IFT 3051 – Projet informatique

Implémentation des Micro Patterns sous Ptidej

Plan

- Présentation des Micro Patterns.
- Implémentation sous Ptidej.
- Problèmes rencontrés.
- Analyse des résultats obtenus.
- Différences avec l’article.
- Évolutions et travaux futurs envisageables.
- Conclusion.

Présentation des Micro Patterns

Les Micro Patterns sont:

- Un moyen de mesurer la qualité d'un logiciel.
- Similaires aux design patterns mais à un niveau d'abstraction inférieur.
- Détectable automatiquement.
- Regroupés en 8 catégories.

Présentation des Micro Patterns

Le micro patterns se définissent au niveau d'une classe.

Un micro pattern est un ensemble de caractéristiques que possède une classe.

Présentation des Micro Patterns

Par exemple le micro pattern « Box » défini toutes les classes avec exactement un champ d'instance modifiable par au moins une méthode statique ou non de la classe.

Un autre exemple le micro pattern « Sink » est une classe dont les méthodes n'appelle ni méthode d'instance, ni méthode statiques d'une autre classe.

Présentation des Micro Patterns

Les catégories des Micro Patterns permettent de les regrouper en assembles aux caractéristiques commune.

Par exemple la catégorie « Controlled Creation » regroupe les classes dont les constructeur font l'objet de protocoles de création spécifique.

Présentation des Micro Patterns

	Main Category	Pattern	Short description	Additional Category
Degenerate Classes	Degenerate State and Behavior	Designator	An interface with absolutely no members.	
		Taxonomy	An empty interface extending another interface.	
		Joiner	An empty interface joining two or more superinterfaces.	
		Pool	A class which declares only static final fields, but no methods.	
	Degenerate Behavior	Function Pointer	A class with a single public instance method, but with no fields.	
		Function Object	A class with a single public instance method, and at least one instance field.	
		Cobol Like	A class with a single static method, but no instance members	
	Degenerate State	Stateless	A class with no fields, other than static final ones.	
		Common State	A class in which all fields are static.	
		Immutable	A class with several instance fields, which are assigned exactly once, during instance construction.	
	Controlled Creation	Restricted Creation	A class with no public constructors, and at least one static field of the same type as the class	
		Sampler	A class with one or more public constructors, and at least one static field of the same type as the class	
Coupling	Wrappers	Box	A class which has exactly one, mutable, instance field.	
		Compound Box	A class with exactly one non primitive instance field.	
		Canopy	A class with exactly one instance field that is assigned exactly once, during instance construction.	Degenerate State
	Data Managers	Record	A class in which all fields are public, no declared methods.	Degenerate Behavior
		Data Manager	A class where all methods are either setters or getters.	
		Sink	A class whose methods do not propagate calls to any other class.	
Inheritance	Base Classes	Outline	A class where at least two methods invoke an abstract method on "this"	Degenerate State
		Trait	An abstract class which has no state.	
		State Machine	An interface whose methods accept no parameters.	Degenerate State and Behavior
		Pure Type	A class with only abstract methods, and no static members, and no fields	
		Augmented Type	Only abstract methods and three or more static final fields of the same type	
		Pseudo Class	A class which can be rewritten as an interface: no concrete methods, only static fields	
	Inheritors	Implementor	A concrete class, where all the methods override inherited abstract methods.	
		Overrider	A class in which all methods override inherited, non-abstract methods.	
		Extender	A class which extends the inherited protocol, without overriding any methods.	

Implémentation sous Ptidej

- Compréhension des micro patterns et de Ptidej.
  - Comment intégrer la détection des micro pattern à Ptidej?
  - Quelle éléments de Ptidej nous permet d'effectué la détection?
- Micro pattern et PADL.

Implémentation sous Ptidej

- Ajout d’un nouveau projet.
- Le patron de conception visiteur exécute nos algorithme de détection.
- Chaque micro pattern est détecté par une classe distincte.
- Une liste des classes par micro pattern est obtenue.

Problèmes rencontrés

- Chaque classe ne fait pas partie d'un micro pattern unique .
- listOfInheritedActors n'est pas listOfImplementedActors!
- Un modèle identique?
- Immuable, une définition imprécise...
- Les classes et interfaces héritées et ré implémentées.

