



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
BACHARELADO EM ENGENHARIA DE SOFTWARE

DAVI CEDRAZ SILVA DE OLIVEIRA

**AGREGAÇÃO DE MEDIDAS PARA AVALIAÇÃO DA MANUTENIBILIDADE DO
MODELO DE *FEATURES* EM LINHAS DE PRODUTO DE SOFTWARE**

QUIXADÁ

2018

DAVI CEDRAZ SILVA DE OLIVEIRA

AGREGAÇÃO DE MEDIDAS PARA AVALIAÇÃO DA MANUTENIBILIDADE DO
MODELO DE *FEATURES* EM LINHAS DE PRODUTO DE SOFTWARE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia de Software
do Campus Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Engenharia de Software.

Orientadora: Prof^a. Dra. Carla Ilane Mo-
reira Bezerra

QUIXADÁ

2018

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- O46a Oliveira, Davi Cedraz Silva de.
Agregação de medidas para avaliação da manutenibilidade do modelo de features em linhas de produto de software / Davi Cedraz Silva de Oliveira. – 2018.
68 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Software, Quixadá, 2018.
Orientação: Profa. Dra. Carla Ilane Moreira Bezerra.
1. Engenharia de linha de produto de software. 2. Modelo de características. 3. Lógica difusa. 4. Software - Controle de qualidade. I. Título.

CDD 005.1

DAVI CEDRAZ SILVA DE OLIVEIRA

AGREGAÇÃO DE MEDIDAS PARA AVALIAÇÃO DA MANUTENIBILIDADE DO
MODELO DE *FEATURES* EM LINHAS DE PRODUTO DE SOFTWARE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia de Software
do Campus Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Engenharia de Software.

Aprovada em: __/__/____

BANCA EXAMINADORA

Prof^a. Dra. Carla Ilane Moreira
Bezerra (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. José Maria da Silva Monteiro Filho
Universidade Federal do Ceará (UFC)

Prof^a. Ma. Rainara Maia Carvalho
Universidade Federal do Ceará (UFC)

Prof. Dr. Ismayle de Sousa Santos
Universidade Federal do Ceará (UFC)

À toda minha família, em especial aos meus pais, meus maiores exemplos, que acreditaram e investiram em mim.

AGRADECIMENTOS

Primeiramente a Deus, pela sua presença, cuidado e amor nos momentos difíceis e nas conquistas.

Aos meus pais, pela confiança, amor e dedicação, que apesar das dificuldades sempre confiaram na minha capacidade, mesmo que eu as vezes não o fizesse.

A minha noiva, pelo enorme carinho e companheirismo, apoiando-me nos desafios e me confortando em todas as dificuldades, dando-me alegria diariamente.

À Prof^a. Dra. Carla Ilane Moreira Bezerra, pela confiança e por sua excelente orientação, tornando a experiência da graduação ainda mais positiva. Também pela mentoria e oportunidades oferecidas, que contribuíram decisivamente para minha formação.

Aos professores participantes da banca examinadora José Maria da Silva Monteiro Filho, Rainara Maia Carvalho e Ismayle de Sousa Santos pelo tempo dedicado, pelas colaborações e sugestões.

Aos especialistas de domínio, pelo tempo concedido e enorme contribuição ao conduzir a revisão por pares neste trabalho.

A todos os meus professores nesse período, que sem exceção, fizeram de cada conhecimento transmitido algo enriquecedor que levarei para sempre.

À Universidade Federal do Ceará, Campus Quixadá, por superar minhas expectativas e fornecer um ambiente acolhedor, com professores excelentes, ensino de alta qualidade e amigos que me proporcionaram uma experiência única.

Aos meus colegas e amigos, que motivaram-me e alegraram-se com mais esta conquista. Em especial Samuel Alves, Ernandes Azevedo, Rodrigo Almeida, Brian Viana, Gleydson Rodrigues e Leo Jaimesson pela amizade e trabalho em equipe durante toda a graduação.

“A quem me pergunta se sou pessimista ou otimista, respondo que o meu conhecimento é de pessimista, mas a minha vontade e a minha esperança são de otimista.”

(Albert Schweitzer)

RESUMO

Linha de Produto de Software (LPS) é uma abordagem que propõe a reutilização sistemática de artefatos e componentes para o desenvolvimento de sistemas que compartilham um conjunto comum de características, ou seja, que pertencem à um mesmo domínio ou família. A variabilidade destas características comuns em uma LPS podem ser gerenciadas por meio do modelo de *features*, um artefato que consiste em uma diagrama em formato de árvore, descrevendo as características (*features*) identificadas nos produtos e as possíveis relações entre elas. Neste contexto, garantir a qualidade do modelo de *features* pode ser essencial para garantir que os erros não se propaguem por todos os produtos de uma LPS. O processo de avaliação da qualidade de um produto ou artefato pode ser feito por meio de medidas, que podem refletir as características, subcaracterísticas ou atributos de qualidade. Porém, os valores isolados de cada medida não permitem acessar uma característica de qualidade do modelo de *features* como um todo, pois a maioria delas abrange diversos aspectos específicos que não estão correlacionados. Desta forma, é possível agregar estas medidas e acessar as respectivas subcaracterísticas por meio de uma única medida agregada que possui as mesmas informações disponíveis de um conjunto de medidas. Neste contexto, este trabalho propõe realizar a agregação de medidas para avaliar a manutenibilidade do modelo de *features* em LPS. Para isso, foi utilizada a teoria de Lógica Fuzzy como técnica para agregação destas medidas, a partir de parâmetros e regras definidas com ajuda de especialistas. Foram selecionadas medidas do catálogo COffEE para compor o conjunto de entrada do processo de agregação resultando um subconjunto do catálogo, o MiniCOffEE, contendo 15 de medidas de manutenibilidade. Como resultado deste trabalho, foi obtida uma nova medida agregada que representa o índice de manutenibilidade de um modelo de *features* (MIFM). Além disso, houve a extensão do *dataset* MACchiATO, adicionando novos modelos de *features* ao repositório e os valores coletados de cada uma das medidas. O MIFM foi aplicado a um conjunto de modelos a fim de avaliar o seu uso. Constatou-se que o valor agregado obtido permite mensurar se um modelo de *features* possui alto ou baixo índice de manutenibilidade, auxiliando o engenheiro de domínio na avaliação da manutenibilidade do modelo de uma forma mais rápida e precisa.

Palavras-chave: Medidas. Linhas de Produto de Software. Avaliação de Qualidade. Modelo de *Features*. Lógica Fuzzy.

ABSTRACT

Software Product Line (LPS) is an approach that proposes the systematic reuse of artifacts and components for development of systems that share a common set of characteristics, that is, that belong to the same domain or family. The variability of these common features in an LPS can be managed by a feature model, an artifact that consists of a tree-shaped diagram, that describes the features identified in the products and the possible relationships between them. In this context, guaranteeing the quality of the feature model may be essential to ensure that errors do not propagate across all products of an LPS. The process of evaluating the quality of a product or artifact can be done using measures, which may reflect the characteristics, subcharacteristics or attributes of quality. However, the isolated values of each measure do not allow access to a quality of the feature model as a whole, since most of the measures cover several specific aspects that are not correlated. Thus, it is possible to aggregate these measures and access the respective subcharacteristics by means of a single aggregate measure that has the same available information as a set of measures. In this context, this work proposes to perform the aggregation of measures in order to evaluate the maintainability of the feature model in LPS. For this, the theory of Fuzzy Logic was used as a technique for aggregation of these measures, from parameters and rules defined with the help of specialists. Measures from the COFFEE catalog were selected to compose the input set of the aggregation process resulting in a subset of the catalog, the MiniCOFFEE, containing 15 maintainability measures. As a result of this work, a new aggregate measure that represents the maintainability index of a features (MIFM) model was obtained. In addition, the dataset MACchiATO was extended, adding new feature models to the repository and the new values collected from each of the measures. The MIFM was applied to a set of models in order to evaluate its use. It was verified that the aggregate measure obtained allows to measure if a features model has a high or low maintainability index, supporting the domain engineer in the evaluation of the maintenance of the feature model in a faster and more precise way.

Keywords: Measures. Software Product Lines. Quality Evaluation. Feature Models. Fuzzy Logic.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Modelo de <i>features</i> de um Telefone Móvel | 20 |
| Figura 2 – Visualização e análise do Número de Features de um Web Game na ferramenta DyMMer | 25 |
| Figura 3 – Variável linguística “Idade” | 28 |
| Figura 4 – Estrutura do sistema <i>fuzzy</i> | 29 |
| Figura 5 – Fluxograma das atividades realizadas durante a execução deste trabalho | 33 |
| Figura 6 – Processo de extensão do <i>dataset</i> MACchiATO. | 37 |
| Figura 7 – Função exportação dos valores das medidas para uma planilha Excel na ferramenta DyMMer. | 39 |
| Figura 8 – Intervalos das funções de pertinência para as medidas de Tamanho | 46 |
| Figura 9 – Implementação dos parâmetros para as funções de pertinência das medidas de manutenibilidade | 50 |
| Figura 10 – Função de pertinência para o Índice de Manutenibilidade do Modelo de Features (MIFM) | 52 |
| Figura 11 – Índice de Manutenibilidade do Modelo de <i>Features Mobile Tourist Planner</i> | 54 |
| Figura 12 – Índice de Manutenibilidade do Modelo de <i>Features Emergency Response Application</i> | 55 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 – MiniCOFFEE: subconjunto de 15 medidas selecionadas do catálogo COFFEE e suas fórmulas de cálculo | 22 |
| Tabela 2 – Valor médios das medidas do MiniCOFFEE aplicadas a modelos de <i>features</i> | 40 |
| Tabela 3 – Medidas agrupadas por subcaracterísticas de manutenibilidade | 43 |
| Tabela 4 – Amostra de regras <i>fuzzy</i> para medidas de Tamanho. | 47 |
| Tabela 5 – Amostra de regras <i>fuzzy</i> para medidas de Complexidade. | 48 |
| Tabela 6 – Amostra de regras <i>fuzzy</i> para medidas de Variabilidade. | 48 |
| Tabela 7 – Amostra de regras <i>fuzzy</i> para as medidas de entrada do MIFM. | 51 |
| Tabela 8 – Resultados da aplicação do MIFM em modelos de <i>features</i> | 53 |
| Tabela 9 – Parâmetros para funções de pertinência para medidas de Tamanho. | 64 |
| Tabela 10 – Parâmetros para funções de pertinência para medidas de Complexidade. . . | 65 |
| Tabela 11 – Parâmetros para funções de pertinência para medidas de Variabilidade. . . . | 65 |
| Tabela 12 – Regras <i>fuzzy</i> para medidas de Tamanho. | 66 |
| Tabela 13 – Regras <i>fuzzy</i> para medidas de Complexidade. | 67 |
| Tabela 14 – Regras <i>fuzzy</i> para medidas de Variabilidade. | 68 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------------|--|
| LPS | Linha de Produto de Software |
| COFFEE | <i>CatalOg of measures for Feature modEl quality Evaluation</i> |
| DyMMer | <i>Dynamic Feature Model Tool Based on Measures</i> |
| MiniCOFFEE | <i>Mini CatalOg of measures for Feature modEl quality Evaluation</i> |
| MAcchiATO | <i>MeAsures dATaset for feaTure mOdel</i> |
| NF | Número de <i>Features</i> |
| NM | Número de <i>Features</i> Mandatórias |
| NTop | Número de <i>Features</i> Top |
| NLeaf | Número de <i>Features</i> Folhas |
| DTMax | Profundidade Máxima |
| CogC | Complexidade Cognitiva |
| FEX | Extensibilidade da <i>Feature</i> |
| FoC | Flexibilidade da Configuração |
| SCDF | <i>Features</i> Dependentes Cíclicas Únicas |
| MCDF | <i>Features</i> Dependentes Cíclicas Múltiplas |
| RDen | Densidade do grafo |
| RoV | Taxa de Variabilidade |
| NVC | Número de Configurações Válidas |
| NGOr | Número de Grupos Or |
| NGXOr | Número de Grupos XOr |

SUMÁRIO

| | | |
|-------|--|----|
| 1 | INTRODUÇÃO | 14 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 17 |
| 2.1 | Linha de Produtos de Software | 17 |
| 2.2 | Modelo de <i>Features</i> | 18 |
| 2.3 | Medidas de Manutenibilidade | 21 |
| 2.4 | Avaliação da Manutenibilidade do Modelo de <i>Features</i> | 23 |
| 2.4.1 | <i>Ferramenta e repositório de modelos de features S.P.L.O.T.</i> | 24 |
| 2.4.2 | <i>Ferramenta DyMMer</i> | 24 |
| 2.5 | Agregação de Medidas | 25 |
| 2.5.1 | <i>Lógica Fuzzy</i> | 26 |
| 3 | TRABALHOS RELACIONADOS | 31 |
| 4 | PROCEDIMENTOS METODOLÓGICOS | 33 |
| 4.1 | Seleção das medidas de manutenibilidade | 34 |
| 4.2 | Aplicação das medidas selecionadas a um conjunto de modelos de <i>features</i> | 34 |
| 4.3 | Agrupamento das medidas | 34 |
| 4.4 | Definição de funções de pertinência e regras <i>fuzzy</i> | 35 |
| 4.5 | Validação das funções de pertinência e regras <i>fuzzy</i> | 35 |
| 4.6 | Implementação do procedimento de agregação na ferramenta jFuzzyLogic | 35 |
| 4.7 | Aplicação e análise da medida agregada | 36 |
| 5 | PROCESSO DE EXTENSÃO DO MACCHIATO E APLICAÇÃO DE MEDIDAS | 37 |
| 5.1 | Coleta dos modelos de <i>features</i> | 37 |
| 5.2 | Importação dos modelos de <i>features</i> na ferramenta DyMMer | 38 |
| 5.3 | Extração do valor das medidas | 38 |
| 5.4 | Análise geral | 40 |
| 6 | PROJETO E DEFINIÇÃO DO SISTEMA <i>FUZZY</i> | 41 |
| 6.1 | Agrupamento das medidas | 41 |
| 6.2 | Projeto das funções de pertinência de entrada e saída das medidas e regras <i>fuzzy</i> | 43 |
| 6.3 | Revisão por pares | 44 |

| | | |
|-----|--|----|
| 6.4 | Definição das funções de pertinência | 45 |
| 6.5 | Definição das regras <i>fuzzy</i> | 47 |
| 7 | AGREGAÇÃO DE MEDIDAS DE MANUTENIBILIDADE | 49 |
| 7.1 | Implementação do sistema <i>fuzzy</i> | 49 |
| 7.2 | Avaliação do Índice de Manutenibilidade | 52 |
| 8 | CONSIDERAÇÕES FINAIS | 56 |
| 8.1 | Ameaças à validade | 57 |
| 8.2 | Contribuições | 57 |
| 8.3 | Trabalhos Futuros | 58 |
| | REFERÊNCIAS | 59 |
| | APÊNDICE A – QUESTIONÁRIO DE PERFIL DO ESPECIALISTA . . | 62 |
| | APÊNDICE B – PARÂMETROS DAS FUNÇÕES DE PERTINÊNCIA DO SISTEMA <i>FUZZY</i> | 64 |
| | APÊNDICE C – REGRAS <i>FUZZY</i> PARA OS GRUPOS DE MEDIDAS . | 66 |

1 INTRODUÇÃO

A qualidade é fundamental para o sucesso de um produto de software. Uma das formas de garantir a qualidade do produto é através da reutilização de componentes previamente desenvolvidos, testados e validados, para que partir deles, outros sistemas sejam construídos (CLEMENTS; NORTHROP, 2002). Neste contexto, a abordagem de Linhas de Produto de Software (LPS) propõe a reutilização sistemática de componentes, que são desenvolvidos como base de uma família de produtos. O Instituto de Engenharia de Software (SEI)¹ define uma LPS como um conjunto de sistemas de software intensivo que compartilham um conjunto comum e gerenciado de características a partir de um conjunto comum de núcleos ativos de forma prescrita. Ou seja, um conjunto de características comuns (*features*) são especificadas, e a partir delas, é desenvolvido um aglomerado comum de artefatos base desses produtos, considerando a sua variabilidade, para serem reutilizados de forma previamente planejada (POHL *et al.*, 2005).

Segundo Benavides *et al.* (2010), um artefato essencial para apoiar o estudo da variabilidade deste conjunto de características comuns em uma LPS é o modelo de *features*. Este artefato consiste em um diagrama em formato de árvore, onde cada nó representa as *features* que podem ser implementadas nos sistemas que serão desenvolvidos na linha de produto. Dessa forma é possível modelar os conjuntos de requisitos dos produtos e relacioná-los por meio de suas características em comum, podendo representar um requisito, uma funcionalidade ou um aspecto de qualidade (BEUCHE; DALGARNO, 2007).

O modelo de *features* permite também descrever as características identificadas nos produtos e as possíveis relações existentes entre elas (BENAVIDES *et al.*, 2010). Por isso, a avaliação de qualidade deste artefato pode ser fundamental para garantir que inconsistências e erros nos estágios iniciais do desenvolvimento não se espalhem por toda a Linha de Produto (POHL *et al.*, 2005). Um bom modelo de *features* pode ajudar os engenheiros a orientar seus esforços e fornecer-lhes o *feedback* necessário durante o desenvolvimento dos produtos de software. No contexto de LPS, o engenheiro de domínio deve estabelecer uma linha de produtos com a construção de componentes com alto nível de manutenibilidade. Este nível, pode ser mensurado através da avaliação da qualidade do modelo de *features* sob essa perspectiva (COLEMAN *et al.*, 1994).

No processo de avaliação da qualidade, uma estratégia popular é o uso das medidas (BAGHERI E., 2011). Todavia, no contexto deste trabalho, entende-se que o valor individual de

¹ <https://www.sei.cmu.edu/productlines>

medidas, por si só, não são suficientes para mensurar a qualidade total de um modelo de *features*. A avaliação deste artefato requer o uso de diferentes medidas e estas comumente possuem faixas amplas de entrada e saída, o que torna mais difícil mensurar um atributo de qualidade durante esse processo (BEZERRA *et al.*, 2016). Portanto, surge a necessidade de combinar essas medidas para obter uma medida agregada que resulte em um processo de avaliação unificada da qualidade.

