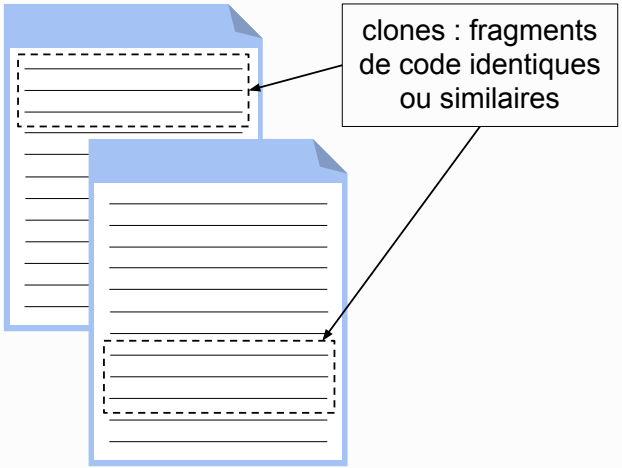
Contributions à l'usage des détecteurs de clones pour des tâches de maintenance logicielle

Alan Charpentier

Duplication de code et maintenance logicielle

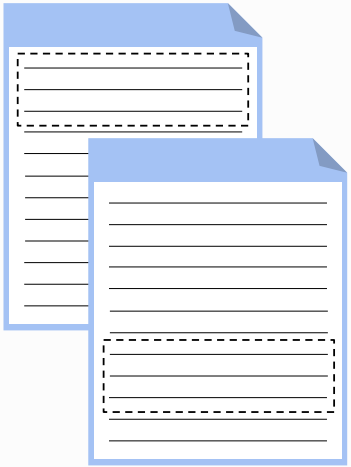
Duplication de code et maintenance



» Elle résulte principalement de l'utilisation de copier-coller.

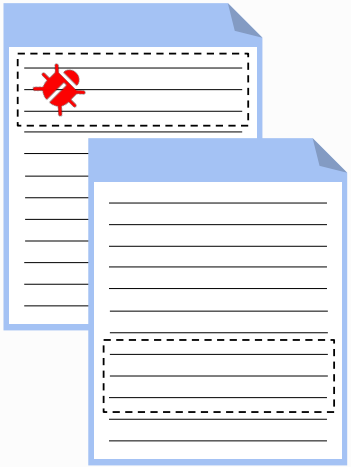
⚠ Elle peut compliquer la maintenance logicielle.

Duplication de code et maintenance corrective



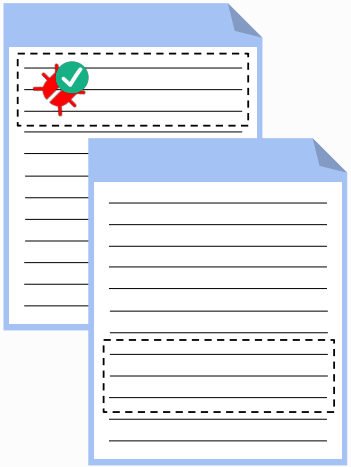
» **Tâche** : correction d'un bogue.

Duplication de code et maintenance corrective



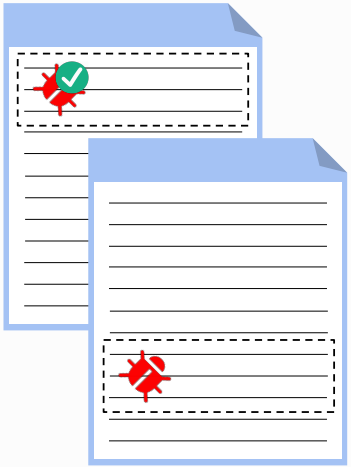
» Identification d'un bogue.

Duplication de code et maintenance corrective



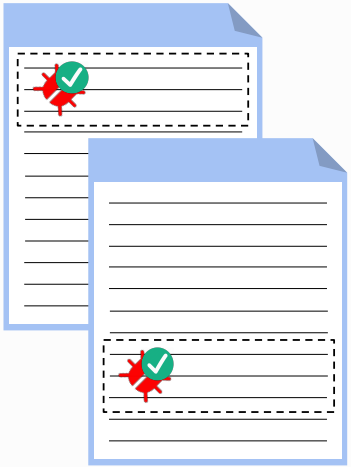
- Identification d'un bogue.
- Correction de ce bogue.

Duplication de code et maintenance corrective



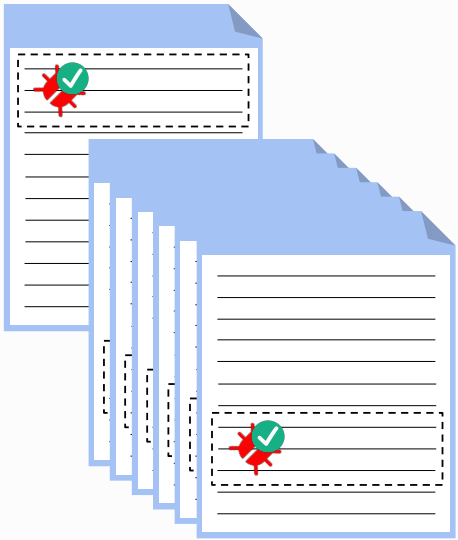
- » Identification d'un bogue.
- » Correction de ce bogue.
- ⚠ Dans quelles autres parties du code propager cette correction ?

Duplication de code et maintenance corrective



» Nécessité de propager la correction dans tous les fragments dupliqués.

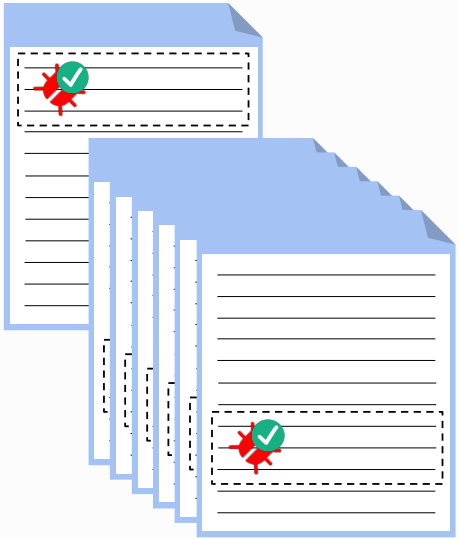
Duplication de code et maintenance corrective



» Nécessité de propager la correction dans tous les fragments dupliqués.

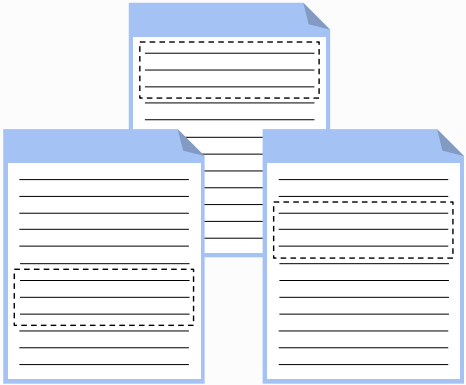
⚠ Ces fragments dupliqués peuvent être très nombreux et ne sont pas connus.

Duplication de code et maintenance corrective



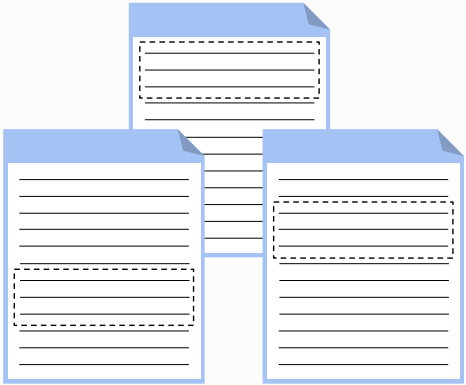
$\Longrightarrow$ Nécessité de pouvoir identifier la duplication de code pour faciliter la maintenance corrective.

Duplication de code et maintenance perfective



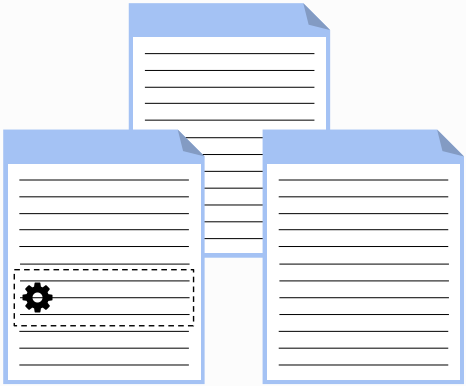
» **Tâche** : réingénierie.