Analyse des résultats obtenus

Qu’avons-nous analysé?

- Sun 1.4.2
- Scala
- MJC
- Ant
- JEdit
- Tomcat
- Poseidon
- JBoss

Analyse des résultats obtenus

- Le package « Shared »
- Quelles classes sont analysées?

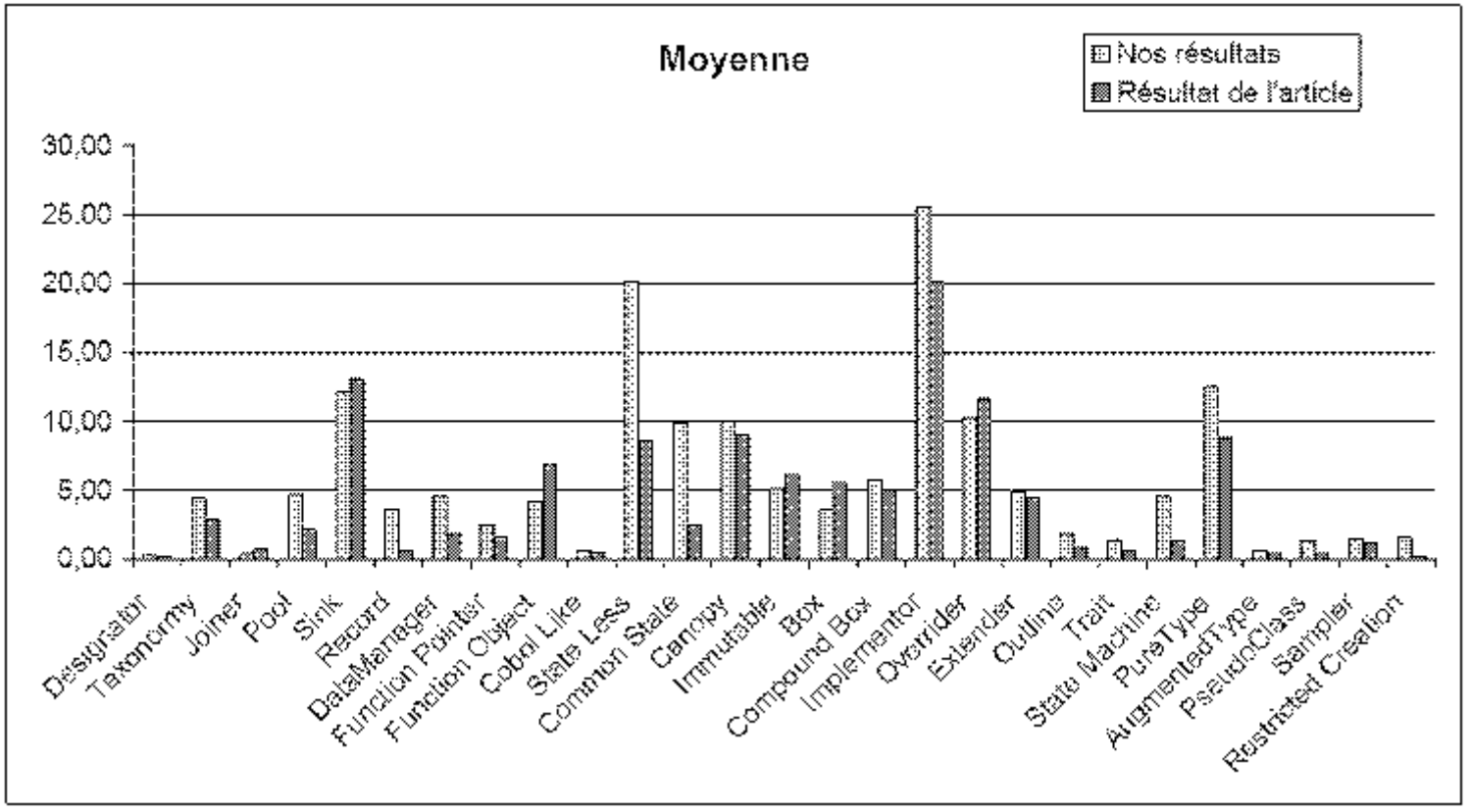
Application :	Nombre de classes que nous avons analysé :	Nombre de classes analysées par les auteurs :
Sun 1.4.2	7200	7525
Scala	802	2678
MJC	870	945
Ant	1487	421
JEdit	367	676
Tomcat	3406	1434
Poseidon	6397	8162
JBoss	4321	13623
Shared	-	5979
Total	24850	41443

