Avaliando a Especificação das Ocorrências de Características Transversais em Software Embarcado

Rodrigo Souza Cavalcanti de Oliveira, Marco Aurelio Wehrmeister[a]
Universidade Tecnológica Federal do Paraná (UTFPR)
Av. Sete de Setembro, 3165 - CEP 80230-901 - Curitiba - PR - Brazil
rodrigoo@alunos.utfpr.edu.br, wehrmeister@utfpr.edu.br
[a]http://orcid.org/0000-0002-1415-5527

***Resumo*—A engenharia de software embarcado vem evoluindo rapidamente ao longo dos últimos anos, em especial para lidar com o aumento da complexidade associada a grande quantidade de requisitos funcionais e características transversais associadas aos requisitos não-funcionais. O processo de identificar e especificar a seleção de pontos onde as características transversais devem ser tratadas vem sendo amplamente discutido desde meados dos anos 90, especialmente no contexto da *Aspect-Oriented Software Development* (AOSD) e modelagem. No entanto, a compreensão de tais especificações é muitas vezes difícil, pois a sintaxe e a semântica das linguagens utilizadas, principalmente as linguagens visuais, não permitem intuir facilmente o significado dos símbolos usados. Tal situação produz artefatos difíceis de compreender e manter. Portanto, o processo de localizar, identificar, e especificar os pontos onde as características transversais devem ser tratadas é chave no projeto de software embarcado, pois pode levar a problemas no desenvolvimento e, principalmente, na integração dos componentes do software do sistema. Este trabalho apresenta uma avaliação empírica com o objetivo de avaliar a compreensão da especificação de seleção de POCT em sistemas embarcados de forma indireta através da avaliação dos efeitos cognitivos e das propriedades de percepção das notações. Foram avaliadas três notações, duas gráficas (JPDD e Theme/UML) e uma textual (AspectJ), que foram usadas para especificar implícita e explicitamente a seleção de POCT (para 15 pontos distintos) em três projetos de software embarcados diferentes que representam aplicações reais no contexto de sistemas de automação. Para tal, propõe-se um modelo de qualidade baseado no framework conceitual "Physics of Notation" (PON). Foram usadas nove métricas para quantificar as propriedades perceptuais na especificação da seleção de POCT, sendo que algumas métricas foram criadas e outras representam conceitos dentro dos princípios do PON. Os resultados dos experimentos mostram como cada uma das notações impacta na compreensão das especificações criadas. Foram encontradas evidências empíricas de que a especificação gráfica das seleções dos POCT tem pouca discriminação sobre a eficácia cognitiva das representações visuais (conforme o framework PON) e, portanto, os indícios apontam para a conclusão de que o uso de notações visuais para especificar a seleção de POCT é um processo inconsciente e sujeito a ambiguidades tanto na especificação dos artefatos quanto na compreensão e interpretação deles.**

***Index Terms*—software embarcado, características transversais, notação visual, propriedades cognitivas, métricas de software**

Este trabalho recebeu o apoio financeiro da Fundação Araucária do Paraná através dos convênios 337/2014 e 34/2019.

I. Introdução

O desenvolvimento de software embarcado geralmente inclui características funcionais e não-funcionais que estão vinculadas a um único serviço ou função. Nos paradigmas tradicionais de decomposição de software, e.g., paradigma de orientação a objetos e/ou procedural, tais responsabilidades são disseminadas por várias unidades de modularização, tanto nas especificações usando modelos visuais quanto na codificação do software embarcado. As características transversais afetam os requisitos funcionais e os não-funcionais levando a um entrelaçamento e espalhamento do código por vários módulos do software, afetando negativamente o princípio de responsabilidade única e também a construção de um código desacoplado e coeso. A literatura apresenta muitas abordagens com o objetivo de resolver esses problemas de modularização e também da quebra de proposições chave da engenharia de software, e.g., a premissa de responsabilidade única. Uma das técnica mais populares é o Desenvolvimento de Software Orientado a Aspectos (em inglês, *Aspect-oriented Software Development* - AOSD) [1]. A AOSD propõe tratar uma característica transversal ao requisito através de unidades de modularização denominadas aspectos. Os aspectos encapsulam toda a implementação responsável por interceptar o fluxo de execução do software e injetar o tratamento da característica transversal nele. Assim, os problemas associados ao entrelaçamento e espalhamento da característica transversal pelo software embarcado são atenuados.

A especificação da seleção de *Pontos de Ocorrência das Características Transversais* (POCT) indica onde, no código fonte, podem ou devem haver características transversais que são inerentes aos requisitos funcionais ou não-funcionais do sistema embarcado. Existem várias linguagens de programação e também notações visuais orientadas a aspectos que apresentam diversos meios linguísticos para identificação e especificação dessas ocorrências. As formas mais comuns de designar a seleção de POCT são baseadas em propriedades léxicas aplicadas em um contexto comportamental ou estrutural das características transversais [2].

Embora existam um número significativo de linguagens, seja

em forma gráfica ou textual, a compreensão dos elementos relacionados à AOSD geralmente requer um conhecimento profundo não só dos conceitos mas também da sintaxe e da semântica da linguagem. No entanto, existem muitas extensões de linguagens consolidadas, e.g., Java e UML, que empregam uma semântica diferente uma da outra para indicar POCT. Deste modo, é difícil entender a especificação de seleções de POCT escritas em uma linguagem adaptada ou em notações visuais que são geralmente adotadas nas abordagens de modelagem da AOSD [3]–[5]. Tal situação compromete o entendimento de "como" e "onde" as características transversais ocorrem e são tratadas. Destaca-se que projeto de sistemas embarcados de tempo real geralmente apresenta vários tipos distintos de características transversais inerentes não somente aos requisitos funcionais e não-funcionais como também as suas restrições [6], e.g., restrições de tempo, concorrência, monitoramento de recursos, segurança, confiabilidade e outros.