Duplication de code et maintenance perfective



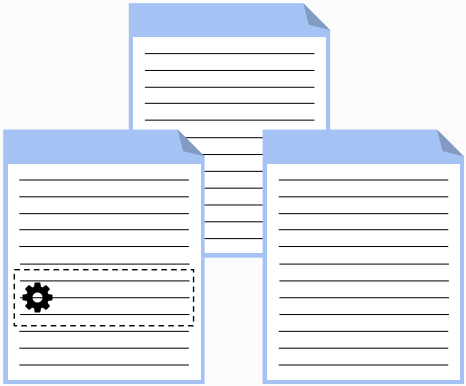
» Identification de plusieurs fragments dupliqués.

Duplication de code et maintenance perfective



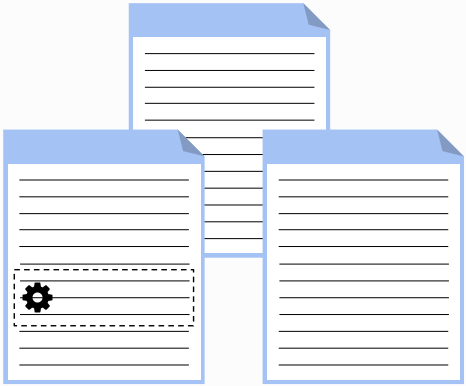
- Identification de plusieurs fragments dupliqués.
- Réingénierie.

Duplication de code et maintenance perfective



- » Identification de plusieurs fragments dupliqués.
- » Réingénierie.
- ⚠ Travail supplémentaire si tous les fragments dupliqués ne sont pas identifiés.

Duplication de code et maintenance perfective



$\Longrightarrow$ Nécessité de pouvoir identifier la duplication de code pour faciliter la maintenance perfective.

Détecteurs de clones

Objectif : Rechercher des **fragments** de code **identiques** ou **similaires**.

- Différentes approches :
  - Textuelle, lexicale, syntaxique et sémantique.
- Complexités croissantes ;
- Différents types de clone identifiés ;
- Définir le domaine de comparaison avec des seuils :
  - Taille et similarité des fragments.

Détecteurs de clones et maintenance

- Comment se comportent les détecteurs de clones dans des tâches de maintenance logicielle ?
- Exemple : réingénierie.

Cas d'utilisation

```
public TreeMap(Tree tree) {
  trees = new TIntObjectHashMap<Tree>();
  for (Tree t: tree.getTrees())
    trees.put(t.getId(), t);
}
```

```
public ActionGenerator(Tree src, Tree dst,
                       MappingStore mappings) {
  this.origSrc = src;
  this.newSrc = this.origSrc.deepCopy();
  this.origDst = dst;

  origSrcTrees = new TIntObjectHashMap<Tree>();
  for (Tree t: origSrc.getTrees())
    origSrcTrees.put(t.getId(), t);

  cpySrcTrees = new TIntObjectHashMap<Tree>();
  for (Tree t: newSrc.getTrees())
    cpySrcTrees.put(t.getId(), t);

  [...]
}
```

Cas d'utilisation

```
public TreeMap(Tree tree) {
  trees = new TIntObjectHashMap<Tree>();
  for (Tree t: tree.getTrees())
    trees.put(t.getId(), t);
}
```

```
public ActionGenerator(Tree src, Tree dst,
                       MappingStore mappings) {
  this.origSrc = src;
  this.newSrc = this.origSrc.deepCopy();
  this.origDst = dst;

  origSrcTrees = new TIntObjectHashMap<Tree>();
  for (Tree t: origSrc.getTrees())
    origSrcTrees.put(t.getId(), t);

  cpySrcTrees = new TIntObjectHashMap<Tree>();
  for (Tree t: newSrc.getTrees())
    cpySrcTrees.put(t.getId(), t);

  [...]
}
```

Cas d'utilisation

```
public TreeMap(Tree tree) {
    X = new TIntObjectHashMap<Tree>();
    for (Tree t: Y.getTrees())
        X.put(t.getId(), t);
}
```

```
public ActionGenerator(Tree src, Tree dst,
                       MappingStore mappings) {
    this.origSrc = src;
    this.newSrc = this.origSrc.deepCopy();
    this.origDst = dst;

    X = new TIntObjectHashMap<Tree>();
    for (Tree t: Y.getTrees())
        X.put(t.getId(), t);

    X = new TIntObjectHashMap<Tree>();
    for (Tree t: Y.getTrees())
        X.put(t.getId(), t);

    [...]
}
```

Cas d'utilisation

```
public TreeMap(Tree tree) {
  trees = new TIntObjectHashMap<Tree>();
  for (Tree t: tree.getTrees())
    trees.put(t.getId(), t);
}
```

```
public ActionGenerator(Tree src, Tree dst,
                       MappingStore mappings) {
  this.origSrc = src;
  this.newSrc = this.origSrc.deepCopy();
  this.origDst = dst;

  origSrcTrees = new TIntObjectHashMap<Tree>();
  for (Tree t: origSrc.getTrees())
    origSrcTrees.put(t.getId(), t);

  origSrcTrees = (new TreeMap(origSrc)).getTrees();

  cpySrcTrees = new TIntObjectHashMap<Tree>();
  for (Tree t: newSrc.getTrees())
    cpySrcTrees.put(t.getId(), t);

  cpySrcTrees = (new TreeMap(newSrc)).getTrees();

  [...]
}
```

Recherche de clones

Premier défi : configurer le détecteur de clones.

- NiCad[1] :
  - *MinLine* ;
  - *MaxLine* ;
  - *UPI* : pourcentage d'items uniques (un item peut représenter un ou plusieurs tokens) ;
  - *rename* : *none*, *blind*, *consistent* ;
  - *abstract* : *none*, *expression*, *condition*, …

1. Roy, C. et Cordy, J., *NICAD : Accurate Detection of Near-Miss Intentional Clones Using Flexible Pretty-Printing and Code Normalization*, ICPC 2008

Limites des détecteurs de clones (1)

- Les paramètres des détecteurs de clones ne sont pas orientés métier ;
- Quelles valeurs pour :
  - réingénierie ?
  - co-évolution ?

Limites des détecteurs de clones (1)

- Les paramètres des détecteurs de clones ne sont pas orientés métier ;
- Quelles valeurs pour :
  - réingénierie ?
  - co-évolution ?

$\Longrightarrow$ Difficulté à configurer un outil pour une tâche de maintenance donnée.

Recherche de clones

Second défi : utiliser les résultats du détecteur de clones.

Taille	Similarité	Renommage	Nombre de clones [1]
3	100%	*blind*	2 228
3	100%	*consistent*	2 171

1. Taille du logiciel en entrée : environ 5 000 lignes de code.

Recherche de clones

Second défi : utiliser les résultats du détecteur de clones.

```
trees = new TIntObjectHashMap<Tree>();
for (Tree t: tree.getTrees())
   trees.put(t.getId(), t);
```

```
cpySrcTrees = new TIntObjectHashMap<Tree>();
for (Tree t: newSrc.getTrees())
   cpySrcTrees.put(t.getId(), t);
```

```
double[][] matrix = new double[lstSrcs.size()][lstDsts.size()];
for (int i = 0; i < lstSrcs.size(); i++)
   for (int j = 0; j < lstDsts.size(); j++)
```

```
boolean[] dstMarks = new boolean[hDsts.size()];
for (int i = 0; i < hSrcs.size(); i++) {
   for (int j = 0; j < hDsts.size(); j++) {
```

Recherche de clones

Second défi : utiliser les résultats du détecteur de clones.

