EKON 17

SONAR

```
10 print “Hello EKON”;
20 Goto 10;
```

“Metrics measure the design of code after it has been written”

Don’t Change Rules during the Game

Kleiner Kommunikation.....

EKON 17

maXbox

Agenda EKON

- What are Metrics ?
- Install SONAR
- Top Five (LCOM4)
- Improve your Code (Metric Based Dashboard)
- Optimisation and other Tools

What are Metrics?

Metrics are for

- Evaluate Object Complexity
- Quantify your code
- Highlight Redesign Needs
- Change Impact Analysis

UEB: 15_pas_designbycontract.txt

EKON 17

maXbox

Metrics deal with

Bad Structure

- General Code Size (in module)
- Cohesion (in classes and inheritance)
- Complexity
- Coupling (between classes or units)
  - Cyclic Dependency, Declare+Definition, ACD-Metric
- Interfaces or Packages (design & runtime)
- Static, Public, Private (inheritance or delegate)

UEB: 10_pas_oodesign_solution.txt

Some Kind of wonderful ?

- Grab the example project files from the Github repository: https://github.com/SonarSource/sonar-examples/tree/master/projects/languages
- There's a Delphi sample in there demonstrating how to configure a sonar-project.properties file that points to your source code directories.

UEB: 8_pas_verwechselt.txt

EKON 17

maXbox

Important metrics to look for

- duplicated_blocks
- violations – info_violations
- public_undocumented_api
- uncovered_complexity_by_tests (it is considered that 80% of coverage is the objective)
- function_complexity_distribution >= 8,
- class_complexity_distribution >= 60
- package_edges_weight

IMPLEMENTED FEATURES

- Counting lines of code, statements, number of files
- Counting number of classes, number of packages, methods, accessors
- Counting number of public API (methods, classes and fields)
- Counting comments ratio, comment lines (including blank lines)
- CPD (code duplication, how many lines, block in how many files)
- Code Complexity (per method, class, file; complexity distribution over methods, classes and files)
- LCOM4 and RFC
- Code colorization

EKON 17

maXbox

IMPLEMENTED FEATURES II

- Unit tests reports
- Assembler syntax in grammar
- Include statement
- Parsing pre-processor statements
- Rules
- Code coverage reports
- Source code highlight for unit tests
- “Dead” code recognition
- Unused files recognition

http://docs.codehaus.org/display/SONAR/Delphi+Plugin

Metric/Review Checklist

1. Standards - are the software standards for name conventions being followed?
2. Are all program headers completed?
3. Are changes commented appropriately?
4. Are release notes Clear? Complete?
5. Installation Issues, Licenses, Certs. Are there any?
6. Version Control, Are output products clear?
7. Test Instructions - Are they any? Complete?
8. "Die andere Seite, sehr dunkel sie ist" - "Yoda, halt's Maul und iß Deinen Toast!"

Install or pitfall

Ensure Java 6 SDK/JRE installed (**Delphi addon doesn't work with Java 7**)
    Download the Sonar application: http://www.sonarsource.org/downloads/
    Install Sonar (or copy/clone)
    Run Sonar (StartSonar.bat)
    Once logged on (default user/pass is admin) install the Delphi add-on via the Update Center list of Available Plugins. See:

http://docs.codehaus.org/display/SONAR/Update+Center

After setting the SONAR_RUNNER_HOME environment variable, exec 'sonar-runner' in the folder with your configured properties file to have
Sonar parse through the Delphi project files.

Sonar in 3 minutes

- http://sonar.codehaus.org/downloads/
- unzip
- sonar.sh console
- mvn sonar:sonar
- http://localhost:9000/
- Download plug-in jar, Copy to extensions/..
- Restart Sonar

EKON 17

maXbox

Top Nine Metrics

1. VOD Violation of Law of Demeter
2. LCOM4 (Lack of Cohesion of Methods)
3. DAC (Data Abstraction Coupling)(Too many responsibilities or references in the field)
4. CC (Complexity Report), McCabe cyclomatic complexity, Decision Points)
5. CBO (Coupling between Objects)→ Modularity

