

Računalom podržano programsko inženjerstvo

CASE pomagala

CASE

❑ CASE

- eng. Computer Aided Software Engineering
- eng. Computer Aided System Engineering

❑ CAISE

- Computer Aided Information System Engineering

❑ Računalom podržano programsko/informacijsko inženjerstvo

- programski sustav za pomoć pri izgradnji IS i razvoju programske opreme
- uporaba alata koji podupiru neku od faza životnog ciklusa
- automatizacija programske opreme (Software Automation)

CASE pomagala

❑ Gornji CASE (Upper CASE, Front-End CASE)

- Automatizacija ili podrška ranih faza - planiranje, analiza i oblikovanje
 - modeliranje zahtjeva
 - oblikovanje baza podataka, prevođenje modela u 3NF
 - generiranje sheme BP (SQL naredbe)

❑ Donji CASE (Lower CASE, Back-End CASE)

- potpora fazama razvoja: detaljni dizajn, konstrukcija, ugradnja, održavanje
- generatori programske opreme (application generator, code generator)
- alati za održavanje programske potpore (version control) i prepravke (reengineering)
- alati za izradu programske dokumentacije (documentation tool)
- alati za povratno inženjerstvo (reverse engineering), npr. analizatori izvornog koda, alati za restrukturiranje koda

❑ Integrirana CASE pomagala (ICASE - Integrated CASE)

- cjeloviti CASE sustavi koji pokrivaju sve faze razvoja, a sadrže i dodatne module za povratno inženjerstvo, nadzor i upravljanje izvedbom te osiguranje kvalitete