Vrai positif

```
trees = new TIntObjectHashMap<Tree>();
for (Tree t: tree.getTrees())
   trees.put(t.getId(), t);
```

```
cpySrcTrees = new TIntObjectHashMap<Tree>();
for (Tree t: newSrc.getTrees())
   cpySrcTrees.put(t.getId(), t);
```

Faux positif

```
double[][] matrix = new double[lstSrcs.size()][lstDsts.size()];
for (int i = 0; i < lstSrcs.size(); i++)
   for (int j = 0; j < lstDsts.size(); j++)
```

```
boolean[] dstMarks = new boolean[hDsts.size()];
for (int i = 0; i < hSrcs.size(); i++) {
   for (int j = 0; j < hDsts.size(); j++) {
```

Limites des détecteurs de clones (2)

- Certains clones sont non pertinents ;
- Trop de clones identifiés pour les vérifier tous.

Limites des détecteurs de clones (2)

- Certains clones sont non pertinents ;
- Trop de clones identifiés pour les vérifier tous.

$\Longrightarrow$ Les résultats des détecteurs de clones ne sont pas directement utilisables.

Clones : état de l'art

Définition(s) de clones

- Problème de recherche sans définition formelle du sujet d'étude :
  - Des clones représentent des fragments de code identiques ou similaires.
- Plusieurs définitions dans la littérature :
  - La plus répandue : taxonomie de Bellon.

Taxonomie de Bellon

» 3 types pour représenter des clones syntaxiques.

Taxonomie de Bellon

Clone de type-1

» Ils représentent des fragments identiques à l'exception des commentaires et de la mise en page.

Taxonomie de Bellon

Clone de type-1

```java
boolean obsoleteForm = (new Random()).nextBoolean();
if (obsoleteForm) {
  System.out.println("Zenzizenzizenzic");
} else {
  System.out.println("x^8");
}
```

```java
boolean obsoleteForm = (new Random()).nextBoolean();
if (obsoleteForm) {
  System.out.println("Zenzizenzizenzic");
} else {
  System.out.println("x^8");
}
```

Taxonomie de Bellon

Clone de type-2

» Ils représentent des fragments identiques à l'exception des types et des identificateurs.

Taxonomie de Bellon

Clone de type-2

```
boolean obsoleteForm = (new Random()).nextBoolean();
if ( obsoleteForm ) {
  System.out.println("Zenzizenzizenzic");
} else {
  System.out.println("x^8");
}
```

```
boolean outofdateForm = (new Random()).nextBoolean();
if ( outofdateForm ) {
  System.out.println("Zenzizenzizenzic");
} else {
  System.out.println("x^8");
}
```

» Renommage de la variable `obsoleteForm` en `outofdateForm`.

Taxonomie de Bellon

Clone de type-3

- Ils représentent des fragments avec des modifications autorisées :
  - instructions ajoutées ;
  - instructions supprimées ;
  - instructions modifiées.

Taxonomie de Bellon

Clone de type-3

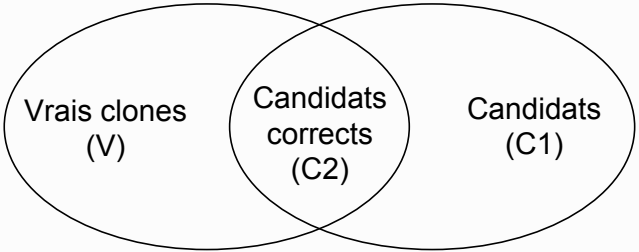
```
boolean obsoleteForm = (new Random()).nextBoolean();
if ( obsoleteForm ) {
  System.out.println("Zenzizenzizenzic");
} else {
  System.out.println("x^8");
}
```

```
Random random = new Random();
boolean outofdateForm = random.nextBoolean();
if ( outofdateForm ) {
  System.out.println("Zenzizenzizenzic");
} else {
  System.out.println("x^8");
}
```

» Extraction de la déclaration de l'objet `random`.

Benchmark de clones

- Ensemble de vrais clones ;
- Support pour calculer la précision et le rappel des détecteurs de clones.



$$Précision = \frac{|C_2|}{|C_1|}$$

$$Rappel = \frac{|C_2|}{|V|}$$

Synthèse

- Nombreuses techniques de détection de clones ;
- Nombreux détecteurs de clones ;
- Résultats des détecteurs de clones pas directement utilisables pour des tâches de maintenance.
- *Benchmarks* pour comparer et évaluer les détecteurs de clones.

Plan

Usage des détecteurs de clones dans des tâches de maintenance.

1. Méthodologie pour comparer et évaluer les détecteurs de clones ;
   - » Correction des *benchmarks* non évaluée.
2. Outils spécialisés dans une tâche.
   - » Les détecteurs de clones actuels sont généralistes.

Méthodologie pour comparer et évaluer les détecteurs de clones

Évaluation d'un *benchmark* de clones

Problématique : Les *benchmarks* de clones sont-ils fiables ?

- *Benchmark* de Bellon :
  - Le plus répandu dans la littérature ;
  - 467 citations.

Benchmark de Bellon [1]

- Bellon a classifié 2% d'un ensemble de 325 935 clones ;
- Corpus de 4 096 vrais clones :
  - Clones de type-1, type-2 et type-3 ;
  - Clones non liés à une tâche ;
  - Fragments syntaxiquement complets ;
  - Fragments avec au moins 6 lignes de code.

1. Bellon, S., Koschke, R., Antoniol, G., Krinke, J. et Merlo, E. *Comparison and evaluation of clone detection tools*. IEEE TSE 2007.

Limites du *benchmark* de Bellon

- Bellon a décidé seul les clones à inclure dans le corpus ;
- Bellon n'était pas un expert des projets.

Approche [1]

- Expérience contrôlée avec 9 groupes de 2 participants ;
- Évaluation d'un sous-ensemble du corpus de Bellon ;
- Sélection de 120 clones pour chaque groupe ;
- Classification des clones en vrais et faux positifs.

1. **Charpentier, A.,** Falleri, J.-R., Lo, D. et Réveillère, L., *An Empirical Assessment of Bellon's Clone Benchmark*, EASE 2015

Questions de recherche

1. Les chercheurs peuvent-ils avoir confiance dans le corpus de Bellon ?
2. Les mesures dérivées du corpus de Bellon sont-elles fiables ?

Expérimentation

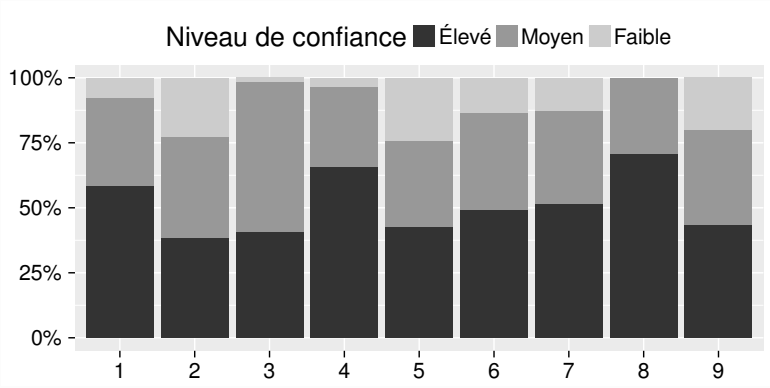
- Sélection aléatoire de 1 080 clones ;
- Répartition aléatoire des participants en 9 groupes ;
- Répartition aléatoire des 1 080 clones en 9 groupes ;
- Avis des participants :
  - *Oui* pour les clones jugés comme étant des vrais positifs ;
  - *Non* pour les autres.
- Provenance des clones inconnue des participants.

Niveau de confiance

- 3 avis pour chacun des 1 080 clones ;
- Niveau de confiance :
  - 3 avis positifs $\Rightarrow$ *élevé* ;
  - 2 avis positifs $\Rightarrow$ *moyen* ;
  - 1 avis positif $\Rightarrow$ *faible* ;
- Définition d'une échelle ordinale : *faible* $<$ *moyen* $<$ *élevé*.

