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

Engenharia de Requisitos

TCC

Condensing Class Diagrams With Minimal Manual Labeling Cost

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 Yang *et al.* (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 o condensamento dos diagrams?
- **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 proposta, denominada *MCCondenser*, foi aplicada ao domínio de **sistemas orientados a objetos desenvolvidos em Java**, com foco na condensação de diagramas de classes gerados via engenharia reversa. Nesse contexto, o estudo busca apoiar a compreensão e documentação de grandes sistemas, reduzindo a complexidade dos diagramas produzidos a partir do código-fonte ao avaliar quais classes são mais relevantes para a visualização e análise.

Para validar o método, foram utilizados nove sistemas de código aberto amplamente conhecidos: ArgoUML, JavaClient, JGAP, JPMC, Mars, Maze, Neuroph, Wro4J e xUML. Esses projetos representam diferentes comunidades e domínios de aplicação, todos implementados em Java, o que demonstra a aplicabilidade da abordagem em distintos contextos de software orientado a objetos.

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

Concentra-se na condensação de **diagramas de classes UML** obtidos por engenharia reversa de sistemas orientados a objetos. O modelo principal é, portanto, o *Class*

Diagram, cuja reconstrução visa condensar a representação estrutural obtida pela engenharia reversa, destacando apenas as classes mais relevantes.

Assim, o artefato de alto nível produzido é um **diagrama de classes condensado**.

O processo se apoia etapas automáticas de análise de métricas e classificação supervisionada/não supervisionada para identificar as "important classes". Assim, o modelo final é um **subconjunto do diagrama UML original**, estruturado de forma a manter a coerência semântica e a utilidade para documentação e manutenção de software.

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

O artigo privilegia o **aspecto estático** do sistema, pois toda a abordagem baseia-se em **métricas estruturais extraídas de diagramas de classes** gerados a partir do código-fonte. O método MCCondenser analisa características como tamanho, acoplamento e centralidade de classes em redes de dependência, sem recorrer a informações de execução ou rastreamento dinâmico.

Essas métricas são calculadas com base em propriedades estruturais do código, como número de métodos, atributos e relações entre classes.

5 RQ4. Quais técnicas e transformações viabilizam o condensamento dos diagrams?

O artigo Yang *et al.* (2016) propõe o método **MCCondenser**, que utiliza uma combinação de técnicas de aprendizado de máquina, **k-means clustering**, **random under-sampling** e **ensemble learning** — para condensar diagramas de classes gerados por engenharia reversa, reduzindo o número de classes exibidas sem comprometer a representatividade do modelo.

Primeiramente, o processo parte da extração de métricas estruturais (tamanho, acoplamento e rede) das classes de um projeto (utilizando o SDmetrics). Essas métricas são normalizadas pelo método **z-score** para padronizar as magnitudes numéricas.

A seguir, aplica-se o algoritmo **k-means clustering** para identificar subconjuntos representativos de classes, que servirão como amostras rotuladas manualmente. Essa etapa constitui a primeira transformação crítica do processo de condensamento, pois permite selecionar amostras diversificadas e representativas de todo o diagrama, reduzindo significativamente o custo de rotulagem manual.

Para lidar com o desequilíbrio entre classes importantes e não importantes, o método utiliza **random under-sampling**.

Por fim, o **ensemble learning** é aplicado para combinar diversos classificadores gerados sobre subconjuntos distintos dos dados, criando um modelo final mais robusto e generalizável.

Esse modelo é responsável por prever automaticamente quais classes são "importantes" e devem permanecer no diagrama condensado.

6 RQ5. Quais ferramentas/frameworks são utilizados?

O artigo faz uso de um conjunto de ferramentas de modelagem e mineração de dados voltadas à engenharia reversa e ao aprendizado de máquina. As principais são:

MagicDraw utilizada para gerar o diagrama de classes a partir do código-fonte. Os autores descrevem essa etapa como o ponto de partida da abordagem:

SDMetrics empregada para extrair métricas estruturais e de acoplamento das classes representadas no diagrama UML.

Random Forest (como base classifier) escolhido como classificador principal para o processo de ensemble learning.

Ferramentas e ambiente computacional os experimentos foram conduzidos em ambiente Windows 7 (64-bit) com CPU Intel Core T6570 e 4GB RAM, demonstrando que a execução do MCCondenser requer baixo poder computacional.

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

A validação da abordagem **MCCondenser** foi conduzida por meio de um **estudo experimental extensivo** realizado sobre nove sistemas de **código aberto desenvolvidos em Java**: ArgoUML, JavaClient, JGAP, JPMC, Mars, Maze, Neuroph, Wro4J e xUML, totalizando **2.640 classes**. O objetivo foi mensurar a eficácia e a eficiência do método em condensar diagramas de classes com o menor custo de rotulagem manual possível.

A métrica central empregada foi a **AUC (Area Under the ROC Curve)**, utilizada em problemas de classificação binária. Seu uso é justificado pela natureza **desequilibrada dos conjuntos de dados**, em que há muito mais classes "não importantes" do que "importantes", pois a AUC mede a capacidade do modelo de ranquear corretamente ambas as categorias.

Com base nessa métrica, o experimento utilizou apenas 10% dos dados como amostras rotuladas, enquanto o restante foi previsto automaticamente. Os autores realizaram dez repetições para reduzir o viés estatístico. O classificador base adotado foi o **Random Forest**, escolhido por sua robustez em comparação a outros algoritmos testados.

Com esse delineamento experimental, os resultados mostraram que o **MCCondenser** superou significativamente os métodos de referência. Essa superioridade foi posteriormente confirmada pelos testes de Wilcoxon e *Cliff's delta*, assegurando a robustez empírica dos resultados. Além disso, a diferença de desempenho entre o uso de 10% e 90% dos dados rotulados foi pequena (queda média de apenas 0,13 na AUC), o que comprova o custo-benefício e a eficiência prática da abordagem, mesmo sob condições de mínima intervenção manual.

Autores / Referência	Linguagem / Domínio	Modelo Gerado	Aspecto	Técnica / Transformação	Ferramenta / Framework	Validação / Estudo de Caso
Yang *et al.* (2016)	Java; sistemas OO	UML Classe (condensado)	Estático; Compreensão/Redocumentação (estrutura)	**Extração de métricas (SDMetrics)** → Normalização (z-score) → **k-means clustering** → Random undersampling → **Ensemble learning (Random Forest)** → diagrama condensado	MagicDraw; SDMetrics; Random Forest; Windows 7	OSS (9 projetos, 2640 classes); AUC=0.73; custo de rótulo=10%; teste de Wilcoxon e Cliff's δ

Tabela 1: Resumo da abordagem de Yang et al. (2016) — MCCondenser

Referências

YANG, X. *et al.* Condensing Class Diagrams With Minimal Manual Labeling Cost. Versão inglesa. *In:* Proceedings of the 40th IEEE Annual International Computers, Software and Applications Conference (COMPSAC 2016). Atlanta, Georgia, USA: IEEE, 2016. p. 22–31. Published in COMPSAC 2016: Proceedings of the 40th IEEE Annual International Computers, Software and Applications Conference.