A practical Approach to Refactoring in .NET

An Introduction

By Enricos Manassis

Enricos.Manassis@ProClaris.com

proclaris
We engineer software. Others just craft it

01/09/2006

Agenda

- Purpose of Refactoring
  - Challenge
  - Quality measures in software
- Approach
- Tools
  - Reverse Engineering
  - Similarity Analysis
  - Code Cleansing
  - Code Metrics

Agenda (continued)

- C# vs. VB.NET
  - Differences in Language
  - Comparison of tools available in C# / VB.NET
- Miscellaneous tools
  - Refactoring support tools
  - Code converters
  - De-compilers / code morphing
- Practical Tool Box

Refactoring Challenge

- Non cohesive classes
  - Ill-defined and ill-assigned responsibilities: the functionality maze
  - Improve class typing (boundary, controller, entity) : UI presentation, UI workflow, Business workers (workflow and rules), Persistence
  - Excessive class coupling
  - Synonym classes, duplicate responsibilities
- Business logic and rules scattered throughout classes
  - Improve structure of code modules
  - Increase decoupling between layers
- Extract class commonalities: improve abstraction
- Decrease code complexity, and code extent
  - “Less Code, Less Bugs”

Quality Measures in software

	Class design and Coding	Software Architecture	Technical Architecture	Functional/ Class Design
Reusability	X	X		X
Maintainability/ Extensibility	X	X		X
Robustness (Reliability, Resilience, Availability)	X	X	X	
Performance	X	X	X	X (cohesiveness)
Scalability		X	X	
Security	X (SQL Injection)	X	X	X (application)

proclaris
We engineer software. Others just craft it

Approach

- Reverse engineering of OO code base
  - Static structure: class diagrams
  - Dynamic structure
    - Sequence (collaboration) diagrams
    - Activity (flow) diagrams for class operations
- Code similarity analysis
- Rule based code cleansing
- Code Quality Metrics

Quality Metrics / Techniques

	Reverse Engineering	Similarity Analysis	Code Cleansing	Code Metrics
Reusability	X	X		X
Maintainability/ Extensibility	X	X		X
Robustness	X		X	X
Performance	X		X	X
Scalability	X			
Security			X	

proclaris
We engineer software. Others just craft it

Tools: Reverse Engineering

- Class structure
  - Class inventory
  - Class static relationships (inheritance)
- Class Dynamics
  - On one responsibility (class member)
  - Sequence diagrams: class interaction of one class member
  - Activity Diagrams: program flow within one class member
- Applicability
  - Improve structure of code modules
  - Increase decoupling between layers
  - Improve class cohesiveness
  - Improve class typing
- Scope
  - Rearranging module structure effectively amounts to re-engineering!

Tools: Reverse Engineering

- Redefine class structure
  - Create new classes
    - UI controllers: to collect the UI workflow (task oriented)
    - Business controllers: to collect the workflow on business objects (process oriented) and business rules spanning business objects
    - Business entities (business objects and business rules on specific business object)
    - Data Type Classes (DTC) == ECompas layer
  - Extract application (UI) workflow logic from UI classes into UI controllers
  - Extract business logic from the maze, assign it to new class members, assign new class members to new classes
- Generate code for new structure
  - Forward engineer of structure
  - Copy paste code logic from initial location to new class members
  - Use aid from code refactoring support tools (Refactor!™ Pro)

Tools: Reverse Engineering

- Tools
  - Borland® Together® Edition for Microsoft® Visual Studio® .NET 2.0 (199€ VAT Excl. Single User)
    - General UML modelling, with forward and reverse engineering
    - Reverse engineer Sequence diagrams only for C#
    - No reverse engineer of flow diagrams
    - Forward engineering support can be an effective aid for rearranging classes and their responsibilities
  - LogicExplorers CodeLogic for C# (295$ US Single User)
    - Only for C#
    - Only reverse engineer (class, sequence and flow diagrams)

Tools: Reverse Engineering

Demos

Code Logic

http://www.logicexplorers.com/index.html

Together HTML report

file:\\C:\DATA\Projects\Anesthesia\Code\ARCsharp\CS\UtilitiesVB\out\doc\index.html

proclaris
We engineer software. Others just craft it

Tools: Similarity Analysis

- Detect similar code patterns in any set of code files
- Useful as a refactor aid to pinpoint places where commonalities exist
- Applicability
  - Synonym classes, duplicate responsibilities
  - Extract commonalities, improve abstraction
  - Decrease code complexity
- Tools
  - redhillconsulting: Simian (99$ US)
    - Command line tool only, but HTML output

Tools: Similarity Analysis

Demo
Simian

http://www.redhillconsulting.com.au/products/simian/

Tools: Rule based code cleansing

- Two types
  - Working on IL code
  - Working on Source Code (VB.NET or C#)
- Rule Categories
  - Library design
  - Localization
  - Naming conventions
  - Performance
  - Security
- Tools
  - Microsoft: FxCop (Free): Works on IL Code
  - FMS Inc. : Total .NET Analyzer (500$ US Single User)
    - Works on source code (VB.NET and C#)