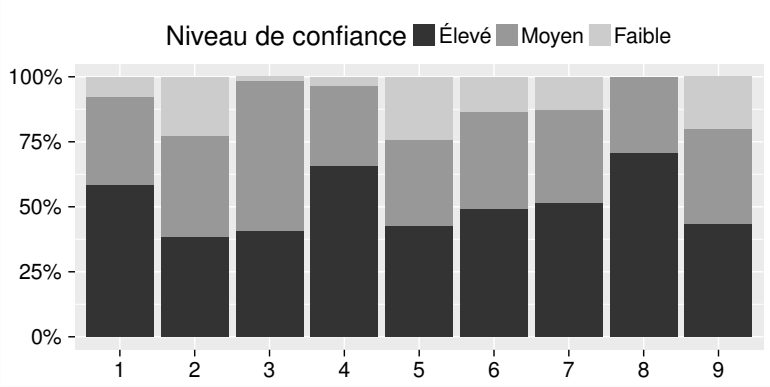
Confiance dans les clones

Échantillon de 1 080 clones



Confiance dans les clones

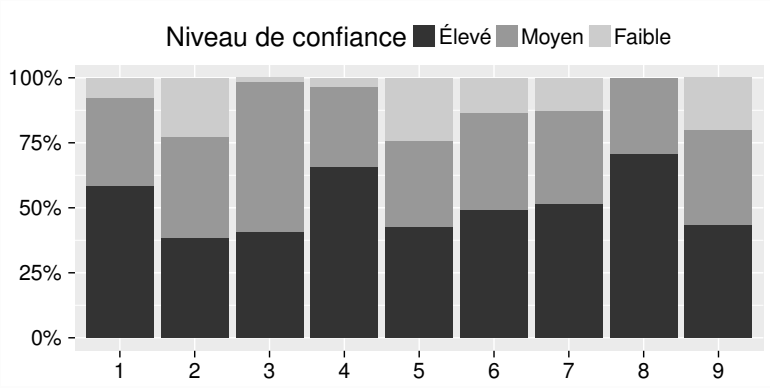
Échantillon de 1 080 clones



» Plus de la moitié des clones n'ont pas un niveau de confiance élevé.

Confiance dans les clones

Échantillon de 1 080 clones



- Plus de la moitié des clones n'ont pas un niveau de confiance élevé.
- Entre 10 et 20% des clones ont un niveau de confiance faible.

Confiance dans les clones

Corpus de Bellon

	Intervalle de confiance (95%)	
Niveau de confiance	**Borne inférieure**	**Borne supérieure**
faible	10%	14%
moyen	34%	40%
élevé	48%	54%

Synthèse

- Un nombre significatif des clones du corpus de Bellon ne sont pas fiables ;
- Les résultats dérivés du corpus de Bellon doivent être interprétés avec précaution ;
- *Benchmark* de Bellon : 467 citations !

- Les *benchmarks* construits par une seule personne ne sont pas fiables ;
- Les *benchmarks* construits sans objectif particulier ne sont pas fiables.

Amélioration des *benchmarks*

Problématique : Comment construire des *benchmarks* fiables ?

- Clones liés à une tâche ;
- Plusieurs juges ;
- Expert.

Approche [1]

- Expérience contrôlée avec plusieurs juges ;
- Sélection du même ensemble de clones pour tous les juges ;
- Classification des clones en vrais et faux positifs pour une tâche donnée ;
- Recherche de recommandations pour construire des *benchmarks* plus fiables.

1. **Charpentier, A.**, Falleri, J.-R., Morandat, F., Ben Hadj Yahia, E. et Réveillère, L., *Raters' reliability in clone benchmarks construction*, ESE

Questions de recherche

1. Les réponses des juges sont-elles cohérentes dans le temps ?
2. Les juges externes sont-ils d'accord les uns avec les autres et avec l'expert d'un projet ?
3. Quelles sont les caractéristiques des clones qui influencent l'accord entre juges externes et experts ?
   - » Caractéristiques des clones : *type*, *taille* et *distance* ;
   - » Projet ;
   - » Vrais/faux positifs.

Expérimentation

Projets

- 2 projets Java :

Projet	Fichiers Java	Lignes de code Java
FastR	343	54 511
GumTree	77	4 750

- 4 juges par projet :
  - 1 expert ;
  - 3 juges externes.

Expérimentation

Détecteur de clones et configuration

- Configuration [1] qui maximise le rappel pour avoir une vue d'ensemble de tous les clones présents dans les projets.
- iClones :
  - Facilement accessible pour répliquer l'étude ;
  - Seul détecteur de clones réputé supportant les génériques de Java ;

1. Wang, T., Harman, M., Jia, Y. et Krinke, J., *Searching for better configurations : A rigorous approach to clone evaluation*, ESEC/FSE 2013

Expérimentation

Sélection des clones

Projet	Clones
FastR	49 911
GumTree	216

$\Longrightarrow$ Nécessité de sélectionner un sous-ensemble.

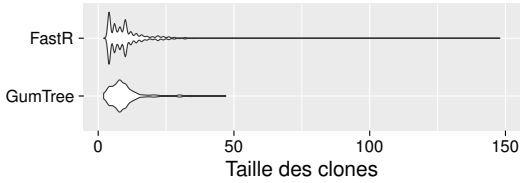
Expérimentation

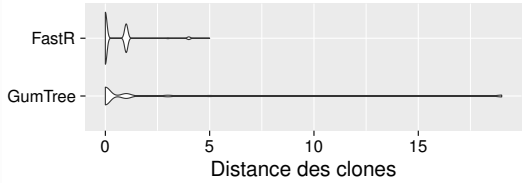
Sélection des clones

- Sélection des clones par rapport aux caractéristiques à étudier :
  - type, taille et distance.
- Création de 2 groupes pour chaque caractéristique :
  - type : *Type-2* et *Type-3* ;
  - taille et distance : pas de définition naturelle de groupe.

Expérimentation

Sélection des clones





- Distribution décalée à droite de la médiane avec une longue queue ;
- Séparation des clones en 2 groupes : la tête et la queue ;
- Production de groupes stables avec l'algorithme *neural-gas* :
  - taille : petits clones / grands clones ;
  - distance : clones proches / clones éloignés.

Expérimentation

Sélection des clones

- 6 groupes ;
- Sélection de 50 clones par groupe.

Projet	Clones	Clones sélectionnés
FastR	49 911	300
GumTree	216	234

Expérimentation

Collecte des données

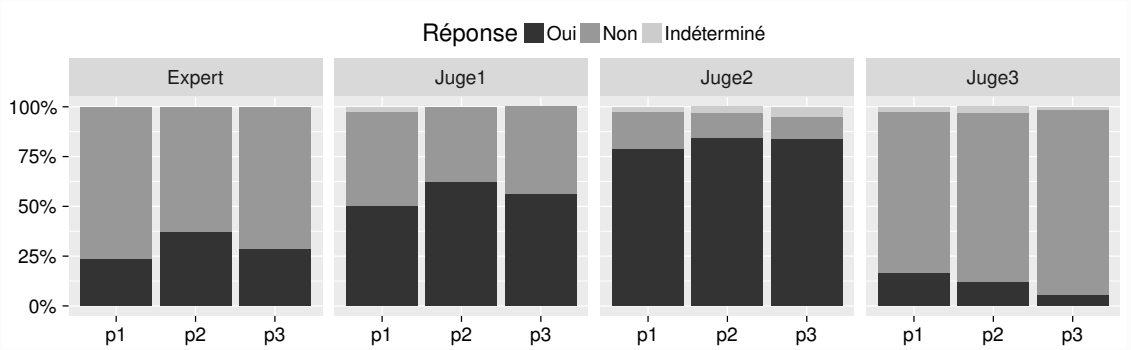
- *Ce clone est-il utile pour une activité de co-évolution ou de réingénierie ?*
  - *Oui*, *Non*, *Indéterminé* (réservée aux juges externes).
- Sélection des clones non connue des participants.

Stabilité des réponses des juges

Hypothèse : Comme l'ordre des clones est aléatoire, la proportion de réponses *oui*, *non* et *indéterminé* devrait rester stable pendant toute l'expérience.

Stabilité des réponses des juges

Stabilité des réponses des juges



$\Longrightarrow$ La classification de clones ne semble pas induire d'effet de fatigue.

Régularité des réponses des juges

Hypothèse : Un même clone présenté plusieurs fois au même juge devrait toujours être classé de la même manière.

» Utilisation des 65 doublons contenus dans les clones du projet GumTree.

