



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

Engenharia de Requisitos

**TCC**

# **WIP: Generating Sequence Diagrams for Modern Fortran**

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 Leatongkam, Nanthamornphong e Rouson (2017):

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 foi aplicada à linguagem Fortran moderna (OO Fortran).

O domínio de aplicação é o da computação científica.

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

Os modelos e artefatos de alto nível gerados são derivados da engenharia reversa de código Fortran orientado a objetos, produzindo representações UML que capturam tanto a estrutura quanto o comportamento do sistema.

Originalmente implementado pela ferramenta ForUML, que extrai de código Fortran OO as estruturas de classes, atributos, métodos e relações de agregação e herança.

Esses diagramas representam o aspecto estático do sistema, isto é, sua organização estrutural e hierarquias de classes.

O trabalho propõe uma extensão da ferramenta ForUML para gerar também diagramas de sequência.

O processo de extração gera um arquivo intermediário em formato XMI, que armazena a estrutura do diagrama UML para importação em ferramentas de modelagem como o ArgoUML.

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

A proposta adota uma abordagem estática, fundamentada na análise do código-fonte Fortran. Concentrando-se exclusivamente na inspeção estrutural e semântica do código.

O propósito declarado pelos autores é auxiliar a compreensão e redocumentação de programas científicos escritos em Fortran OO, cujo ciclo de vida é longo e frequentemente carece de documentação de design.

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

Se baseiam em três etapas principais: definição de regras, extração e geração de modelo.

A primeira etapa consiste em definir regras formais de mapeamento entre elementos da linguagem Fortran OO e notações de diagramas UML de sequência. Ou seja, o sistema identifica padrões sintáticos no código (como chamadas de método, criação de objetos e controle de fluxo) e os converte em elementos comportamentais UML.

Na segunda etapa um módulo de extração percorre o código-fonte Fortran e constrói uma árvore de nós (tree node structure) que representa a hierarquia de classes e suas relações. Durante essa travessia, o código é validado quanto à gramática Fortran, garantindo consistência sintática antes da geração dos diagramas. Essa estrutura intermediária atua como um modelo abstrato do sistema, análogo a um Abstract Syntax Tree (AST) estendida com semântica de classes e interações.

Por fim, na terceira etapa o processo de transformação produz um modelo intermediário XMI, padrão utilizado para representar diagramas UML em formato intercambiável. Esse arquivo é então importado por ferramentas de modelagem como o ArgoUML, que renderiza os diagramas de sequência resultantes.

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

O processo se baseia em três etapas principais: definição de regras de mapeamento, extração estrutural e geração do modelo UML.

Na primeira etapa, são definidas regras formais que associam elementos da linguagem Fortran OO a notações de diagramas UML. O sistema identifica padrões sintáticos, como chamadas de método, criação de objetos e estruturas de controle, e os converte em elementos comportamentais UML.

Na segunda etapa, um módulo de extração percorre o código-fonte e constrói uma estrutura em árvore de nós (tree node structure) que representa classes, métodos e relações. Durante essa travessia, o código é validado segundo a gramática Fortran, assegurando consistência sintática. Essa estrutura atua como um modelo abstrato do sistema, análogo a uma AST (Abstract Syntax Tree) enriquecida com semântica de classes e interações.

Por fim, a terceira etapa gera um modelo intermediário XMI, padrão da OMG para representação de diagramas UML. Esse arquivo é então importado por ferramentas como o ArgoUML, que renderiza automaticamente os diagramas de sequência.

## **7 RQ5. Quais ferramentas/frameworks são utilizados?**

As ferramentas e frameworks empregados concentram-se na extensão da infraestrutura já existente da ferramenta ForUML (Fortran UML Reverse Engineering Tool), complementada por tecnologias padrão de modelagem UML e interoperabilidade via XMI.

O ForUML é a ferramenta-base do estudo, ela é capaz de analisar código-fonte Fortran orientado a objetos e gerar diagramas UML de classes. E será agora estendida para incluir diagramas de sequência.

O XMI é o formato de intercâmbio padronizado pela OMG (Object Management Group), responsável por definir as especificações UML utilizadas no mapeamento.

Ferramenta open-source usada para visualização dos modelos UML gerados pela ForUML é o ArgoUML é compatível com o padrão XMI (XML Metadata Interchange), o que permite importar o modelo intermediário e renderizar automaticamente o diagrama.

As regras de transformação entre código Fortran e notação UML seguem o padrão de especificação UML do OMG. Isso garante conformidade semântica e sintática com o metamodelo UML e integração com outras ferramentas compatíveis.

## **8 RQ5. Quais ferramentas/frameworks são utilizados?**

As ferramentas empregadas concentram-se na extensão da infraestrutura da ForUML (Fortran UML Reverse Engineering Tool), complementada por tecnologias padrão de modelagem UML e interoperabilidade via XMI.

A ForUML é a ferramenta-base do estudo: capaz de analisar código-fonte Fortran OO e gerar diagramas UML de classes, sendo agora expandida para incluir diagramas de sequência.

O XMI (XML Metadata Interchange) é o formato de intercâmbio padronizado pela OMG (Object Management Group), utilizado para garantir a compatibilidade e a conformidade com o metamodelo UML.

A visualização dos diagramas é realizada no ArgoUML, ferramenta open-source compatível com o padrão XMI, que permite importar e renderizar automaticamente

os modelos gerados.

As regras de transformação seguem as especificações da UML 2.x da OMG, assegurando aderência semântica e integração com outros frameworks de modelagem.

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

O estudo é apresentado como um work in progress, descrevendo as etapas conceituais e o plano de testes para a nova funcionalidade de geração de diagramas de sequência no ForUML.

O artigo menciona que o projeto foi publicado no GitHub para incentivar o uso e colaboração da comunidade científica.

<b>Autores / Referência</b>	<b>Linguagem / Domínio</b>	<b>Modelo Gerado</b>	<b>Aspecto</b>	<b>Técnica / Transforma- ção</b>	<b>Ferramenta / Fra- mework</b>	<b>Validação / Estudo de Caso</b>
Leatongkam, Nanthaa- mornphong e Rouson (2017)	Fortran OO; computação científica e engenharia	UML Classe; UML Sequên- cia; modelo intermediário XMI	Estático — compreensão e redocumen- tação	Regras de mapeamento código → UML (OMG); parsing está- tico; árvore sintática; ge- ração XMI → importação ArgoUML	ForUML (extensão); ArgoUML; padrão OMG UML/XMI;	<i>Work in progress</i>

Tabela 1: Resumo das abordagens

## Referências

LEATONGKAM, A.; NANTHAAMORNPHONG, A.; ROUSON, D. W. WIP: Generating Sequence Diagrams for Modern Fortran. *In*: 2017 IEEE/ACM 12th International Workshop on Software Engineering for Science (SE4Science). Buenos Aires, Argentina: IEEE, 2017. p. 22–25.