Foreword by Sam Guckenheimer

Domain-Specific Development

with Visual Studio DSL Tools

Microsoft .net Development Series

Steve Cook
Gareth Jones
Stuart Kent
Alan Cameron Wills

Domain-Specific Development with Visual Studio DSL Tools

Domain-Specific Development with Visual Studio DSL Tools

Table of Contents

Contents

List of Figures

List of Tables

Foreword

Preface

About the Authors

1 Domain-Specific Development

- Introduction
- Domain-Specific Development
- Examples
  - Software Defined Circuitry
  - Embedded Systems
  - Device Interfaces
  - Software Development Process Customization
- Benefits
- Languages
- Textual DSLs
- Graphical DSLs
  - Conventions for Representing Structure
  - Conventions for Representing Behavior
- Aspects of Graphical DSLs
  - Notation

Pearson

Table of Contents

Domain Model

Generation

Serialization

Tool Integration

Putting It All Together

DSLs in Visual Studio

The Customization Pit

UML

Summary

2 Creating and Using DSLs

Introduction

Process: Incremental Development of DSLs

Generalizing an Application: Identify Variability, Discover DSLs

Top-Down and Bottom-Up

Developing the DSL: From Sketches to Domain Model

Domain Model and Presentation Are Separate

Refining the DSL

Driving the Framework from the DSL

Using the DSL

Evolving the DSLs

Interpretive Architectures

Creating a DSL in Visual Studio

Creating a DSL Authoring Solution in Visual Studio

Trying Out the DSL Solution

Defining the DSL

Generating the Code for the Designer

Adding to the DSL

Constraints

Customizing the Explorer Window

Table of Contents

    - Customizing the Properties Window
    - Custom Code for the Designers
    - Serialization Format of the DSL File
    - Driving Applications from the DSL
    - Deployment
  - A Second DSL: The Project Definition DSL
  - Architecture of the DSL Tools
    - The Generated Code
    - DSL Tools Architectural Layers
    - The Framework Assemblies
    - Content of the DSL Project
    - Content of the DslPackage Project
  - Summary
- 3 Domain Model Definition
  - Introduction
  - The Domain Model Designer
  - The In-Memory Store
  - Domain Classes
  - Domain Relationships
    - Embeddings
    - Multiplicity
    - References
    - Relationship Derivation
  - Generating a Designer with No Shapes
  - The Generated Code
    - Using the Generated Code
  - More about Domain Classes
    - DomainClassInfo
  - More about Domain Properties

Table of Contents

Calculated Properties

DomainPropertyInfo

More on Domain Relationships and Roles

Accessing Links

More on Relationship Derivation

DomainRelationshipInfo and DomainRoleInfo

More about the Store

Looking Up Elements

Partitions

Rules

DomainModelInfo

Summary

4 Presentation

Introduction

Graphical NotationOverview

Diagram and Editor

Diagram

Editor

Designer

Custom Editor

Shapes

Kinds of Shapes

Shape Maps

Connectors

Connector Anatomy and Appearance

Connectors and Inheritance

Connector Maps

Advanced Connector Maps

Decorators

Table of Contents

Kinds of Decorators

Positioning

Decorator Maps

Customizing the Graphical Notation in Code

Multiline Text Decorators

Variable Image Shape

Set a Background Picture

Set Custom Connection Points

Change Routing Style of Connectors

Explorer

Default Appearance

Changing the Window Icon and Label

Customizing the Appearance of Nodes

Hiding Nodes

Customizing the Explorer through Code

Properties Window

Default Appearance of Properties Window

Categories, Names, and Descriptions

Hiding Properties and Making Them Read-Only

Forwarding Properties

Customizing the Properties Window through Code

Summary

5 Creation, Deletion, and Update Behavior

Introduction

Element Creation

The Toolbox

Element Merge Directives

Custom Element Merge Directives

Re-Parenting with Element Merge Directives

Table of Contents

Custom Element Tool Prototypes

Connection Builders

Multiple Source and Target Role Directives

Multiple Link Connect Directives

Custom Connection Builders

Element Deletion

Default Delete Propagation Rules

Controlling Delete Propagation

Customizing Delete Propagation

Summary

6 Serialization

Introduction

Saving and Loading Models and Diagrams

Model XML File Format

Elements and Properties

Relationships

Relationship Derivation

Cross-Referencing

Using Guids as References

Using Qualified Names as References

References to Links

Diagram XML File Format

Versioning and Migration

The XML Schema

Customization

Modifying XML Element Names

Element Data

Implementing Your Own Serializer

Table of Contents

Generated Serialization Code

Customized Serialization Code

Impact of Customization on the Schema

Summary

7 Constraints and Validation

Introduction

Choosing Hard or Soft Constraints?