A revisão de literatura em Bezerra *et al.* (2015), identificou características (e.g., Manutenibilidade, Confiabilidade, Eficiência de Desempenho) e sub-características (e.g., Acurácia, Consistência) que estão relacionadas com o modelo de *features*. A partir disso, foi proposto o *CatalOg of measures for Feature modEl quality Evaluation* (COFEE), um catálogo contendo 32 medidas de Manutenibilidade. Em uma extensão deste trabalho, uma análise realizada em Bezerra *et al.* (2016) demonstrou ainda que existem correlações entre as medidas do COFEE, e que nem todas as 32 medidas são necessárias para suportar a avaliação da Manutenibilidade de um modelo de *features*, podendo-se utilizar apenas 15 delas. Os autores do catálogo concluíram também, que a maioria das medidas foram definidas a nível de aspectos individuais (e são frequentemente computadas individualmente), por isso não conseguem indicar um nível de Manutenibilidade e apoiar adequadamente a avaliação da manutenibilidade do modelo de *features* como um todo.

Sendo assim, uma forma encontrada de acessar a Manutenibilidade deste artefato, em um processo de avaliação de qualidade utilizando medidas, é através de métodos de agregação. Este trabalho tem como objetivo selecionar o subconjunto de 15 medidas do catálogo COFEE e utilizar Lógica *Fuzzy* como método de agregação de medidas a fim de obter uma nova medida agregada: o Índice de Manutenibilidade do Modelo *Features*. Esta nova medida obtida pode ajudar a indicar se o modelo de *feature* possui boa ou ruim manutenibilidade. Os valores da nova medida agregada podem ser utilizados por engenheiros de domínio como dados adicionais quantitativos sobre o modelo de *features* e assim, apoiar ou fornecer credibilidade às opiniões dos especialistas no processo de avaliação da qualidade de uma LPS.

Este trabalho está organizado da seguinte forma: na Seção 2 são apresentados os conceitos principais que fundamentam este trabalho; a Seção 3 apresenta trabalhos que tratam de estudos de caso e técnicas de agregação de medidas de qualidade; Na Seção 4 são apresentados os passos executados e a abordagem utilizada para a realização deste trabalho, assim como os procedimentos aplicados no processo de agregação de medidas de manutenibilidade. A Seção 5 descreve a etapa de aplicação de medidas de manutenibilidade em modelos de *features* e a

extensão do *dataset* MACchiATO. A Seção 6 discorre sobre o projeto e definição do sistema *fuzzy* com ajuda de especialistas. A Seção 7 apresenta de forma detalhada o processo de agregação de medidas utilizando lógica *fuzzy* e os resultados da aplicação da nova medida agregada. Por fim, a Seção 8 apresenta as limitações, contribuições, conclusões e possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados os conceitos principais para o desenvolvimento deste trabalho. Na Seção 2.1 são introduzidos os conceitos referentes à Linha de Produto de Software. Na Seção 2.2 são abordados os modelo de *features* no que corresponde à sua definição, notações, características, e restrições em LPS. Na Seção 2.3 são abordadas as medidas de manutenibilidade e em seguida na Seção 2.4, serão apresentados alguns conceitos e ferramentas sobre a avaliação do modelo de *features* de uma LPS. A última Seção 2.5 trata do processo de agregação de medidas e discorre sobre a técnica de agregação utilizando Lógica *Fuzzy*.

2.1 Linha de Produtos de Software

O reuso de software é uma estratégia que vem sendo utilizada amplamente por médias e grandes empresas com o objetivo principal de agilizar e reduzir os custos de desenvolvimento (PACHECO *et al.*, 2015). Nessa abordagem, o produto é desenvolvido baseado em um conjunto de componentes reutilizáveis, para que a partir deles, outros sistemas sejam construídos na própria organização ou em outras. Para a construção de sistemas pertencentes a mesma família, ou seja, que possuem o mesmo domínio, é possível sistematizar o reuso de software com a aplicação do conceito de Linha de Produto de Software (LPS) (POHL *et al.*, 2005).

Por meio de LPS, empresas conseguiram alcançar uma produção de software de baixo custo e alta qualidade e esta têm se consolidado como uma abordagem fortemente adotada pela indústria de software nos últimos anos (SILVA *et al.*, 2011). Este conceito traz a ideia básica de desenvolver sobre um grupo de produtos que compartilham um conjunto comum de recursos gerenciados (WEISS; LAI, 1999). A estratégia de LPS proporciona uma redução no tempo de desenvolvimento, diminuição de custos e melhoria da qualidade no processo de criação de um produto de software (COHEN, 2002). Uma estratégia que têm permitido que organizações sejam capazes de criar famílias de produtos para domínios específicos por meio do desenvolvimento e reuso de software de maneira sistematizada.

O processo de desenvolvimento de software, no paradigma de Linha de Produto de Software, ocorre basicamente em duas etapas (POHL *et al.*, 2005):

- Engenharia de Domínio: etapa em que se estabelece o que é comum e o que é variável da linha de produtos. Onde são definidos os ativos da base da LPS (todos os tipos de artefatos de software: requisitos, *design*, testes, etc).

- Engenharia de Aplicação: os produtos da linha são derivadas a partir do processo de engenharia de domínio, considerando a gestão da variabilidade da linha de produtos e seu desenvolvimento de acordo com as necessidades de personalização e desenvolvimento.

Em consequência do processo de engenharia de aplicação, surge o conceito de variabilidade, que consiste em explorar as características que variam em relação aos diversos produtos da linha. Este conceito está relacionado às possibilidades de mudança ou personalização da LPS (CAPILLA; BOSCH, 2011). Os ativos base de uma LPS são elementos prontos para serem reutilizados na construção do produto, que podem ser componentes de software, modelos de documentos utilizados em processos, padrões de projeto, documento de requisitos, arquitetura de linha de produto, etc (LINDEN *et al.*, 2007). Estes artefatos são customizados para a instanciação de produtos por meios de mecanismos de variabilidade que satisfazem as necessidades de um segmento de mercado particular ou missão (CLEMENTS; NORTHROP, 2002).

Para apoiar a gestão da variabilidade, existem artefatos que caracterizam funcionalidades dos sistemas de software a serem desenvolvidos na linha, como o documento de Arquitetura da Linha de Produto (ALP) e o modelo de *features*, que é desenvolvido durante a etapa de Engenharia de Domínio, na fase de requisitos, e representa as características dos possíveis produtos a serem gerados e as relações existentes entre as mesmas (BABAR *et al.*, 2010). Um erro neste artefato, pode se propagar para as outras etapas do desenvolvimento da linha e para outros produtos (BEZERRA *et al.*, 2016). Por isso, os ativos de base e artefatos de referência, em especial o modelo de *features*, são de extrema importância para descrever aquilo que o produto de software deve atender dentro do escopo de domínio estabelecido e garantir sua qualidade.

2.2 Modelo de *Features*

O modelo de *features* representa as *features* do domínio e variabilidade de uma LPS. A *feature* em uma LPS, corresponde a uma característica do produto da linha que é visível ao usuário final (SOCHOS I. PHILIPPOW, 2004). A definição dessas *features* pode variar em relação a complexidade e diversidade dos domínio de aplicação, além de ser utilizada para especificar um conjuntos de requisitos funcionais e avaliar a sua qualidade (BENAVIDES D., 2010).

Neste modelo, é possível representar a informação dos possíveis produtos de uma LPS, em termos de características e relacionamentos entre eles (BENAVIDES D., 2010). As-

sim como em termos de características e relacionamentos, as similaridades e variabilidades de produtos de um determinado domínio comum entre os produtos de uma LPS podem ser especificadas, já que a *feature* pode representar muito bem uma funcionalidade ou um atributo de qualidade (BEUCHE; DALGARNO, 2007). A representação do conjunto de características de um LPS são organizadas hierarquicamente em uma estrutura de árvore, onde a raiz é o conceito principal do produto e os ramos representam as *features*. Para elaboração do modelo e representação das características e relações entre elas, existem as notações do método FODA (do inglês *Feature-Oriented Domain Analysis*), que permitem representar as *features* como:

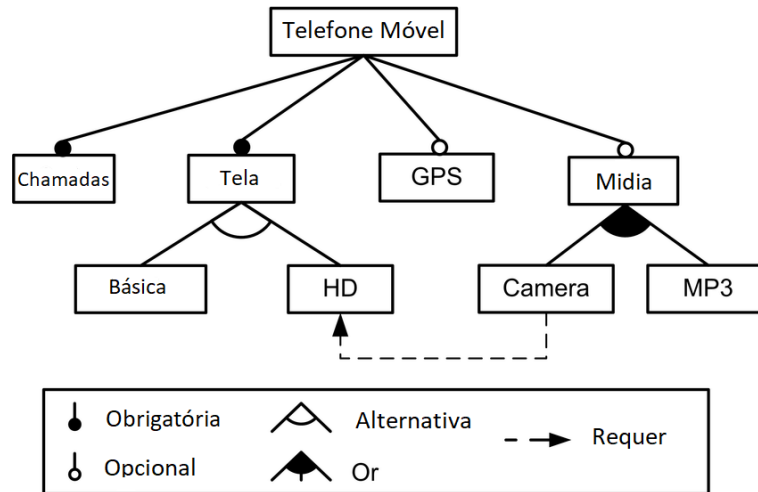
- **Obrigatória:** quando um determinado tipo de *feature* deve estar presente em todos os membros da LPS;
- **Opcional:** a presença da *feature* é opcional nos produtos membros da linha;
- **Alternativa:** composta de um conjunto de *features*, significa que é possível escolher uma ou mais *features* que serão utilizadas;
- **Or:** a *feature* Or é exclusiva, ou seja, a inclusão de uma *feature* deste tipo em um produto exclui outra.

Também são representadas no modelo as relações entre as *features*: a *feature* raiz é uma parte de todos os produtos dentro da linha de produtos de software. Uma *feature* filha estará presente no modelo representando uma característica do produto se a sua *feature* pai também faz parte do produto (BENAVIDES D., 2010). Além das relações de parentesco, o modelo também pode conter restrições entre as suas *features* (KANG *et al.*, 1990). Estas restrições podem ser: (i) se uma *feature* A requer uma *feature* B, a inclusão de A no produto da linha implica a inclusão de B; (ii) se uma *feature* A exclui uma *feature* B, ambas as *features* não farão parte do mesmo produto da linha.

A Figura 1 apresenta um exemplo do modelo de *features* de um telefone móvel (PAREJO *et al.*, 2016). As *features* de chamadas (Calls) e tela (Screen) são obrigatórias. 'GPS' e Mídia (Media) são opcionais, ou seja, as *features* opcionais podem ou não estar presentes nos produtos desta linha. A escolha de qual Mídia é de valor alternativo (deve conter pelo menos uma das funcionalidades: Câmera ou MP3) e a escolha do tipo da tela é alternativo obrigatório (apenas uma dos tipos de tela: Básica ou HD). Essas relações entre as *features* estão sempre presentes nesses modelos (KANG *et al.*, 1990). Os relacionamentos “requer” e “exclui” são conhecidos como restrições transversais, por exemplo, a *feature* Câmera “requer” o tipo de tela HD presente no produto. Ou seja, se um recurso for selecionado, o recurso requerido também

deve ser selecionado.

Figura 1 – Modelo de *features* de um Telefone Móvel



Fonte: adaptada de Parejo *et al.* (2016)

A notação FODA também permite que os modelos sejam definidos com base na cardinalidade entre *features* e *sub-features*, evitando ambiguidades e facilitando a representação das características do produto da linha (BENAVIDES D., 2010). A cardinalidade indica o número de ocorrências da *feature* em um produto através de um intervalo $[n..m]$, onde **n** é o limite mínimo e **m** o limite máximo. Esse intervalo é usado para limitar a quantidade de *features* filhas que devem ser inseridas em um produto caso uma *feature* pai seja selecionada.

- $[0..1]$: pode-se escolher uma ou nenhuma *feature* do conjunto de *sub-features*;
- 1: uma *feature* deve ser escolhida do conjunto de *sub-features*;
- $[0..*]$: nenhuma ou várias *features* podem ser selecionadas do conjunto de *i*;
- $[1..*]$: uma ou várias *features* podem ser selecionadas do conjunto de *sub-features*.

Dessa forma, o modelo de *features* representa um artefato vital em uma Linha de Produto de Software fornecendo uma visão esquemática da representação de semelhanças, restrições, relacionamentos e pontos de variabilidade dentro desta família de sistemas (MENDONCA, 2009). O artefato auxilia na definição das *features* comuns dos produtos, assim como diminuir a distância entre os usuários finais e os desenvolvedores com relação aos requisitos do produto que está sendo desenvolvido, por isso está diretamente ligada a qualidade da LPS.

2.3 Medidas de Manutenibilidade

O padrão ISO/IEC 12207 define a qualidade como “o grau em que um sistema, componente ou processo atende aos requisitos especificados e as necessidades do cliente ou expectativas do usuário” (ISO/IEC:2008,). No contexto da avaliação da qualidade, a medição de software é um processo contínuo de definição, coleta e análise de dados sobre o desenvolvimento de software e de seus produtos (SOLINGEN *et al.*, 2002). Uma estratégia eficaz para medir a qualidade de um artefato ou componente é o uso de medidas ou métricas, possibilitando uma avaliação quantitativa dos aspectos de qualidade dos produtos desenvolvidos a fim de melhorar o processo de desenvolvimento de software.

As medidas de qualidade podem refletir as características, sub-características ou atributos de qualidade de um produto ou artefato (KAN, 2002). Enquanto a estrutura de uma LPS se modifica constantemente, gerando mudanças constantes no modelo de *features*, o que impacta diretamente em uma das características de qualidade mais críticas para este artefato: a Manutenibilidade (BAGHERI E., 2011). As definições padrão do (IEE, 1990) para os conceitos de manutenção e manutenibilidade de um software são:

- **Manutenção:** o processo de modificação de um sistema de software ou componente após a entrega, para corrigir falhas, melhorar o desempenho ou outros atributos e se adaptar a um ambiente alterado; e
- **Manutenibilidade:** a facilidade com que um sistema ou componente de software pode ser modificado para corrigir falhas, melhorar o desempenho ou outros atributos, e se adaptar a um ambiente alterado.

Essas características (e.g. Confiabilidade, Usabilidade, Manutenibilidade, Portabilidade, Compatibilidade e Segurança) em um produto de software ou artefato podem ser complexas, uma única métrica pode não capturar vários aspectos de qualidade e suas subcaracterísticas (FENTON, 1994). A fim de mensurar a manutenibilidade dos modelos de *features* em LPS, o trabalho de Bagheri e Gasevic (2011) propõe abranger a maior quantidade possível de características de um modelo de *features* de maneira simples para obter uma série de medidas de manutenibilidade. Para obter estas medidas, são consideradas desde métricas estruturais do modelo de *features*, como o número total de *features* (NF) presentes no modelo, assim como características relacionadas a variabilidade do artefato, como a Flexibilidade de Configuração (FoC), por exemplo, que é a proporção do número de recursos opcionais sobre todos os recursos disponíveis no modelo de *features* (BAGHERI E., 2011).

Sabendo-se que uma medida permite mensurar quantitativamente um atributo de qualidade (FINKELSTEIN, 2003), o catálogo de medidas proposto por Bezerra et al. (2015) foi aplicado em seis modelos de *features* em diversos domínios de aplicação, demonstrando também que as características estruturais deste artefato podem gerar medidas que refletem os atributos de manutenibilidade e que seus resultados auxiliam no processo de avaliação da qualidade (BEZERRA *et al.*, 2015).

O estudo realizado em (BEZERRA *et al.*, 2016) analisou as correlações entre as 32 medidas do catálogo COFFEE, e concluíram que não são necessárias todas as 32 medidas para avaliar a qualidade do modelo de *features*, e apontam um subconjunto de 15 medidas que supostamente são suficientes para tal. Neste trabalho, estas medidas foram selecionadas para compor o processo de agregação a fim de alcançar um índice de manutenibilidade para avaliação do modelo de *features*. Este é um subconjunto selecionado do catálogo COFFEE o qual neste trabalho é denominado de MiniCOFFEE. A Tabela 1 apresenta as definições e formulas de cálculos das 15 medidas deste subconjunto.

Tabela 1 – MiniCOFFEE: subconjunto de 15 medidas selecionadas do catálogo COFFEE e suas fórmulas de cálculo

| Acrônimo | Nome da medida | Forma de cálculo |
|----------|--|---|
| NF | Número de features | Número total de features no modelo |
| NLeaf | Features Folhas | Número de features que não possuem filhas |
| NM | Features Obrigatórias | Número de features obrigatórias no modelo |
| NTop | Número de top features | Número de features descendentes diretas da raiz |
| NGXOr | Número de grupos XOR | Número de pontos de variação com relacionamento XOR |
| DT Max | Densidade máxima da árvore | Número de features com a maior distância da raiz do modelo de feature |
| CogC | Complexidade Cognitiva | Número de pontos de variância |
| FEX | Extensibilidade da Feature | NLeaf + SCDF + MCDF |
| FoC | Flexibilidade da configuração | NO/NF |
| SCDF | Features dependentes de ciclos únicos | A soma de todas features participantes em restrições de features filhas que possuem pontos variantes com [1..1] |
| MCDF | Múltiplos ciclos dependentes de features | A soma de todas features participantes em restrições de features filhas que possuem pontos variantes com cardinalidade [1..*] |
| RDen | Coefficiente de densidade-conectividade | Número médio de features (não-pais) referenciados nas restrições de um a outra feature |
| RoV | Proporção de Variabilidade | Soma dos números médios de feature filhas em cada um dos nós |
| NVC | Configurações Válidas | Número de configurações possíveis e válidas do modelo de features |
| NGOr | Número de grupos do tipo Or | Número de pontos de variação com relacionamento Or |

Fonte: elaborado pelo Autor.

2.4 Avaliação da Manutenibilidade do Modelo de *Features*

Mensurar completamente a Manutenibilidade de um software é um desafio considerável, pois os sistemas são cada vez mais complexos e há um aumento na proporção de desenvolvedores realizando atividades inerentes a manutenção de software como parte do processo de desenvolvimento (COLEMAN *et al.*, 1994). Rosenberg (1998) afirma que, quando medidas são usadas no processo de avaliação da qualidade, a interpretação final pode ser resultado de um senso comum e experiência, a depender do que consiste um valor aceitável ou não, referente aos requisitos da empresa, da experiência dos desenvolvedores ou opinião de especialistas. Os resultados receberão uma taxa de aceitação ou rejeição entre eles, ou seja, o grau em que os especialistas concordam com as estimativas obtidas, obtendo faixas de classificações para avaliação da qualidade (ROSENBERG, 1998).