Analyse des résultats obtenus

Collection	JRE 1.4.2	Struts	MJC	Ant	Jena	Tomcat	Poseidon	JBoss	Moyenne	Total	Moyenne	Total
											Article	
Designator	0.40	0.06	0.37		0,08	0,23	0.47	0.45	0.28	0.39	0.20	0.20
Taxonomy	7.61	3.62	2.21	3.03	3.27	2.36	3.43	3.78	4.48	6.69	2.90	3.50
Joiner	0.42	1.25	0.11	0.40	0,08	0,15	0.92	0.14	0.42	0.47	0.70	1.20
Pool	7.25	5.24	4.60	3.15	1.01	2.35	3.05	2.98	4.63	6.62	2.10	2.30
Sink	23.42	12.47	13.45	7,83	7,37	15,90	14,67	10,34	12,05	14,87	13,40	13,60
Record	6.19	3.48	2,50	1,69	2.72	2.78	4,42	4.78	5,57	4,49	0,60	0.80
DataManager	7.79	3.58	4.37	2.43	3.54	3.03	5.79	5.09	4,59	6.53	1,90	1.90
Function Pointer	4.39	1.12	1.84	1.94	3,27	3.29	2.63	4.05	2.51	3.25	1.60	1.60
Function Object	3.84	3.37	4.27	2.29	2.46	4.62	5.33	5.11	4.28	6.13	6.60	6.50
Cobol Like	0.85	1.09	1.81	0.94	0,27	0.56	0.28	0.13	0.64	0.64	0.50	0.60
State Less	28.59	16.09	18.21	12.64	14.71	20.89	23.06	23.26	20.14	23.13	8.60	3.90
Common State	18.64	4.61	7,04	7,26	6,56	10.21	10.55	9,67	6,91	12,32	2,50	9,80
Canopy	5.11	12.22	10,83	8,53	11.72	7.72	12,18	9.03	8,95	8,53	8,00	7.70
Immutable	3.19	1.02	4.71	7.12	10,09	6,34	3.17	4.29	6.06	6.12	6.10	3.10
Box	4.35	1.12	5.29	4.79	2,72	4.78	2.45	3.60	3.63	3.72	5.80	6.00
Compound Box	6.54	3.52	6.67	3.83	5,91	5,09	5.25	5.33	6.76	5.29	6.50	4.40
Implementor	31.33	45.86	30.69	24.08	17.17	24.40	18.78	24.11	26.54	26.23	30.20	21.20
Overrider	17.47	8.48	10.50	7.60	7.08	8,04	11.19	13.48	10.22	12.54	11.70	10.80
Extender	4.67	4.67	5,40	1.05	7.08	6.59	7.29	9,60	4,84	4,09	3,40	4,00
Outline	2.71	1.28	2.24	1.14	3.08	1.29	2,13	1.61	1,76	2.07	0,90	0.90
Trait	2.98	1.27	1.15	0.87	0,54	1,32	1.64	1.27	1.35	1.79	0.80	0.80
State Machine	4.84	7.11	4.31	6.98	1.38	3.41	4.03	3.44	4.58	4.35	1.40	1.80
PureType	10.24	3.35	6.74	19.26	5.18	14.03	18.05	15.02	10.51	14.05	6.60	3.80
AugmentedType	0.74	3.05	0.11	2.15	0,06	1.26	0.32	0.53	0.84	0.75	0.40	0.40
PseudoClass	4.14	1.37	0.40	0.47	8.54	8.75	1.89	1.07	1.53	2.05	0.40	0.40
Sampler	1.18	4.25	2.18	0.64	3.05	1.59	1.33	3.75	3,47	1,29	1,20	1,00
Restricted Creation	1.89	6.53	1.56	0,18	0.54	0.23	1,38	1.11	1,54	1,44	0,20	1.60
Coverage	93	93	74	74	[illegible]	79	93	61	77.92		71.20	

