COOLPROFS

Afstudeeropdracht HBO – AI-gedreven Diagramanalyse voor C4-Architecturen

Achtergrond

Om de architectuur binnen software systemen te visualiseren gebruiken wij het C4 model concept. Hiermee bereiken wij alle stakeholders door duidelijk en inzichtelijke diagrammen te genereren met onze Cool Model. We ontwikkelen een intern product die softwarearchitecturen visualiseert op basis van het C4-model (Context, Container, Component, Code). Binnen onze C4-visualisatietool willen we een tweede pijler toevoegen: een intelligente analysemodule die de kwaliteit van architectuurdiagrammen automatisch beoordeelt. Naast lay-outverbetering richten we ons nu op structurele kwaliteit: domeinscheiding (bounded contexts), koppelingen en afhankelijkheden, en het vroegtijdig signaleren van risico's zoals dependency magnets en circulaire dependencies.

Opdrachtomschrijving

Ontwikkel en onderzoek een AI-gedreven analysemodule die C4-diagrammen en bijbehorende metadata/codebasis analyseert op kwaliteitscriteria. De module detecteert problemen in domeinscheiding en afhankelijkheden, geeft gerichte verbetersuggesties en kan deze suggesties toelichten met hulp van Large Language Models (LLMs).

Doelen

1. Automatische beoordeling van domeinscheiding en afhankelijkheden in C4-diagrammen.
2. Detectie van dependency magnets, circulaire dependencies en laagschendingen (layering violations).
3. Genereren van concrete, contextspecifieke verbetersuggesties met onderbouwing via LLMs.
4. Integreerbaar prototype (PoC) binnen onze toolchain.

Onderzoeksvragen

1. Welke metriek(en) zijn geschikt om domeinscheiding en modulariteit in C4-diagrammen te beoordelen (bijv. afferent/efferent coupling, instability, modularity/community-detectie)?
2. Hoe detecteren we dependency magnets (componenten met buitensporig hoge centraliteit/degree) en circulaire dependencies effectief en efficiënt?
3. Hoe kunnen LLMs ingezet worden voor:
    a. Het verklaren van de gevonden issues in natuurlijke taal;
    b. Het doen van gerichte refactors (bijv. hersnijden van bounded contexts, interface-extractie, event-driven ontkoppeling)?
4. Hoe kwantificeren we de kwaliteit van de verbetersuggesties (precisie/recall, architect-review, task-driven evaluatie)?

Aanpak en Methode

- Dataverzameling & Representatie: Parse C4-bronnen (Structurizr DSL/JSON, YAML, of interne export) naar een graafrepresentatie (nodes = context/container/component, edges = afhankelijkheden).
- Metrieken & Detectie: Bereken o.a. afferent (Ca) en efferent (Ce) coupling, instability I=Ce/(Ca+Ce), betweenness/degree centrality, edge cut, modularity; detecteer cycli (SCCs) en layered-violation rules.
- Domeinscheiding: Vergelijk community-detectie (bijv. Louvain/Leiden) met gedefinieerde bounded contexts; flag overlap/lekkage.
- AI/LLM-rol: Gebruik LLM met RAG over architectuur- en domeindocumentatie om findings te duiden en verbetersuggesties te genereren (patroonbibliotheek: anti-corruption layer, façade, domain events, CQRS, etc.).Agentic Orchestratie: Een eenvoudige 'AI agent' die iteratief issues ophaalt, prioriteert (op impact/risk), en suggesties produceert inclusief acceptatiecriteria.
- Validatie: Seed testcases met bekende problemen; vergelijk detectie met ground truth; laat 2–3 engineers een expert review doen.

Scope & Randvoorwaarden

- Inputformaten: minimaal Structurizr JSON of DSL-export; idealiter ook directe koppeling met onze interne tool.
- Uitvoer: JSON-rapport + human-readable uitleg en suggesties; optioneel annotaties/overlays op het diagram.
- Niet-doelen: Volledige automatische hertekening (valt buiten scope, kan als stretch).

Deliverables

1. Onderzoeksrapport (achtergrond, ontwerpkeuzes, evaluatie, beperkingen).
2. Werkend PoC (CLI of webmodule) dat:
    a. C4-data inleest;
    b. Metrieken berekent en issues detecteert (magnets, cycli, layer-violations, domein-lekkage);
    c. LLM-gedreven uitleg en verbetersuggesties oplevert (incl. voorbeeld refactor-stappen en architectuurpatronen).
3. Testset met synthetische en (geanonimiseerde) real-world voorbeelden + meetrapport (precisie/recall, runtime).
4. Demo & presentatie aan het team.

Evaluatiecriteria

- Technisch: Correctheid van detectie (cycli, magnets), performance, codekwaliteit.
- UX/Adoptie: Duidelijkheid van rapporten en bruikbaarheid van adviezen voor engineers/architecten.

- AI-kwaliteit: Nauwkeurigheid, traceerbaarheid (bronverwijzing via RAG), en veiligheid (hallucinatiebeperking).
- Wetenschappelijk: Degelijke onderbouwing van metriekkeuze en evaluatieset.

Technische Richting (suggestie)

- Taal: Python of TypeScript.
- Graph: NetworkX (Python) of graphlib/graphology (TS); SCC-detectie (Tarjan/Kosaraju), centrality.
- LLM-integratie: OpenAI of lokaal model; RAG over interne docs; prompt-templates per issue-type.
- Packaging: REST-endpoint of module die onze tool kan aanroepen; JSON-rapport + HTML-viewer.

Planning (indicatief, 20 weken)

Wk 1–3: Verdieping & datamodel • Wk 4–6: Metrieken & detectie • Wk 7–10: LLM/RAG + agent-loop • Wk 11–14: PoC & integratie • Wk 15–17: Validatie & meetset • Wk 18–20: Rapport & demo.

Acceptatiecriteria (Definition of Done)

- Kan ten minste 3 issue-typen automatisch detecteren (cycli, magnets, domein-lekkage of layer-violation).
- Levert per issue een begrijpelijke uitleg + minimaal 1 concrete verbetersuggestie.
- Rapportage exporteerbaar (JSON + leesbare samenvatting).

Ethiek & Veiligheid

- Geen broncode of bedrijfsgeheimen naar externe services zonder expliciete toestemming.
- Pseudonimiseer/anonimiseer data voor onderzoek.
- Log LLM-prompts/responsen voor herleidbaarheid; vermijd onjuiste zekerheidsclaims.

Waar kom je te werken

CoolProfs ontwikkelt maatwerk bedrijfsapplicaties voor en met organisaties zoals WoningNet, Stedin, a.s.r, Gemeentes, logistieke vervoerders, agribedrijfen en klanten in de financiële sector. Dit doen we met het (low-code) ontwikkelplatform OutSystems.

Ook zijn we continu bezig met het ontwikkelen van nieuwe features en tools. Ons R&D team onderzoekt nieuwe technologieën en vertaalt deze naar praktische add-ons, waar onze klanten en andere OutSystems-gebruikers over de hele wereld blij van worden. Als afstudeerde ga je meedraaien in het R&D team en je hebt dan ook beschikking tot de benodigde tooling en omgevingen.