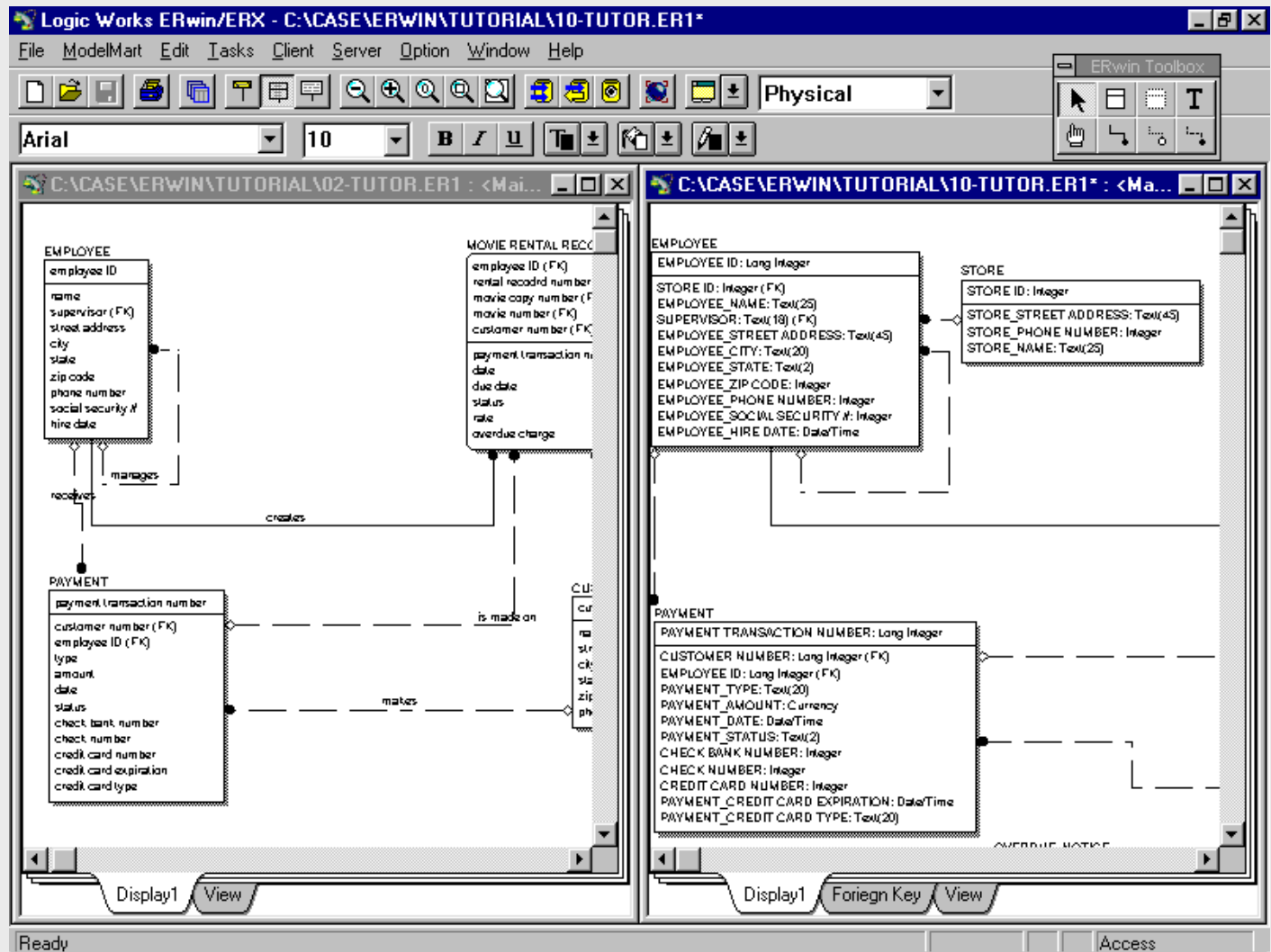
Funkcije pomagala i razvojni procesi

□ Slijede primjeri (dostupnih?) pomagala

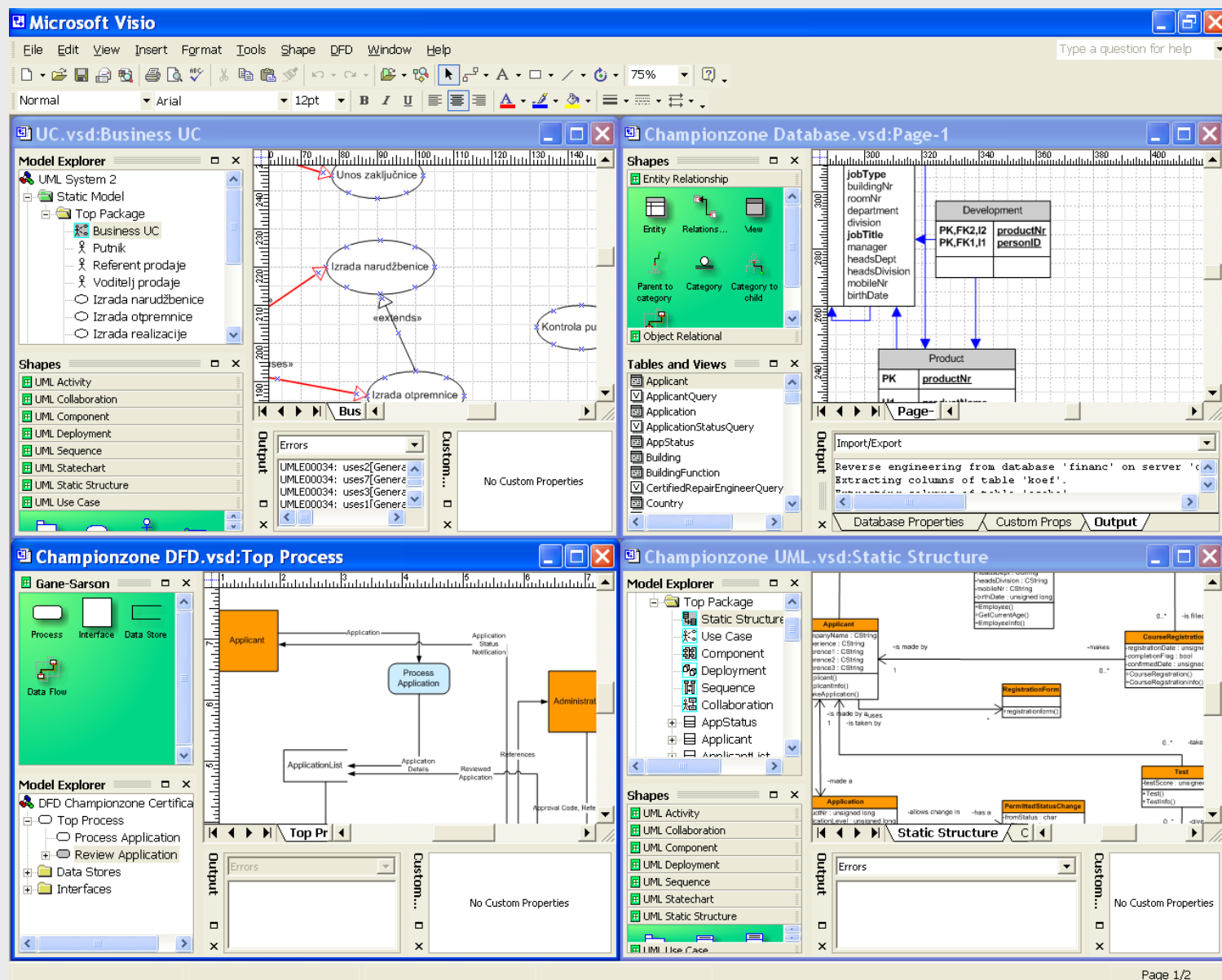
Alati za reinženjerstvo			•	
Alati za testiranje			•	•
Alati za debugiranje			•	•
Alati za analizu programa			•	•
Alati za procesiranje koda		•	•	
Potporna metodologije	•	•		
Alati za prototipiranje	•			•
Upravljanje konfiguracijom		•	•	
Upravljanje promjenama	•	•	•	•
Alati za dokumentiranje	•	•	•	•
Alati za modeliranje	•	•	•	•
Alati za planiranje	•	•	•	•
	Specifikacija	Dizajn	Implementacija	Testiranje