O estado da arte atual apresenta dois tipos de abordagens para especificar o tratamento das características transversais: as notações gráficas e as textuais. A especificação da seleção de POCT através da notação gráfica é uma área de pesquisa que tem sido bastante explorada nos últimos anos [7]–[9]. Por exemplo, o *Join Point Designation Diagram* (JPDD) [2] é uma técnica interessante que foi criada para identificar e especificar a seleção de POCT através da forma gráfica. O JPDD propõe uma extensão da UML [10] cujo objetivo é especificar as seleções de POCT de uma forma independente do sistema ou da linguagem orientada a aspectos. Outra abordagem a ser destacada é a *Theme/UML* [11], que também é baseada na UML mas que atende a separação simétrica das características transversais; a separação assimétrica é suportada pela maioria das abordagens AOSD. Os desenvolvedores especificam ligações (*bindings*) entre os temas para indicar onde a seleção de POCT deve ser inserida no fluxo de execução do software.

Contudo, mesmo sendo a UML amplamente aceita na indústria e na academia, ainda não existe uma semântica precisa quando se trata da especificação de seleções de POCT [12], [13], até mesmo porque a UML foi criada para atender ou dar suporte ao paradigma orientado a objetos. Consequentemente, embora existam abordagens que estendem a semântica da UML para suportar a AOSD [2], [11], [12], o resultado é uma especificação imprecisa e ambígua das seleções de POCT, devido a pouca discriminação sobre a eficácia cognitiva das representações visuais [14]. Isso aumenta de forma indireta a dificuldade no desenvolvimento, compreensão e manutenção dos artefatos gerados para sistemas embarcados.

Sendo assim, este artigo apresenta uma avaliação empírica que compara três notações - duas gráficas, JPDD e Theme/UML, e uma textual, AspectJ - para especificar implícita ou explicitamente a seleção de POCT no software embarcado. O objetivo é avaliar a compreensão da especificação das seleções de POCT em sistemas embarcados, seja ela gráfica ou textual, de forma indireta através da avaliação dos efeitos cognitivos e das propriedades de percepção das notações. Essa é a contribuição principal do trabalho.

Para isso, este trabalho propõe um modelo de qualidade baseado nas premissas conceituais do framewok *Physics of Notations* (PON) [14]. Tal modelo inclui nove métricas com o objetivo de quantificar propriedades físicas perceptivas (i.e. sintaxe) das notações usadas para especificar a seleção de POCT em software embarcado. Tal modelo de qualidade e suas métricas também são contribuições relevantes desse trabalho.

Para a avaliação, foram utilizados quinze POCT comumente encontradas no desenvolvimento de software de sistemas embarcados [6] e que foram especificados em três estudos de caso representando sistemas reais [12], [13]. Cada POCT foi especificado usando as três notações mencionadas. Os resultados iniciais mostram evidências empíricas de que a especificação gráfica das seleções de POCT tem pouca discriminação sobre a eficácia cognitiva das representações visuais e, portanto, os indícios apontam para a conclusão de que o uso de notações visuais para especificar a seleção de POCT é um processo inconsciente e sujeito a ambiguidades, tanto na especificação dos artefatos, quanto na sua compreensão e interpretação.

Esse artigo está organizado da seguinte forma. A Seção 2 apresenta uma visão geral sobre o framework PON. A Seção 3 discute os trabalhos relacionados e analisa os problemas em aberto. A Seção 4 apresenta o modelo de avaliação proposto para a especificação da seleção de POCT. A Seção 5 apresenta a avaliação empírica e analisa seus resultados, enquanto a Seção 6 discute as possíveis ameaças à validade dos resultados e da avaliação empírica. Finalmente, a Seção 7 apresenta as conclusões e indica direções para os trabalhos futuros.

II. A Física das Notações

O framework conceitual *Physics of Notations* (PON) [14] fornece uma base lógico-científica para o design das notações visuais na engenharia de software. Moody [14] afirma que as notações visuais são parte integrante da engenharia de software, mas os pesquisadores e designers de notações têm historicamente ignorado ou subestimado questões importantes quanto a eficácia cognitiva da representação visual.

Ele afirma que, embora vários trabalhos na literatura avaliem e comparam as notações, os detalhes quanto a sintaxe das representações visuais são raramente discutidos. Para preencher essa lacuna, o PON aborda as propriedades físicas e perceptivas das notações (sintaxe), em vez de suas propriedades lógicas (semânticas). As propriedades perceptivas são analisadas à luz da Teoria do Design [15] e envolve nove princípios. Este trabalho usou os 5 princípios mais relevantes para a especificação das seleções de POCT em sistemas embarcados. São eles:

- ***Clareza Semiótica*:** deve haver uma correspondência 1:1 entre construções semânticas e símbolos gráficos;
- ***Discriminabilidade Perceptiva*:** símbolos diferentes devem ser claramente distinguíveis um do outro;
- ***Transparência Semântica*:** a aparência das representações visuais deve sugerir seu significado;
- ***Expressividade Visual*:** o uso de todas as possibilidades e capacidades de variáveis visuais;
- ***Economia gráfica*:** o número de símbolos gráficos diferentes devem ser cognitivamente gerenciável.

Os demais princípios do PON não foram considerados relevantes para o objetivo e escopo deste trabalho. O princípio de *Gerenciamento de Complexidade* é apenas aplicado quando o sistema não faz uso de modularidade devido a integração heterogênea ou outros motivos, o que não é o caso do contexto de seleção de POCT para sistemas embarcados. O princípio de *Integração Cognitiva* não é relevante para a análise apresentada neste trabalho pois é aplicado apenas quando múltiplos diagramas são usados para representar um sistema, o que não é o caso da seleção de POCT. Por fim, o princípios do *Ajuste Cognitivo* e da *Codificação Dupla* não foram abordados pois considerou-se que nenhuma das notações avaliadas contempla as características associadas a estes princípios. O primeiro sugere usar dois dialetos diferentes nas notações visuais, um para usuários experientes e outro para iniciantes. O segundo sugere usar textos como algo redundante dentro da notação para reforçar o significado dos elementos visuais.