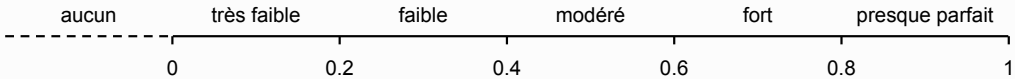
Réponses inconsistantes

Juge	Nombre
expert	1
juge1	11
juge2	3
juge3	7

$\Longrightarrow$ Une session d'entraînement serait bénéfique aux juges externes.

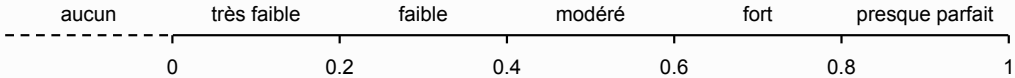
Accord entre juges externes

	FastR		GumTree	
Juges	κ	**Accord**	κ	**Accord**
juges externes	-0,12	aucun	-0,05	aucun



Accord entre juges externes et experts

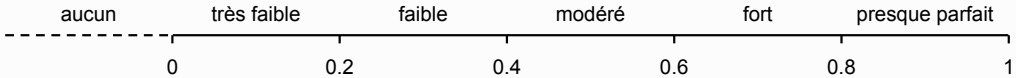
	FastR		GumTree	
Juges	κ	**Accord**	κ	**Accord**
expert et juge1	0,24	faible	0,42	modéré
expert et juge2	0,09	très faible	0,13	très faible
expert et juge3	0,16	très faible	0,28	faible



Accord entre juges externes et experts

Juges	FastR		GumTree	
	κ	Accord	κ	Accord
expert et juge1	0,24	faible	0,42	modéré
expert et juge2	0,09	très faible	0,13	très faible
expert et juge3	0,16	très faible	0,28	faible

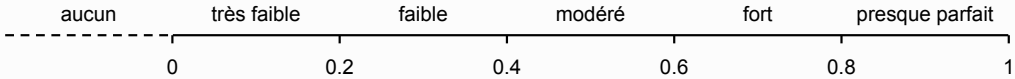
» Pas d'accord élevé.



Accord entre juges externes et experts

	FastR		GumTree	
Juges	κ	**Accord**	κ	**Accord**
expert et juge1	0,24	faible	0,42	modéré
expert et juge2	0,09	très faible	0,13	très faible
expert et juge3	0,16	très faible	0,28	faible

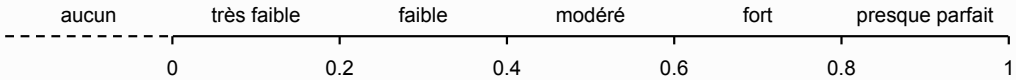
- Pas d'accord élevé.
- Le choix d'un juge externe a un impact.



Accord entre juges externes et experts

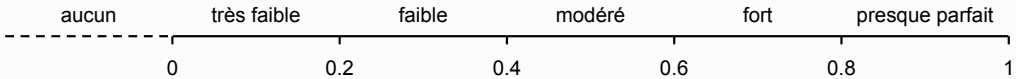
	FastR		GumTree	
Juges	κ	**Accord**	κ	**Accord**
expert et juge1	0,24	faible	0,42	modéré
expert et juge2	0,09	très faible	0,13	très faible
expert et juge3	0,16	très faible	0,28	faible

- Pas d'accord élevé.
- Le choix d'un juge externe a un impact.
- Le choix d'un projet a un impact.



Accord entre juges externes et experts

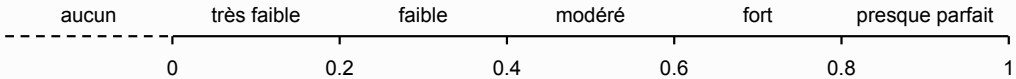
	FastR		GumTree	
Juges	κ	**Accord**	κ	**Accord**
expert et juge1	0,24	faible	0,42	modéré
expert et juge2	0,09	très faible	0,13	très faible
expert et juge3	0,16	très faible	0,28	faible
expert et majorité	**0,25**	faible	**0,44**	modéré



Accord entre juges externes et experts

Juges	FastR		GumTree	
	κ	Accord	κ	Accord
expert et juge1	0,24	faible	0,42	modéré
expert et juge2	0,09	très faible	0,13	très faible
expert et juge3	0,16	très faible	0,28	faible
expert et majorité	**0,25**	faible	**0,44**	modéré

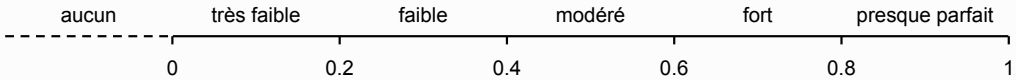
» Toujours pas d'accord élevé.



Accord entre juges externes et experts

Juges	FastR		GumTree	
	κ	Accord	κ	Accord
expert et juge1	0,24	faible	0,42	modéré
expert et juge2	0,09	très faible	0,13	très faible
expert et juge3	0,16	très faible	0,28	faible
expert et majorité	**0,25**	faible	**0,44**	modéré

- Toujours pas d'accord élevé.
- Vote de la majorité toujours une meilleure stratégie.



Accord entre juges

$\Longrightarrow$ Utiliser un seul juge externe pour évaluer des clones n'est pas fiable ;

$\Longrightarrow$ Utiliser le vote de la majorité est un moyen d'atténuer le faible accord entre juges.

Impact de caractéristiques sur l'accord

Méthodologie

Hypothèse nulle : Le nombre de réponses identiques et différente avec l'expert est indépendant de chaque caractéristique.

Hypothèse alternative : Ce nombre n'est pas indépendant.

- Test statistique du χ^2 pour évaluer nos hypothèses ;
- V de Cramer pour mesurer la taille de l'effet ;
- Utilisation du vote de la majorité.

Impact de caractéristiques sur l'accord

Observations

- *Taille* et *Projet* : seules caractéristiques avec un effet sur les 2 projets ;
- *Taille* :
  - Effet modéré à fort ;
- *Projet* :
  - Effet faible ;

Impact de caractéristiques sur l'accord

Observations

- *Taille* et *Projet* : seules caractéristiques avec un effet sur les 2 projets ;
- *Taille* :
  - Effet modéré à fort ;
- *Projet* :
  - Effet faible ;

⟹ Les petits clones doivent être jugés par un expert ;

⟹ Les clones de certains projets ne peuvent pas être jugés par des juges externes.

Synthèse

- Identification de plusieurs recommandations pour construire des *benchmarks* plus fiables ;
- Mais pas ou peu d'accord entre plusieurs juges sur un nombre significatif de clones.

$\Longrightarrow$ Difficile de construire des *benchmarks* de clones.

$\Longrightarrow$ Les *benchmarks* de clones ne peuvent pas aider les développeurs à choisir l'outil et la configuration les plus adaptés à une tâche.

Outils spécialisés dans une tâche

Développement d'un outil spécialisé

Problématique : Des détecteurs spécialisés peuvent-ils aider les développeurs dans des tâches de maintenance ?

- Cas d'étude :
  - Identification et suppression de clones dans du code CSS.

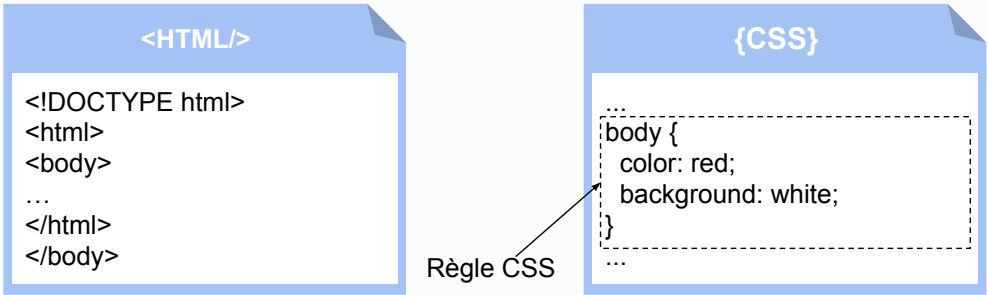
Langage CSS

- Utilisé pour décrire la présentation de documents HTML ;
- Une des principales technologies pour le développement web ;
- Utilisé par plus de 95% [1] de tous les sites web.