Primjer alata za oblikovanje BP

- ❑ CA AllFusion ERwin Data Modeler – dizajn i reinženjersvo BP, različite notacije (IDEF1X, IE, DM)



MS Visio - univerzalno pomagalo za oblikovanje



Integracija CASE alata u razvojnu okolinu

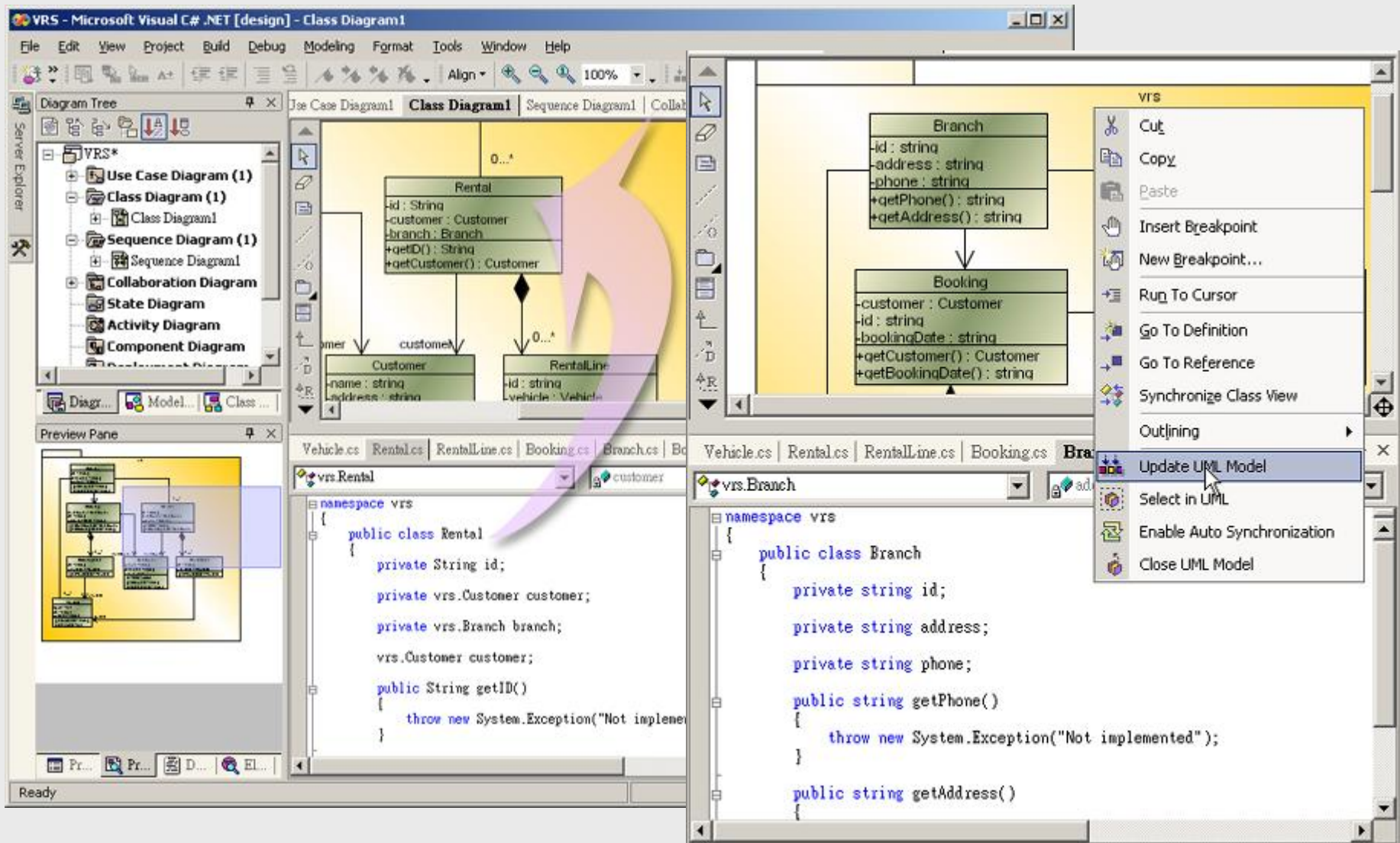
❑ Visual Paradigm for UML

- Svi UML dijagrami
- Analiza teksta
- CRC kartice

❑ Visual Paradigm for SDE (Smart Development Environment)

- integracija s Visual Studio, Eclipse, JBuilder, ...
- reverzno inženjerstvo
- generiranje izvornog koda
- Community Edition
- <http://www.visual-paradigm.com>

Reverzno inženjerstvo



Generiranje izvornog koda

The screenshot displays the Microsoft Visual C# .NET [design] - Booking.cs environment. The interface includes a menu bar (File, Edit, View, Project, Build, Debug, Modeling, Tools, Window, Help), a toolbar, and several panes:

- Diagram Tree:** Shows the project structure with 'VRS*' containing 'Use Case Diagram (1)', 'Class Diagram (1)', 'Sequence Diagram (1)', 'Collaboration Diagram', 'State Diagram', 'Activity Diagram', and 'Component Diagram'.
- Class Diagram:** The main diagram area shows a UML class diagram with classes 'Booking', 'BookingLine', 'Vehicle', and 'Customer'. 'Booking' has attributes '-customer: Customer', '-id: string', and '-bookingDate: string', and methods '+getCustomer(): Customer' and '+getBookingDate(): string'. 'BookingLine' has attributes '-id: string', '-vehicle: Vehicle', and '-vip: bool', and method '+getVehicle(): Vehicle'. 'Vehicle' has attributes '-type: int', '-id: string', and '-model: string', and methods '+getModel(): string' and '+getType(): int'. 'Customer' has attributes '-name: string', '-address: string', and '-vip: bool', and method '+isVIP(): bool'. A multiplicity of '0..*' is shown for the 'BookingLine' association.
- Preview Pane:** Shows a preview of the class diagram.
- Code Editor:** Displays the source code for 'Booking.cs' and 'BookingLine.cs'. The 'Booking.cs' code is as follows:

```
namespace vrs
{
    public class Booking
    {
        private vrs.Customer customer;
        private string id;
        private string bookingDate;
        vrs.Customer customer;

        public vrs.Customer getCustomer()
        {
            throw new System.Exception("Not implene");
        }
    }
}
```