Segundo Etxeberria e Sagardui (2005), os atributos de qualidade de uma LPS estão relacionadas à sua variabilidade, flexibilidade e complexidade. Dessa forma, o processo de avaliação da qualidade em Linhas de Produto de Software se concentra nas variabilidades, garantindo que estas são tangíveis e oferecem flexibilidade suficiente para derivar os produtos específicos da linha e não impor custos de desempenho inaceitáveis ao desenvolvimento (CLEMENTS; NORTHROP, 2002). Portanto, a avaliação da qualidade de artefatos que suportam a gestão da variabilidade como o modelo de *feature* pode ser fundamental para garantir que erros ou inconsistências nos estágios iniciais do desenvolvimento não se espalhem por toda a Linha de Produto.

Existem técnicas e ferramentas para avaliação da qualidade de uma LPS e do modelo de *features*. Etxeberria e Sagardui (2005) classifica, entre as técnicas existentes, como técnicas de medição para avaliação quantitativa as simulações, o uso de modelos matemáticos e experimentais e por conseguinte, quaisquer técnicas que alcançam atributos de qualidades específicos, como a agregação medidas. Algumas ferramentas como o repositório *online* de modelos de *features* S.P.L.O.T. (MENDONCA *et al.*, 2009), FAMILIAR (ACHER *et al.*, 2013) e a DyMMER (BEZERRA *et al.*, 2016) apoiam o processo de avaliação da qualidade dos modelos de *features* dando suporte a análise automática destes modelos.

A medição de software pode então ser efetivamente utilizada para apoiar de forma quantitativa a avaliação da qualidade de uma LPS e dos seus artefatos. Para tanto, é necessário identificar um conjunto de medidas que podem ser usadas; quais as características de qualidade a serem avaliadas; adotar um protocolo de avaliação e definir e mensurar os atributos de qualidade

com base nos valores obtidos das medidas consideradas no processo (BOEHM *et al.*, 1976). É dessa maneira que, neste trabalho, a avaliação da qualidade embasada na medição de software pode permitir por exemplo, mensurar a manutenibilidade do modelo de *features*.

2.4.1 Ferramenta e repositório de modelos de *features* S.P.L.O.T.

A ferramenta S.P.L.O.T.¹, pode ser utilizada para criação e edição de modelos de *features* online que pode ser utilizada pela comunidade de pesquisadores em Linhas de Produto de Software (MENDONCA *et al.*, 2009). É possível criar um novo modelo de *features* ou editar um existente. Os modelos criados podem ser salvos no repositório da ferramenta e são armazenados no repositório apenas os modelos consistentes, que possuam pelo menos uma configuração válida e modelos corretos, que não possuam *features* mortas. Também é possível realizar uma análise automatizada de um modelo salvo na ferramenta ou de algum que tenha sido criado no formato SXFM (do inglês *Simple XML Feature Model Fomat*) e possui diversas medidas de qualidade previamente computadas para apoiar a avaliação da qualidade do modelo de *features*.

Atualmente o repositório, conta com 1.052 modelos disponíveis para *download*, edição ou configuração de produto. Este repositório foi utilizado no desenvolvimento deste trabalho durante o processo de coleta de modelo de *features* para construção de um conjunto de dados contendo o valor das medidas selecionadas no processo de agregação aplicadas a um conjunto de modelos de *features*

2.4.2 Ferramenta DyMMer

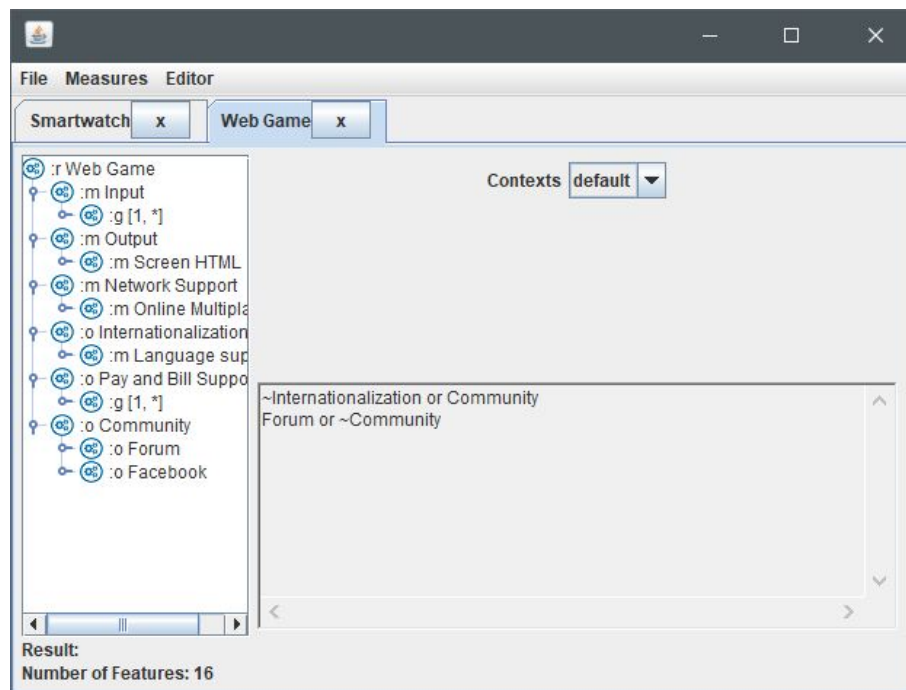
Dentre as ferramentas disponíveis para apoiar o processo de avaliação da qualidade do modelo de *features*, a ferramenta DyMMer trabalha muito bem com aplicar de forma automatizada um conjunto de medidas de qualidade em modelos de *features* diversificados e realizar exportação dos valores e resultados obtidos para serem analisados. Os modelos podem ser descritos no formato de arquivo (XML) ou extraídos do repositório de modelo de *features* S.P.L.O.T e importados na ferramenta. É possível também verificar de forma automática a integridade dos modelos de *features*; visualizar e editar o modelo de *features* importado conforme o contexto selecionado; realizar análise de diferentes medidas de qualidade e exportar os resultados encontrados (BEZERRA *et al.*, 2016).

¹ <http://www.splot-research.org/>

A arquitetura da ferramenta consiste em três camadas: (i) camada de visualização de modelos de *features*; (ii) camada de edição dos modelos *features*; (iii) camada de exportação de dados, que permite a exportação dos valores gerados a partir das medidas de qualidade aplicadas aos modelos para uma planilha no formato Microsoft Office Excel.

Como ilustrado na Figura 2, a DyMMer é capaz de abrir modelos de *features* para serem representados. Do lado esquerdo, surge uma lista hierarquizada. No centro da janela, os contextos podem ser selecionados e logo abaixo é possível visualizar as restrições que as *features* do modelo possuem. A ferramenta também é capaz de exportar os resultados das medidas de qualidade. Neste trabalho, as medidas de manutenibilidade apresentadas na Seção 2.3 são previamente computadas. As camadas de visualização e exportação da ferramenta permitirão analisar um conjunto de modelos de *features* e obter o valor de cada uma das medidas especificadas.

Figura 2 – Visualização e análise do Número de Features de um Web Game na ferramenta DyMMer



Fonte: elaborado pelo Autor.

2.5 Agregação de Medidas

No contexto da avaliação da qualidade de um artefato ou produto de software, as medidas propostas individualmente por um modelo de qualidade podem não refletir todas as

características ou deixar de refletir um determinado atributo de qualidade que se deseja mensurar (DASSO; FUNES, 2012). Essas medidas são frequentemente definidas para obter resultados de componentes de software individuais (métodos, classes, etc.) e não são facilmente aplicáveis para avaliar pacotes ou sistemas inteiros (MORDAL *et al.*, 2013). Para tanto, é necessário agregar os resultados dessas métricas. Por exemplo, para um determinado componente de software ou um artefato como o modelo de *features*, é preciso obter os resultados de todas as medidas individuais consideradas. Porém, dada uma característica de qualidade composta de sub-caraterísticas como a manutenibilidade, é preciso agregar todos os resultados de todas as medidas em um só valor.

A ideia principal do processo de agregação das medidas é suportar a avaliação da qualidade de forma quantitativa, tornando possível obter um grau de consenso entre as informações disponíveis nos valores de entrada (medidas selecionadas) através do cálculo de um valor final (nova medida agregada) (VASILESCU *et al.*, 2010). Dessa forma, esse processo pode ser realizado a partir da seleção de diferentes medidas e técnicas de agregação e apoiar a avaliação da qualidade do modelo de *features*, por exemplo, em mensurar as características principais deste artefato que são relacionadas a manutenibilidade e que são acessadas através das medidas.

Mordal *et al.* (2013) faz uma análise sobre as técnicas de agregação de medidas de software mais utilizadas na indústria: (i) Agregação por média aritmética, que se baseia na computação da média aritmética dos resultados obtidos individualmente. Porém, alerta que o cálculo da média suaviza esses resultados, que podem vir a não ser suficientemente representativos; (ii) A agregação por média ponderada, que pode ser usada atribuindo um peso na métrica ou componente no cálculo da média, destacando assim um componente de baixa qualidade ou um componente crítico no processo de agregação. Por causa da atribuição de um peso em um determinado componente na métrica, os resultados agregados podem apresentar problemas pontuais, como por exemplo: o retorno de um valor agregado que sugere a diminuição da qualidade do software, quando na realidade, houve melhoria (retornando um falso positivo). Uma alternativa para médias simples e ponderadas é o uso da Lógica *Fuzzy* como técnica de agregação medidas.

2.5.1 Lógica Fuzzy

Uma proposição lógica, tradicionalmente, possui dois valores lógicos variáveis: ou é completamente falsa (0) ou é completamente verdadeira (1). Na teoria dos conjuntos *fuzzy*

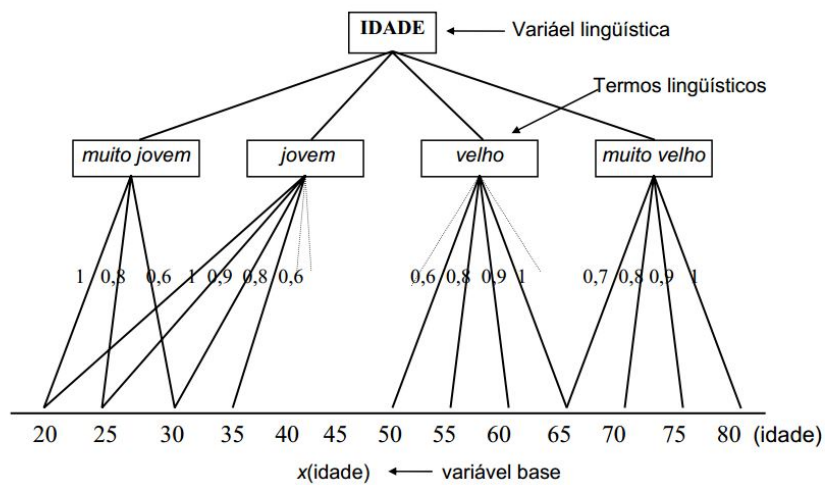
existe um grau de pertinência entre os valores 0 e 1. Dessa forma, na lógica *fuzzy*, uma premissa varia entre parcialmente verdadeira ou parcialmente falsa (KLIR; YUAN, 1995). Esta lógica é comumente utilizada para representar modelos de raciocínio impreciso, que buscam simular a habilidade humana de tomar decisões racionais, em ambientes de incertezas e imprecisões (ZADEH, 1988).

A lógica *fuzzy* pode se utilizar de predicados *fuzzy* (velho, raro, perigoso, etc.), quantificadores *fuzzy* (muito, pouco, quase tudo, usualmente, e semelhante) e valores-verdade *fuzzy* (totalmente verdadeiro, mais ou menos verdadeiro, muito falso, etc.) (ZIMMERMANN, 1997). Ou seja, em um sistema *fuzzy* é possível incorporar conceitos subjetivos que são inerentes à mente humana, tais como alto, baixo, velho e novo. Estes conceitos são incorporados em forma de variáveis linguísticas conhecidas pela base de conhecimento do sistema *fuzzy*, onde a pertinência ou não de um elemento a um conjunto se dá de forma gradual e não simplesmente binária como na lógica tradicional.

A função de pertinência é um dos componentes cruciais de um sistema *fuzzy*, pois as operações são definidas em conformidade com ela, juntamente com as regras *fuzzy* (RUONING; XIAOYAN, 1992). As regras *fuzzy* definem os intervalos nas funções de pertinência para cada uma das entradas e, conseqüentemente, definem o intervalo *fuzzy* para as variáveis de saída a partir de uma variável linguística.

Por exemplo, considerando um sistema *fuzzy* em que há uma variável linguística (x) identificada por “Idade”, e o resultado está no intervalo $[0, 100]$. Os termos linguísticos possíveis deste sistema, como “velho”, “jovem”, “muito velho” fazem parte de um conjunto *fuzzy* e o valor de entrada é a idade em anos de vida. A partir de um conjunto de regras *fuzzy* definidas para estes termos, será atribuída à Idade um termo linguístico correspondente, como ilustra a Figura 3.

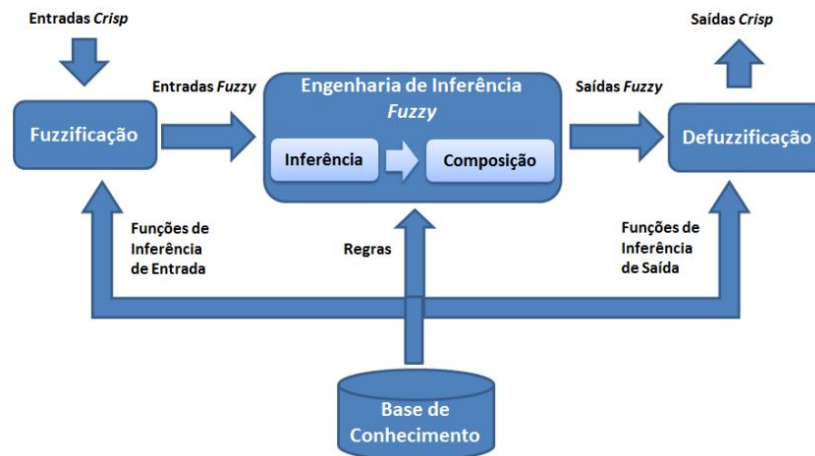
Figura 3 – Variável linguística “Idade”



Fonte: Belchior (1997).

Portanto, cada termo linguístico, definida em um intervalo para o valor da variável linguística de entrada, tem seu estado interpretados como números *fuzzy* específicos (e.g. 20 anos é considerado "muito jovem", 35 "jovem", 65 considerado "velho" e 80 anos considerado "muito velho"). Os termos linguísticos representam valores aproximados de uma variável linguística, relacionados a uma aplicação específica ao domínio aplicado (BELCHIOR, 1997). Dessa forma, a aplicação desta lógica resulta em sistemas *fuzzy* capazes de produzir respostas mais rápidas e “suaves” que os sistemas convencionais (COX, 1995) e são úteis para problemas ou aplicações complexas, que envolvam descrições humanas ou pensamento indutivo, como a avaliação da qualidade e o contexto deste trabalho, isto é, a agregação de medidas de manutenibilidade.

A Figura 4 ilustra a estrutura de um sistema *fuzzy*. No processo chamado de *Fuzzificação*, o conjunto de entradas (informações numéricas) são traduzidas para variáveis linguísticas que consistem nos nomes dados aos conjuntos *fuzzy* dentro do domínio (e.g. baixo, médio ou alto). De acordo com as definições das regras *fuzzy*, advindas da base de conhecimento e domínio de interesse para serem utilizadas no processo de inferência, são geradas as funções de pertinência (relações das variáveis linguísticas com os intervalos entre 0 e 1) para cada uma das entradas. As regras são consideradas no processo de inferência onde cada saída *fuzzy* será agregada em um único conjunto *fuzzy* no processo de composição.

Figura 4 – Estrutura do sistema *fuzzy*

Fonte: adaptada de Hettiarachchi *et al.* (2016).

O uso da lógica *fuzzy* tem sido utilizada como método de agregação de medidas e atende bem a problemas de engenharia de software (PIZZI, 2013). Yadav e Yadav (2015) propõe uma metodologia para agregação de medidas baseadas em lógica *fuzzy* que consiste basicamente na: (i) seleção de medidas; (ii) definição das funções de pertinência de entradas e saídas das medidas; (iii) definição das regras *fuzzy* e (iv) execução da *fuzzificação* e *defuzzificação*, obtendo novas medidas agregadas. Dentro deste contexto, baseado nas definições para os componentes *fuzzy* em Hettiarachchi *et al.* (2016), as principais definições para cada um dos componentes e etapas do processo são:

- **Base de Conhecimento:** A base de conhecimento contém o conjunto de regras *fuzzy* selecionadas e os parâmetros definidos para as funções de pertinência. Em um sistema *fuzzy*, as regras possuem um papel essencial uma vez que são definidas de acordo com o conhecimento de especialistas sobre o domínio.
- **Fuzzificação:** É o processo responsável por traduzir a informação de entrada para o domínio *fuzzy*, transformando a entrada *crisp* em uma entrada para as funções de pertinência, ou seja, as informações numéricas das medidas são traduzidas para variáveis linguísticas.
- **Inferência:** O processo de inferência utiliza as entrada *fuzzy* das funções de pertinência para determinar a saída das funções de pertinência usando regras formuladas a partir da base de conhecimento.
- **Composição:** O processo de composição agrega todas as saídas *fuzzy* em um conjunto *fuzzy* único, portanto, resulta na agregação de um conjunto de entrada de medidas em um

único valor agregado.

- **Defuzzificação:** É a conversão de um conjunto *fuzzy* em um único valor numérico ou em um vetor de valores. É calculada uma saída usando o conjunto *fuzzy* produzido pelos processos de inferência e composição, resultando em função de pertinência e um valor para medida agregada de acordo com as medidas de entrada.

Tudo isso permite que as medidas selecionadas em um processo de agregação utilizando este método venham compor o conjunto de entradas que serão *fuzzificadas*, inferidas, agregadas e *defuzzificadas* em um único número. A introdução de técnicas de agregação com diferentes abordagens e os desafios elencados por Mordal *et al.* (2013) na aplicação de agregação com médias simples e ponderadas demonstram a necessidade de atentar para a conformidade da técnica escolhida com o resultado agregado desejado. Especialmente para o modelo de *features* de uma LPS, um bom processo de agregação deve refletir todas as sub-características acessadas pelas medidas, o quanto for possível. Neste trabalho, assim como em Bezerra *et al.* (2018), será utilizado a Lógica *Fuzzy* como técnica de agregação de medidas.