III. Trabalhos Relacionados

Vários trabalhos propuseram abordagens distintas para especificar a seleção de POCT. Por exemplo, AspectJ [16] é uma linguagem de programação textual que faz extensão da linguagem Java para definir um mecanismo dinâmico de intercepção transversal na execução de um programa. Esse mecanismo é suportado por construções denominadas pontos de junção que representam os POCT. AspectJ é a base de muitos trabalhos que visam lidar com características transversais e especificar seleções de POCT. Todavia, entender a sua semântica não é trivial, pois é necessário um esforço adicional para aprender a sintaxe, os símbolos e as regras para o uso correto da AspectJ.

A notação JPDD [2] é uma abordagem de modelagem para visualizar seleções de POCT com base na semântica da UML. Os engenheiros criam diagramas dos eventos do sistema, símbolos de objetos e arestas de fluxo de objetos através da adoção de símbolos de ação que representam um fluxo de execução para atingir um POCT. Embora a JPDD permita especificar seleções de POCT de maneira uniforme e independente da linguagem de programação, o uso dos símbolos da UML no JPDD não é de fácil compreensão pois difere bastante do uso tradicional dos mesmos elementos. Além disso, os engenheiros precisam ter uma grande experiência em AOSD para projetar os JPDDs adequadamente no contexto de requisitos funcionais e não funcionais.

A técnica Theme/UML [11], [17] é uma abordagem baseada em UML que suporta a separação simétrica de características transversais em vez de separação assimétrica. A última é suportada na maioria das abordagens de AOSD. Fragmentos de comportamento e/ou estruturas que representam uma característica transversal são encapsulados em um único tema. A integração entre temas é definida como uma ligação que indica quais elementos de um tema são afetados por elementos de outros temas. De alguma forma, isso é semelhante à especificação dos pontos de junção e a relação entre elementos funcionais afetados por aspectos [16]. No entanto, a especificação de como um tema afeta outros temas não é adequada devido à falta de escalabilidade, ou seja, em um sistema com uma grande quantidade de características transversais, a especificação do relacionamento de vínculo para expressar a integração desses temas dificulta na compreensão, manutenção e na evolução do modelo.

Além disso, o trabalho publicado em [18] examina a literatura sobre características transversais para reunir uma lista de métricas de qualidade existentes para avaliar a eficácia de diferentes técnicas. Aqueles autores concluíram que existe uma lacuna crítica de busca pelas melhores práticas quanto a especificação de seleções de POCT, dificultando a avaliação dos métodos e técnicas por meio da avaliação empírica. A falta de parâmetros de referência adequados afeta a comparação objetiva e a validação empírica das pesquisas no tema.

Em [19], um experimento controlado compara a notação visual (JPDD) e uma notação textual (Tracematch [20]) para especificar seleções de POCT em relação a capacidade das técnicas de facilitar a compreensão de expressões lógicas. Esse trabalho envolveu 35 alunos que realizaram dois tipos de tarefas de compreensão com base em 28 seleções de POCT criadas artificialmente. Cada participante realizou as mesmas tarefas. Segundo aqueles autores, os resultados do experimento indicaram que a JPDD é melhor em relação ao Tracematch na maioria dos casos. O presente trabalho tem semelhanças com [19] pois compara notações gráficas e textuais com o objetivo de quantificar a compreensão da especificação de seleções de POCT. No entanto, nosso trabalho é complementar e se concentra em avaliar as propriedades perceptivas das notações, e não a semântica das especificações de seleção de POCT.

Uma questão importante no entendimento de uma especificação de seleção de POCT é a falta de conhecimento sobre o paradigmas voltados ao encapsulamento das características transversais, e.g., AOSD, a partir dos quais o conceito de ocorrências de características transversais é derivado. Na AspectJ, as ocorrências de características transversais são pontos bem definidos na execução do programa, por exemplo, variáveis, chamadas de métodos, construtores e outros elementos comportamentais. Contudo, essa definição não é abrangente o suficiente para o nível de modelagem e existe uma complexidade associada quando se pretende modelar uma seleção de POCT. Por exemplo, nas linguagens de modelagem, é necessário unificar conceitos suportados em várias linguagens de programação em elementos de modelagem mais abstratos. Portanto, as linguagens de modelagem são mais ricas em termos de conceitos (i.e., elementos de modelagem) que podem servir como pontos de ocorrências de características transversais. Porém, a relação entre os níveis de abstração usados na modelagem e na execução do software é menos clara do que no caso das linguagens de programação.

Uma boa compreensão é uma meta qualitativa importante quando se pretende abstrair um conceito computacional em uma notação, seja ela gráfica ou textual. Portanto, considerando a grande ocorrência de características transversais no software embarcado, é essencial haverem mecanismos para avaliar a especificação de seleções de POCT. Tais mecanismos devem fornecer informações quantitativas para permitir uma avaliação mais precisa dos artefatos. Até onde sabemos, este

trabalho é o primeiro a avaliar notações distintas (i.e., notações gráficas, e.g., JPDD e Theme/UML, e também notações textuais, e.g., AspectJ) para seleção de POCT seguindo as premissas do framework PON. Além disso, nenhum outro estudo empírico apresentado no estado da arte da engenharia de software analisa as propriedades perceptivas de notações relacionadas a AOSD através de métricas quantitativas como as propostas neste trabalho.

IV. Métricas e Modelo de Qualidade para Avaliar as Propriedades Cognitivas

Atualmente, pode-se encontrar na literatura de engenharia de software várias métricas e modelos de qualidade que podem ser usados para extrair informações quantitativas de artefatos de software. No entanto, até onde sabemos, não existe um modelo de avaliação ou de qualidade criado a partir da perspectiva das premissas do framework PON que possibilita quantificar as propriedades perceptivas das notações gráficas e/ou textuais, possibilitando avaliar os seus efeitos cognitivos.