1. `http://w3techs.com/technologies/overview/site_element/all` (11 Octobre 2016)

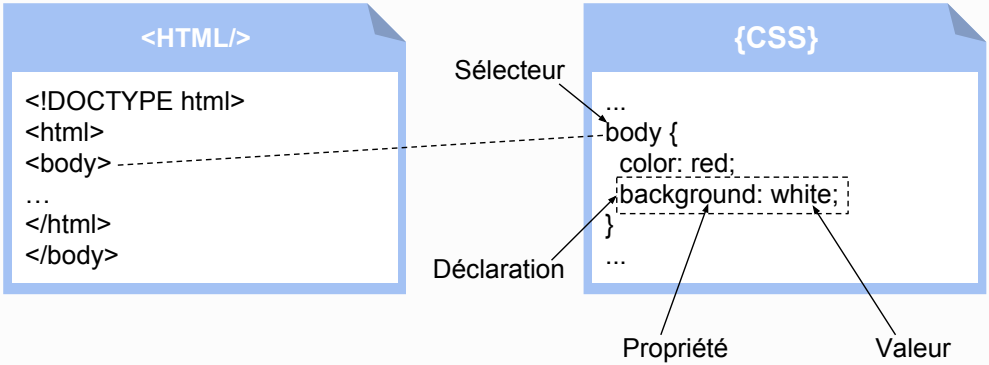
Langage CSS

» Fichier CSS : ensemble de règles ;



Langage CSS

- Fichier CSS : ensemble de règles ;
- Règle CSS :
  - Sélecteurs qui pointent sur des éléments HTML ;
  - Déclarations qui définissent le style à appliquer.



Langage CSS

- Avantage :
  - Langage simple
    - Peu de constructions : sélecteurs, propriétés, valeurs

Langage CSS

- Avantage :
  - Langage simple
    - Peu de constructions : sélecteurs, propriétés, valeurs
- Inconvénients :
  - Langage trop simple ;
  - Manque de mécanismes avancés pour éviter la duplication de code :
    - Mauvais pour la réutilisation ;
    - Mauvais pour la maintenance ;
    - Mauvais pour l'évolution.

Langage CSS

- Avantage :
  - Langage simple
    - Peu de constructions : sélecteurs, propriétés, valeurs
- Inconvénients :
  - Langage trop simple ;
  - Manque de mécanismes avancés pour éviter la duplication de code :
    - Mauvais pour la réutilisation ;
    - Mauvais pour la maintenance ;
    - Mauvais pour l'évolution.
- Préprocesseurs CSS pour faciliter le développement et la maintenance de code CSS.

Préprocesseurs CSS

- Mécanismes de réutilisation de haut-niveau : variables, fonctions, mixins ;
- Compilés en code CSS ;
- Utilisés par plus de 50% des développeurs [1] ;
- Utilisés par des projets d'envergure : Bootstrap, Foundation, …

1. Étude récente basée sur plus de 13 000 réponses de développeurs web.

Utilisation d'un mixin en Sass

» Les mixins sont l'un des principaux mécanismes de réutilisation.

{CSS}

```
div#main {
  margin: 5px;
  padding: 20px;
  font-size: 30px;
}
div#content {
  margin: 5px;
  padding: 5px;
}
```

{SASS}

```
@mixin m($p) {
  padding: $p;
}

div#main {
  @include m(20px);
  margin: 5px;
  font-size: 30px;
}
div#content {
  @include m(5px);
  margin: 5px;
}
```

Migration de code CSS

Existant

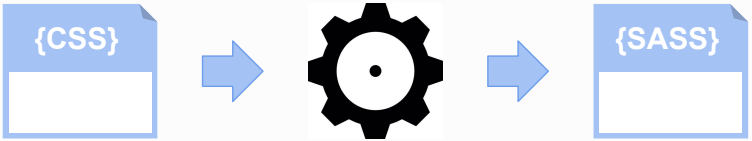
- » Recherche manuelle de mixins ;
- » Réingénierie manuelle ;

⟹ Tâche laborieuse.

Notre approche

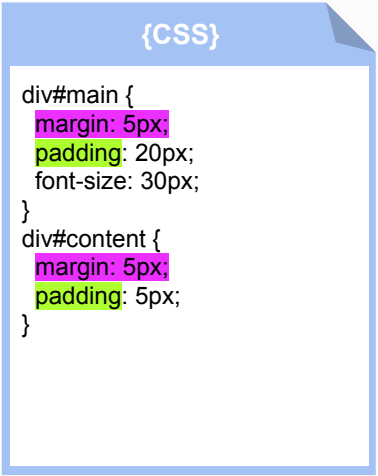
- » Extraction automatique de mixins ;
- » Génération de code Sass.

Approche [1]



1. **Charpentier, A.**, Falleri, J.-R. et Réveillère, L., *Automated Extraction of Mixins in Cascading Style Sheets*, ICSME 2016

Cas d'utilisation simple



Étape 1 : contexte formel

{CSS}

```
div#main {
  margin: 5px;
  padding: 20px;
  font-size: 30px;
}
div#content {
  margin: 5px;
  padding: 5px;
}
```

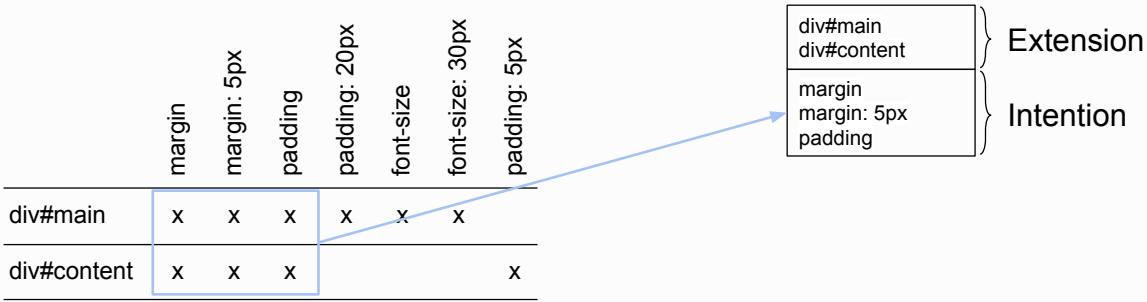
	margin	margin: 5px	padding	padding: 20px	font-size	font-size: 30px	padding: 5px
div#main	x	x	x	x	x	x	
div#content	x	x	x				x

Étape 2 : concept formel

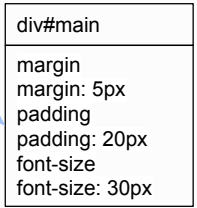
» Collection maximale d'objets partageant un nombre maximal d'attributs ;

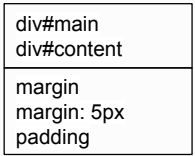
Étape 2 : concept formel

- Collection maximale d'objets partageant un nombre maximal d'attributs ;
- Concept formel ≡ (extension, intension) :
  - extension : tous les objets qui partagent les attributs de l'insention ;
  - insention : tous les attributs partagés par les objets de l'extension.



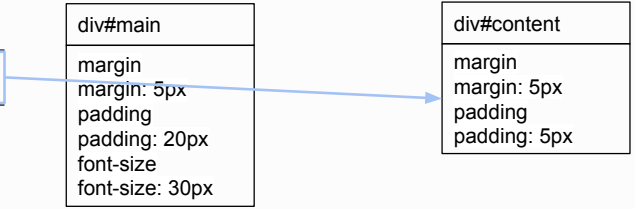
Étape 2 : identifier les concepts formels