3 TRABALHOS RELACIONADOS

Foram identificados estudos que auxiliam o desenvolvimento deste trabalho e possuem similaridades e diferenças com o trabalho proposto. Estes estudos são apresentados nesta seção, assim como suas principais contribuições e relações com o presente trabalho.

Bezerra *et al.* (2016) realizaram um estudo de caso com o objetivo de investigar como as medidas podem ser aplicadas ao processo de avaliação de qualidade do modelo de *features* em Linhas de Produto de Software. Utilizando o catálogo COFFEE (proposto em Bezerra *et al.* (2015)) e o S.P.L.O.T (repositório de modelos de *features* (MENDONCA *et al.*, 2009)), os autores obtiveram o MACchiATO (*MeAsures dATaset for feaTure mOdel*), um *dataset* contendo os valores das 32 medidas do catálogo em 218 modelos de *features* extraídos do repositório. Essas medidas foram agregadas utilizando a técnica de análise PCA (do inglês *Principal Component Analysis*) para obter um grupo menor de medidas contendo as mesmas informações que as demais. Com o uso de um software estatístico, foi feito o cálculo do desvio padrão e da proporção de variância e proporção cumulativa para o do conjunto de variáveis obtido (medidas agregadas). As análises realizadas demonstraram que existe uma correlação entre as medidas do COFFEE e que nem todas as 32 medidas são necessárias para revelar a qualidade de um modelo de *features*, indicando um subconjunto de apenas 15 medidas que podem ser usadas. Após a execução, foi constatado que o conjunto de 9 novas medidas agregadas obtidas, ainda continha boa parte das informações no conjunto de medidas inicial. No entanto, a partir da conclusão dos autores, usando apenas as medidas agregadas obtidas, não é possível determinar qual modelo possui melhor manutenibilidade.

Desse modo, a fim de estender este estudo, neste trabalho será definida uma abordagem para agregação das 15 medidas selecionadas no catálogo COFFEE para obter uma nova medida agregada que represente o índice de manutenibilidade do modelo de *feature* e apoie a avaliação da qualidade deste artefato em Linhas de Produto de Software.

Em Bezerra *et al.* (2018) investigamos como agregar medidas utilizadas na avaliação da qualidade do modelo de *features*. Utilizando um catálogo de 13 medidas, obtiveram um conjunto de dados com valores dessas medidas aplicados a 30 modelos de *features* extraídos da literatura. Em seguida, os autores aplicaram a técnica de Lógica *Fuzzy* para agregar essas medidas, produzindo 4 novas medidas agregadas. As medidas agregadas obtidas podem ser aplicadas para avaliar as características de qualidade do modelo de *features* relacionadas ao seu: tamanho; estabilidade; flexibilidade e dinamicidade. A fim de avaliar o uso das medidas

agregadas, os autores aplicaram as 4 novas medidas no conjunto de dados anterior. Os resultados sugeriram que as medidas agregadas são válidas e podem ajudar o engenheiro de domínio a avaliar características de qualidade dos modelos de *features*.

Conforme a metodologia utilizada no estudo apresentado acima, o presente trabalho utiliza Lógica Fuzzy como técnica de agregação de medidas selecionadas. Porém, de modo que a nova medida agregada obtida possa abranger todas as subcaracterísticas de manutenibilidade do modelo de *features*, permitindo mensurar a manutenibilidade geral do modelo a partir de um único valor agregado.

Oliveira e Belchior (2002) apresentam uma ferramenta chamada AdeQuaS, que tem o objetivo de dar suporte às etapas do processo de avaliação de software. Visto que existem muitas dificuldades no processo de avaliação de um produto ou artefato, que se iniciam no planejamento e na definição de como será realizada, até a especificação de atributos de qualidade relevantes ao objeto de avaliação, definição de qual método de avaliação e medição é mais adequado, estabelecimento da combinação de métricas e medidas que quantifiquem adequadamente os atributos em resultados reais e etc. A ferramenta é baseada no Modelo *Fuzzy* de Avaliação de Qualidade de Software (MFAQS) e propõe apoiar este processo em obter resultados mais reais sobre o grau de qualidade de atributos subjetivos através do julgamento de um grupo de especialistas. Além disso, permite obter resultados quantitativos, que traduzam valores reais pela consolidação do julgamento de diversos avaliadores, com níveis de experiência e pontos de vista diferentes, para atributos subjetivos. Apesar do MFAQS ser flexível em várias situações e resolver de modo satisfatório o problema de agregação de medidas, sua utilização manual exige um enorme esforço. Portanto a ferramenta *fuzzy* AdeQuaS, baseada nesse modelo, automatiza o processo de avaliação da qualidade, para se obter resultados mais eficientes e em um menor espaço de tempo. Além de apresentar um exemplo de automatização do processo de avaliação de Especificações de Requisitos de Software (ERS).

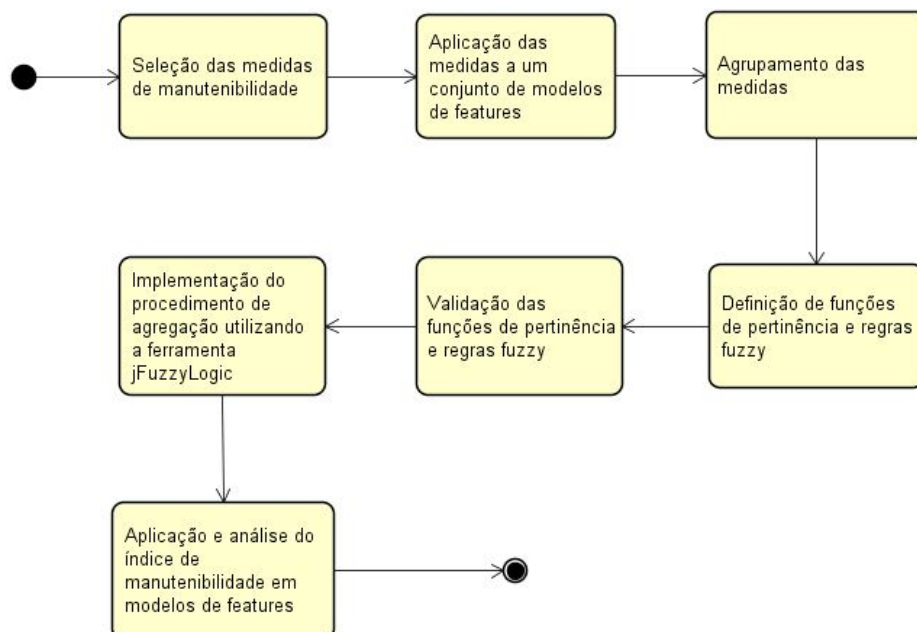
Neste trabalho, será utilizada uma ferramenta que permite automatizar o processo de agregação de medidas de manutenibilidade, mas que ainda não automatiza a etapa de definição de parâmetros e regras *fuzzy* do sistema *fuzzy* de agregação de medidas. Em trabalhos futuros, a metodologia aplicada poderá ser adequada para basear-se no modelo MFAQS e utilizar a ferramenta AdeQuaS para apoiar o processo de agregação aqui utilizado. Principalmente em obter os resultados subjetivos indicados por especialistas nas definições do sistema *fuzzy* e reduzir o tempo e esforço destes procedimentos.

4 PROCEDIMENTOS METODOLÓGICOS

Nesta Seção, são apresentados os passos estabelecidos na execução deste trabalho. A Figura 5 apresenta o fluxograma das etapas realizadas, a partir da seleção das medidas de manutenibilidade até os resultados obtidos na análise e aplicação da medida agregada aplicada aos modelos de *features*.

Inicialmente, foram selecionadas medidas de manutenibilidade para compor os valores de entrada do processo de agregação. Em seguida, foram obtidos os valores dessas medidas para um conjunto de modelo de *features*, resultando na extensão do *dataset* MAcchiaTO. Logo após, as medidas selecionadas foram agrupadas em 3 grupos. Para cada um deles, foram definidas funções de pertinência e regras *fuzzy* para compor a base de conhecimento do processo de agregação. Em seguida, estes parâmetros e regras foram validados por especialistas em Linhas de Produto de Software e Avaliação da Qualidade através de uma Revisão por Pares. Por fim, a base de conhecimento composta pelas funções e regras definidas foram codificadas na ferramenta de execução do sistema *fuzzy*, resultando no processo agregação onde obteve-se a nova medida agregada: o índice de manutenibilidade dos modelos de *features*.

Figura 5 – Fluxograma das atividades realizadas durante a execução deste trabalho



Fonte: elaborado pelo Autor.

4.1 Seleção das medidas de manutenibilidade

A primeira etapa da realização deste trabalho foi a seleção de medidas de manutenibilidade para compor o conjunto de entradas do processo de agregação. Para isso, foi necessário identificar as subcaracterísticas de manutenibilidade que são relevantes para a avaliação do modelo de *features* e quais medidas acessam essas características. Foi utilizado o catálogo de medidas COFFEE (do inglês *CatalOg of measures for Feature modEl quality Evaluation*) que é composto por 32 medidas de qualidade para avaliação da manutenibilidade do modelo de *features* e também identifica as subcaracterísticas de manutenibilidade que são específicas para avaliação do modelo de *features* (e.g. Extensibilidade, Flexibilidade, Complexidade Estrutural e Variabilidade) e quais medidas relacionadas (BEZERRA *et al.*, 2015).

Com base na análise de correlação realizada por Bezerra *et al.* (2016) entre as 32 medidas do catálogo COFFEE, 15 medidas do catálogo são suficientes para acessar a manutenibilidade de um modelo de *features*. Estas 15 medidas foram selecionadas e especificadas na Seção 2.3, resultaram em um subconjunto do catálogo COFFEE: o MiniCOFFEE. Dessa forma, o subconjunto de 15 medidas proposto foi utilizado neste trabalho para compor os valores de entrada do processo de agregação e obtenção do índice de manutenibilidade.

4.2 Aplicação das medidas selecionadas a um conjunto de modelos de *features*

Para obter dados relevantes sobre as medidas selecionadas no passo anterior, foi realizada a coleta dos valores de cada uma das medidas do MiniCOFFEE aplicadas a um conjunto de modelos de *features* retirados do repositório S.P.L.O.T, como especificado na Seção 2.4.1. O primeiro passo, foi estender o *dataset* MACchiATO obtido em Bezerra *et al.* (2016), que continha o valor das 32 medidas do catálogo COFFEE aplicados em 218 modelos de *features* retirados do S.P.L.O.T. Dessa forma, foram coletados novos modelos do repositório e realizada a exportação dos valores das medidas do MiniCOFFEE utilizando ferramenta DyMMer. No total, as medidas foram aplicadas em 326 modelos de *features*, resultando em uma extensão do *dataset*.

4.3 Agrupamento das medidas

Neste passo, as medidas do MiniCOFFEE foram subdivididas com base nas subcaracterísticas de Manutenibilidade as quais estão relacionadas: Complexidade Estrutural, Analisabilidade, Extensibilidade, Flexibilidade, Modularidade, Complexidade Cognitiva e a Variabilidade

Estática do modelo de *features* (BEZERRA *et al.*, 2015). As medidas selecionadas foram divididas em três grandes grupos: Tamanho, Complexidade e Variabilidade, ou seja, foram agrupadas as medidas que fornecem dados sobre a estrutura do modelo, as que permitem mensurar a complexidade e as que estão relacionadas com flexibilidade e extensibilidade do modelo *features*, respectivamente.

4.4 Definição de funções de pertinência e regras *fuzzy*

De acordo com a metodologia proposta por Yadav e Yadav (2015), foi necessário definir, para cada uma das medidas selecionadas e agrupadas, as funções de pertinência e os termos linguísticos que determinam o parâmetros *fuzzy* de uma determinada medida no sistema. Além disso, foram definidas regras *fuzzy* que permitem definir no sistema uma declaração condicional para cada uma das medidas em dos grupos definidos no passo da Seção 4.3.

4.5 Validação das funções de pertinência e regras *fuzzy*

A fim de validar as funções, parâmetros e regras definidos no passo anterior, foi realizada uma revisão por pares com 3 especialistas em Engenharia de Domínio de Linhas de Produto de Software. Os revisores tiveram acesso a uma planilha contendo as medidas do MiniCOFFEE agrupadas em Tamanho, Complexidade e Variabilidade, assim como as suas funções de pertinência e regras *fuzzy*. Dessa forma, o especialista participante pôde indicar, para cada medida, o nível de concordância com os parâmetros e regras definidos, adicionar comentários e sugestões além de reportar inconsistências ou ambiguidades.

4.6 Implementação do procedimento de agregação na ferramenta jFuzzyLogic

Para realizar os processos de *fuzzificação*, inferência e *defuzzificação*, descritos na Seção 2.5, foi utilizada a ferramenta jFuzzyLogic que permite declarar controladores lógicos, definir as entradas, saídas, funções e regras de um sistema fuzzy. A ferramenta permitiu também declarar as medidas selecionadas no passo da Seção 4.1 como as entradas do sistema, além de automatizar as etapas de inferência, composição e associação dessas entradas. Ao final do procedimento, o processo de *defuzzificação* resulta em um único valor numérico, que representa a nova medida obtida através do processo de agregação.

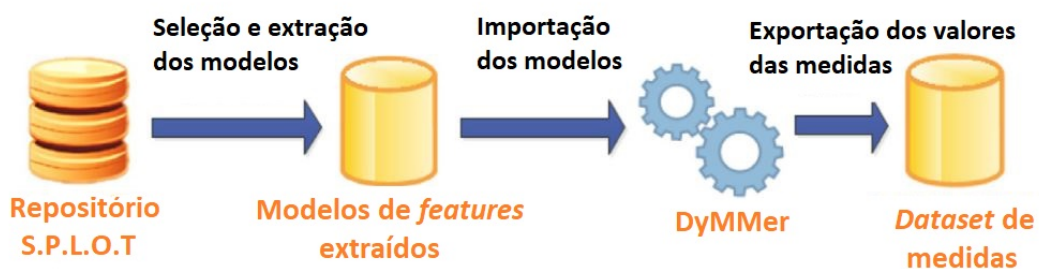
4.7 Aplicação e análise da medida agregada

Neste passo, a fim de validar a utilização da medida agregada obtida no passo anterior, a nova medida foi aplicada a um conjunto de 20 modelos de *features*, resultando em um valor numérico do índice de manutenibilidade para cada um dos modelos. Foi utilizada a ferramenta jFuzzyLogic para calcular o valor da nova medida e avaliar à conformidade resultado em relação as parâmetros e regras definidos, considerando os aspectos de tamanho, complexidade e variabilidade da estrutura do modelo de *features* e avaliar se a nova medida permite inferir se este possui boa ou ruim manutenibilidade. De modo geral, após a análise e avaliação dos resultados, conclui-se que a medida resultando do processo de agregação é útil para apoiar a avaliação da manutenibilidade do artefato.

5 PROCESSO DE EXTENSÃO DO MACCHIATO E APLICAÇÃO DE MEDIDAS

Como mencionado na Seção 3, o *dataset* MACchiATO contém os valores de 32 medidas de manutenibilidade aplicadas a 218 modelos de *features*, assim como as estatísticas dos valores obtidos incluindo o valor médio e o desvio padrão das medidas em cada um dos modelos. Nesta Seção, é descrito como feita a coleta de 108 novos modelos de *features* do repositório S.P.L.O.T e a aplicação das medidas de manutenibilidade nestes modelos, resultando em uma extensão do *dataset* MACchiATO. A mesma metodologia utilizada no processo de geração do MACchiATO foi utilizada no seu processo de extensão, como ilustra a Figura 6. Os passos deste processo são descritos a seguir.

Figura 6 – Processo de extensão do *dataset* MACchiATO.



Fonte: adaptada de Bezerra *et al.* (2016).

5.1 Coleta dos modelos de *features*

O repositório S.P.L.O.T possuía, até a data em que foi realizada a coleta dos modelos de *features*, 1.052 modelos disponíveis para *download*. Os modelos armazenados no repositório foram extraídos de publicações acadêmicas e outras fontes relevantes ou editados por outros autores utilizando a funcionalidade de edição de modelos de *features* online da ferramenta.

Um subconjunto de todos esses modelos de *features* armazenados no repositório S.P.L.O.T foi selecionado manualmente. Para garantir a integridade do *dataset* MACchiATO durante a coleta de novos modelos de *features*, foram determinados alguns critérios de filtro:

- Foram selecionados somente os modelos de *features* relacionados a software (por exemplo, modelos de recursos relacionados a carro, bicicleta, notebook, etc, foram descartados);
- Foram descartados modelos de *features* inconsistentes, com números de *features* e confi-

gurações próximas ao limite aceito pela ferramenta ou modelos duplicados;

- O modelo de *features* selecionado precisa estar disponível para download no formato SXFM (do inglês *Simple XML Feature Model Fomat*), pois é o mesmo formato aceito pela ferramenta DyMMer.
- O modelo de *features* selecionado não pode representar um mesmo produto de um modelo já presente no *dataset* MACchiATO.

Estes filtros foram necessários porque a construção inicial *dataset* MACchiATO destinou-se a fornecer informações reais sobre modelos de *features* de Linhas de Produto de Software (BEZERRA *et al.*, 2016). Portanto, modelos duplicados ou inconsistentes poderiam trazer ameaças à validade para o propósito de utilização deste conjunto de dados. Após esse processo, 108 novos modelos de *features* foram selecionados, armazenados e organizados junto aos modelos já existentes, totalizando 326 modelos de *features*.