Assim, além de contribuir com uma avaliação da compreensão da especificação de seleção de POCT em sistemas embarcados, este trabalho propõem um novo modelo de qualidade baseado no framework PON. Um conjunto de métricas denominado *Metrics Suite for PON conceptual framework* (MS4PON) foi criado. O objetivo deste conjunto de métricas é quantificar as propriedades perceptivas de notações gráficas e textuais na especificação de seleções de POCT. As métricas do MS4PON foram combinadas em um modelo de qualidade apresentado na Figura 1. Esse modelo de qualidade faz uso de 5 princípios da teoria associada ao framework PON. O MS4PON quantifica os elementos, símbolos e a sintaxe usados na aplicação de uma notação gráfica ou textual para a especificação da seleção de POCT. As métricas são:

- **Number of elements (NE):** essa métrica contabiliza o número de elementos contidos no modelo de seleção de POCT, independentemente da sua especificação ser em formato textual ou gráfico. Quanto maior o NE, mais complexo será a especifição
- **Number of Logical Expressions (NLE):** essa métrica é definida como a quantidade de expressões lógicas agnósticas[1] contidas na especificação da seleção de POCT. As expressões lógicas indicam os critérios de seleção implícitos ou explícitos para a seleção dos POCT. Nos nossos experimentos, os valores NLE são iguais para cada um dos POCT, independente da notação que foi aplicada. O objetivo é mensurar a expressão lógica que representa o critério de seleção que deve ser especificado usando uma notação. O NLE permite gerar valores relativos viabilizando analises comparativas entre notações e fornecendo resultados mais precisos e homogêneos.
- **Number of Visual Distance (NVD):** essa métrica quantifica o número de elementos diferentes que são especificados na seleção de POCT. Em geral, quanto maior o número de elementos diferentes disponíveis na notação, maior é o NVD e, consequentemente, a especificação de seleção de POCT é compreendida com maior precisão.

[1]i.e., expressões lógicas abstratas e independentes da notação ou modelo de seleção de POCT

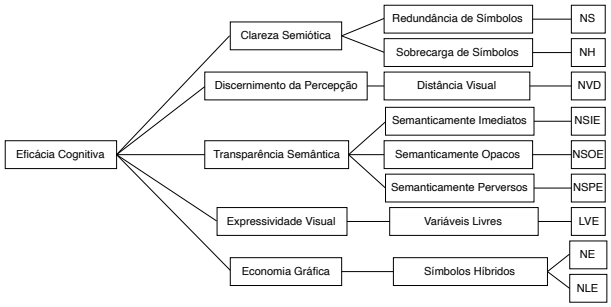

Figura 1. Modelo de qualidade para avaliar as propriedades cognitivas

- **Number of Semantically Immediate Elements (NSIE):** essa métrica determina o número total de elementos semanticamente imediatos especificados na seleção de POCT. Um elemento é semanticamente imediato se for fácil inferir seu significado a partir de sua aparência ou mnemônico. Quanto maior o NSIE, mais fácil será de entender a especificação do POCT em uma determinada notação. Um elemento ou símbolo semanticamente imediato deve ter em sua natureza uma associação com: (i) *similaridades funcionais*, por exemplo, o método elemento "before()" no AspectJ usa um termo que expressa o conceito de fazer algo antes (a palavra "before") trazendo uma similaridade funcional; ou (ii) *metáforas*, por exemplo, um desenho de um boneco pode representar uma pessoa nos diagramas de casos de uso da UML; ou (iii) quaisquer *associações culturais*, por exemplo, um grande "X" em um diagrama de sequência da UML pode significar uma comunicação fechada ou interrompida.
- **Number of Semantically Opaque Elements (NSOE):** essa métrica contabiliza o número de elementos semanticamente opacos especificados na seleção de POCT. Um elemento é semanticamente opaco (ou convencionado) se houver uma convenção puramente arbitrária entre a sua aparência e o seu significado. Quanto maior for o NSOE, mais elementos neutros ou convencionados são usados na especificação. Por exemplo o uso excessivo de retângulos nos diagramas da notação gráfica Theme/UML corresponde a um uso excessivo de elementos convencionados que, por sua vez, são puramente arbitrários quanto aos seus significados. NSOE representa o ponto zero na escala entre o NSIE e o NSPE, pois não aumenta a percepção cognitiva, mas também não sugere erroneamente um significado diferente de sua aparência, i.e., é um elemento convencionado que deve ser aprendido [14].
- **Number of Semantically Perverse Elements (NSPE):** essa métrica quantifica o número de elementos semanticamente perversos especificados na seleção de POCT. Um elemento pode ser semanticamente perverso (ou ter falso mnemônico), quando o seu significado é oposto à sua aparência. Quanto maior o NSPE, pior será o entendimento da especificação. Um elemento ou símbolo semanticamente perverso possui em sua natureza algumas

propriedades léxicas associadas a mnemônicos falsos que podem sugerir um significado totalmente diferente de sua aparência para um um leitor iniciante. Algumas convenções da UML apresentam essa propriedade indesejável, e.g., o símbolo "merge" de pacotes da UML [14].

- **Level of Expressiveness (LVE):** essa métrica determina o número de variáveis visuais usadas na notação para especificar a seleção de POCT. A LVE contabiliza a quantidade usada de variáveis de comunicação em notações visuais, e.g., forma, textura, cor, etc. Conforme [14], os valores de LVE variam de zero a oito. Portanto, quanto maior for o LVE, mais a notação aproveita a vantagem de ser uma notação gráfica. Quanto menor o LVE, mais próximo de uma notação textual é a notação visual.
- **Number of Synographs (NS):** essa métrica é definida como o número total de elementos sinográficos especificados na seleção de POCT. Elementos sinográficos aparecem quando há uma redundância no uso de símbolos ou elementos para representar a mesma construção semântica, e.g., interfaces na UML podem ser representadas de duas formas diferentes. Quanto maior o NS, pior é a compreensão da especificação. Uma notação gráfica satisfaz os requisitos de um sistema de notação quando se fornece uma correspondência individual entre símbolos e seus conceitos associados [14]. O elemento sinográfico é uma anomalia que ocorre quando não existe essa correspondência individual, levando à redundância de símbolos e diminuindo a compreensão da especificação ou notação.
- **Number of Homographs (NH):** essa métrica determina o número total de elementos homógrafos na especificação da seleção de POCT. Um homógrafo ocorre quando há vários elementos semânticos distintos que podem ser representadas pelos mesmos símbolos ou elementos. Quanto maior for o NH, pior é para entender a especificação. Esse é o pior tipo de anomalia em uma notação, seja ela gráfica ou textual, pois leva à ambiguidade e a potenciais erros de interpretação [14]. Por exemplo, Theme/UML evidencia essa anomalia pois a mesma convenção gráfica (e.g., retângulo) é usada para representar vários elementos semanticamente distintos: *subject*, *class*, *lifelines*, etc.