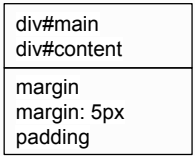




Étape 2 : identifier les concepts formels

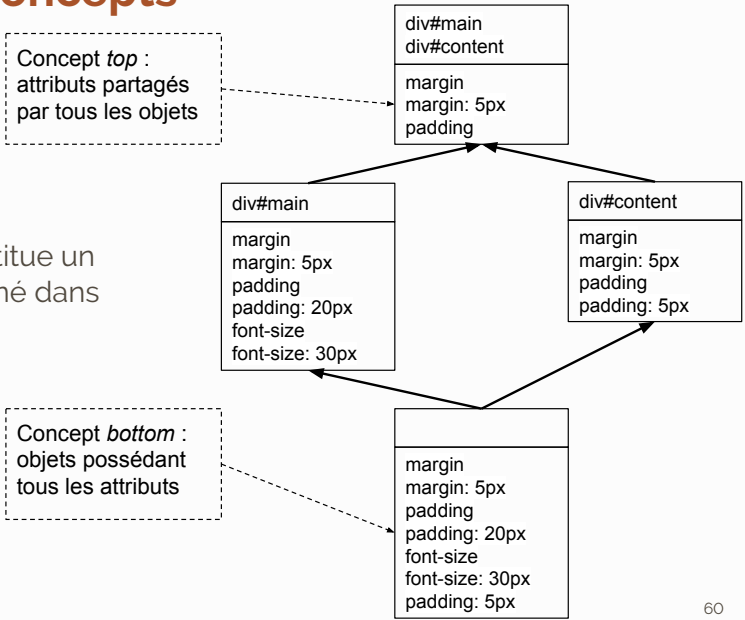
	margin	margin: 5px	padding	padding: 20px	font-size	font-size: 30px	padding: 5px
div#main	x	x	x	x	x	x	
div#content	x	x	x				x



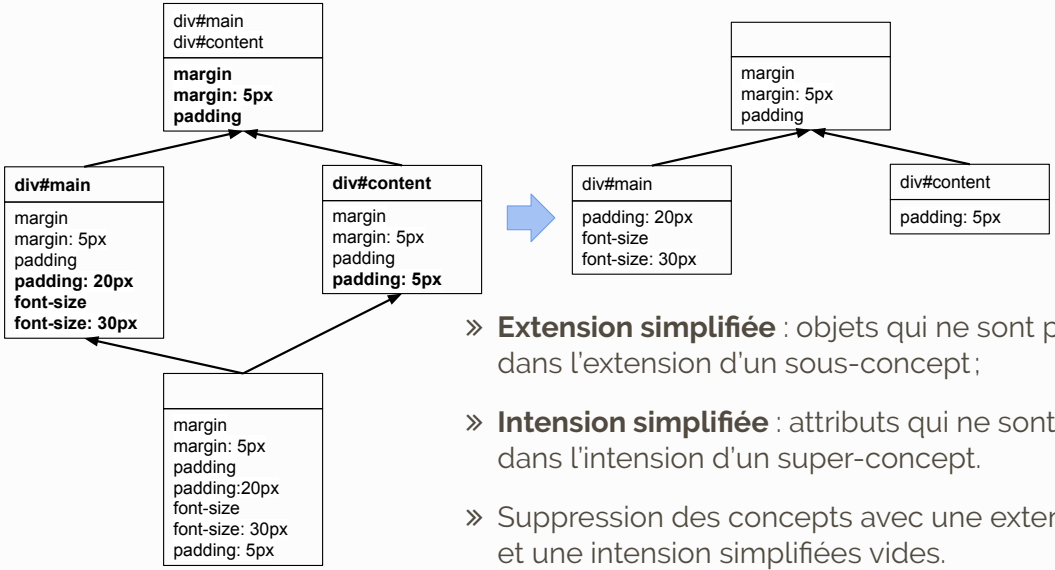


Étape 2 : treillis de concepts

» L'ensemble des concepts constitue un ordre partiel et peut être ordonné dans un treillis.

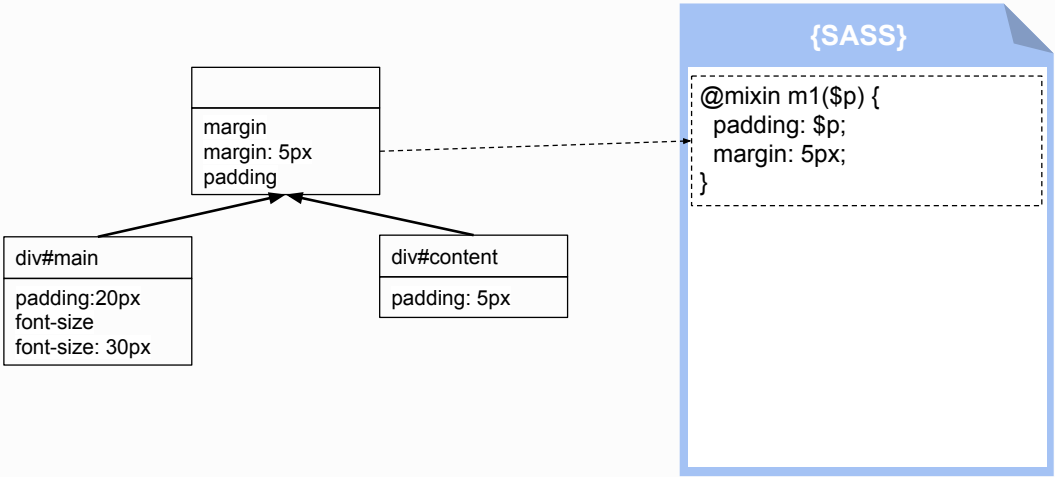


Étape 3 : sous-hiérarchie de Galois

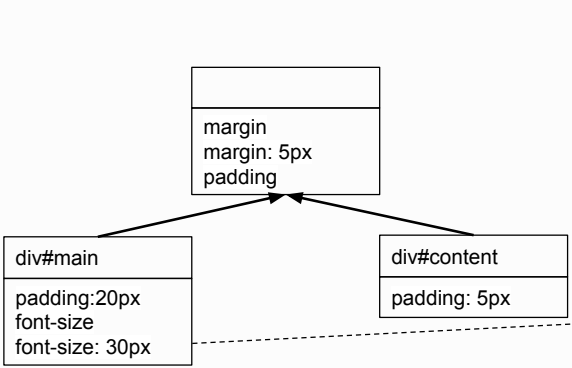


- **Extension simplifiée** : objets qui ne sont pas dans l'extension d'un sous-concept ;
- **Intension simplifiée** : attributs qui ne sont pas dans l'intension d'un super-concept.
- Suppression des concepts avec une extension et une intension simplifiées vides.

Étape 4 : mixins et règles CSS



Étape 4 : mixins et règles CSS

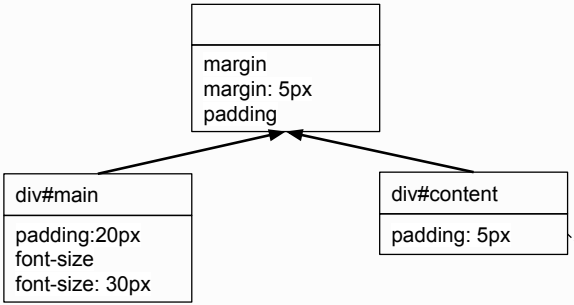


{SASS}

```
@mixin m1($p) {
  padding: $p;
  margin: 5px;
}

div#main {
  @include m1(20px);
  font-size: 30px;
}
```

Étape 4 : mixins et règles CSS



{SASS}

```
@mixin m1($p) {
  padding: $p;
  margin: 5px;
}

div#main {
  @include m1(20px);
  font-size: 30px;
}

div#content {
@include m1(5px);
}
```

Implémentation

- Outil libre ;
- Disponible sur GitHub : `https://github.com/acharpen/mocss` ;
- Contrôle sur le code généré :
  - Filtrage des mixins selon des critères métiers :
    - nombre de paramètres ;
    - nombre et type des déclarations factorisées ;
    - nombre d'utilisation.
- Évaluation :
  - Performance : moins de 2 secondes sur 108 fichiers CSS issus de vraies applications web (11 secondes au maximum) ;
  - Pertinence : études de cas.

Études de cas

Méthodologie

» Participation de 4 développeurs experts en CSS et Sass.

Observations

- ⊕ Identification de mixins pertinents ;
- ⊕ Filtrage des mixins intéressant.
- ⊖ Critères de filtrage des mixins parfois insuffisants.

Synthèse

- Approche automatisée pour l'identification de mixins dans du code CSS ;
- Contrôle fin sur le code généré pour s'adapter aux besoins des développeurs ;
- Études de cas démontrant que notre outil peut aider les développeurs à éliminer la duplication de code.