5.2 Importação dos modelos de *features* na ferramenta DyMMer

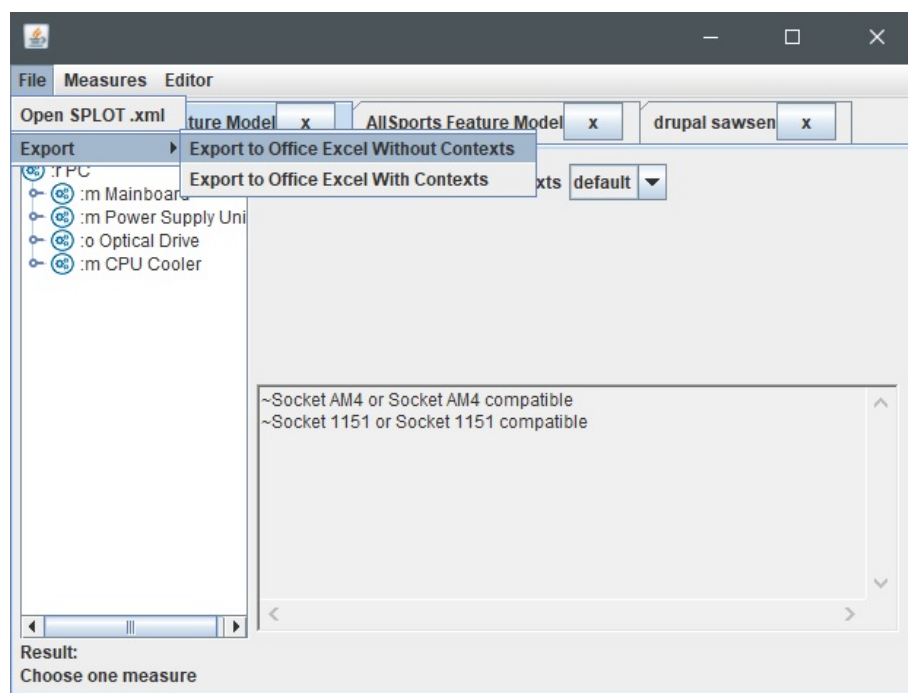
Para realizar a importação dos modelos de *features* selecionados no passo anterior, foi explorada a funcionalidade de Importação de modelos com base em estrutura de arquivos da ferramenta DyMMer e a visualização do modelo de *features* exibido pela aplicação, como ilustrado na Figura 2. Neste passo, cada um dos modelos são importados na ferramenta manualmente e processados de maneira automática pela DyMMer, verificando também possíveis erros de importação ou de formatação do arquivo. Dessa forma, a ferramenta processa um arquivo de modelo de *features* importado e cria um modelo na memória, permitindo realizar o cálculo das medidas de qualidade no passo seguinte (Seção 5.3). Foi possível constatar que atualmente a ferramenta limita-se a importar e processar apenas de 30 modelos de cada vez, o que inviabilizaria a execução deste procedimento para um conjunto maior de dados.

5.3 Extração do valor das medidas

O *dataset* MACchiATO é originalmente composto de um conjunto de 218 modelos de *features* obtidos do repositório S.P.L.O.T e uma planilha compilada contendo os valores das 32 medidas de qualidade do catálogo COFEE para cada um dos 218 modelos. Além de manter um repositório de modelo de *features*, a ferramenta *online* S.P.L.O.T, permite calcular automaticamente 11 medidas de qualidade para estes modelos. Porém, estas medidas representam

apenas um subconjunto do catálogo COFFEE. Por isso a ferramenta DyMMer foi desenvolvida (BEZERRA *et al.*, 2016), permitindo a exportação dos valores das 32 medidas de manutenibilidade presentes no catálogo da seguinte forma: (i) os modelos de *features* são carregados são processados pela ferramenta DyMMer para calcular os valores de medida de qualidade. (ii) Logo após, os valores obtidos de cada medida são compilados e armazenados no formato de planilha do *Microsoft Office Excel*, como ilustrado abaixo na Figura 7.

Figura 7 – Função exportação dos valores das medidas para uma planilha Excel na ferramenta DyMMer.



Fonte: elaborado pelo Autor.

O processo de extração de valores das medidas foi aplicado ao conjunto de 108 novos modelos de *features* obtidos durante a coleta descrita na Seção 5.1. Os valores calculados para cada um dos modelos foram armazenados e acrescentados à planilha original do *dataset*, resultando em uma versão estendida¹ do MACchiATO em ambas as partes: um conjunto maior de modelos de *features* coletados do repositório S.P.L.O.T e os valores das medidas para cada novo modelo.

¹ <http://bit.ly/newMACchiATO>

5.4 Análise geral

Ao final da extensão do MACchiATO, o *dataset* agora contém os valores das medidas do catálogo COFFEE aplicados a um total de 326 modelos de *features*. Neste passo, é possível observar que os novos valores interferem nas estatísticas do *dataset*. A Tabela 2 apresenta um comparativo do valor médio de 15 medidas do catálogo COFFEE (MiniCOFFEE), aplicadas ao conjunto de 218 modelos de *features* do MACchiATO antes da extensão e o valor médio obtido para cada medida depois da adição de 108 novos modelos de *features*.

Tabela 2 – Valor médios das medidas do MiniCOFFEE aplicadas a modelos de *features*

| Medida | Antes | Depois |
|--------|-------|--------|
| NF | 31.55 | 28.11 |
| NM | 9.61 | 8.70 |
| NTop | 5.32 | 5.19 |
| NLeaf | 22.09 | 19.64 |
| DTMax | 4.30 | 4.16 |
| CogC | 4.38 | 4.07 |
| FEX | 24 | 21.61 |
| FoC | 0.26 | 0.24 |
| SCDF | 0.78 | 0.96 |
| MCDF | 1.12 | 1.01 |
| RDen | 0.73 | 0.68 |
| RoV | 2.71 | 2.65 |
| NVC | 2.07 | 1.39 |
| NGOr | 2.39 | 2.08 |
| NGXOr | 1.99 | 1.98 |

Fonte: elaborado pelo Autor.

De modo geral, a extensão MACchiATO contribui positivamente para manutenção e atualização do *dataset*. No entanto, mais modelos de *features* poderiam ser extraídos do repositório S.P.L.O.T, assim como outras medidas de manutenibilidade poderiam ser adicionadas ao *dataset* MACchiATO e implementadas na ferramenta DyMMer, gerando uma atualização de maior impacto e potencializar o uso destes repositórios e ferramentas.

Ao final deste processo, os valores referentes ao subconjunto de 15 medidas de manutenibilidade que compõe o MiniCOFFEE foram armazenados separadamente, assim como as estatísticas de valor médio e desvio padrão das medidas para cada um dos 326 modelos de *features* armazenados, a fim de serem utilizados neste trabalho nas Seções 6 e 7.

6 PROJETO E DEFINIÇÃO DO SISTEMA FUZZY

Neste trabalho, a teoria de lógica *fuzzy* é utilizada para agregar um subconjunto de 15 medidas de manutenibilidade do catálogo COFFEE. A metodologia proposta por Yadav e Yadav (2015) para realizar este processo requer a definição das funções de pertinência de entrada e saída das medidas e o projeto das regras *fuzzy* que irão compor a base de conhecimento do sistema, como ilustra a Figura 4.

Esta Seção descreve os passos realizados após a seleção das medidas que serão agregadas, especificadas na Tabela 1: (i) como as medidas selecionadas foram agrupadas de acordo com as subcaracterísticas de manutenibilidade, (ii) o projeto da função de pertinência e das regras *fuzzy*, (iii) a realização de um processo de revisão por pares para indicar o nível de concordância de especialistas em engenharia de domínio de Linhas de Produto de Software e avaliação da qualidade de modelos de *features* com os parâmetros e regras projetados; e por fim, (iv) a definição final da função de pertinência e regras *fuzzy* a serem utilizados no processo de agregação de medidas e compor o sistema *fuzzy*. Cada uma destas etapas é discutida em detalhes nas próximas seções.

6.1 Agrupamento das medidas

Como visto na Seção 2.3, a Manutenibilidade é um atributo de qualidade do modelo de *features* e possui subcaracterísticas que podem ser acessadas a partir de medidas. Neste etapa, foi necessário identificar as subcaracterísticas de Manutenibilidade e quais medidas do MiniCOFFEE estão relacionadas com cada uma delas. Segundo Bezerra *et al.* (2015) as subcaracterísticas de manutenibilidade específicas para avaliação do modelo de *features* são: Extensibilidade, Flexibilidade, Complexidade Estrutural e Variabilidade .

A subcaracterística de qualidade Extensibilidade pode ser representada pela medida Extensibilidade da *feature* (FEX) e permite mensurar a capacidade de um modelo de *feature* ser estendido e o nível de esforço necessário para realizar a extensão. A subcaracterística Flexibilidade se refere a capacidade que um modelo de *features* tem de responder a possíveis mudanças internas ou externas, de forma que pode ser representada pela medida Flexibilidade da Configuração (FoC). A subcaracterística Complexidade Estrutural representa a complexidade do modelo de *features* no tocante a sua estrutura. Dentre as medidas de complexidade estrutural, estão presente no MiniCOFFEE o Número de *Features* (NF), Número de *Features* Mandatórias

(NM), Profundidade de Árvore (DT Max), Número de *Features* Top (NTop) e a Densidade do Grafo (RDen). Para mensurar a subcaracterística de Variabilidade Estática, estão presentes as medidas: Taxa de Variabilidade (RoV), Número de Grupos Or (NGOr), Número de Grupos XOr (NGXOr), e a Taxa de *Features* XOr (RXOr). A Variabilidade em si é a capacidade de configuração e personalização do modelo de *features*, ou seja, quanto mais baixo for o valor das medidas de variabilidade estática, menor a Variabilidade.

Além destas, algumas das medidas selecionadas podem representar a Analisabilidade, Complexidade Cognitiva e Modularidade do modelo de *features*. A subcaracterística Analisabilidade está relacionada a medida Número de *Features* Folhas (NLeaf), ou seja, quanto maior for a NLeaf, maior é sua analisabilidade. A subcaracterística Complexidade Cognitiva pode ser acessada através da medida Complexidade Cognitiva (CogC) e representa quantitativamente quão fácil o artefato pode ser entendido. Por fim, a subcaracterística de Modularidade está relacionada as medidas *Features* Dependentes Cíclicas Únicas (SCDF) e *Features* Dependentes Cíclicas Múltiplas (MCDF). Portanto, quanto menor o número de SCDF e MCDF, melhor é a modularidade do modelo de *features*.

A partir dessas definições, as 15 medidas de manutenibilidade do MiniCOFFEE foram agrupadas em três grandes grupos (ver Tabela 3): (i) algumas medidas que permitem mensurar o Tamanho do modelo de *features*, relacionadas a Complexidade Estrutural e Analisabilidade; (ii) medidas que permitem determinar a Complexidade geral do modelo, relacionadas com as subcaracterísticas de qualidade Complexidade Cognitiva, Estrutural e Flexibilidade; e (iii) as medidas que fornecem dados sobre a Variabilidade do modelo *features*, também relacionadas a Complexidade Estrutural, Extensibilidade, Modularidade e Variabilidade Estática.

Um modelo de qualidade é composto de características, subcaracterísticas e atributos de qualidade (ISO/IEC, 2011). Os atributos de qualidade são propriedades mensuráveis de uma entidade e as medidas de qualidade são utilizadas para medir, de forma direta ou indireta, as subcaracterísticas e as características de qualidade em um produto de software ou artefato. Partindo destas definições, o agrupamento das medidas realizado nesta etapa resultou em três grupos de medidas que podem representar atributos de qualidade ligados à Manutenibilidade do modelo de *features*: Tamanho, Complexidade e Variabilidade. Portanto, esses grupos foram definidos pois representam atributos que permitem mensurar as subcaracterísticas de manutenibilidade, gerais e específicas de um modelo de *features* através das medidas selecionadas.

Tabela 3 – Medidas agrupadas por subcaracterísticas de manutenibilidade

| Grupo | Subcaracterísticas | Medidas |
|----------------------|---|--|
| Tamanho | Complexidade Estrutural; Analisabilidade | Profundidade da Árvore (DT Max); Número de Features (NF); Número de features folhas (NLeaf); |
| Complexidade | Complexidade Cognitiva; Complexidade Estrutural; Flexibilidade; | Complexidade Cognitiva do Modelo de Features (CogC); Número de Features Top (NTop); Densidade do Grafo (RDen); Flexibilidade da Configuração (FoC); |
| Variabilidade | Complexidade Estrutural; Extensibilidade; Modularidade; Variabilidade Estática; | Número de Features Mandatórias (NM); Extensibilidade da Feature (FEX); Features Dependentes Cíclicas Únicas (SCDF); Features Dependentes Cíclicas Múltiplas (MCDF); Taxa de Variabilidade (RoV); Número de Configurações Válidas (NVC); Número de Grupos Or (NGOr); Número de Grupos XOr (NGXOr) |

Fonte: elaborado pelo Autor.

6.2 Projeto das funções de pertinência de entrada e saída das medidas e regras *fuzzy*

O processo de agregação utilizando lógica *fuzzy* transforma a medida (entrada *crisp*) em uma entrada *fuzzy* com base na função de pertinência de entrada e os termos linguísticos para o conjunto de entradas (notação de valor baixo, médio ou alto). Como visto na Seção 2.5.1, a função de pertinência é essencial no processo de *fuzzificação* do sistema, pois a partir dos parâmetros definidos, o valor de cada medida será automaticamente atribuído a um intervalo e *defuzzificado* em um único valor numérico durante os processos de inferência e composição (ver Figura 4). Por exemplo, os parâmetros para função de pertinência da medida Número de *Features* (NF) no sistema pode ser no mínimo 0 e no máximo 50, sendo o seu valor considerado baixo caso seja menor que 3; considerado médio caso esteja entre 3 e 5 e considerado alto caso seja maior que 5.

Dessa forma, para cada grupo de medidas (definidos na Seção 6.1) foi realizado o projeto dos parâmetros das funções de pertinência com base nas estatísticas do *dataset* obtido na Seção 5. Assim foi possível determinar o valor médio de cada medida aplicada ao conjunto de 326 modelos de *features*, considerando também o desvio padrão dos valores de cada medida para determinar os intervalos que os valores serão considerados "Baixo", "Médio" ou "Alto" no

processo de *fuzzificação*.

Além dos parâmetros utilizados para desenvolver a função de pertinência, o processo de inferência no sistema *fuzzy* utiliza as funções definidas para cada medida de entrada, para determinar as funções de saída com base nas regras *fuzzy* da base de conhecimento (YADAV; YADAV, 2015). Neste trabalho, as regras *fuzzy* foram inicialmente projetadas com base na análise do conjunto de dados obtidos na extensão do *dataset* MACchiATO. Portanto, da mesma forma que foram definidas as funções de pertinência, para cada grupo de medidas, foram projetadas regras que serão computadas durante o processo de composição e resumir os valores de entrada em um valor final. Um exemplo de regra a ser definida, para o grupo de medidas Tamanho, é que se os valores para DTMax, NLeaf e NF são altos, então podemos considerar que o valor numérico resultante da composição das medidas deste grupo também será alto.

6.3 Revisão por pares

Na etapa de *fuzzificação*, os valores das variáveis de entrada são usados para determinar o nível em que cada entrada *crisp* é atribuída na função de pertinência e traduzida em informação de entrada para o domínio *fuzzy* com base nas regras definidas. O projeto inicial destas funções e regras foram discutidos na a Seção anterior, todavia, segundo Yadav e Yadav (2015), as funções de pertinência podem ser geradas com a ajuda de especialistas do domínio ou a partir de dados reais. As regras *fuzzy* também podem ser projetadas a partir da análise de um conjunto de dados, mas em geral, são definidas com o suporte de especialistas.

Sendo assim, é fundamental determinar se os parâmetros para as funções de pertinência e regras *fuzzy* projetadas para o sistema de agregação estão corretos e identificar possíveis erros nestas definições antes da execução dos processos de inferência, *fuzzificação* e *defuzzificação* na etapa de agregação de medidas. Para isso, foi realizada uma revisão por pares a fim de avaliar o nível de concordância dos especialistas com as definições sobre a base de conhecimento do sistema *fuzzy*. A abordagem de revisão por pares consiste na avaliação de um documento, código ou artefato tanto pelo seu autor quanto por um ou mais revisores (CHISM, 1999). Esses revisores fazem comentários ou sugerem revisões no artefato analisado, a fim de avaliar o seu conteúdo técnico e a sua qualidade e contribuindo para a validade do conteúdo examinado.

Nesta etapa do trabalho, três especialistas em engenharia de domínio de Linhas de Produto de Software e avaliação da qualidade conduziram a revisão por pares como revisores dos parâmetros das funções de pertinência e regras *fuzzy* projetadas na Seção 6.2. Dentre eles, um

doutor e dois estudantes de mestrado, todos com mais de 2 anos experiência em Engenharia de Domínio de LPSs, um deles com pelo menos 5 anos de experiência no domínio. O conhecimento em Engenharia de Domínio de LPS e modelo de *features* é considerado alto para dois dos três especialistas, enquanto apenas um indicou possuir conhecimento médio. Em relação ao conhecimento em avaliação da qualidade, dois especialistas indicaram possuir pelo menos 2 anos de experiência na área, e somente um indicou possuir nenhuma experiência. Todos os especialistas informaram ter trabalho com pelo menos 2 projetos envolvendo Linhas de Produto de Software anteriormente.

O processo de revisão dos especialistas foi realizado da seguinte forma: (i) os especialistas foram convidados a participar e a preencher um formulário de revisão com o perfil dos especialistas (ver Apêndice A); (ii) foi enviado a cada especialista participante uma planilha¹ contendo os parâmetros para as funções de pertinência e regras *fuzzy* projetadas na Seção 6.2 e sugeridas para cada medida agregada; (iii) os especialistas julgaram as definições para cada medida tomando como base os dados coletadas na extensão do *dataset* MACchiATO e seu conhecimento sobre domínio; (iv) as discordâncias indicadas pelos especialistas na planilha foram categorizadas em fato incorreto, ambiguidade ou inconsistência; e por fim, (v) a partir das sugestões, comentários e novas regras *fuzzy* sugeridas pelos especialistas, foi possível iniciar o processo de definição final das funções pertinência e regras de da lógica *fuzzy* para cada uma das medidas do MiniCOFFEE.