As métricas são combinadas em um modelo de qualidade para avaliar os cinco princípios definidos no framework PON. O princípio da clareza semiótica (*Semiotic Clarity*) é definida pela redundância (*Symbol Redundancy*) e sobrecarga (*Symbol Overload*) de símbolos. O primeiro é impactado pelo número de sinógraficos (NS) encontrados na especificação de seleções de POCT, enquanto o último pelo número de homógrafos (NH). O princípio do discernimento da percepção (*Perceptual Discriminability*) é definido pela distância visual que, por sua vez, é diretamente afetado pelo número de elementos diferentes usados (NVD). O princípio da transparência semiótica (*Semiotic Transparency*) é influenciado pelo número de elementos considerados semanticamente imediatos (NSIE), semanticamente opacos (NSOE) e semanticamente perversos (NSPE). Idealmente, uma notação deve apresentar o maior número possível de elementos semanticamente imediatos e zero elementos semanticamente perversos. O princípio da expressividade visual (*Visual Expressiveness*) é influenciado pela quantidade de variáveis livres (*Free Variables*) que é diretamente afetada pelo número de canais de comunicação visual (LVE). Uma notação sem variáveis de comunicação visuais é chamada de notação não visual (ou textual), enquanto uma notação que usa todas as variáveis de comunicação visuais é considerada visualmente saturada [14]. Por fim, o princípio da economia gráfica (*Graphic Economy*) é afetado pela quantidade de símbolos híbridos (*Hybrid Symbols*) que são medidos pelo NE e NLE. Uma notação aumenta sua complexidade proporcionalmente ao aumento da quantidade de elementos que são especificados por expressão lógica.

Para ilustrar como essas métricas são calculadas sobre as notações gráficas ou textuais, um exemplo simples de especificação da seleção de POCT é apresentado a seguir. Os POCT a serem selecionados são as ações de chamada para métodos cujos nomes começam com *set* (métodos *setter*) das classes cujos objetos são acessados simultaneamente por vários objetos ativos (i.e., threads/processos). Este POCT será nomeado como *JPExclusiveSetCalls*. Destaca-se que objetivo do MS4PON (e deste exemplo especificamente) não é a comparação das notações (gráficas ou textuais), mas sim a avaliação da compreensão da especificação da seleção de POCT, de forma indireta, através da avaliação dos efeitos cognitivos associados com as propriedades de percepção das notações medidas através das métricas propostas.

A Figura 2 apresenta o JPDD que especifica a seleção dos POCT mencionados. Este JPDD especifica a seleção de mensagens relacionadas aos métodos com nomes que começam com "set" e é enviado para objetos de classes que são anotadas com o estereótipo *<<MutualExclusionResource>>* [21]. A Figura 3 mostra exatamente a mesma especificação da seleção de POCT porém descrita usando a notação Theme/UML. Adicionalmente, a Listagem 1 apresenta essa mesma seleção de POCT através da linguagem AspectJ que cria uma analogia dos elementos usados por meio de uma representação textual com a mesma intenção conceitual que foi expressa através das notações gráficas JPDD e Theme/UML.

A Tabela I apresenta as métricas calculadas para essas três especificações do *JPExclusiveSetCalls*. Como mencionado, o NLE mede o critério de seleção independente da notação que deve ser especificado usando as três notações. Assim, para o *JPExclusiveSetCalls*, esse critério é "`(selecione o elemento que é igual a ação`

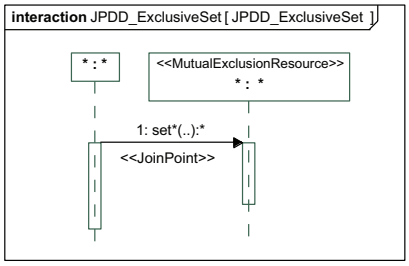

Figura 2. JPDD: seleção das mensagens cujo nome começa com "set"

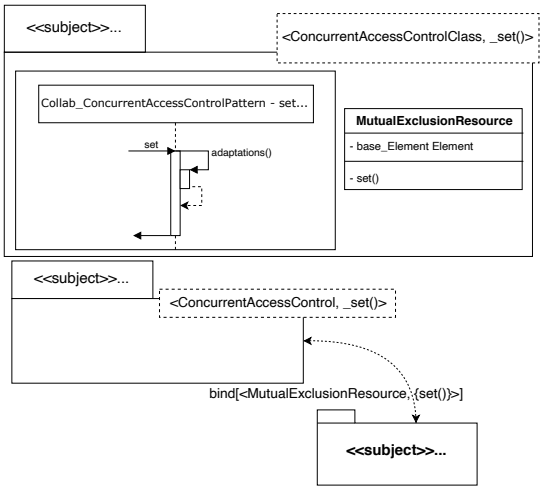

Figura 3. Theme/UML: seleção das mensagens cujo nome começa com "set"

de enviar mensagem) E (todos os métodos E nome que começam com 'set') E (todas as classes E qualquer nome E classe anotada com <<MutualExclusionResource>>)". Portanto, o NLE é o mesmo para as três especificações e, sendo assim, é possível fazer uma combinação do NLE com as outras métricas para gerar valores relativos, i.e., taxa calculadas com o valor de outras métricas dividido por NLE.

