**DOKUMENTACJA**

**WRANGLER**

**ERLANG**

**Olszewska Małgorzata**

**Polnik Wojciech**

1. **Co to jest ?**

Nie, nie jest to tylko marka spodni. ☺ Wrangler to też interaktywne narzędzie refaktoryzacji dla języka Erlang , zintegrowane w obu formach dla emacs i Eclipse as a part of the Erlang IDE, ErlIDE plugin for Erlang.

1. **Refaktoryzacja.**

Proces poprawy konstrukcji programu, który nie zmienia jego działania zewnętrznego. Zmianie ulega sposób otrzymania wyniku. Robimy to po to aby nasz program mógł być w przyszłości łatwo utrzymywany i rozszerzalny.

W refaktoryzacji zależy nam na zmianie kodu całego programu a nie pojedynczej funkcji. Dlatego zapotrzebowanie na takie narzędzia jest oczywiste.

/\*Podczas refaktoryzacji ręcznej wspierające narzędzie do refaktoryzacji jest bezcenne, ponieważ jest bardziej niezawodne. \*/

1. **Refaktoryzacja Wranglerem.**

Refaktoryzacja Wrangler’a obejmuje zmiany strukturalne , takie jak funkcje zmiany nazwy zmiennej i , ekstrakcji modułu funkcji i uogólnienia . Wrangler rozpoznaje makra w kodzie i może być używany na jednym pliku lub w całym projekcie .

Wrangler może być również wykorzystywany do zlokalizowania i usunięcia klonów kodu i poprawienia struktury modułu projektów .

*•* renaming of variables, functions and modules;

-function extraction: an identified expression is made for the body of a new function  
definition;

*•* introduction of new macro definitions and to fold against macro definitions.  
*•* inlining or ‘unfolding’ a function application, and its dual, ‘folding’ an expression  
into a function application;  
*•* introduction of local variables to name an identified expression, and dually, to inline  
variable definitions;

*•* function generalisation, as shown above;

Taka refaktoryzacja powoduje nie tylko zmiany w kodzie funkcji, zmienia też definicje funkcji, jej wywołanie oraz eksportowanie.

1. **Cechy Wranglera.**

* wpiera testy: EUnit , QuickCheck i Common Test(test jest refaktoryzowany wraz z kodem)
* /\*In particular,  
  we have implemented transformations for *state data* in various kinds of generic state  
  machines in both the OTP library and the QuickCheck.\*/
* The code inspector highlights  
  local ‘code smells’, and a number of reports highlight issues in the module structure of  
  projects, including circular inclusions and other potential faults
* The code clone detection facilities can be used on large multi-module projects to report on  
  code clones and how they can be removed; clone detection can be preformed incrementally  
  on larger code bases, for example, as part of a continuous integration approach to software  
  construction
* The Emacs and Eclipse versions provide a preview of the effects of a refactoring and also  
  support multi-level ‘undo’ once refactorings have been performed.
* Wrangler has been recently extended with a framework that allows users to define for themselves refactorings and code inspection functions that suit their needs. These are defined using a template- and rule-based program transformation  
  and analysis API
* Wrangler also supports a domain-specific language that allows users  
  to script composite refactorings, test them and apply them on the fly

User-defined refactorings and scripts are not ‘second-class citizens’: like the existing Wrangler refactorings, user-defined refactorings benefit from features such as results  
preview, layout preservation, selective refactoring, undo and so on.

* -Wrangler also provides a set of process-based refactorings, including the introduction of process naming, instead of providing access to a process only through a dynamically generated ‘process id’.

1. **Kod do refaktoryzacji z błędami składni.**

Wrangler akceptuje programy Erlanga z błędami składniowymi lub z definicjami makr, które nie mogą być przetwarzane przez epp\_dodger . W przypadku , że program bierze pod uwagę błędy składniowe lub niesparsowane makra, funkcje / atrybuty , do których należą te błędy składni nie dotyczą procesu refaktoryzacji, jednak komunikaty ostrzegawcze z prośbą o ręczną kontrolę tych części programu są podane po refakoryzacji.

1. **Podsumowując:**

* Target the full language.
* Produce readable results.
* Workflow integration – popularny edytor.
* Support incomplete code.
* Support user decisions.
* Support user-driven extension.
* Integrate with other tools.

1. **Implementation.**

Program text is parsed into an abstract syntax tree, which  
is then argumented with additional semantic information. Walking this tree allows the computation of the preconditions for the refactoring, and if these conditions hold then the tree is transformed; it is finally rendered back into program source code.

1. **Instalacja**