6.4 Definição das funções de pertinência

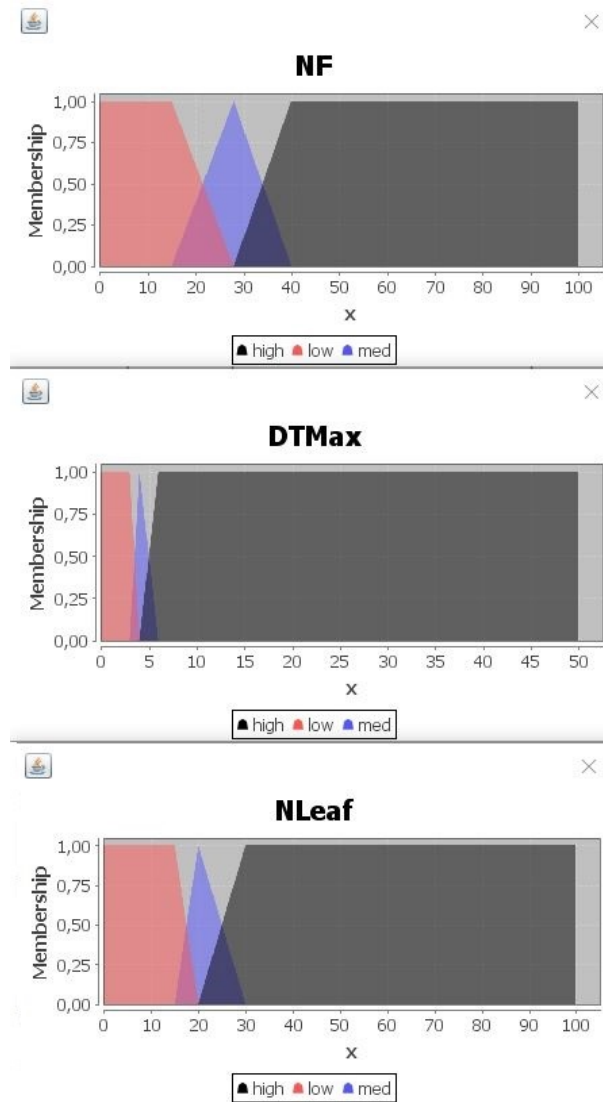
Segundo Yadav e Yadav (2015), uma função de pertinência apresentada em formas triangulares e trapezoidais simplifica o processo de computação (realizado na Seção 7) e permite que o conhecimento dos especialistas do domínio, participantes da revisão por pares, sejam representados de forma adequada nos intervalos definidos. A partir da especificação dos parâmetros com o suporte de especialistas (ver Apêndice B), esses parâmetros foram computados na ferramenta jFuzzyLogic e foram obtidas as funções de pertinência do sistema *fuzzy* para cada grupo de medidas.

A partir da definição dos parâmetros para as medidas Profundidade da Árvore (DT Max), Número de *Features* (NF) e Número de *Features* folhas (NLeaf), foi obtido automaticamente os intervalos das funções de pertinência para o grupo de medidas relacionadas ao tamanho

¹ <http://bit.ly/PeerReviewAggregationMetrics>

do modelo de *features*. A Figura 8 apresenta um exemplo dos intervalos de função de pertinência destas medidas.

Figura 8 – Intervalos das funções de pertinência para as medidas de Tamanho



Fonte: elaborado pelo Autor.

O mesmo foi realizado para o grupo de medidas relacionadas a Complexidade do modelo de *features*, obtendo funções de pertinência definidas a partir dos parâmetros das medidas Complexidade Cognitiva do Modelo de *Features* (CogC), Número de *Features* Top (NTop), Densidade do Grafo (RDen) e Flexibilidade da Configuração (FoC), e por fim, também para o grupo de medidas relacionadas a Variabilidade: Número de *Features* Mandatórias (NM), Extensibilidade da *Feature* (FEX), *Features* Dependentes Cíclicas Únicas (SCDF), *Features* Dependentes

Cíclicas Múltiplas (MCDF), Taxa de Variabilidade (RoV), Número de Configurações Válidas (NVC), Número de Grupos Or (NGOr) e XOr (NGXOr).

6.5 Definição das regras *fuzzy*

Neste trabalho, as regras *fuzzy* também foram projetadas com a ajuda dos especialistas do domínio e foram são definidas considerando as características referentes à cada uma das medidas selecionadas para compor o processo de agregação. Para determinar o índice de Tamanho, Complexidade e Variabilidade, foram determinados estados linguísticos: muito baixo (VL), baixo (L), médio (M), alto (H) ou muito alto (VH). Todas as condições para as regras *fuzzy* definidas são apresentadas no apêndice C deste trabalho. Uma amostra da implementação dessas regras são apresentadas nas Tabelas 4, 5 e 6.

Por exemplo, se os valores para DTMax, NLeaf e NF são altos, então o tamanho do modelo de *features* também pode ser considerado é alto. Cada uma das medidas de entrada possui três estados linguísticos, ou seja, baixo (L), médio (M) e alto (H).

Tabela 4 – Amostra de regras *fuzzy* para medidas de Tamanho.

| Regras | Regras Fuzzy |
|--------|--|
| 1 | SE DTMax é L e NLeaf É L e NF É L ENTÃO o Tamanho é VL |
| 2 | SE DTMax é M e NLeaf É L e NF É L ENTÃO o Tamanho é L |
| ... | ... |
| 9 | SE DTMax é M e NLeaf É H e NF É M ENTÃO o Tamanho é H |
| 10 | SE DTMax é M e NLeaf É H e NF É H ENTÃO o Tamanho é H |
| 11 | SE DTMax é H e NLeaf É H e NF É H ENTÃO o Tamanho é VH |

Fonte: elaborado pelo Autor.

Para o grupo de medidas de Complexidade, um exemplo é: se os valores para RDen, CogC, NTop são baixos e FoC é alto, então a complexidade do modelo de *features* pode ser considerada baixa. Cada uma das medidas de entrada deste grupo também possui três estados linguísticos, ou seja, baixo (L), médio (M) e alto (H).

Tabela 5 – Amostra de regras *fuzzy* para medidas de Complexidade.

| Regras | Regras Fuzzy |
|--------|---|
| 1 | SE RDen É L e CogC É L e NTop É L e FoC É H ENTÃO a Complexidade É VL |
| 2 | SE RDen É L e CogC É L e NTop É M e FoC É L ENTÃO a Complexidade É M |
| ... | ... |
| 32 | SE RDen É H e CogC É H e NTop É L e FoC É M ENTÃO a Complexidade É H |
| 33 | SE RDen É H e CogC É L e NTop É M e FoC É L ENTÃO a Complexidade É H |
| 34 | SE RDen É H e CogC É L e NTop É H e FoC É L ENTÃO a Complexidade É VH |

Fonte: elaborado pelo Autor.

Por fim, o mesmo acontece para o grupo de medidas de Variabilidade, onde se os valores para NM, SCDF, MCDF, NGXOr e NGOR forem baixos e ao mesmo tempo FEX, RoV e NVC forem altos, implica que a variabilidade do modelo de *features* pode ser considerada muito alta. O três estados linguísticos, baixo (L), médio (M) e alto (H), também foram determinadas para cada uma das medidas do grupo.

Tabela 6 – Amostra de regras *fuzzy* para medidas de Variabilidade.

| Regras | Regras Fuzzy |
|--------|--|
| 1 | SE NM É L e FX É H e SCDF É L e MCDF É L e RoV é H e NVC é H e NGXOr é L e NGOR é H ENTÃO a Variabilidade É VL |
| 2 | SE NM É L e FX É H e SCDF É L e MCDF É L e RoV é M e NVC é M e NGXOr é L e NGOR é H ENTÃO a Variabilidade É M |
| ... | ... |
| 37 | SE NM É H e FX É M e SCDF É M e MCDF É M e RoV é M e NVC é M e NGXOr é H e NGOR é H ENTÃO a Variabilidade É H |
| 38 | SE NM É H e FX É M e SCDF É H e MCDF É H e RoV é L e NVC é L e NGXOr é H e NGOR é L ENTÃO a Variabilidade É VH |
| 39 | SE NM É H e FX É M e SCDF É H e MCDF É H e RoV é L e NVC é M e NGXOr é M e NGOR é L ENTÃO a Variabilidade É VH |

Fonte: elaborado pelo Autor.

7 AGREGAÇÃO DE MEDIDAS DE MANUTENIBILIDADE

Nesta Seção, é descrito o processo de agregação das medidas e discutido como os resultados podem ajudar na avaliação da manutenibilidade do modelo de *features*. Como discutido na Seção 2.5, entende-se que valores individuais de medidas não são suficientes para caracterizar a manutenibilidade geral do modelo de *features*. Cada medida está relacionada a uma subcaracterística de qualidade e muitas vezes, correlacionadas com outras medidas. Portanto, a fim de obter um valor numérico que permita mensurar uma característica de qualidade e suas subcaracterísticas, é possível aplicar uma técnica de agregação de medidas. O objetivo principal do processo de agregação é obter um grau de consenso entre as informações disponíveis nas medidas que compõe o conjunto de entrada do processo e o valor agregado obtido no final (MORDAL *et al.*, 2013).

A teoria de lógica *fuzzy* foi utilizada como técnica de agregação neste trabalho e aplicada ao conjunto de 15 medidas do subconjunto MiniCOFFEE através da ferramenta jFuzzyLogic. De acordo com a metodologia utilizada para este processo, (i) as medidas foram selecionadas, (ii) agrupadas de acordo com as subcaracterísticas de manutenibilidade, (iii) logo após foram projetadas e definidas funções de pertinência e regras *fuzzy* para cada medida com apoio de especialistas e (iii) realizada a execução do processo de *fuzzificação* e *defuzzificação*, inferência e composição, obtendo assim um único valor agregado. As próximas seções descrevem com detalhes cada etapa deste processo.

7.1 Implementação do sistema *fuzzy*

Para realizar os procedimentos de *fuzzificação*, inferência, composição e *defuzzificação*, as medidas selecionadas para compor o conjunto de entradas, os parâmetros das funções de pertinência e as regras *fuzzy* foram computadas na ferramenta jFuzzyLogic. Esta ferramenta é uma biblioteca *open source* para sistemas *fuzzy*, desenvolvida por Cingolani e Alcalá-Fdez (2013) com o objetivo de facilitar o desenvolvimento de controladores lógicos para sistemas *fuzzy* na linguagem Java. A escolha dessa ferramenta se deu principalmente por possuir documentação adequada e código aberto¹, facilitando seu uso e incentivando a colaboração entre a comunidade.

Para obter o valor agregado das 15 medidas de manutenibilidade do subconjunto MiniCOFFEE, cada medida é considerada como uma entrada real do sistema *fuzzy*. Na primeira

¹ <http://jfuzzylogic.sourceforge.net>

etapa deste processo, as medidas são *fuzzificadas* em entradas *fuzzy*, ou seja, os parâmetros definidos para as funções de pertinência de cada medida são computados pela ferramenta e são determinados automaticamente os intervalos apropriados para os valores de saída, como exemplificado na Figura 8. A implementação desta etapa para as 15 medidas de manutenibilidade é apresentada na Figura 9. Como exemplo, é possível observar que as medidas NF e NM serão *fuzzificadas* nos termos linguísticos baixo, médio ou alto, de acordo com os parâmetros definidos pelos especialistas e o mesmo acontece para todas as medidas.

Figura 9 – Implementação dos parâmetros para as funções de pertinência das medidas de manutenibilidade

```

1 FUNCTION_BLOCK maintainability_index
2
3 VAR_INPUT           // Define input measures
4     NF : REAL;
5     NM : REAL;
6     NTop : REAL;
7     NLeaf : REAL;
8     DTMax : REAL;
9     CogC : REAL;
10    FEX : REAL;
11    FoC : REAL;
12    SCDF : REAL;
13    MCDF : REAL;
14    RDen : REAL;
15    RoV : REAL;
16    NVC : REAL;
17    NGOr : REAL;
18    NGXOr : REAL;
19 END_VAR
20
21 VAR_OUTPUT         // Define output measure
22    MIFM : REAL;
23 END_VAR
24
25 FUZZIFY NF // Fuzzify input measure in : {'low', 'med', 'high'}
26     TERM low := (0, 1) (15, 1) (28, 0) ;
27     TERM med := (15, 0) (28, 1) (40, 0);
28     TERM high := (28, 0) (40, 1) (100, 1);
29 END_FUZZIFY
30
31 FUZZIFY NM // Fuzzify input measure in : {'low', 'med', 'high'}
32     TERM low := (0, 1) (5, 1) (9, 0) ;
33     TERM med := (5, 0) (9, 1) (15, 0);
34     TERM high := (9, 0) (15, 1) (50, 1);
35 END_FUZZIFY

```

Fonte: elaborado pelo Autor.

Nesta implementação, as regras *fuzzy* definidas para cada grupo de medidas são inseridas na ferramenta em diferentes blocos, mas consideradas no processo de inferência de de

forma única. Partindo do seguinte pressuposto: quanto maior o valor obtido para os atributos de qualidade de tamanho, complexidade e variabilidade de um modelo de *features*, menor será o seu índice de manutenibilidade, para cada condicional, o termo linguístico resultante para MIFM é o inverso ao termo linguístico da regra original. Por exemplo, uma regra que antes resultava no termo linguístico "muito baixo" para um conjunto de medidas de tamanho, nesta etapa, essa regra agora irá implicar um termo linguístico "muito alto" para a nova medida MIFM. O mesmo ocorre para as medidas de complexidade e variabilidade. Assim, para cada valor obtido de cada medida, um nível de veracidade é calculado e aplicado para o valor de saída com base em cada regra, garantindo que variável de saída pertence ao intervalo apropriado da função de pertinência. A Tabela 7 ilustra esta implementação apresentando uma amostra de como estas regras foram declaradas na ferramenta jFuzzyLogic.

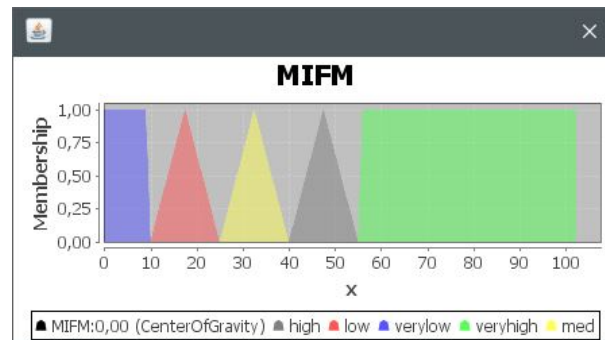
Tabela 7 – Amostra de regras *fuzzy* para as medidas de entrada do MIFM.

| Regras | Todas as regras Fuzzy |
|--------|--|
| 1 | SE DTMax é M e NLeaf é L e NF é L ENTÃO o MIFM é H |
| 2 | SE DTMax é M e NLeaf é H e NF é M ENTÃO o MIFM é L |
| ... | ... |
| 37 | SE RDen é M e CogC é L e NTop é L e FoC é M ENTÃO o MIFM é M |
| 38 | SE RDen é H e CogC é L e NTop é H e FoC é L ENTÃO o MIFM é VL |
| ... | ... |
| 79 | SE NM é M e FEX é L e SCDF é H e MCDF é H e RoV é L e NVC é L e NGXOr é M e NGOR é L ENTÃO o MIFM é VH |
| 80 | SE NM é H e FEX é M e SCDF é H e MCDF é H e RoV é M e NVC é M e NGXOr é H e NGOR é L ENTÃO o MIFM é VL |

Fonte: elaborado pelo Autor.

O valor de saída MIFM (Índice de Manutenibilidade do Modelo de *Features*) é obtido então a partir do processo de inferência, composição e *defuzzificação*. No processo de inferência, as entradas *fuzzy* são aplicadas em cada regra da Tabela 7, para determinar o nível de pertinência do valor obtido com as funções de cada medida. A etapa de composição é responsável por combinar todas as funções de pertinência obtidas a partir de cada variável de saída e formar uma única função de pertinência para variável de saída, ou seja, para a medida agregada MIFM. A Figura 10 apresenta esta função de pertinência obtida para o Índice de Manutenibilidade do Modelo de *Features*, que sugere que o valor de MIFM pode ser classificado como muito baixo, baixo, médio, alto ou muito alto.

Figura 10 – Função de pertinência para o Índice de Manutenibilidade do Modelo de Features (MIFM)



Fonte: elaborado pelo Autor.

Durante o processo de agregação, os intervalos obtidos para cada valor das medidas de entrada são combinados (considerando as funções de pertinência de entrada e saída) e a variável de saída é *defuzzificada*. Neste caso, o processo de *defuzzificação* produz um valor numérico para MIFM, e a função de pertinência resultante da etapa anterior é *defuzzificada* em um único número, que representa o valor do Índice de Manutenibilidade do Modelo de *Features*. A medida agregada obtida é capaz de indicar a manutenibilidade de um modelo de *features*, com base em diferentes aspectos relacionados ao tamanho, complexidade e variabilidade. Estes grupos de medidas de entrada englobam as subcaracterísticas de manutenibilidade específicas ao modelo de *features*: Extensibilidade, Flexibilidade, Complexidade Estrutural e Variabilidade Estática, além da Analisabilidade, Complexidade Cognitiva e Modularidade.

Portanto, com base na função de pertinência obtida para a nova medida agregada, resultante do processo de agregação utilizando Lógica *Fuzzy* e das regras *fuzzy* definidas com apoio de especialistas, o Índice de Manutenibilidade do Modelo de *Features* é calculado a partir dos valores das 15 medidas de manutenibilidade, retornando um único valor numérico que permite mensurar se o modelo de *features* possui boa manutenibilidade ou não. Ou seja, quanto maior o valor de MIFM para um modelo, melhor é sua manutenibilidade.

7.2 Avaliação do Índice de Manutenibilidade

Neste trabalho, o processo de agregação de medidas foi projetado com base nos dados extraídos de especialistas, como apresentado na Seção 6, considerando o grau de concordância com os parâmetros e regras que compõem a base de conhecimento do sistema *fuzzy* que incorpora este processo. Ainda assim, a fim de avaliar o uso da nova medida agregada, o MIFM foi aplicado

a uma amostra de 20 modelos de *features* de Linhas de Produto de Software reais, extraídos do repositório S.P.L.O.T durante o procedimento realizado na Seção 5.1. Para cada modelo coletado, foram calculados os valores das 15 medidas de manutenibilidade do subconjunto de medidas MiniCOFFEE, e adicionados na ferramenta.