Na JPDD (Figura 2), o NE indica a quantidade dos elementos visíveis usados na especificação, por exemplo, a Figura 2 mostra duas *lifelines*, dois retângulos verticais definidos pela UML como símbolo para comportamento, uma linha de conexão que significa uma ação de envio de mensagem e dois estereótipos como elementos de texto (<<JoinPoint>>, <<MutualExclusionResouce>>). Os outros símbolos, por exemplo "$*$, :;", não são reconhecidos como elementos, pois os mesmos são atributos do próprio elemento, por exemplo, o "*" significa o nome da classe na *lifeline*; outro exemplo seriam os "1: set * (..): *" que indicam símbolos léxicos usados no modelo, i.e., não são elementos em si como o símbolo que representa o relacionamento de conexão com uma seta que simboliza o envio da mensagem. Na AspectJ (Listagem 1), o NE considera os elementos textuais usados para descrever o aspecto em si: o pointcut (linha 2); a chamada do método "set" (que é um POCT implícito); a classe *MutualExclusionResource*; o método *set(..)*; os símbolos *!within* e *before()*.

Por sua vez, a notação usada na Theme/UML (Figura 3) apresentou um NE surpreendentemente grande na especificação de seleção de POCT. Todos os retângulos que significam classes, atributos, métodos, *subjects*, *lifelines*, comportamentos e relacionamentos de conexões que enviam símbolos de mensagens foram contabilizados. Além disso, a aplicação da Theme/UML, apresentou um alto valor para o NH devido à sobrecarga de retângulos usados na especificação, causando uma anomalia relacionada ao grande número de elementos homógrafos, i.e., o uso da mesma convenção gráfica para expressar vários elementos distintos. Por outro lado, o JPDD tem o NH igual a zero, uma vez que os elementos utilizados na especificação têm uma correspondência de um

Listagem 1
ASPECTJ: SELEÇÃO DAS CHAMADAS DE MÉTODOS "SET"

```
1 aspect ConcurrentAccessControl{
2   pointcut pcBeforeWrite() :
3     call(* MutualExclusionResource.set(..))
        && !within(ConcurrentAccessControl);
4   before(): pcBeforeWrite() {
5      //Crosscutting Concerns Implementation
6     }
7 }
```

Tabela I
VALORES DAS MÉTRICAS MS4PON PARA JPEXCLUSIVESETCALLS

Notation	NE	NLE	NVD	NSIE	NSOE	NSPE	LVE	NS	NH
JPDD	7	6	4	0	4	1	1	0	0
AspectJ	7	6	7	1	7	0	0	0	0
Theme/UML	24	6	3	0	6	1	2	1	10

para um entre o símbolo e sua semântica. O mesmo ocorreu no uso da notação textual AspectJ onde o NH é zero pois nenhum elemento da linguagem textual viola os requisitos *sine qua non* para se denominar como um sistema notacional [14].

Os critérios mencionados para calcular as métricas do MS4PON demonstram que é possível aplicar as métricas propostas para avaliar as especificações da seleção dos POCT, independentemente de usar uma notação gráfica ou textual. Portanto, o elemento que pode ser visto em uma especificação gráfica também é um elemento que pode ser visto em uma especificação textual. A diferença está na forma de como o elemento é expresso e apresentado.

V. EXPERIMENTOS E RESULTADOS

Esta seção discute os experimentos empíricos realizados para avaliar a especificação da seleção de POCT usando duas notações gráficas (JPDD e Theme/UML) e uma notação textual (AspectJ). Essas três notações foram escolhida devido a grande quantidade de citações de trabalhos relacionados e por serem amplamente conhecidas dentro do contexto da AOSD [1]. O objetivo é avaliar os efeitos cognitivos e as propriedades de percepção dessas notações, empregando o modelo de qualidade proposto através do cálculo das métricas MS4PON descritas na seção anterior.

A fonte de dados para essas experiências inclui 15 POCT comumente encontrados no desenvolvimento de software para sistemas ciber-físicos e sistemas embarcados de tempo real [6]. Esses POCT foram criados pelos próprios autores, em três estudos de caso realizados em trabalhos anteriores [12], [13]: (i) o sistema de controle de movimento de um veículo aéreo não tripulado; (ii) o sistema de controle de um sistema industrial de empacotamento; e (iii) o sistema de controle de movimento de uma cadeira de rodas automatizada. Os experimentos realizados neste trabalho consistem em criar três especificações para cada uma das quinze POCT usando as notações JPDD, Theme/UML e AspectJ, totalizando 45 especificações distintas. Em seguida, as métricas do MS4PON foram calculadas sobre cada especificação distinta para todas as seleções de POCT. Embora essas notações usadas permitam

Tabela II
MÉTRICAS PARA TODAS AS ESPECIFICAÇÕES DE SELEÇÃO DE POCT

Notação	NE	NLE	NVD	NSIE	NSOE	NSPE	LVE	NS	NH
JPDD	87	72	55	0	55	22	1,333	0	16
AspectJ	201	72	201	28	201	0	0	0	0
Theme/UML	360	72	45	0	90	15	2	15	150

uma especificação completa baseada na AOSD, os experimentos e seus resultados consideram apenas a parte das notações relacionadas a especificação das seleções de POCT.

A Tabela II mostra os valores das métricas do MS4PON calculadas a partir das 45 especificações de seleção de POCT. Os valores representam a soma dos valores individuais de cada métrica para cada especificação, com exceção do valor da LVE, que é calculado como um valor médio. A LVE representa a expressividade visual da notação de acordo com o framework PON [14], cujo valor deve variar de zero a oito. A soma da métricas LVE de todas as especificações não representa adequadamente o conceito de expressividade visual porque indicaria frequentemente uma notação saturada (i.e., valor maior que 8) conforme o número de especificações aumenta.

As métricas da MS4PON quantificam os recursos de uma notação usados para especificação de seleção de POCT. No entanto, para melhorar a análise e a comparação dos resultados, foram criadas taxas (i.e., valores relativos) através da associação entre as métricas. A Tabela III mostra os valores das métricas dividos por NE ou NLE ou ambos. As taxas permitem quantificar melhor a eficácia cognitiva de uma notação sob a perspectiva do framework PON. O objetivo é identificar, para cada métrica, como a propriedade de percepção é distribuída pelos elementos descritos nas especificações.

