



Ciência de Dados e I.A.
Escola de Matemática Aplicada
Fundação Getúlio Vargas

Engenharia de Requisitos

TCC

**Towards New Hybrid Approach of
the Reverse Engineering of UML
Sequence Diagram**

Aluno: Isabela Yabe
Orientador: Rafael de Pinho André
Escola de Matemática Aplicada, FGV/EMAp
Rio de Janeiro - RJ.

Rio de Janeiro, 2025

1 Revisão literária

Artigo revisado Baidada e Jakimi (2016):

A revisão tem o objetivo de compreender o estado da arte das abordagens de engenharia reversa que partem de código-fonte e produzem artefatos de alto nível, como diagramas UML. Para garantir uma análise sistemática e comparável entre diferentes propostas, foram definidas perguntas de pesquisa (*Research Questions — RQs*) que orientam a coleta e síntese dos dados extraídos dos estudos selecionados.

- **RQ1.** Em quais linguagens e domínios as abordagens que partem de código-fonte foram aplicadas?
- **RQ2.** Quais modelos/artefatos de alto nível são gerados?
- **RQ3.** Qual aspecto é privilegiado (estático, dinâmico, híbrido) e com qual objetivo (compreensão, redocumentação, migração, qualidade)?
- **RQ4.** Quais técnicas e transformações viabilizam a passagem do código para o modelo de alto nível?
- **RQ5.** Quais ferramentas/frameworks são utilizados?
- **RQ6.** Como as abordagens são validadas e com que qualidade prática?

2 RQ1. Em quais linguagens e domínios as abordagens que partem de código-fonte foram aplicadas?

A abordagem é declaradamente voltada a linguagens Java. As evidências incluem (i) um exemplo de *control flow graph* (CFG) de um método Java e (ii) a geração de traços por execução em máquina virtual para programas Java, como explicitado pelos autores.

3 RQ2. Quais modelos/artefatos de alto nível são gerados?

O trabalho gera Diagramas de Sequência UML de alto nível (HLS), com uso de operadores de interação (seq, alt, opt, loop, par).

Os HLS são construídos com os operadores de combined SD do UML 2: seq, alt, opt, loop, par.

4 RQ3. Qual aspecto é privilegiado (estático, dinâmico, híbrido) e com qual objetivo (compreensão, redocumentação, migração, qualidade)?

É proposto uma abordagem híbrida (estático + dinâmico).

O objetivo principal é compreensão e redocumentação, diagramas de sequencia, com efeito direto em manutenção/evolução.

5 RQ4. Quais técnicas e transformações viabilizam a passagem do código para o modelo de alto nível?

A abordagem propõe um *pipeline* híbrido composto por quatro etapas articuladas por uma representação intermediária (IR) comportamental baseada em *Colored Petri Nets* (CPN), culminando na geração do *UML Sequence Diagram*.

Na primeira etapa, realiza-se a análise estática do código-fonte por meio da construção de um Control Flow Graph (CFG), que permite identificar condições de execução e derivar valores de entrada capazes de cobrir diferentes comportamentos do sistema.

Em seguida, essas entradas são utilizadas para executar o sistema e coletar traços de execução (*execution traces*) por instrumentação, máquina virtual (no caso de Java) ou depurador customizado. Diversas execuções são realizadas para capturar variações comportamentais, seguidas por uma filtragem de ruído que remove eventos irrelevantes.

Os traços resultantes são convertidos em uma representação intermediária comportamental, estruturada como uma rede de Petri colorida (CPN). Nessa representação, lugares correspondem a diagramas de sequência básicos e transições a operadores de controle (*alt*, *loop*, *seq*, *par*). Um algoritmo incremental percorre os traços linha a linha para construir a CPN e identificar paralelismos por contagem de *threads*.

Por fim, aplica-se a transformação de CPN para UML Sequence Diagram, guiada por regras de mapeamento que traduzem operadores e fluxos em *combined fragments* na notação UML 2.x.

6 RQ5. Quais ferramentas/frameworks são utilizados?

O padrão-alvo é a UML 2.x (OMG), utilizada como linguagem de modelagem de destino. A coleta de traços de execução é realizada por meio de instrumentação do código, uso da máquina virtual Java (JVM) ou de um depurador customizado, dependendo do ambiente analisado.

A representação intermediária é construída sobre o formalismo de Colored Petri Nets (CPN), o que sugere o uso de ferramentas da família CPN Tools para manipulação, análise e simulação das redes.

A transformação final para o Diagrama de Sequência UML é conduzida por regras de mapeamento baseadas em modelo, alinhadas às especificações da UML 2.x, possivelmente implementadas sobre um framework de transformação como o Eclipse Modeling Framework (EMF) ou ferramenta equivalente.

7 RQ6. Como as abordagens são validadas e com que qualidade prática?

Não há validação reportada ou experimento, estudo de caso, métrica, análise de desempenho ou comparação sistemática.

Autores / Referência	Linguagem / Domínio	Modelo Gerado	Aspecto	Técnica / Transformação	Ferramenta / Framework	Validação / Estudo de Caso
Baidada e Jakimi (2016)	OO (ex.: Java); multithread	UML SD (HLSD; seq/alt/opt/loop/par/partition)	Híbrido; compreensão + documentações + traços (filtro); traços→CPN; CPN→UML SD	CFG→entradas; Sem execuções + traços (filtro); traços→CPN; CPN→UML SD	Sem ferramenta nominal; UML 2.x; instrumentação/VM/debugger; CPN (IR)	Sem validação; futuro

Tabela 1: Resumo das abordagens

Referências

BAIDADA, C.; JAKIMI, A. Towards a New Hybrid Approach of the Reverse Engineering of UML Sequence Diagram. *In: 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. Beijing, China: IEEE, 2016. p. 164–168.