Tools: Rule based code cleansing

Demo

FxCop

http://www.gotdotnet.com/team/fxcop/

Tools: Code Metrics

- Features
  - Analyze .NET assemblies or source code of an application and generates design quality metrics.
- Applicability
  - Measure the quality of a design in terms of its extensibility, reusability and maintainability
  - Identify the modules that are the most risky, most unstable
  - Understand assembly dependencies

Tools: Code Metrics

- Tools
  - NDepend (Free)
    - Porting of JDepend to the .NET framework
    - Works on IL code
    - Very complete and conveniently accessible documentation on the code metrics measured
    - Command line tool only

Tools: Code Metrics

Demo
NDepend

http://smacchia.chez.tiscali.fr/NDepend.html

proclaris
We engineer software. Others just craft it

C# vs. VB.NET

- Differences in Language
  - Mainly syntactic
  - A few semantic
    - E.g. Optional parameters in VB.NET, Unsafe code in C#
  - No runtime performance advantage to C#
  - Faster compilation in C#
- References
  - http://www.codeproject.com/dotnet/vbnet_c__difference.asp
  - http://support.microsoft.com/default.aspx?scid=kb;EN-US;Q308470
  - http://www.harding.edu/USER/fmccown/WWW/vbnet_csharp_comparison.html

C# vs. VB.NET

Comparison of tools available in C# / VB.NET

- www.sourceforge.net
  - Projects written in C#: 612
  - Projects written in VB.NET: 23
- www.programmersheaven.com
  - Sourcecode & Tools C#: 66 (26 development tools)
  - Sourcecode & Tools VB.Net AND VB: 42 (12 development tools)
- Microsoft
  - C#: http://msdn.microsoft.com/vcsharp/programming/tools/default.aspx
  - VB.NET: http://msdn.microsoft.com/vbasic/downloads/tools/default.aspx

Tools: Refactor Support

- IDE support for rearranging code structure and logic
  - Inline Temp
  - Replace Temp with Query
  - Split Temporary Variable
  - Move initialization to declaration
  - Split initialization from declaration
  - Move declaration near reference
  - Create strongly typed collections
  - Convert Dynamic SQL to Command Parameters
  - Convert Dynamic SQL to Stored Procedures Automatically
  - Generate Calling Code to Existing Stored Procedure
  - Create Namespace for Selected Class(s) and move class to namespace
  - Manage Regions

Tools: Refactor Support

- Tools
  - DevExpress: Refactor! Pro (99$ US All Incl.)
    - C# and VB.NET
    - Professional and IDE neutral version of Refactor embedded by Microsoft in VS.NET 2005
    - Support for custom configuration
  - KnowDotNet: NET Refactor (59$ US All Incl.)

Tools: Refactor Support

Demo
Refactor!

http://msdn.microsoft.com/VBasic/Downloads/2005/Tools/Refactor/3MinDemo.Aspx

proclaris
We engineer software. Others just craft it

Tools: Code Converters

- Convert VB.NET source code to C#
  - Source-to-source code
  - Preserves code comments
- Tools
  - VB Conversions (98.73€ All Incl.)
    - VB to C# converter
    - Code cleanser: gives advises and explanation for better and more compatible VB.NET code
    - Translates VB.NET to C# (minimizing use of VisualBasic Namespace)
    - Convert multiple projects at once

Tools: Code Converters

Demo
VBConversions

http://www.vbconversions.com/

proclaris
We engineer software. Others just craft it

Tools: De-Compilers

- Decompile IL code to high level language
- Refactor and code optimization
  - Intelligently produce high level code by recognizing high level constructs
- Can be used for code morphing
  - Code compiled from VB.NET can be decompiled to C#

Tools: De-Compilers

- Tools
  - Aisto: Reflector for .NET (Free)
    - C# and VB.NET output
    - Numerous productivity Add-Ins
    - No refactoring!
  - Jungle Creatures, Inc. : Decompiler.NET (550$ US All Incl., 40% volume discounts for more than 5 licenses)
    - C# output
    - Decompiler, Obfuscator, Language Translator, and Refactoring
  - 9rays.net: Spices.Decompiler (493$ US All Incl.)
    - C# and VB.NET output
    - Only de-compilation with refactoring
    - 3 month licence for 153$ US All Incl.
  - Remotesoft : Salamander (1000$ US All Incl.)
    - C# and VB.NET output
    - Only de-compilation with refactoring

Tools: De-Compilers

Demo
Decompiler.NET

http://www.junglecreatures.com/DesktopDefault.aspx?tabindex=2&tabid=3

proclaris
We engineer software. Others just craft it

Practical Tool Box

	Name	URL
Reverse Engineering	CodeLogic (295$ US All Incl.)	http://www.logicexplorers.com/index.html
Refactor Support	Refactor! Pro (99$ US All Incl.)	http://www.devexpress.com/Products/NET/Refactor/
Similarity Analysis	Simian (99$ US All Inc.)	http://www.redhillconsulting.com.au/products/simian/
Code Cleansing	FxCop (Free)	http://www.gotdotnet.com/team/fxcop/
Code Metrics	NDepend (Free)	http://smacchia.chez.tiscali.fr/NDepend.html
Code Converter	VBConversions (98.73€ All Incl.)	http://www.vbconversions.com/

proclaris
We engineer software. Others just craft it