A taxa NVD/NE indica quantos elementos diferentes aparecem para cada elemento usado na especificação. Para essa taxa a notação textual AspectJ apresentou um valor mais alto em comparação com as notações gráficas JPDD e Theme/UML. Isso significa que a linguagem AspectJ emprega mais elementos diferentes para cada elemento usado na especificação de seleção de POCT. O mesmo raciocínio é válido para a relação NVD/NLE que considera o número de elementos diferentes por expressão lógica. Quanto maior a distância visual por expressão lógica, melhor o entendimento da especificação. No entanto, é importante destacar que um bom valor para essas proporções mencionadas deve ser próximo a um (1), i.e., a notação possui apenas um elemento distinto por expressão lógica. Valores menores que 1 indicam que existem elementos implícitos na especificação e valores maiores que 1 indicam que a notação é saturada. Ambas as situações dificultam de alguma forma o entendimento da especificação.

As taxas relacionadas ao NSIE indicam quantos elementos semanticamente imediatos são usados por elemento na especificação ou por expressão lógica. Essas proporções identificam quão fácil é inferir o significado correto dos elementos descritos na especificação pela aparência dos elementos e símbolos. Por exemplo, a proporção NSIE/NE para o AspectJ é um pouco mais alta do que nas outras técnicas, ou seja, a AspectJ possui mais elementos (na especificação) que

Tabela III
TAXAS DAS MÉTRICAS MS4PON

Taxas	AspectJ	JPDD	Theme/UML	Valores Ideais*
NE/NLE	2.790	1.210	5.000	1
NVD/NE	1.000	0.632	0.125	[1,2)
NVD/NLE	2.792	0.764	0.625	[1,2)
NSIE/NE	0.139	0.000	0.000	>0
NSIE/NLE	0.389	0.000	0.000	>0
NSOE/NE	1.000	0.632	0.250	0
NSOE/NLE	2.792	0.764	1.250	0
NSPE/NE	0.000	0.250	0.040	0
NSPE/NLE	0.000	0.306	0.208	0
LVE	0.000	1.333	2.000	8
NS/NE	0.000	0.000	0.042	0
NS/NLE	0.000	0.000	0.208	0
NH/NE	0.000	0.184	0.417	0
NH/NLE	0.000	0.222	2.083	0

* A notação ideal não deve ter nenhum homógrafos, sinógrafos, ou elementos perversos ou opacos que são elementos convencionados arbitrariamente. Por outro lado, quanto maior o número de elementos semanticamente imediatos, melhor a compreensão.

apresentam entendimento semântico implícito que a JPDD e a Theme/UML. Isso pode ter ocorrido devido ao fato de que JPDD e Theme/UML são extensões da UML. De acordo com [14], a maioria das notações que estendem a UML não foi projetada considerando as propriedades perceptivas e cognitivas que uma notação visual deve ter.

Ao analisar as taxas apresentadas na Tabela III, percebe-se que AspectJ é mais eficaz em relação a compreensão das especificações de selecão de POCT em comparação com JPDD e Theme/UML. Esses resultados empíricos indicam que a especificação de seleções de POCT através de notações gráficas apresenta pouco discernimento na eficácia cognitiva das representações visuais. Portanto, essa análise realizada sob a perspectiva do framework PON aponta para o entendimento de que o uso de notações gráficas para a especificação da seleção de POCT é um processo inconsciente, pois as especificações são de alguma maneira difíceis de se entender quando os engenheiros de software não tem experiência suficiente na notação gráfica usada.

VI. LIMITAÇÕES

Embora os resultados obtidos forneçam o suporte adequado para as conclusões deste trabalho, alguns fatores podem ter influenciado a avaliação realizada, como é habitual em qualquer estudo empírico. Os fatores de ameaças à validade interna da avaliação podem surgir do fato de que os resultados da eficácia cognitiva podem ser influenciados pela experiência de quem lê as especificações, no caso, os autores deste trabalho. Entretanto, consideramos o risco desse fato afetar os resultados obtidos como baixo, uma vez que o modelo de qualidade proposto e os resultados experimentais são baseados na perspectiva do framework PON. Esse risco também é atenuado, porque as POCT usadas no experimento são de estudos de caso do mundo real, e não criados artificialmente.

Os resultados alcançados podem ter relação direta com a qualidade das especificações criadas. A falta de experiência em lidar com com características transversais (tanto na especificação quanto na implementação do software embarcado), e também a no uso das notações visuais, pode levar a

criação de especificações com baixa qualidade, o que impacta na sua compreensão. O risco de resultados obtidos não representarem a realidade é baixo pois os autores possuem bastante experiência nos tema mencionados. Além disso, o framework PON e as métricas MS4PON avaliam as propriedades físicas perceptivas (i.e. sintaxe) das notações e não sua semântica, o que nos permite ter confiança nos resultados alcançados.

Outra ameaça à validade pode estar relacionada a generalidade dos resultados, uma vez que estão limitados aos três casos de estudos do domínio de sistemas embarcados. O modelo de qualidade proposto e as métricas MS4PON são genéricas e podem ser usadas para avaliar outras notações visuais e textuais. No entanto, dentro do escopo da especificação de seleção de POCT, a variedade de tipos das ocorrências usadas nos experimentos nos permite acreditar que resultados semelhantes serão obtidos em outros domínios de aplicação.

VII. Conclusões e Trabalhos Futuros

Este trabalho avalia a compreensão da especificação de seleções de POCT de forma indireta, através da avaliação das propriedades de percepção de notações gráficas e textuais para três técnicas distintas (AspectJ, JPDD e Theme/UML). Para isso, foram propostos um modelo de qualidade e um conjunto de métricas baseados nas premissas do framework conceitual *Physics of Notations* (PON) [14].

Os resultados experimentais mostraram evidências empíricas de que a especificação gráfica da seleção de POCT apresenta baixo discernimento na eficácia cognitiva das representações visuais. As notações gráficas avaliadas nos experimentos apresentam um grande número de elementos homógrafos para especificar as seleções dos POCT. Os homógrafos são anomalias que levam à ambiguidade e potencializam o erro nas interpretações dos modelos [14].