Conclusion

- **Thèse**
  - Usage des détecteurs de clones dans des tâches de maintenance logicielle.
- **Objectifs**
  - Évaluer la méthodologie pour comparer et évaluer les détecteurs de clones ;
  - Évaluer des détecteurs de clones spécialisés dans une tâche.
- **Résultats**
  - Les *benchmarks* de clones sont durs à mettre en œuvre ;
  - Un détecteur de clones spécialisé peut aider les développeurs.

Perspectives

» **Outil : spécialisation plus avancée**

Perspectives

» **Outil : spécialisation plus avancée**

» Filtrage des mixins :
  - » Nombre de paramètres ;
  - » Nombre d'utilisation ;
  - » Nombre de propriétés et déclarations factorisées.

⊕ Paramètres métiers ;

⊖ Paramètres haut-niveau.

Perspectives

» **Outil : spécialisation plus avancée**

Propriétés non liées

```
@mixin m($c, $m) {
  color: $c;
  margin: $m;
}
```

Propriétés liées

```
@mixin m($fs, $fw) {
  font-style: $fs;
  font-weight: $fw;
}
```

Perspectives

- **Outil : spécialisation plus avancée**
  - Approche semi-automatique ;
  - Adapter les outils aux particularités d'un individu / d'une équipe ;
  - Analyser le code déjà produit par un individu / une équipe.

Fiabilité des mesures dérivées du corpus de Bellon

Méthodologie

Méthodologie : Comparer des mesures de précision et rappel à partir de différents corpus.

- 2 scénarios envisageables :
  - Clones avec un niveau de confiance supérieur ou égal à *moyen* ;
  - Clones avec un niveau de confiance égal à *élevé*.
- 2 nouveaux corpus :
  - *F* contenant 954 clones.
  - *G* contenant 553 clones.

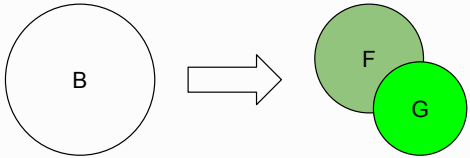
Fiabilité des mesures dérivées du corpus de Bellon

Méthodologie

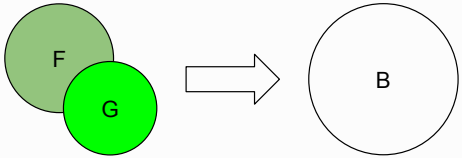
Analyse pire cas en 3 étapes :

1. Un outil fictif t trouve exactement les clones d'un corpus C_1 ;
2. t a donc une précision et un rappel de 1 sur C_1 ;
3. Le corpus n'est plus C_1 mais C_2.

» Diminution de la précision :



» Diminution du rappel :



Fiabilité des mesures dérivées du corpus de Bellon

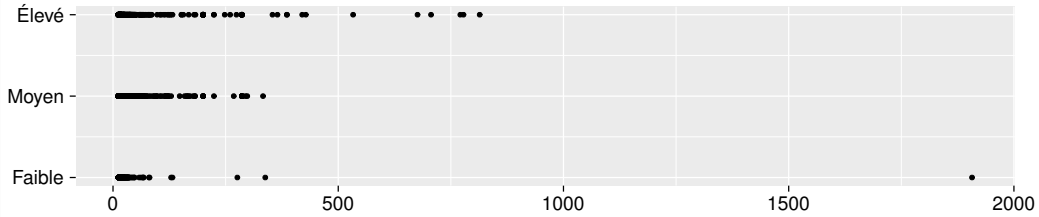
Observations

- Diminution possible de **0,12** si le niveau de confiance est fixé à **moyen** ;
- Diminution possible de **0,49** si le niveau de confiance est fixé à **élevé**.

$\Longrightarrow$ Si des mesures de précision ou rappel sont dans ces intervalles alors les résultats doivent être interprétés avec précaution.

Impact des caractéristiques sur le niveau d'accord

H_0 : Aucune corrélation entre la taille d'un clone et son niveau de confiance.
H_a : Plus la taille d'un clone est élevée plus son niveau de confiance est grand.

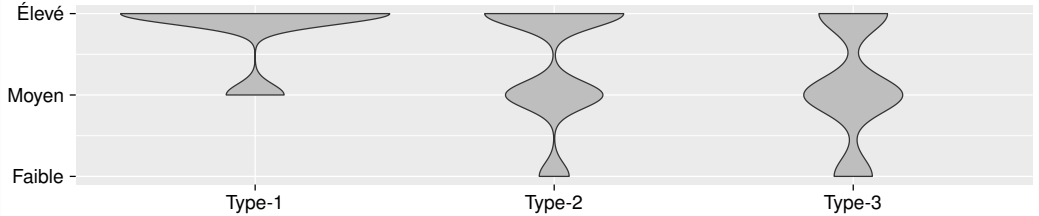


» Test de corrélation de Spearman : résultat significatif ;

» *Bootstrap* (95%) : taille de l'effet très faible.

Impact des caractéristiques sur le niveau d'accord

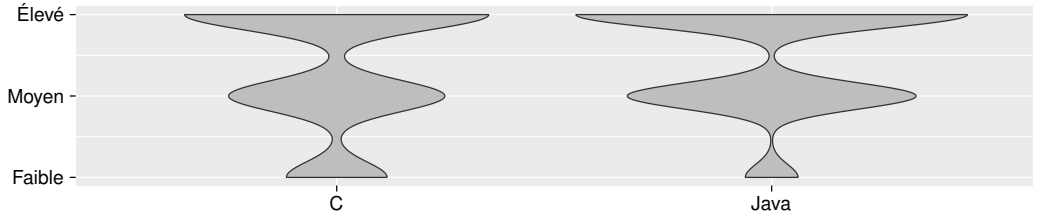
H_0 : Aucune corrélation entre le type d'un clone et son niveau de confiance.
H_a : Plus le type d'un clone est grand moins son niveau de confiance est élevé.



- Test de corrélation de Spearman : résultat significatif ;
- *Bootstrap* (95%) : taille de l'effet modérée et négative.

Impact des caractéristiques sur le niveau d'accord

H_0 : Le langage de programmation n'a pas d'impact sur le niveau de confiance.
H_a : Le langage de programmation a un impact sur le niveau de confiance.

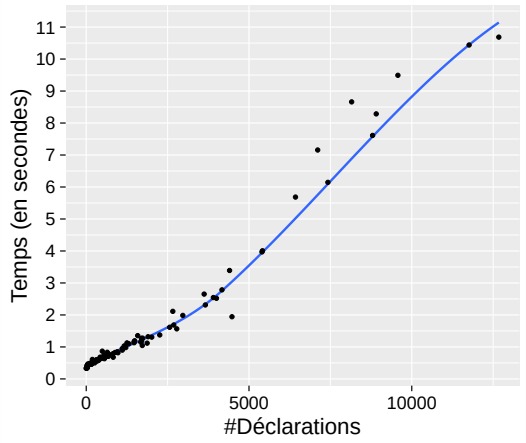


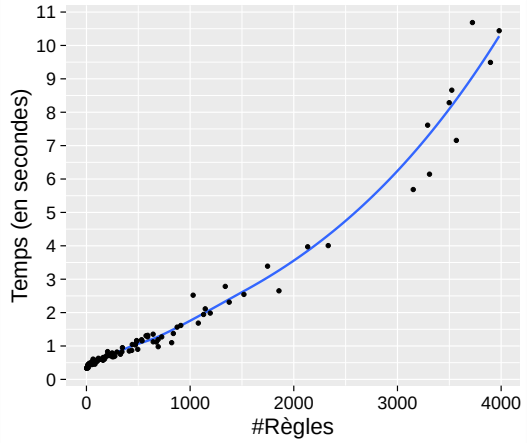
- Test de Wilcoxon-Mann-Whitney : différence significative ;
- Delta de Cliff : effet négligeable.

Évaluation des performances

- 108 fichiers CSS ;
- 10 exécutions sur chaque fichier ;
- Configuration par défaut ;
- Machine de test : 2,10Ghz Core i7-4600U avec 16GB de mémoire.

Évaluation des performances





Évaluation des seuils

- 108 fichiers CSS ;
- Valeurs allant de 0 to 10 ;
- Évaluation individuelle de chaque seuil.

Évaluation des seuils