The 'BookingLine.cs' code is partially visible:

```
namespace vrs
{
    public class BookingLine
    {
        private string id;
        private Vehicle ve
        Vehicle vehicle;

        public Vehicle get
        {
            throw new Syst
        }
    }
}
```

A right-click context menu is open over the 'BookingLine' class, showing options such as 'Open Specification...', 'Model Properties', 'Stereotype', 'Abstract', 'Leaf', 'Root', 'Active', 'Visibility', 'New Attribute', 'New Operation', 'Code Engineering', 'Synchronize Relationships with Model', 'Realize all Interfaces', 'Update to Code' (highlighted), 'Browse Code', 'Edit', 'Cut', 'Copy', 'Delete', 'Send to', 'Presentation Options', 'Fit Size', 'Order', 'Format', and 'Select in Tree'.

Pomagalo za praćenje rada aplikacije

❑ Visual Logger (freeware)

- <http://www.axtools.com>
- može se koristiti prilikom razvoja (debugging i trace) ili nakon isporuke aplikacije za praćenje njenog rada i kasniju analizu zabilježenih informacija (logovi se pošalju programerima na analizu)

```
static void Main(string[] args)
```

```
{
```

```
    AxVisualLogger.VisualLoggerClass logger = new AxVisualLogger.VisualLoggerClass();
```

```
    logger.Register("Ime test aplikacije", true);
```

```
    logger.AddEntry("Praćenje aplikacije uključeno");
```

```
    logger.AddEntry("Aplikacija pokrenuta");
```

```
    try
```

```
    {
```

```
        logger.AddEntry("Otvaram i čitam datoteku test.txt");
```

```
        File.ReadAllText("test.txt");
```

```
    }
```

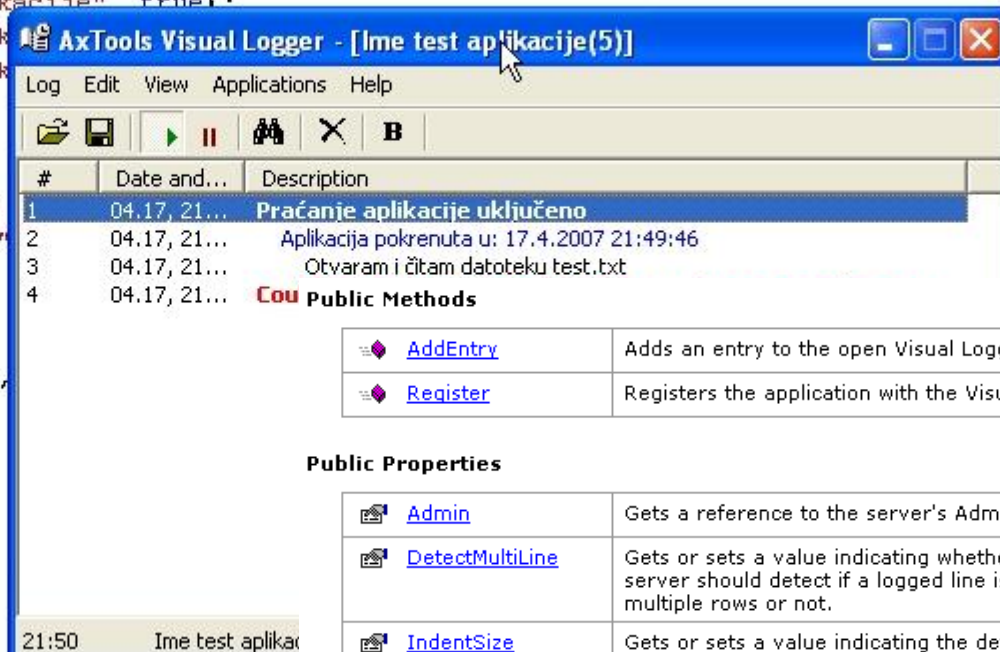
```
    catch (Exception ex)
```

```
    {
```

```
        logger.AddEntry(ex.Message);
```

```
    }
```

```
}
```



◆ AddEntry	Adds an entry to the open Visual Logger list.
◆ Register	Registers the application with the Visual Logger server.