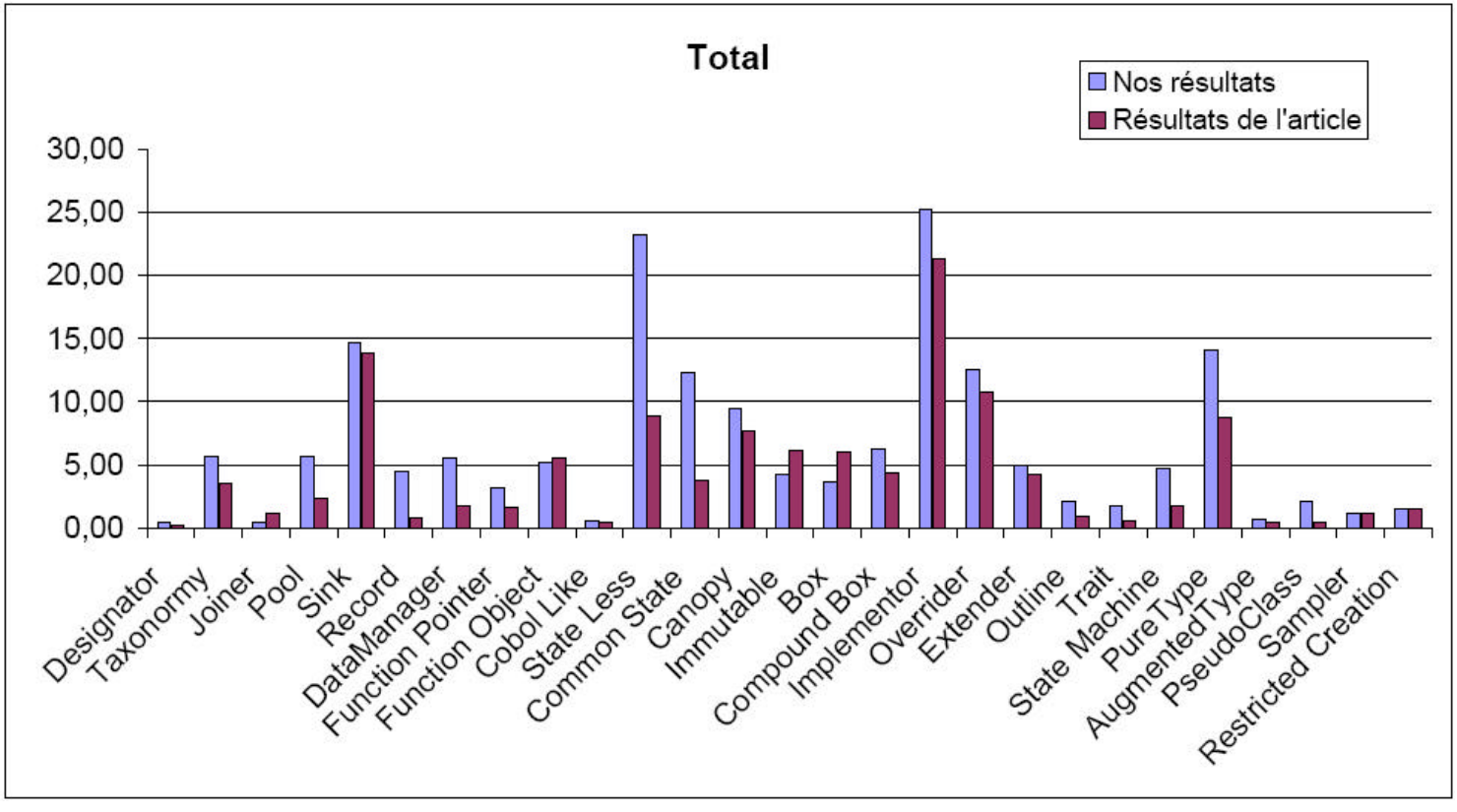
Analyse des résultats obtenus

Les moyennes obtenues



Analyse des résultats obtenus

Les totaux obtenues



Différences avec l'article

- Le micro pattern Stateless
- Le micro pattern Common State

Évolutions et travaux futurs

- Intégration à l’interface graphique de Ptidej.
- Validation de nos résultats à la main
- Corpus des classes du paquet « Shared »