Choices Made by the DSL Tools

Soft Constraints in the DSL Tools

Validation Methods

Enabling Validation

Invoking Validation

Custom Validation Categories

Inheriting Validation Behavior

Validation Output

Using Validation Outside the IDE

Validation Against External Data

Hard Constraints in the DSL Tools

Rules

Putting Together Hard and Soft Constraints

Summary

8 Generating Artifacts

Introduction

Artifact Generation Styles

Extensible Stylesheet Language Transformations

Making Use of the Domain-Specific API

A Template-Based Approach

Complex Relationships and Round-Tripping

Table of Contents

The Templatization Process

The First Cut Template

Generation-Specific Model Data

Starting to Build a Library

Syntax of a Text Template

Directives

Custom Directives

Control Block Types

Problems of Large-Scale, Real-World Artifact Generation

Advanced Customizations

Text Templating Architecture

Custom Hosting

Custom Directive Processor

Custom Orchestration

Summary

9 Deploying a DSL

Introduction

Files Needed to Install a Designer

Getting StartedCreating a Setup Project

Setup Project Contents

Customizing Setup

Customizing InstallerDefinition.dslsetup

Customizing settings.ini

Customizing Strings.wxl

Customizing Product.ico

The .dslsetup Format

<dslPackage>

<licenseAgreement>

<supportingFiles>

Table of Contents

<vsItemTemplates>
<dslSchemas>
<vsProjectTemplates>
<mergeModules>
<textTemplates>
Refreshing the Installation Files
Package Load Key
Deploying Text Templates for Code Generation
Creating a Project Template from the Debugging Project
Using a Text Template Include File
Including Text Templates in the VS Item Template
Summary
10 Advanced DSL Customization
Introduction
Tools for Customization
Partial Classes
Double DerivedThe Generation Gap
Custom Constructors
Customization Switches
Custom Overrides
Responding to Changes
Property Handlers "On Value Changed/Changing"
Calculated Domain Properties
Custom Storage Domain Properties
Notify Value Change
Propagating Change from Model to Shape: OnAssociatedPropertyChanged
Rules
Store Events
.NET Event Handlers

Table of Contents

Event Overrides

Bounds Rules

Summary of Change Propagation and Constraint Techniques

DSL Shell Architecture

How to Add a Menu Command

Add a Command Id for Each Command

Increment Menu Resource Index

Add Commands to Command Set

Define the Command Handlers

Good Practices for Command Handlers

Build and Run

Providing Handlers for Standard Commands

Building the DSL Diagram into Another Interface

Implementing Copy and Paste

The Copy Method

The Paste Method

Registering the Menu Handlers

Shape Containers

Child Shapes

A DSL Using Nested Child Shapes

Shape Containment Using Rules

Summary

11 Designing a DSL

Introduction

Identifying Variability

Bottom-Up or Top-Down?

Feature Trees

Feature Trees and DSLs

Developing the Domain Model

Table of Contents

Sketch Domain Snapshots

Domain Model from Snapshots

Developing the Notation

Project Definition Notation

Issue State Notation

Familiar Notations

Defining Validation Constraints

Internal Consistency

Consistency with External Data and Models

Developing and Evolving the Framework

Generation versus Interpretation

Evolving a Generic Framework

Driving a Framework from the DSL

Testing

Validation Constraints

Generator Templates

Generated Code

Rules

Language Definition

Evolving a DSL

What Makes a Good DSL?

Appropriate Notation: An Example with Regular Expressions

Candidate Notations

Graphs Are Not Syntax Trees

Summary

Conclusion

Index