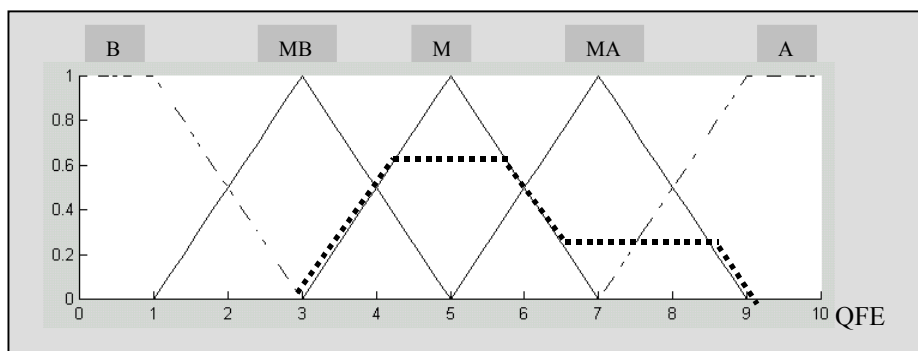


Figura 6. Resultado do método de inferência da segunda camada

### 3.6 O Processo de Defuzzificação da Segunda Camada

O método de *defuzzificação* da segunda camada também é pelo *Centro-do-Máximo (C-o-M)*. Aplicando o *C-o-M* no exemplo, obtemos o seguinte resultado:

$$F^{-1}(qfe) = \frac{0,4 \cdot 4,5 + 0,6 \cdot 6,5 + 0,285 \cdot 7 + 0,285 \cdot 7}{0,4 + 0,6 + 0,285 + 0,285} = 5,73$$

Portanto, para o exemplo adotado, como resultado final do processo de avaliação, temos que o valor *defuzzificado* é 5,73, o que equivale a dizer que a qualidade final da especificação de requisitos avaliada (*QFE*) é média com grau de pertinência 0,63.

### Conclusão

O modelo de avaliação de especificação de requisitos apresentado flexibiliza os valores dos atributos de qualidade através da *fuzzificação* dos mesmos, o que acreditamos ser mais adequado do que uma avaliação rígida baseada em valores discretos. Defendemos a idéia de que um modelo de avaliação de especificação de requisitos deve levar em conta o alto grau de incerteza que se tem sobre os requisitos no início do processo de desenvolvimento do software, e o alto grau de imprecisão, que conseqüentemente passa a existir, sobre a avaliação de um documento de especificação de tais requisitos. Isso torna-se possível quando trabalhamos com um modelo de avaliação que utiliza valores linguísticos como baixo, médio e alto para a avaliação dos atributos de qualidade da especificação de requisitos.

Várias especificações de requisitos representadas através de diagramas de fluxo de dados e diagramas entidade-relacionamento foram avaliadas adotando o modelo apresentado, com a obtenção de resultados mais satisfatórios do que aqueles obtidos numa avaliação rígida, baseada apenas em valores discretos.

Uma restrição com relação à adoção de regras de associação como método de inferência para o modelo proposto é o número elevado de regras que precisam ser definidas para cobrir todas as possíveis combinações de entrada dos atributos de qualidade [7]. Uma saída apontada foi a divisão dos atributos em grupos, criando camadas de filtragem durante o processo de avaliação. Para seis atributos de qualidade conseguimos reduzir de 729 para 79 o número de regras necessárias, obtendo um grau de confiança satisfatório para os resultados obtidos, conforme observamos no exemplo apresentado. Esta divisão em grupos é uma possível solução para que o modelo possa cobrir um número maior de atributos de qualidade.

No momento, está em curso a implementação de um software que automatize o modelo de avaliação proposto.

## Referências Bibliográficas

- [1] Jacobson, I., Booch, G. and Rumbaugh, J., *The Unified Software Development Process*, Addison-Wesley, 1998.
- [2] Pressman, R. S., *Engenharia de Software*, 3ª Edition, Makron Books, 1995.
- [3] Davis, A. et al., "Identifying and Measuring Quality in a Software Requirements Specification" in *Software Requirements Engineering*, 2<sup>nd</sup> Ed., IEEE CS Press, 1997, pp 164-175.
- [4] Institute of Electrical and Electronics Engineers, *IEEE Guide to Software Requirements Specifications*, Standard 830-1984, New York: IEEE Computer Society Press, 1984.
- [5] Pedrycz, W. and Gomide, F., *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press, 1998.
- [6] Shaw, I. S. e Simões, M. G., *Controle e Modelagem Fuzzy*, Editora Edgard Blücher Ltda, Co-edição com FAPESP, 1999.
- [7] Combs, W. E. et al., *Combinatorial Rule Explosion Eliminated by a Fuzzy Rule ConFfiguration*, IEEE Transactions on Fuzzy Systems, Vol. 6, N. 1, Feb 1998.