EKON 17

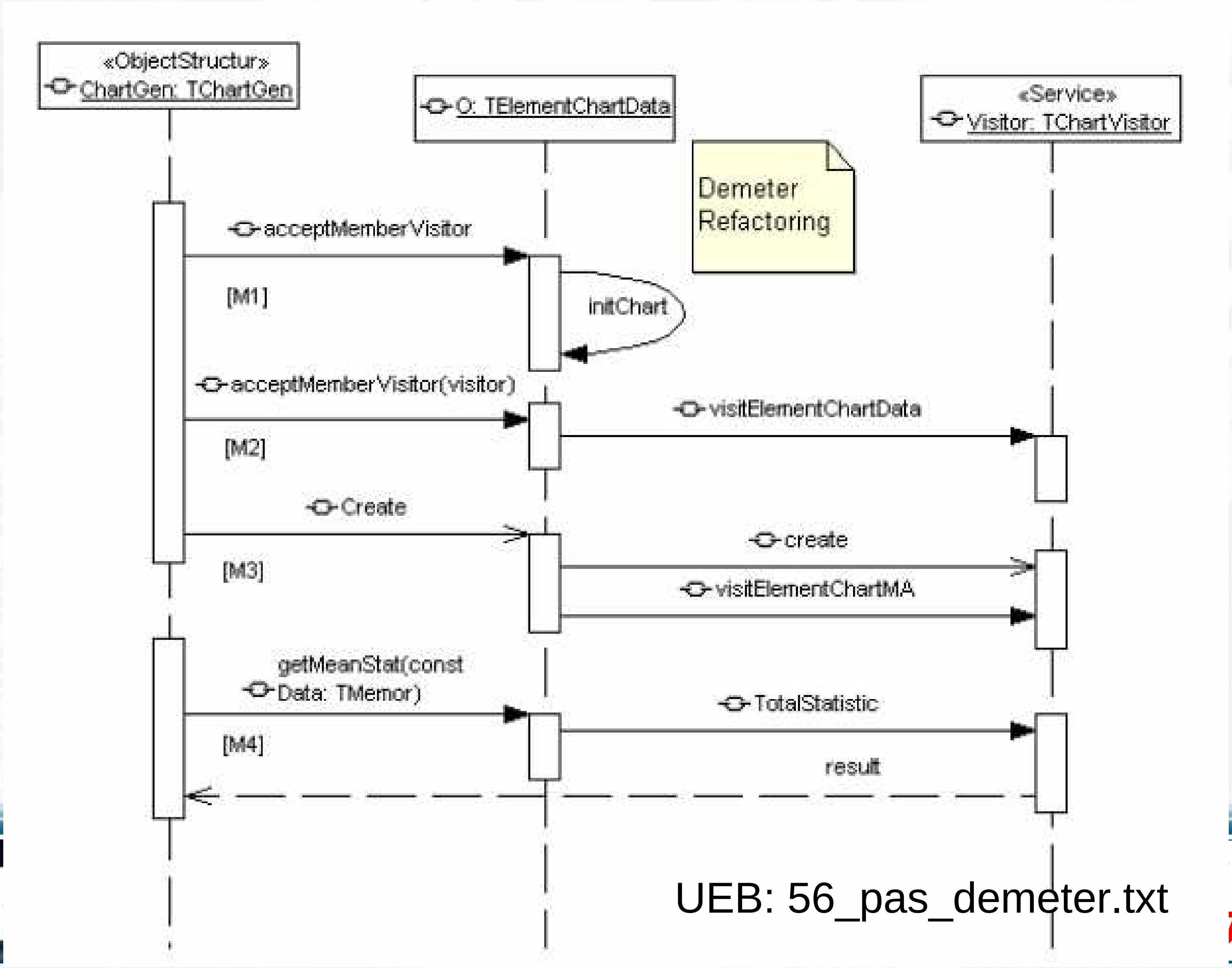
Top Nine II

6. PUR (Package Usage Ratio) access information in a package from outside
7. DD Dependency Dispersion (SS, Shotgun Surgery (Little changes distributed over too many objects or procedures))
8. CR Comment Relation
9. MDC (Module Design Complexity (Class with too many delegating methods)

EKON 17

maXbox

Demeter as SEQ



UEB: 56_pas_demeter.txt

LCOM – cohesion

•LCOM4 measures the number of "connected components" in a class. A connected component is a set of related methods and fields.