Além disso, a distância visual das notações gráficas avaliadas ficou aquém do esperado. Os resultados mostram um valor máximo de 0,764 elementos diferentes por expressão lógica. Isso significa que a maioria dos elementos visuais são iguais ou semelhantes, afetando negativamente a compreensão e a cognição dos engenheiros. Por outro lado, a notação textual avaliada (AspectJ) mostra uma razão de 2,792, indicando que, para cada expressão lógica, existem mais de dois elementos diferentes, i.e., a distância visual entre os elementos na AspectJ é melhor que em JPDD e Theme/UML.

O grande número de elementos semanticamente perversos (NSPE) nas notações gráficas avaliadas chama a atenção. Na JPDD e particularmente Theme/UML, a especificação da seleção de POCT inclui elementos ou símbolos cuja natureza possui um falso mnemônico. Isso pode induzir um engenheiro de software iniciante a inferir um significado diferente do que se apresenta na especificação. Além disso, o nível de expressividade (LVE) das notações gráficas avaliadas se apresentou muito baixo, próximo a de uma notação textual, pois elas apresentam poucas variáveis de comunicação visual (e.g., forma, textura, cores, etc.). No entanto, os indícios apontam que a JPDD e a Theme/UML não se beneficiam das vantagens esperadas de uma linguagem visual. Moody [14] afirma que isso é semelhante ao que ocorre com a maioria das notações visuais usadas na engenharia de software.

Como trabalho futuro, planeja-se avaliar outras notações gráficas e textuais para especificar a seleção de POCT mais complexas. Além disso, considerando que as notações gráficas não melhoraram o entendimento da seleção de POCT, pretende-se criar em uma nova técnica para especificação de POCT através de um modelo que inclui os elementos textuais e gráficos (i.e., dual coding [14]), visando aumentar a eficácia cognitiva de suas propriedades de percepção.

Referências

[1] R. Filman, T. Elrad, S. Clarke, and M. Akşit, Eds., *Aspect-Oriented Software Development*, 1st ed. Addison-Wesley Professional, 2004.

[2] D. Stein *et al.*, "Join point designation diagrams: a graphical representation of join point selections," *Int. Journal of Software Engineering and Knowledge Engineering*, vol. 16, no. 03, pp. 317–346, 2006.

[3] T. Elrad *et al.*, "Aspect-oriented modeling: Bridging the gap between implementation and design," in *Generative Programming and Component Engineering*, D. Batory *et al.*, Eds. Springer, 2002, pp. 189–201.

[4] J. Kienzle, W. Al Abed, and J. Klein, "Aspect-oriented multi-view modeling," in *Proc. of AOSD'09*. ACM, 2009, pp. 87–98.

[5] S. Yang and Z. Wei-Dong, "Aspect-oriented modeling in concurrent system," in *IEEE Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. IEEE, 2019, pp. 836–840.

[6] E. P. Freitas *et al.*, "DERAF: A high-level aspects framework for distributed embedded real-time systems design," in *Early Aspects: Current Challenges and Future Directions*. Berlin, Heidelberg: Springer, 2007, pp. 55–74, doi: 10.1007/978-3-540-76811-1_4.

[7] M. Mongiovì *et al.*, "Specifying and identifying widely used crosscutting concerns," *Knowledge-Based Systems*, vol. 126, pp. 20 – 32, 2017.

[8] V. Acreţoaie, H. Störrle, and D. Strüber, "VMTL: a language for end-user model transformation," *Software & Systems Modeling*, vol. 17, no. 4, pp. 1139–1167, 2018.

[9] M. U. Khan *et al.*, "AspectOCL: using aspects to ease maintenance of evolving constraint specification," *Empirical Software Engineering*, vol. 24, no. 4, pp. 2674–2724, 2019.

[10] Object Management Group (OMG), "Unified Modeling Language version 2.5.1," 2017, https://www.omg.org/spec/UML/2.5.1/.

[11] S. Clarke, "Extending standard uml with model composition semantics," *Science of Computer Programming*, vol. 44, no. 1, pp. 71–100, 2002.

[12] M. A. Wehrmeister, E. P. de Freitas, A. P. D. Binotto, and C. E. Pereira, "Combining aspects and object-orientation in model-driven engineering for distributed industrial mechatronics systems," *Mechatronics*, vol. 24, no. 7, pp. 844 – 865, 2014, doi: 10.1016/j.mechatronics.2013.12.008.

[13] M. A. Wehrmeister, C. E. Pereira, and F. J. Rammig, "Aspect-oriented model-driven engineering for embedded systems applied to automation systems," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2373–2386, 2013, doi: 10.1109/TII.2013.2240308.

[14] D. Moody, "The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering," *IEEE Transactions on Software Engineering*, vol. 35, no. 6, pp. 756–779, 2009.

[15] D. Jones and S. Gregor, "The anatomy of a design theory," *Journal of the Association for Information Systems*, vol. 8, no. 5, p. 1, 2007.

[16] G. Kiczales *et al.*, "An overview of aspectj," in *European Conference on Object-Oriented Programming*. Springer, 2001, pp. 327–354.

[17] E. Baniassad and S. Clarke, "Theme: An approach for aspect-oriented analysis and design," in *26th Int. Conf. on Software Engineering*. IEEE, 2004, pp. 158–167.

[18] R. R. McFadden and F. J. Mitropoulos, "Survey of aspect mining case study software and benchmarks," in *2013 Proceedings of IEEE Southeastcon*, April 2013, pp. 1–5.

[19] D. Stein and S. Hanenberg, "Comparison of a visual and a textual notation to express data constraints in aspect-oriented join point selections: A controlled experiment," in *Int. Conf. on Program Comprehension*. IEEE, 2011, pp. 141–150.

[20] C. Allan *et al.*, "Adding trace matching with free variables to aspectj," *Sigplan Notices - SIGPLAN*, vol. 40, pp. 345–364, 10 2005.

[21] Object Management Group (OMG), "UML profile for Modeling and Analysis of Real-Time Embedded systems (MARTE), version 1.2," 2018, https://www.omg.org/spec/MARTE/.