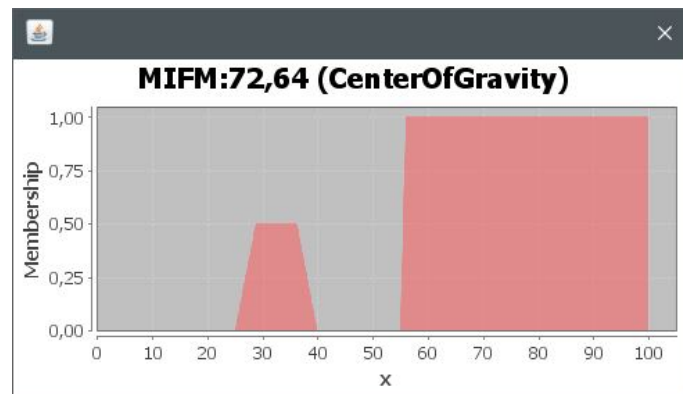
Dessa forma, a partir desses valores, o valor do Índice de Manutenibilidade foi obtido dos 20 modelos de *features*. De acordo com a função de pertinência obtida para a medida agregada MIFM durante o processo de composição e *defuzzificação*, o valor da nova medida possui um intervalo de 0 a 100, que permite classificar a manutenibilidade do modelo de *features* em: muito baixa, baixa, média, alta ou muito alta. A partir da análise dos valores de MIFM obtidos na aplicação, foi possível observar que: nenhum dos modelos possuem o índice de manutenibilidade considerado muito baixo; 3 deles possuem manutenibilidade classificada como baixa; 6 modelos possuem o índice de manutenibilidade médio; enquanto 4 modelos obtiveram o índice de manutenibilidade alto; e 7 modelos tiveram o índice de manutenibilidade classificado como muito alto. Os resultados desta aplicação são apresentados na Tabela 8.

Tabela 8 – Resultados da aplicação do MIFM em modelos de *features*.

| Modelos de Features | MIFM |
|---------------------------------|-------------|
| Telecommunication System | 47.49 |
| Heroku | 36.28 |
| Jetbrains | 33.80 |
| VR Application | 32.500 |
| Agent Illustration System | 32.49 |
| Weather Station | 77.72 |
| Mobile Tourist Planner | 72.64 |
| iNavBasic | 70.60 |
| Student Attendance System | 69.51 |
| Amazon Web Services | 42.721 |
| Tree Stem Deformation Simulator | 11.49 |
| Emergency Response Application | 17.49 |
| Voice Recording System | 73.25 |
| Poker Game | 77.72 |
| Card Payment Terminal | 40.20 |
| ESCA Embeeded System | 47.5 |
| Simple Drawing Tools | 74.18 |
| VMS | 34.64 |
| SWAMP Platform | 32.49 |
| Text To Speech App | 19.91 |

Apenas 15% dos modelos de *features* avaliados possui manutenibilidade baixa. 30% dos modelos possui o índice de manutenibilidade considerado médio, enquanto a maioria dos modelos de *features* (55%) possuem manutenibilidade considerada alta ou muito alta, o que indica que esses modelos, em sua maioria, são de tamanho, complexidade e variabilidade baixa. Por exemplo, para o modelo de *features Mobile Tourist Planner* que possui os valores NF (10), NM (3), NTop (4), NLeaf (7), DTMax (3), CogC (2), FEX (7), FoC (0.1), SCDF (0), MCDF (0), RDen (0), RoV (2.2), NVC (28), NGOR (1) e NGXOR (1), o valor de MIFM obtido para este modelo foi de 72.64, como apresenta a Figura 11 em que o centro de gravidade da medida MIFM é localizado no intervalo muito alto.

Figura 11 – Índice de Manutenibilidade do Modelo de *Features Mobile Tourist Planner*



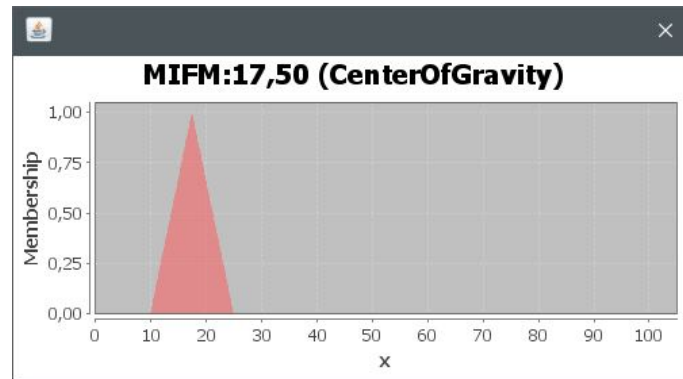
Fonte: elaborado pelo Autor.

Segundo os parâmetros definidos com ajuda de especialistas, é possível analisar a estrutura do modelo de *features Mobile Tourist Planner* e inferir que se trata de um modelo consideravelmente pequeno, pois possui valores baixos para medidas relacionadas ao tamanho (NF, NLeaf), assim com também possui valores considerados baixos para medidas relacionadas a Complexidade Estrutural, como NTop e RDen, Complexidade Cognitiva (CogC) e valores baixos para medidas relacionadas a Variabilidade (e.g. NVC e NM). Estes valores indicam que o modelo possui baixos índices de complexidade e baixos índices de variabilidade. Isso têm um impacto positivo no índice de manutenibilidade do modelo (72.64), classificando-o com manutenibilidade muito alta de acordo com a função de pertinência definida automaticamente para a medida MIFM.

Por outro lado, a aplicação do MIFM no modelo de *features Emergency Response Application*, que possui os NF (43), NM (22), NTop (3), NLeaf (32), DTMax (4), CogC (5),

FEX (32), FoC (0.13), SCDF (0), MCDF (0), RDen (0), RoV (2.93), NVC (23040), NGOr (4) e NGXOr (1) retorna um MIFM de valor 17.50, como mostra a Figura 12.

Figura 12 – Índice de Manutenibilidade do Modelo de *Features Emergency Response Application*



Fonte: elaborado pelo Autor.

Para o modelo de *features Emergency Response Application*, o centro de gravidade da medida MIFM é localizado no intervalo baixo, ou seja, indica que o modelo possui baixa manutenibilidade. Este resultado também se mostra consistente com o projeto das funções de pertinência e das regras *fuzzy* definidas com ajuda de especialistas a partir da revisão por pares. Por exemplo, os números elevados de configurações válidas (NVC) e de número de *features* (NF) permite inferir, respectivamente, que se trata de um modelo de *features* de variabilidade e tamanho considerado alto. As medidas de entrada agrupadas pelas subcaracterísticas de complexidade não possuem valores altos neste modelo, o que justifica a classificação do valor obtido de MIFM como um índice baixo em vez de classificá-lo muito baixo, dada sua alta variabilidade e tamanho.

Portanto, é possível argumentar que a nova medida agregada pode ser usada por engenheiros de domínio para inferir sobre a manutenibilidade geral de um modelo de *features*, uma vez que o processo de agregação desta medida utiliza como entrada medidas de manutenibilidade que abrangem as subcaracterísticas deste atributo de qualidade e foram agrupadas em relação aos aspectos gerais de tamanho, variabilidade e complexidade agregados em um único valor, em vez de utilizar o resultado individual de cada medida para realizar a avaliação da manutenibilidade deste artefato.

8 CONSIDERAÇÕES FINAIS

Neste trabalho foi realizada a agregação de 15 medidas de manutenibilidade do modelo de *features* em Linhas de Produto de Software, obtendo uma nova medida agregada: o Índice de Manutenibilidade do Modelo de *Fatures* (MIFM). Para isso, foi utilizada a teoria de Lógica *Fuzzy* como técnica de agregação de medidas, de acordo com a metodologia proposta por Yadav e Yadav (2015), que consistiu em (i) selecionar as medidas que irão compor o conjunto de entrada; (ii) definir funções de pertinência de entrada e saída das medidas; (iii) projetar as regras *fuzzy*; (iv) executar o processo de *fuzzificação*, inferência, composição e *defuzzificação* do conjunto de medidas de entrada obtendo uma nova medida agregada e (v) aplicar a nova medida obtida a um conjunto de modelos de *features* a fim de avaliar seu uso.

Um dos pontos positivos para realização deste processo foi a proveitosa curva de aprendizado sobre medição de software, qualidade de software, linhas de produtos de software e lógica *fuzzy*, além de contribuir propondo uma nova medida que pode ser usada para apoiar a avaliação da qualidade do modelo de *features* em LPS.

Outro ponto positivo foi a contribuição com a extensão e atualização do *dataset*, pois além de utilizá-lo, foram adicionados 108 novos modelos e o respectivos valores das 32 medidas ao conjunto de dados, além de utilizar as estatísticas de valor médio e desvio padrão das medidas para as definições do sistema *fuzzy*. Os parâmetros das funções de pertinência de cada medida do MiniCOFFEE foram projetados e posteriormente definidos com o apoio de especialistas através de uma revisão por pares. A fase de execução utilizando as definições do sistema *fuzzy* foi possível graças a ferramenta jFuzzyLogic, onde foram implementada as definições do processo de agregação e obtida a medida agregada.

A medida MIFM foi aplicada em um conjunto de 20 modelos de *features* retirados do repositório S.P.L.O.T. A análise dos valores coletados para cada um dos modelos sugere que a medida agregada MIFM pode apoiar a avaliação da manutenibilidade do modelo de *features*, assim como também foram identificados algumas melhorias e limitações no processo de agregação e validação dos resultados.

Com a realização deste trabalho, espera-se que a nova medida agregada possa auxiliar engenheiros de domínio e pesquisadores na redução do tempo e esforço da avaliação da manutenibilidade do modelo de *features* a partir de um só valor numérico. E ainda, que a abordagem para agregação de medidas adotada possa ser reutilizada em mais medidas para acessar outras características de qualidade.

Esta Seção descreve as considerações do finais do trabalho. Primeiramente serão apresentadas as possíveis ameaças à validade dos resultados obtidos. Em seguida, são apresentadas as contribuições deste trabalho, e por fim, os possíveis trabalhos futuros são levantados.

8.1 Ameaças à validade

São discutidas nesta Seção as ameaças que podem interferir na validade dos resultados obtidos neste trabalho. A primeira possível ameaça deste trabalho é sobre a efetividade da revisão por pares realizada, uma vez que a quantidade de participantes foi baixa, contando apenas com três especialistas. Embora os participantes possuam conhecimentos específicos que envolvem o domínio do estudo, a baixa variedade de opiniões e sugestões tendem a impactar na acurácia dos parâmetros e regras definidos, e conseqüentemente nos resultados obtidos.

Outra ameaça identificada é a questão do conhecimento dos especialistas que participaram da revisão. A proposta de obter um único valor numérico que permita mensurar várias subcaracterísticas de manutenibilidade ao mesmo tempo é algo complexo, e faz necessário que o conhecimento dos especialistas em avaliação da qualidade do modelo de *features* fosse específico para estas subcaracterísticas.

Uma terceira possível ameaça está relacionada com o número pequeno de modelo de *features* coletados para estender o *dataset* MACchiATO e a quantidade de modelos reais. Os modelos foram utilizados para obter as estatísticas utilizadas nas definições do sistema *fuzzy* e para aplicar a nova medida agregada obtida (MIFM), mas muitos se tratam de modelos de *feature* representativos, retirados do repositório S.P.L.O.T. Desse modo, a aplicação da abordagem realizada neste trabalho poderia ser aplicada em um conjunto de modelos de *features* reais, resultando em uma maior confiabilidade do valores médios das medidas e validade do uso da nova medida agregada aplicada a estes modelos.

8.2 Contribuições

As principais contribuições identificadas com a realização deste trabalho foram:

- Extensão do número de modelo de *features* do repositório e dos valores das medidas do *dataset* MACchiATO aplicadas aos novos modelos;
- Definição de parâmetros para funções de pertinência de 15 medidas de manutenibilidade do subconjunto do catálogo COFFEE (MiniCOFFEE). Estes parâmetros podem ser reutilizados

em outros processos de agregação de medidas utilizando lógica *fuzzy* ou como base para uma revisão por pares a fim de obter novos parâmetros;

- Definição de funções de pertinência e regras *fuzzy* para cada medida do subconjunto de medidas MiniCOFFEE, permitindo classificar o valor de uma medida aplicada a um modelo de *feature* como baixo, médio ou alto;
- Obtenção de uma nova medida agregada que apoia quantitativamente a avaliação da manutenibilidade geral do modelo de *features* em Linhas de Produto de Software, utilizando como entrada os valores de 15 medidas de manutenibilidade.

8.3 Trabalhos Futuros

A partir dos resultados obtidos e contribuições identificadas neste trabalho, foi possível observar melhorias e possíveis trabalhos futuros. Ao utilizar a Lógica *Fuzzy* como técnica de agregação, fez-se necessário definir funções de pertinência a partir de dados reais ou com a ajuda de especialistas do domínio, o que resultou em uma atividade extensa no processo: a aplicação de uma revisão por pares ou algum procedimento de validação externa. Por isso, uma possível melhoria seria utilizar um conjunto de modelo de *features* reais, a fim de utilizar os dados estatísticos da aplicação das medidas nestes modelos e realizar a definição dos parâmetros. No entanto, mesmo utilizando de dados reais, ainda seria necessário a ajuda de especialistas para definir as regras *fuzzy* para variável de entrada e saída do processo de agregação.

Dessa forma, um trabalho futuro seria investigar outras técnicas de agregação de medidas que permitam realizar o processo de maneira analítica e automatizada. Investigar outras técnicas de agregação de medidas adequadas podem reduzir os esforços dos especialistas que participarem de qualquer tipo de revisão, pois a dificuldade da definição de funções de pertinências e regras *fuzzy* é diretamente proporcional ao tamanho do conjunto de medidas de entrada. Um outro trabalho possível seria utilizar uma abordagem de validação externa com especialistas, a fim de avaliar o uso da nova medida agregada obtida. Além disso, outro trabalho possível seria realizar a adaptação dos procedimentos metodológicos, e aplicação dessa metodologia em medidas para avaliação de modelos de *features* com ou sem contexto, permitindo avaliar modelos também em Linhas de Produto de Software Dinâmicas.

REFERÊNCIAS

- ACHER, M.; COLLET, P.; LAHIRE, P.; FRANCE, R. B. Familiar: A domain-specific language for large scale management of feature models. **Science of Computer Programming**, Elsevier, v. 78, n. 6, p. 657–681, 2013.
- BABAR, M. A.; CHEN, L.; SHULL, F. Managing variability in software product lines. **IEEE software**, IEEE, v. 27, n. 3, p. 89–91, 2010.
- BAGHERI E., G. D. Assessing the maintainability of software product line feature models using structural metrics. **Softw. Qual. Cont.**, p. 579–612, 2011.
- BELCHIOR, A. D. **Um modelo fuzzy para avaliação da qualidade de software**. Tese (Doutorado) — Engenharia de Sistemas e Computação, COPPE, Universidade, 1997.
- BENAVIDES, D.; SEGURA, S.; RUIZ-CORTÉS, A. Automated analysis of feature models 20 years later: A literature review. **Information Systems**, Elsevier, v. 35, n. 6, p. 615–636, 2010.
- BENAVIDES D., S. S. R.-C. A. Automated analysis of feature models 20 years later: a literature review. **Inform. Syst.** **35** (6), p. 615–636, 2010.
- BEUCHE, D.; DALGARNO, M. Software product line engineering with feature models. **Overload Journal**, v. 78, p. 5–8, 2007.
- BEZERRA, C.; ANDRADE, R.; MONTEIRO, J. M. Measures for quality evaluation of feature models. Schaefer, I., Stamelos, I. (Eds.), **Software Reuse for Dynamic Systems in the Cloud and Beyond**. In: **Lecture Notes in Computer Science, 8919**., Springer International Publishing, p. 282–297, 2015.
- BEZERRA, C.; ANDRADE, R.; MONTEIRO, J. M. Exploring quality measures for the evaluation of feature models: a case study. **The Journal of Systems and Software**, Elsevier, p. 1–20, 2016.
- BEZERRA, C. I.; ANDRADE, R.; MONTEIRO, J.; CEDRAZ, D. Aggregating measures using fuzzy logic for evaluating feature models. In: ACM. **Proceedings of the 12th International Workshop on Variability Modelling of Software-Intensive Systems**. [S.l.], 2018. p. 35–42.
- BEZERRA, C. I.; ANDRADE, R. M.; MONTEIRO, J. M. S. Measures for quality evaluation of feature models. In: SPRINGER. **International Conference on Software Reuse**. [S.l.], 2015. p. 282–297.
- BEZERRA, C. I.; BARBOSA, J.; FREIRES, J. H.; ANDRADE, R.; MONTEIRO, J. M. Dymmer: a measurement-based tool to support quality evaluation of dspl feature models. In: ACM. **Proceedings of the 20th International Systems and Software Product Line Conference**. [S.l.], 2016. p. 314–317.
- BOEHM, B. W.; BROWN, J. R.; LIPOW, M. Quantitative evaluation of software quality. In: IEEE COMPUTER SOCIETY PRESS. **Proceedings of the 2nd international conference on Software engineering**. [S.l.], 1976. p. 592–605.
- CAPILLA, R.; BOSCH, J. The promise and challenge of runtime variability. **Computer**, IEEE, v. 44, n. 12, p. 93–95, 2011.

- CHISM, N. V. N. **Peer Review of Teaching. A Sourcebook.** [S.l.]: ERIC, 1999.
- CINGOLANI, P.; ALCALÁ-FDEZ, J. jfuzzylogic: a java library to design fuzzy logic controllers according to the standard for fuzzy control programming. p. 61–75, 2013.
- CLEMENTS, P.; NORTHROP, L. **Software product lines.** [S.l.]: Addison-Wesley, 2002.
- COHEN, S. **Product line state of the practice report.** [S.l.], 2002.
- COLEMAN, D.; ASH, D.; LOWTHER, B.; OMAN, P. Using metrics to evaluate software system maintainability. **Computer**, IEEE, v. 27, n. 8, p. 44–49, 1994.
- COX, E. D. **Fuzzy logic for business and industry.** [S.l.]: Charles River Media, Inc., 1995.
- DASSO, A.; FUNES, A. Software quality metrics aggregation. In: **13th Argentine Symposium on Software Engineering, ASSE.** [S.l.: s.n.], 2012. p. 312–323.
- ETXEGERIA, L.; SAGARDUI, G. Product-line architecture: New issues for evaluation. In: SPRINGER. **International Conference on Software Product Lines.** [S.l.], 2005. p. 174–185.
- FENTON, N. Software measurement: A necessary scientific basis. **IEEE Transactions on software engineering**, IEEE, v. 20, n. 3, p. 199–206, 1994.
- FINKELSTEIN, L. Widely, strongly and weakly defined measurement. **Measurement**, Elsevier, v. 34, n. 1, p. 39–48, 2003.
- HETTIARACHCHI, C.; DO, H.; CHOI, B. Risk-based test case prioritization using a fuzzy expert system. **Information and Software Technology**, Elsevier, v. 69, p. 1–15, 2016.
- IEE, E. Ieee standard glossary of software engineering terminology. 1990.
- ISO/IEC. **Systems and Software Engineering: Systems and Software Quality Requirements and Evaluation (SQuaRE): System and Software Quality Models.** [S.l.]: ISO/IEC, 2011.
- ISO/IEC:2008. **ISO/IEC 12207:2008. Systems and software engineering software life cycle processes. Technical Report.** [S.l.: s.n.].
- KAN, S. H. **Metrics and models in software quality engineering.** [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 2002.
- KANG, K.; COHEN, S.; HESS, J.; NOVAK, W.; PETERSON, A. **Feature-Oriented Domain Analysis (FODA) Feasibility Study.** Pittsburgh, PA, 1990.
- KLIR, G.; YUAN, B. **Fuzzy sets and fuzzy logic.** [S.l.]: Prentice hall New Jersey, 1995. v. 4.
- LINDEN, F. Van der; SCHMID, K.; ROMMES, E. The product line engineering approach. In: **Software Product Lines in Action.** [S.l.]: Springer, 2007. p. 3–20.
- MENDONCA, M. **Efficient reasoning techniques for large scale feature models.** Dissertação (Ph.D. Thesis) — University of Waterloo, Canada, 2009.
- MENDONCA, M.; BRANCO, M.; COWAN, D. Splot: software product lines online tools. In: ACM. **Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications.** [S.l.], 2009. p. 761–762.