•There should be only one such component in each class. If there are 2 or more components, the class should be split into so many smaller classes.

(Lack of cohesion of methods)

EKON 17

maXbox

DAC or Modules of Classes

Large classes with to many references
- More than seven or eight variables
- More than fifty methods
- You probably need to break up the class in Components (Strategy, Composite, Decorator)

```
TWebModule1 = class(TWebModule)
  HTTPSoapDispatcher1: THTTPSoapDispatcher;
  HTTPSoapPascalInvoker1: THTTPSoapPascalInvoker;
  WSDLHTMLPublish1: TWSDLHTMLPublish;
  DataSetTableProducer1: TDataSetTableProducer;
```

CC

- Check Complexity

```
function IsInteger(TestThis: String): Boolean;
begin
  try
    StrToInt(TestThis);
  except
    on EConvertError do
      result:= False;
  else
    result:= True;
  end;
end;
```

Ueb: 164_code_reviews.txt

EKON 17

maXbox

CBO I

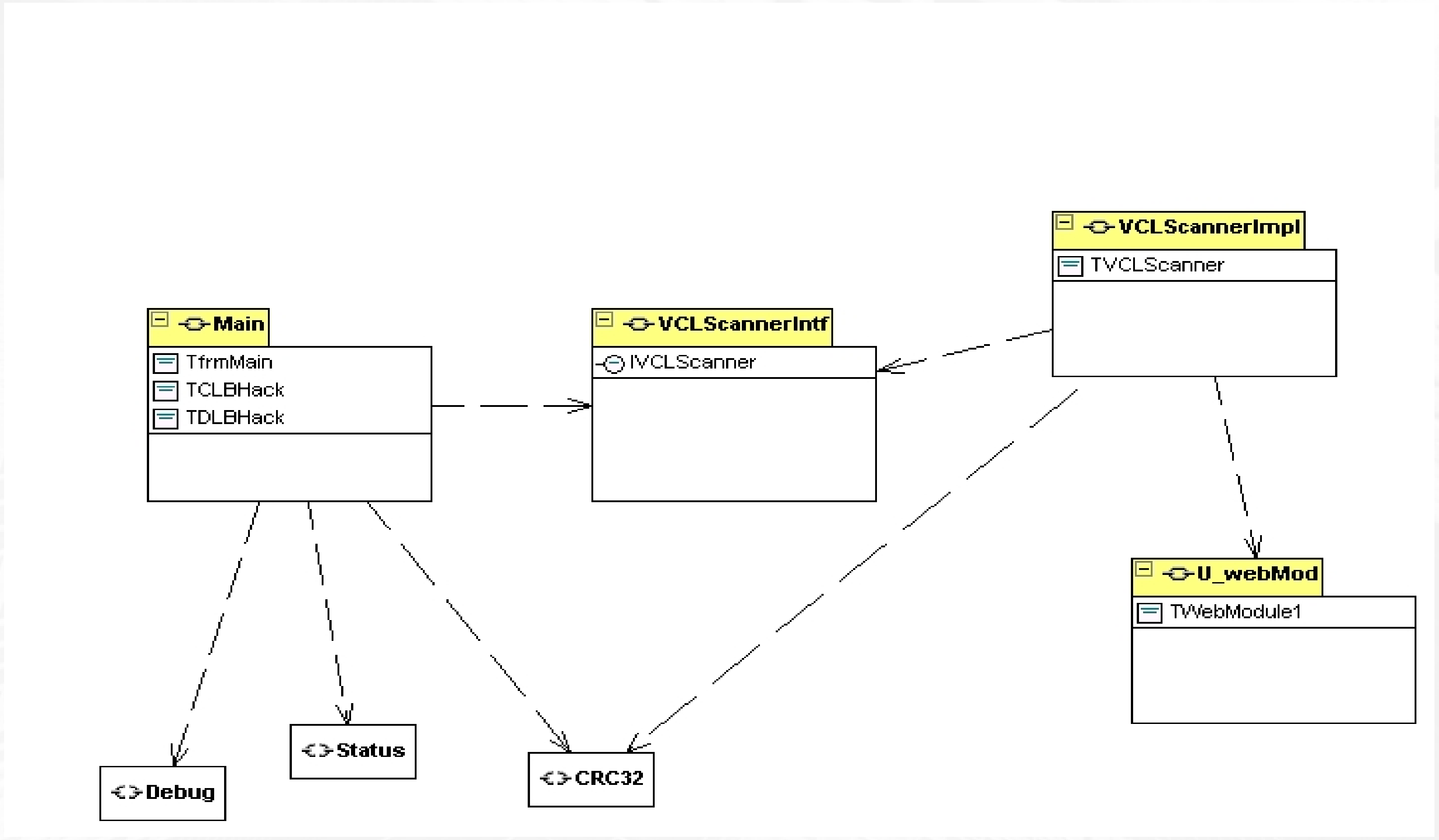
• CBO measures the number of classes to which a class is coupled. According to remarks and comments on CBO and coupling, we include coupling through inheritance.