Public Properties

📁 Admin	Gets a reference to the server's Admin object.
📁 DetectMultiLine	Gets or sets a value indicating whether the Visual Logger server should detect if a logged line is spanned across multiple rows or not.
📁 IndentSize	Gets or sets a value indicating the default text nesting level in the Visual Logger list.
📁 UseIndentBar	Gets or sets a value indicating whether the logger should use pipe (" ") characters to outline the text nesting.

Generatori koda [Herrington]

❑ **Prevođenje koda (Code munging)**

- prijevod programskog koda napisanog u jednom programskom jeziku u drugi, regularnim izrazima ili jednostavnom obradom (parsiranjem) i ugrađenim predlošcima (templates)

❑ **Ekspanzija koda (Inline code expanding)**

- generiranje supstitucijom posebnih oznaka (markups)

❑ **Generiranje miješanim kodom (Mixed-code generation)**

- čita izvorni kod i zamjenjuje ga npr. analizom posebne vrste komentara

❑ **Djelomično generiranje razreda (Partial-class generation)**

- generiranje skupova razreda temeljem apstraktnih definicija

❑ **Generiranje slojeva (Tier or layer generation)**

- generiranje cjelokupnog sloja aplikacije, npr. temeljem UML modela (model-driven generation)

CodeSmith

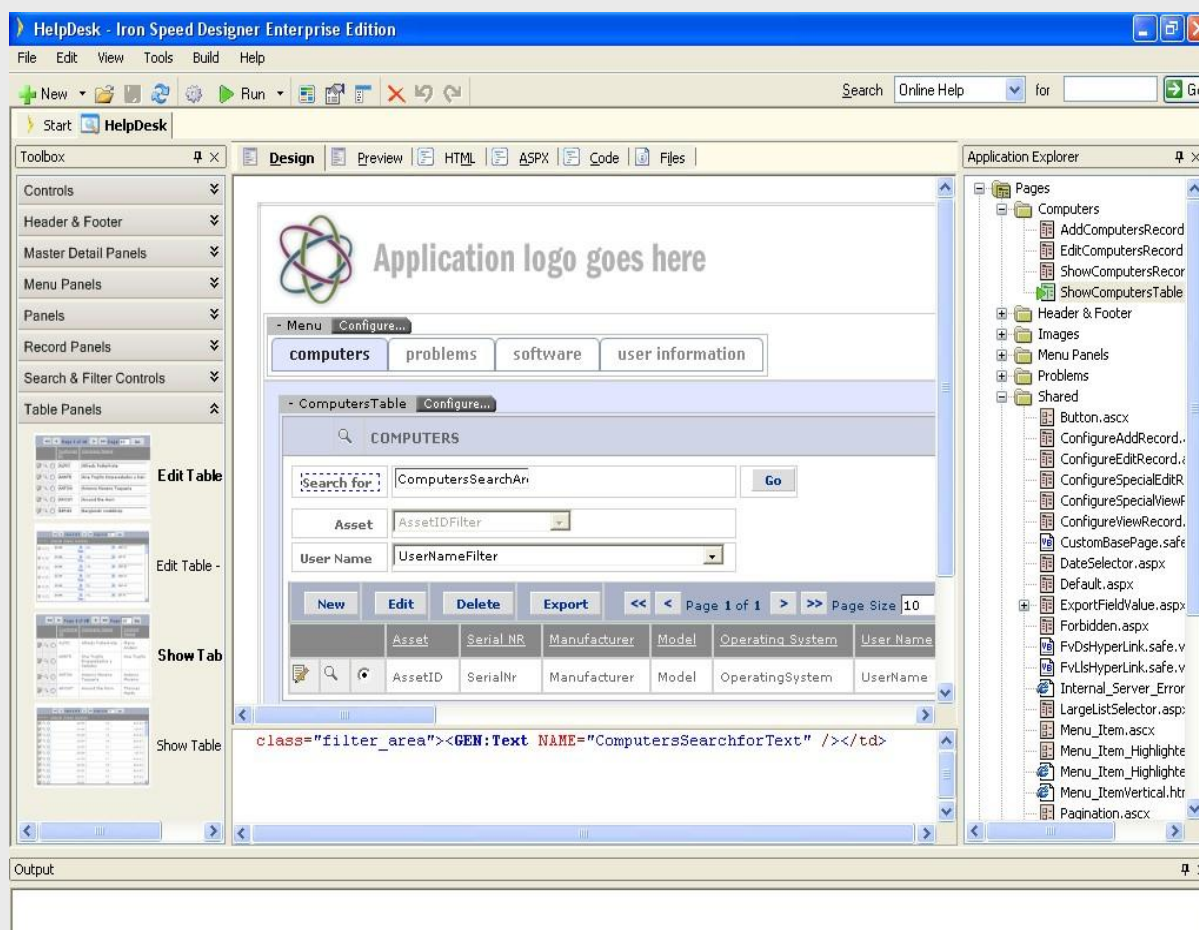
- ❑ Predlošci za C#, VB, J# ...
- ❑ Prilagodba sintaksom sličnom ASP.NET

```
4 Description: Create a list of properties from a database table
5 --%>
6 <%@ CodeTemplate Language="C#" TargetLanguage="C#" Debug="False" Description="Create a list of
7 <%@ Property Name="SourceTable" Type="SchemaExplorer.TableSchema" Category="Context" Descript
8 <%@ Map Name="CSharpAlias" Src="System-CSharpAlias" Description="System to C# Type Map" %>
9 <%@ Assembly Name="SchemaExplorer" %>
10 <%@ Import Namespace="SchemaExplorer" %>
11
12 <% foreach (ColumnSchema column in this.SourceTable.Columns) { %>
13 private <%= CSharpAlias[column.SystemType.FullName] %> _<%= StringUtil.ToCamelCase(column.Nam
14
15 public <%= CSharpAlias[column.SystemType.FullName] %> <%= StringUtil.ToPascalCase(column.Name
16 {
17     get { return _<%= StringUtil.ToCamelCase(column.Name) %>; }
18     set { _<%= StringUtil.ToCamelCase(column.Name) %> = value; }
19 }
20
21 <% } %>
```


