

## **Approche Générique pour la Migration de la Partie Client (Front end) d'Application Web : Application à la Migration de GWT vers Angular.**

Stage proposé en collaboration entre le laboratoire LIRMM (<http://www.lirmm.fr/>) et l'entreprise Berger Levraut (<https://www.berger-levraut.com/fr/>)

### **Encadrants :**

- Abdelhak-Djamel Seriai, MCF Dept Info, FDS, UM, <http://www.lirmm.fr/~seriai/>, [seriai@lirmm.fr](mailto:seriai@lirmm.fr)
- Anas Shatnawi, ingénieur de recherche chez Berger Levraut (BL), <https://www.berger-levraut.com/fr/>
- Pascal Zaragoza, Doctorant Chez BL, <https://www.berger-levraut.com/fr/>

### **Contexte**

Google Web Toolkit (GWT) est un ensemble d'outils logiciels développé par Google, permettant de créer et maintenir des applications web dynamiques mettant en œuvre JavaScript, en utilisant le langage et les outils Java. Lors de la phase de développement, l'application est écrite en Java de façon classique, dans un environnement de développement intégré Java, et peut être déboguée avec les outils Java habituels. Une fois l'application prête à être déployée, le compilateur GWT la traduit en pur JavaScript, avec support automatique et transparent pour les principaux navigateurs (Internet Explorer, Firefox, Chrome, Safari, Opera). Le code JavaScript généré utilise des techniques d'HTML dynamique et de manipulation du DOM (Document Object Model) pour les aspects dynamiques de l'interface.

Angular est un cadriciel (framework) côté client, open source, basé sur TypeScript. Il est développé par Google et permet la création d'applications Web et plus particulièrement de ce qu'on appelle des « Single Page Applications » : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action. Le Framework est basé sur une architecture du type MVC et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités. Un type d'architecture qui a largement fait ses preuves et qui permet une forte maintenabilité et une amélioration du travail collaboratif.

### **Objectif et étapes**

L'objectif de ce projet TER M1 est d'implémenter une approche existante de migration de la partie front d'une application web développée en GWT vers Angular. Il est structuré en 4 étapes :

- 1) Réaliser une étude bibliographique liée à la problématique de migration d'interfaces graphiques d'applications logicielles et en particulier la migration de la partie Front des applications Web.
- 2) Implémenter l'approche existante (voir Refs. Bibliographique) de transformation d'un code source GWT vers un code source en Angular. Ceci inclut :

- Implémenter les métas-modèles GWT et Angular d'une application et analyser le code source en GWT pour pouvoir extraire le modèle de ce code source instance de ce méta-modèle et la génération du code source Angular à partir d'un modèle, instance du méta-modèle implémenté.

- Implémenter un modèle de correspondance entre les éléments des méta-modèles GWT et Angular et implémenter les procédures (algorithmes) de passage d'un modèle GWT vers un modèle Angular.

**3)** Implémenter une interface IHM Web (SaaS : Software as Service) permettant de fournir les services de migration proposés.

**4)** Proposer quelques éléments de généralisation de l'approche proposée vers la transformation d'un code JSP ou ASP vers Angular.

#### **Pourquoi choisir ce projet :**

---

- Acquérir des compétences certaines en développement logiciel de manière générale et en développement web en particulier (exemple, Angular).
- Maîtrise de plusieurs outils de compréhension de code tels que Spoon ou Java JDT. Ces outils sont très utilisés dans le monde des entreprises pour l'étape d'évolution et de maintenance de logiciels.
- Comprendre un des enjeux majeurs dans le monde du développement logiciel, à savoir pouvoir évoluer des logiciels existants.
- Découvrir plusieurs thématiques dans le monde du développement logiciel : la rétro ingénierie (reverse engineering), la réingénierie (reengineering), l'analyse du code source, les algorithmes génétiques, etc.
- Avoir une collaboration avec le monde des entreprises (l'entreprise BL, un des grands éditeurs logiciels en France).
- Etc.

Si vous avez des questions, contactez Abdelhak-Djamel Seriai  
[seriai@lirmm.fr](mailto:seriai@lirmm.fr)

#### **Références Bibliographiques**

---

- 1) Benoît Verhaeghe, Nicolas Anquetil, Stéphane Ducasse, Abderrahmane Seriai, Laurent Deruelle, Mustapha Derras: Migrating GWT to Angular 6 using MDE. SATToSE 2019
- 2) Benoît Verhaeghe, Anne Etien, Nicolas Anquetil, Abderrahmane Seriai, Laurent Deruelle, Stéphane Ducasse, Mustapha Derras: GUI Migration using MDE from GWT to Angular 6: An Industrial Case. SANER 2019: 579-583
- 3) GWT (Google Web Toolkit) - Développez des Applications Internet Riches (RIA) en Java. Damien Picard. Octobre 2010
- 4) Angular : Développez vos applications web avec le framework JavaScript de Google. Daniel Djordjevic, Sébastien Ollivier, William Klein. Editions ENI. Août 2017.
- 5) Google Web Toolkit : [https://fr.wikipedia.org/wiki/Google\\_Web\\_Toolkit](https://fr.wikipedia.org/wiki/Google_Web_Toolkit)
- 6) Spoon - Source Code Analysis and Transformation for Java : <http://spoon.gforge.inria.fr/>
- 7) AST : [https://www.eclipse.org/articles/article.php?file=Article-JavaCodeManipulation\\_AST/index.html](https://www.eclipse.org/articles/article.php?file=Article-JavaCodeManipulation_AST/index.html)
- 8) Parseur AST dans JDT : <http://www.programcreek.com/2011/01/a-complete-standalone-example-of-astparser/>
- 9) AST View : <http://www.eclipse.org/jdt/ui/astview/index.php>
- 10) Patron visiteur : [https://fr.wikipedia.org/wiki/Visiteur\\_\(patron\\_de\\_conception\)](https://fr.wikipedia.org/wiki/Visiteur_(patron_de_conception))