الجمهورية الشعبية الديمقراطية الجزائرية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
المدرسة العليا للإعلام الآلي 8 ماي 1945 - سيدي بلعباس
Ecole Supérieure en Informatique-08 Mai

Mémoire

En Vue de l'obtention du diplôme de **Master**

Filière: **Informatique**

Spécialité: **Système d'Information et Web (SIW)**

Thème

Approaches and tools for static code analysis in Angular applications: state of the art

---

Présenté par:
Belhadj Mohammed El Amin

Soutenu le: **Sept, 2025**
Devant le jury composé de :

Pr. AMAR BENSABER Djamel	President
Pr. BENSLIMANE Sidi Mohammed	Encadreur
Dr. HLAD Nicolas	Encadreur
Dr. VERHAEGHE Benoit	Encadreur
Dr. BEZZAOUCHA Fatima Souad	Examinatrice

Année Universitaire: 2024/2025

Abstract

Angular, built on TypeScript, has become a leading framework for developing large-scale and modular web applications. Its structure—combining TypeScript components, HTML templates, and dependency injection—improves productivity but also raises new challenges for software quality. Static code analysis is a key practice to address these challenges, offering early detection of defects, vulnerabilities, and maintainability issues. This thesis presents a state of the art on static analysis approaches and tools in the context of Angular applications. It reviews existing solutions such as ESLint, SonarQube, and Fortify, as well as Angular-specific analyzers like Codelyzer and angular-eslint. Their strengths and limitations are examined with regard to false positives, limited coverage of framework-specific constructs, and integration into DevOps/CI pipelines. The study also explores model-driven engineering approaches, such as Moose and FAMIX, which provide higher-level representations and visualization capabilities. The analysis shows that static analysis is indispensable for quality assurance in Angular projects but remains incomplete. Current tools struggle with template–component bindings, redundancy detection, and security vulnerabilities specific to the framework. Perspectives for improvement include hybrid approaches combining static and dynamic analysis, better handling of Angular templates, stronger DevSecOps integration, and the reinforcement of open-source collaboration. In short, this work highlights both the maturity and the limits of static analysis for Angular/TypeScript, and outlines the directions needed to make it more precise, reliable, and adapted to modern development practices.

Résumé

Angular, basé sur TypeScript, s'est imposé comme un cadre majeur pour le développement d'applications web modulaires et de grande envergure. Sa structure combinant composants TypeScript, modèles HTML et mécanismes d'injection de dépendances favorise la productivité mais soulève également de nouveaux défis en matière de qualité logicielle. L'analyse statique du code constitue une pratique essentielle pour répondre à ces enjeux, en permettant la détection précoce de défauts, de vulnérabilités et de problèmes de maintenabilité. Ce mémoire présente un état de l'art des approches et outils d'analyse statique dans le contexte des applications Angular. Il examine les solutions existantes telles qu'ESLint, SonarQube ou Fortify, ainsi que les analyseurs spécifiques à Angular comme Codelyzer et angular-eslint. Leurs atouts et limites sont analysés au regard des faux positifs, de la couverture partielle des constructions propres au framework et de leur intégration dans les chaînes DevOps/CI. Le travail explore également les approches issues de l'ingénierie dirigée par les modèles, notamment Moose et FAMIX, qui offrent des représentations abstraites et des capacités avancées de visualisation. L'étude montre que l'analyse statique est indispensable pour assurer la qualité des projets Angular, mais qu'elle reste incomplète. Les outils actuels rencontrent des difficultés face aux liens entre composants et modèles, à la détection de redondances ou encore aux vulnérabilités spécifiques au framework. Les perspectives d'amélioration concernent l'adoption d'approches hybrides combinant analyse statique et dynamique, un meilleur traitement des modèles Angular, une intégration renforcée dans les pratiques DevSecOps et la valorisation des collaborations open source. En résumé, ce travail met en évidence à la fois la maturité et les limites de l'analyse statique pour Angular/TypeScript, tout en traçant les pistes nécessaires pour la rendre plus précise, fiable et adaptée aux pratiques modernes de développement logiciel.

الملخص

أصبح أنغولار المبني على تايب سكريبت من بين الأطر الأكثر استخداماً في تطوير تطبيقات الويب واسعة النطاق والمبنية على مكوّنات مستقلة. هذه البنية، التي تجمع بين مكوّنات تايب سكريبت وقوالب إتش.تي.إم.إل وآليات حقن الاعتمادية ، تعزز الإنتاجية لكنها تطرح في المقابل تحديات جديدة على مستوى جودة البرمجيات. في هذا السياق، يُعد التحليل الساكن للشفرة ممارسة أساسية للكشف المبكر عن العيوب، الثغرات الأمنية، ومشاكل القابلية للصيانة يقدّم هذا العمل دراسة حالة شاملة (دراسة في حالة الفن) حول المقاربات والأدوات الخاصة بالتحليل الساكن في بيئة تطبيقات أنغولار. حيث يستعرض الأدوات التقليدية مثل إي.إس.لينت و سوناركيوب و فورتي فاي ، بالإضافة إلى الأدوات المخصّصة لأنغولار مثل كودلايزر و أنغولار-إي.إس.لينت . كما يتناول حدود هذه الأدوات من حيث كثرة النتائج الخاطئة (الإيجابيات الكاذبة)، ضعف التغطية لبعض الخصائص الخاصة بأنغولار، وصعوبات إدماجها في سلاسل التطوير المستمر وعمليات التشغيل. إلى جانب ذلك، يناقش العمل مساهمات مقاربات الهندسة الموجّهة بالنماذج مثل موس و فاميكس ، التي توفّر مستويات أعلى من التجريد وإمكانيات متقدمة في التصور والتحليل. تُظهر النتائج أن التحليل الساكن أصبح ضرورياً لضمان جودة مشاريع أنغولار، لكنه يظل غير كافٍ. فالأدوات الحالية ما زالت تواجه صعوبات في تحليل الترابط بين المكوّنات والقوالب، في كشف الشفرات المكرّرة أو غير المستعملة، وفي التعامل مع الثغرات الأمنية الخاصة بالإطار. أما آفاق التطوير المستقبلية فتتمثل في: اعتماد مقاربات هجينة تجمع بين التحليل الساكن والتحليل الديناميكي، تحسين قدرات تحليل القوالب في أنغولار، تعزيز الدمج في بيئات الأمن المدمج في التطوير ، وتطوير التعاون في إطار البرمجيات مفتوحة المصدر. وباختصار، يوضّح هذا العمل أن التحليل الساكن للشفرة في سياق أنغولار/تايب سكريبت قد بلغ مرحلة النضج، لكنه ما زال يتطلب مقاربات أكثر دقّة وموثوقية وتوافقاً مع الممارسات الحديثة في هندسة البرمجيات.