Iron Speed

❑ Generator .NET Web Aplikacija

- generira GUI, web page logic, DAL
- do 80% of aplikacije
- ima čarobnjaka, predloške, prilagodbu izbornika, ...



MyGeneration (besplatan!)

❑ Zasnovan na predloščima

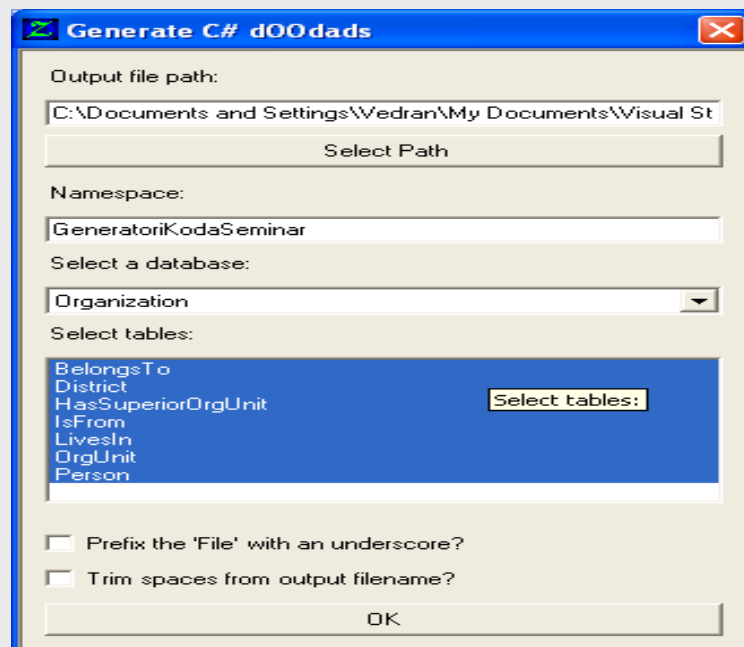
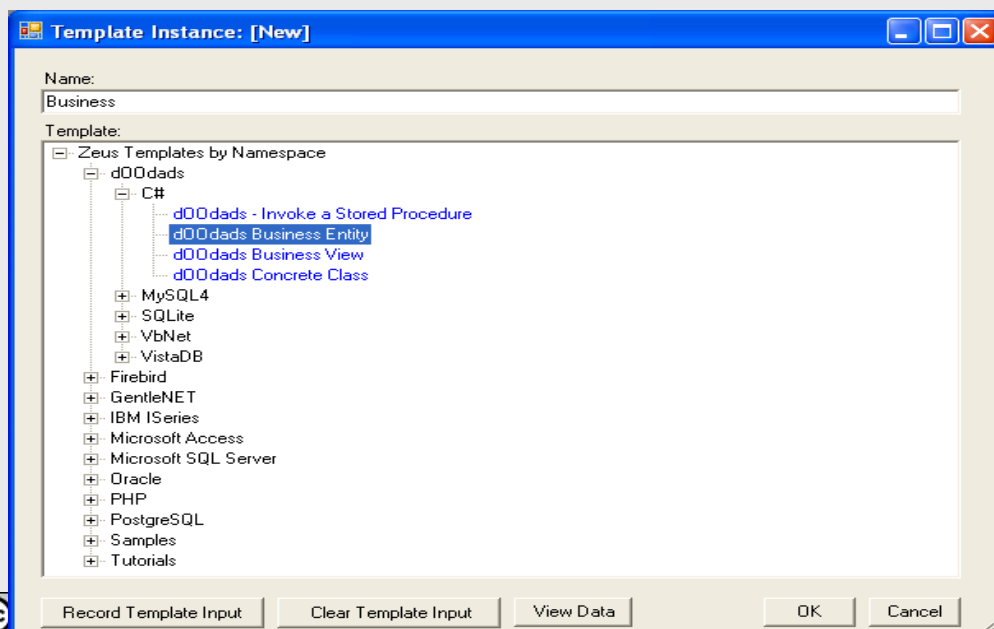
- prilagodba korištenjem JScript, VBScript, C#, VB.NET.

❑ Podrška arhitekturama

- dOOdads, EntitySpaces, EasyObjects.NET/EntLib, Gentle.NET, Opf3, NHibernate, Microsoft's DAAB, DotNetNuke, iBatis.

❑ Podrška bazama podataka

- Microsoft SQL, Oracle, IBM DB2, PostgreSQL, Microsoft Access, FireBird, Interbase, VistaDB, SQLite, MySQL, Advantage, Pervasive



Pouke glede generatora

☐ Pravilno korišteni

- Skraćuju vrijeme kodiranja
- Smanjuju broj pogrešaka
- Povećavaju učinkovitost prilagodbe programa
- Doprinosu pisanju konzistentnog i kvalitetnog koda

☐ Mogući rizici

- Ovisnost o dobavljaču alata
- Loše aplikacije kao rezultat loših predložaka
- Zloraba alata ili uporaba neadekvatnih alata

Reference

❑ C# School Book – Free ebook

- <http://www.programmersheaven.com/2/CSharpBook>

❑ Objects, Classes, and Structs (C# Programming Guide)

- Objects, Classes, Structs, Inheritance, ...
- <http://msdn2.microsoft.com/en-us/library/ms173109.aspx>

❑ Upravljanje projektima

- www.pmi.org
- <http://www.pmi-croatia.hr>
- <http://www.construx.com/>
- <http://office.microsoft.com/en-us/help/HA011361531033.aspx>

❑ IDE , CASE

- <http://www.codegeneration.net/>
- <http://www-306.ibm.com/software/awdtools/developer/rosexde/>
- <http://www.axtools.com>
- <http://www.eclipse.org/>
- <http://staruml.sourceforge.net/en/>
- <http://www.visible.com/>
- <http://www.visual-paradigm.com>