Two classes are considered coupled, if methods declared in one class call methods or access attributes defined in the other class.

EKON 17

maXbox

PUR Package Usage Ratio



Finally you can measure:

- Duplicated code
- Coding standards
- Unit tests
- Complex code
- Potential bugs
- Comments
- Design and architecture

UEB: 33_pas_cipher_file_1.txt

Metric based Sonar Dashboard

- ● You cannot improve what you don’t measure
- ● What you don’t measure, you can’t prove
- ● Broken Window Theory
- Sonar Dashboard
- – Lines of code
- – Code Complexity
- – Code Coverage
- – Rules Compliance
- ● Time Machine
- ● Clouds & Hot spots

http://docs.codehaus.org/display/SONAR/Plugin+Library

EKON 17

maXbox

Refactoring Use

Entity	Refactoring Function	Description
Package	Rename Package	Umbenennen eines Packages
Class	**Move Method**	Verschieben einer Methode
Class	**Extract Superclass**	Aus Methoden, Eigenschaften eine Oberklasse erzeugen und verwenden
Class	Introduce Parameter	Ersetzen eines Ausdrucks durch einen Methodenparameter
Class	**Extract Method**	Heraustrennen einer Codepassage
Interface	Extract Interface	Aus Methoden ein Interface erzeugen
Interface	Use Interface	Erzeuge Referenzen auf Klasse
Component	Replace Inheritance with Delegation	Ersetze vererbte Methoden durch Delegation in innere Klasse
Class	Encapsulate Fields	Getter- und Setter einbauen
Model	Safe Delete	Delete a Class with Refrences

EKON 17

maXbox

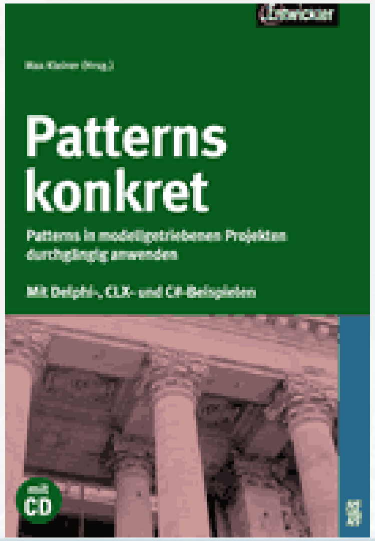
Audits & Metric Links:

- Delphi XE Tool: Together
- http://www.modelmakertools.com/
- Report Pascal Analyzer: http://www.softwareschule.ch/download/pascal_analyzer.pdf
- *Refactoring* Martin Fowler (1999, Addison-Wesley)
- http://nemo.sonarsource.org/ (Live Sonar)
- Model View in Together: *www.softwareschule.ch/download/**delphi**2007_modelview.pdf*

Q&A and may the source be with you

max@kleiner.com

www.softwareschule.ch


Entwickler
Max Kleiner (Hrsg.)
Patterns konkret
Patterns in modellgetriebenen Projekten durchgängig anwenden
Mit Delphi-, CLX- und C#-Beispielen
mit CD

EKON 17

maXbox