- MORDAL, K.; ANQUETIL, N.; LAVAL, J.; SEREBRENIK, A.; VASILESCU, B.; DUCASSE, S. Software quality metrics aggregation in industry. **Journal of Software: Evolution and Process**, Wiley Online Library, v. 25, n. 10, p. 1117–1135, 2013.
- OLIVEIRA, K. R. de; BELCHIOR, A. D. **AdeQuaS**: ferramenta fuzzy para avaliação da qualidade de software. Tese (Doutorado) — Universidade de Fortaleza, 2002.
- PACHECO, C. L.; GARCIA, I. A.; CALVO-MANZANO, J. A.; ARCILLA, M. A proposed model for reuse of software requirements in requirements catalog. **Journal of Software: Evolution and Process**, Wiley Online Library, v. 27, n. 1, p. 1–21, 2015.
- PAREJO, J. A.; SÁNCHEZ, A. B.; SEGURA, S.; RUIZ-CORTÉS, A.; LOPEZ-HERREJON, R. E.; EGYED, A. Multi-objective test case prioritization in highly configurable systems: A case study. **Journal of Systems and Software**, Elsevier, v. 122, p. 287–310, 2016.
- PIZZI, N. J. A fuzzy classifier approach to estimating software quality. **Information Sciences**, Elsevier, v. 241, p. 1–11, 2013.
- POHL, K.; BÖCKLE, G.; LINDEN, F. J. van D. **Software product line engineering: foundations, principles and techniques**. [S.l.]: Springer Science & Business Media, 2005.
- ROSENBERG, L. H. Applying and interpreting object oriented metrics. NASA Software Assurance Technology Center (SACT), 1998.
- RUONING, X.; XIAOYAN, Z. Extensions of the analytic hierarchy process in fuzzy environment. **Fuzzy sets and Systems**, Elsevier, v. 52, n. 3, p. 251–257, 1992.
- SILVA, F. A. P. da; NETO, P. A. d. M. S.; GARCIA, V. C.; MUNIZ, P. F. Linhas de produtos de software: Uma tendência da indústria. 2011.
- SOCHOS I. PHILIPPOW, M. R. P. **Feature-oriented development of software product lines: Mapping feature models to the architecture**. Springer, 2004.
- SOLINGEN, R. V.; BASILI, V.; CALDIERA, G.; ROMBACH, H. D. Goal question metric (gqm) approach. **Encyclopedia of software engineering**, Wiley Online Library, 2002.
- VASILESCU, B.; SEREBRENIK, A.; BRAND, M. van den. Comparative study of software metrics' aggregation techniques. **Proceedings of the International Worskhop Benevol**, v. 2010, 2010.
- WEISS, D. M.; LAI, C. T. R. **Software product-line engineering: a family-based software development process**. [S.l.]: Addison-Wesley Reading, 1999. v. 12.
- YADAV, H. B.; YADAV, D. K. A fuzzy logic based approach for phase-wise software defects prediction using software metrics. **Information and Software Technology**, Elsevier, v. 63, p. 44–57, 2015.
- ZADEH, L. A. Fuzzy logic. **Computer**, IEEE, v. 21, n. 4, p. 83–93, 1988.
- ZIMMERMANN, H. Operators in models of decision making. **Fuzzy information engineering**, New York: Wiley, p. 471–496, 1997.

APÊNDICE A – QUESTIONÁRIO DE PERFIL DO ESPECIALISTA

Neste apêndice, é descrito o formulário do perfil dos especialistas que participaram da revisão por pares para definição dos parâmetros e regras *fuzzy* na Seção 6.

Perfil do Especialista:

Nome:

E-mail:

Perguntas

1. **Qual seu maior nível de formação acadêmica?**
 - a) Doutorado
 - b) Estudante de Mestrado
 - c) Especialização
 - d) Graduação

2. **Qual posição você ocupa atualmente?**
 - a) Professor
 - b) Pesquisador
 - c) Estudante de Doutorado
 - d) Estudante de Mestrado

3. **Quanto tempo de experiência possui em Engenharia de Domínio de Linhas de Produtos de Software?**
 - a) Acima de 5 anos
 - b) Mais do que 2 e menos de 5 anos
 - c) Menos de 2 anos
 - d) Nenhum

4. **Quanto tempo de experiência possui em Avaliação da Qualidade?**
 - a) Acima de 5 anos
 - b) Mais do que 2 e menos de 5 anos
 - c) Menos de 2 anos
 - d) Nenhum

5. Como você classifica seu nível de conhecimento em Linha de Produtos de Software?

Nenhum: 0

Excelente: 5

a) 0

b) 1

c) 2

d) 3

e) 4

f) 5

6. Como você classifica seu nível de conhecimento em Modelos de *Features* ?

Nenhum: 0

Excelente: 5

a) 0

b) 1

c) 2

d) 3

e) 4

f) 5

7. Quantos projetos (pesquisa e/ou indústria) envolvendo Linhas de Produtos de Software você participou ?

a) Mais do que 5 projetos

b) Entre 3 a 5 projetos

c) Menos que 2 projetos

d) Nenhum

**APÊNDICE B – PARÂMETROS DAS FUNÇÕES DE PERTINÊNCIA DO SISTEMA
FUZZY**

Neste apêndice são apresentadas as definições dos parâmetros das funções de pertinência projetadas com a ajuda de especialista do domínio, resultado obtido do processo de projeto e definição descrito no Seção 6 deste trabalho. As Tabela 9, 10 e 11 apresentam os parâmetros para as funções de pertinência para cada uma das 15 medidas selecionadas e agrupadas em Tamanho, Complexidade e Variabilidade.

Tabela 9 – Parâmetros para funções de pertinência para medidas de Tamanho.

| Medidas de Tamanho Parâmetros | |
|--|--------------------------|
| Medidas | Parâmetros |
| DTMax | Baixo ≤ 3 |
| | $3 < \text{Médio} < 6$ |
| | Alto ≥ 6 |
| NLeaf | Baixo ≤ 15 |
| | $15 < \text{Médio} < 30$ |
| | Alto ≥ 30 |
| NF | Baixo ≤ 15 |
| | $15 < \text{Médio} < 40$ |
| | Alto ≥ 40 |

Fonte: elaborado pelo Autor.

Tabela 10 – Parâmetros para funções de pertinência para medidas de Complexidade.

| Medidas de Complexidade | |
|--------------------------------|----------------------------|
| Parâmetros | |
| Medidas | Parâmetros |
| RDen | Baixo $\leq 0,5$ |
| | $0,5 < \text{Médio} < 1$ |
| | Alto ≥ 1 |
| CogC | Baixo ≤ 3 |
| | $3 < \text{Médio} < 8$ |
| | Alto ≥ 8 |
| NTop | Baixo ≤ 3 |
| | $3 < \text{Médio} < 6$ |
| | $6 < \text{Alto} \leq 7$ |
| FoC | Baixo $\leq 0,2$ |
| | $0,2 < \text{Médio} < 0,4$ |
| | Alto $\geq 0,4$ |

Fonte: elaborado pelo Autor.

Tabela 11 – Parâmetros para funções de pertinência para medidas de Variabilidade.

| Medidas de Variabilidade | | | |
|---------------------------------|--------------------------|----------------|----------------------------|
| Parâmetros | | | |
| Medidas | Parâmetros | Medidas | Parâmetros |
| NM | Baixo ≤ 5 | RoV | Baixo ≤ 2 |
| | $5 < \text{Médio} < 15$ | | $2 < \text{Médio} < 4$ |
| | Alto ≥ 15 | | Alto ≥ 4 |
| FEX | Baixo ≤ 20 | NVC | Baixo ≤ 350 |
| | $20 < \text{Médio} < 35$ | | $300 < \text{Médio} < 700$ |
| | Alto ≥ 35 | | Alto ≥ 700 |
| SCDF | Baixo ≤ 1 | NGXOr | Baixo ≤ 2 |
| | $1 < \text{Médio} < 3$ | | $2 < \text{Médio} < 5$ |
| | Alto ≥ 3 | | Alto ≥ 5 |
| MCDF | Baixo ≤ 1 | NGOr | Baixo ≤ 2 |
| | $1 < \text{Médio} < 4$ | | $2 < \text{Médio} < 5$ |
| | Alto ≥ 4 | | Alto ≥ 5 |

Fonte: elaborado pelo Autor.

APÊNDICE C – REGRAS *FUZZY* PARA OS GRUPOS DE MEDIDAS

Este apêndice apresenta todas as regras *fuzzy* projetadas e definidas com a ajuda de especialista do domínio obtidas através do procedimento descrito no Seção 6 deste trabalho. As regras *fuzzy* são definidas na forma de uma declaração condicional SE-ENTÃO. A parte SE da regra são as medidas de entrada e o termo linguístico (baixo, médio ou alto), e a parte ENTÃO é um dos cinco estados linguísticos possíveis (muito baixo, baixo, médio, alto ou muito alto) para o aspecto relacionado ao grupo de medidas: Tamanho, Complexidade, Variabilidade.

São apresentados nas Tabela 12, 13 e 14 a listagem dos termos linguísticos de cada medida e o estado linguístico resultante de cada grupo relacionado.

Tabela 12 – Regras *fuzzy* para medidas de Tamanho.

| Regra | Medidas de entrada | | | Grupo |
|-------|--------------------|-------|-------|------------|
| | DTMax | Nleaf | NF | Tamanho |
| 1 | Baixo | Baixo | Baixo | Baixo |
| 2 | Médio | Baixo | Baixo | Baixo |
| 3 | Baixo | Médio | Baixo | Baixo |
| 4 | Alto | Baixo | Baixo | Médio |
| 5 | Baixo | Médio | Médio | Médio |
| 6 | Médio | Baixo | Médio | Alto |
| 7 | Médio | Médio | Médio | Médio |
| 8 | Alto | Médio | Médio | Alto |
| 9 | Médio | Alto | Médio | Alto |
| 10 | Médio | Alto | Alto | Muito Alto |
| 11 | Alto | Alto | Alto | Muito Alto |

Fonte: elaborado pelo Autor.

Tabela 13 – Regras *fuzzy* para medidas de Complexidade.

| Regras | Medidas de entrada | | | | Grupo |
|--------|--------------------|-------|-------|-------|--------------|
| | RDen | CogC | NTop | FoC | Complexidade |
| 1 | Baixo | Baixo | Baixo | Alto | Muito Baixo |
| 2 | Baixo | Baixo | Médio | Baixo | Médio |
| 3 | Baixo | Baixo | Alto | Baixo | Médio |
| 4 | Baixo | Baixo | Baixo | Alto | Baixo |
| 5 | Baixo | Médio | Alto | Médio | Médio |
| 6 | Baixo | Médio | Baixo | Médio | Médio |
| 7 | Baixo | Médio | Médio | Médio | Médio |
| 8 | Baixo | Médio | Alto | Médio | Médio |
| 9 | Baixo | Médio | Alto | Baixo | Alto |
| 10 | Baixo | Alto | Baixo | Alto | Baixo |
| 11 | Baixo | Alto | Baixo | Médio | Médio |
| 12 | Baixo | Alto | Médio | Médio | Médio |
| 13 | Baixo | Baixo | Alto | Baixo | Baixo |
| 14 | Baixo | Alto | Alto | Médio | Alto |
| 15 | Baixo | Baixo | Alto | Baixo | Alto |
| 16 | Médio | Baixo | Baixo | Médio | Baixo |
| 17 | Médio | Baixo | Baixo | Médio | Médio |
| 18 | Médio | Baixo | Baixo | Médio | Médio |
| 19 | Médio | Baixo | Baixo | Alto | Baixo |
| 20 | Médio | Médio | Baixo | Baixo | Baixo |
| 21 | Médio | Médio | Médio | Médio | Médio |
| 22 | Médio | Médio | Médio | Baixo | Alto |
| 23 | Médio | Médio | Alto | Médio | Médio |
| 24 | Médio | Alto | Baixo | Médio | Médio |
| 25 | Médio | Alto | Médio | Médio | Alto |
| 26 | Médio | Baixo | Alto | Baixo | Alto |
| 27 | Alto | Baixo | Baixo | Médio | Baixo |
| 28 | Alto | Baixo | Médio | Baixo | Médio |
| 29 | Alto | Médio | Baixo | Baixo | Médio |
| 30 | Alto | Médio | Médio | Médio | Médio |
| 31 | Alto | Médio | Alto | Médio | Alto |
| 32 | Alto | Alto | Baixo | Médio | Alto |
| 33 | Alto | Baixo | Médio | Baixo | Muito Alto |
| 34 | Alto | Baixo | Alto | Baixo | Muito Alto |

Fonte: elaborado pelo Autor.

Tabela 14 – Regras *fuzzy* para medidas de Variabilidade.

| Regras | Medidas de entrada | | | | | | | | Grupo |
|--------|--------------------|-------|-------|-------|-------|-------|-------|-------|---------------|
| | NM | FEX | SCDF | MCDF | RoV | NVC | NGXOr | NGOr | Variabilidade |
| 1 | Baixo | Alto | Baixo | Baixo | Alto | Alto | Baixo | Alto | Muito Alto |
| 2 | Baixo | Alto | Baixo | Baixo | Médio | Médio | Baixo | Alto | Médio |
| 3 | Baixo | Alto | Baixo | Baixo | Baixo | Baixo | Baixo | Baixo | Baixo |
| 4 | Baixo | Alto | Baixo | Baixo | Alto | Médio | Médio | Alto | Alto |
| 5 | Baixo | Alto | Baixo | Baixo | Médio | Alto | Baixo | Médio | Alto |
| 6 | Baixo | Alto | Baixo | Baixo | Baixo | Baixo | Alto | Médio | Médio |
| 7 | Baixo | Alto | Baixo | Baixo | Alto | Alto | Alto | Alto | Muito Alto |
| 8 | Baixo | Alto | Baixo | Médio | Médio | Médio | Baixo | Médio | Médio |
| 9 | Baixo | Alto | Baixo | Médio | Alto | Alto | Baixo | Médio | Alto |
| 10 | Baixo | Alto | Baixo | Médio | Baixo | Alto | Baixo | Médio | Médio |
| 11 | Baixo | Alto | Baixo | Médio | Alto | Alto | Alto | Alto | Alto |
| 12 | Baixo | Alto | Baixo | Alto | Baixo | Baixo | Baixo | Baixo | Baixo |
| 13 | Baixo | Alto | Baixo | Alto | Baixo | Baixo | Médio | Baixo | Baixo |
| 14 | Baixo | Alto | Baixo | Alto | Baixo | Baixo | Alto | Baixo | Baixo |
| 15 | Baixo | Alto | Baixo | Alto | Médio | Médio | Alto | Médio | Médio |
| 16 | Baixo | Alto | Médio | Baixo | Alto | Alto | Alto | Alto | Alto |
| 17 | Baixo | Alto | Médio | Baixo | Baixo | Baixo | Médio | Médio | Baixo |
| 18 | Baixo | Alto | Médio | Baixo | Alto | Alto | Baixo | Alto | Muito Alto |
| 19 | Baixo | Alto | Médio | Médio | Médio | Médio | Médio | Médio | Médio |
| 20 | Médio | Alto | Médio | Médio | Baixo | Baixo | Baixo | Baixo | Baixo |
| 21 | Médio | Alto | Médio | Médio | Médio | Médio | Baixo | Médio | Médio |
| 22 | Médio | Alto | Médio | Médio | Alto | Alto | Baixo | Médio | Médio |
| 23 | Médio | Alto | Médio | Alto | Alto | Alto | Baixo | Médio | Muito Alto |
| 24 | Médio | Alto | Alto | Alto | Baixo | Baixo | Baixo | Baixo | Médio |
| 25 | Médio | Alto | Alto | Alto | Médio | Médio | Baixo | Médio | Médio |
| 26 | Médio | Médio | Alto | Alto | Alto | Alto | Baixo | Médio | Alto |
| 27 | Médio | Médio | Alto | Alto | Baixo | Baixo | Alto | Baixo | Médio |
| 28 | Médio | Baixo | Alto | Alto | Alto | Alto | Baixo | Médio | Muito Alto |
| 29 | Médio | Baixo | Alto | Alto | Médio | Médio | Baixo | Médio | Médio |
| 30 | Médio | Baixo | Alto | Alto | Baixo | Baixo | Médio | Baixo | Muito Baixo |
| 31 | Médio | Baixo | Médio | Médio | Baixo | Baixo | Médio | Baixo | Baixo |
| 32 | Médio | Baixo | Médio | Médio | Alto | Alto | Baixo | Médio | Médio |
| 33 | Alto | Baixo | Baixo | Baixo | Médio | Médio | Médio | Médio | Baixo |
| 34 | Alto | Baixo | Baixo | Baixo | Alto | Alto | Baixo | Alto | Muito Alto |
| 35 | Alto | Médio | Baixo | Baixo | Baixo | Baixo | Alto | Baixo | Muito Baixo |
| 36 | Alto | Médio | Médio | Médio | Médio | Médio | Médio | Médio | Médio |
| 37 | Alto | Médio | Médio | Médio | Médio | Médio | Alto | Alto | Médio |
| 38 | Alto | Médio | Alto | Alto | Baixo | Baixo | Alto | Baixo | Muito Alto |
| 39 | Alto | Médio | Alto | Alto | Médio | Médio | Alto | Baixo | Muito Alto |