

Visualization and modeling SDK

The visual studio SDK formerly known as DSL-Tools

The screenshot displays the Microsoft Visual Studio IDE with the DSL Designer tool open. The main workspace shows a UML-like diagram titled "Classes and Relationships" for a DSL named "FamilyTreeModel".

Classes and Relationships Diagram:

- FamilyTreeModel** (DomainClass):
 - Has a relationship **FamilyTreeModelHasElements** with **FamilyTreeModel** (DomainClass) via **Elements** (DomainRelationship). Multiplicity: 0..* to 1..1.
 - Has a relationship **FamilyTreeModelHasTowns** with **Town** (DomainClass) via **Towns** (DomainRelationship). Multiplicity: 0..* to 1..1.
- Person** (DomainClass):
 - Has a relationship **Heritage** with **Person** (DomainClass) via **Parents** (DomainRelationship) and **Children** (DomainRelationship). Multiplicity: 0..* to 0..*.
 - Has a relationship **Residence** with **Town** (DomainClass) via **Towns** (DomainRelationship) and **Persons** (DomainRelationship). Multiplicity: 0..* to 0..*.
- Town** (DomainClass):
 - Has a relationship **Residence** with **Person** (DomainClass) via **Persons** (DomainRelationship).

Diagram Elements Panel:

- PersonShape** (GeometryShape):
 - Domain Properties: Name, Birth, Death.
 - Decorators: NameDecorator, BirthDecorator, DeathDecorator, Dash.
- ParentConnector** (Connector):
 - Domain Properties: Name.
 - Decorators: NameDecorator.
- Language1Diagram** (Diagram):
 - Domain Properties: Name.
- TownShape** (GeometryShape):
 - Domain Properties: Name.
 - Decorators: NameDecorator.
- ResidentConnector** (Connector):
 - Domain Properties: Name.
 - Decorators: NameDecorator.

Toolbox: Contains various DSL modeling tools such as Pointer, Domain Class, Named Domain Class, Geometry Shape, Compartment Shape, Image Shape, Connector, Port Shape, Swimlane, Embedding Relationship, Reference Relationship, Inheritance, and Diagram Element Map.

Properties Window: Shows the properties for the selected "DslDefinition Dsl" element, including Code (AccessModifiers, Custom Attr), Definition (Company Name, MDC, Name, Namespace, Package Name, Package GUID, Product Name), and Documentation (Notes).

Error List: Shows 0 Errors, 0 Warnings, and 0 Messages.

History in a nutshell

- 2004: Initial discussion about DSL
 - Alan Cameron Wills
 - Steve Cook (UML)

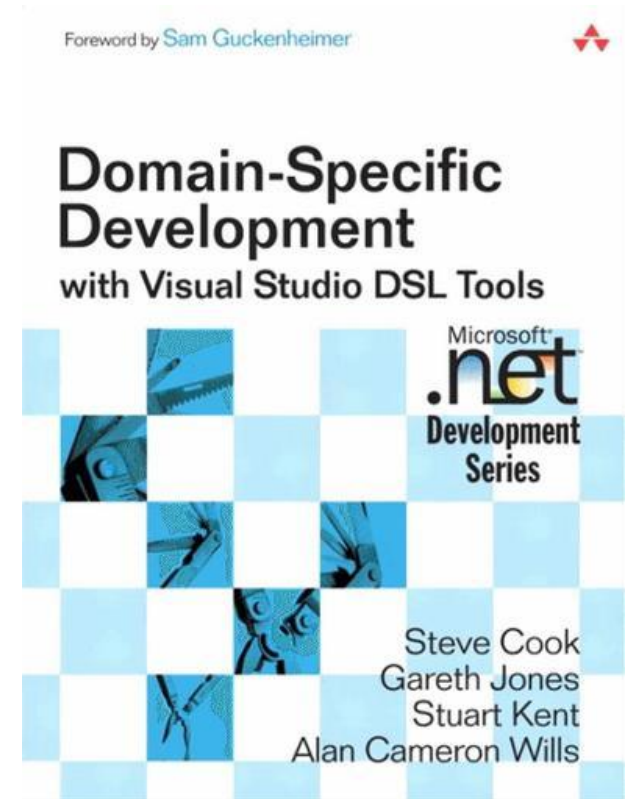
My colleague Alan Cameron Wills makes several excellent points about why not to base domain specific languages on UML. There really is very little common semantic ground between a conveyor belt and anything in UML.

History in a nutshell

- 2004: Initial discussion about DSL
- 2005: DSL Tools added to Visual Studio

History in a nutshell

- 2004: Initial discussion about DSL
- 2005: DSL Tools added to Visual Studio
- 2006: Domain-specific Development

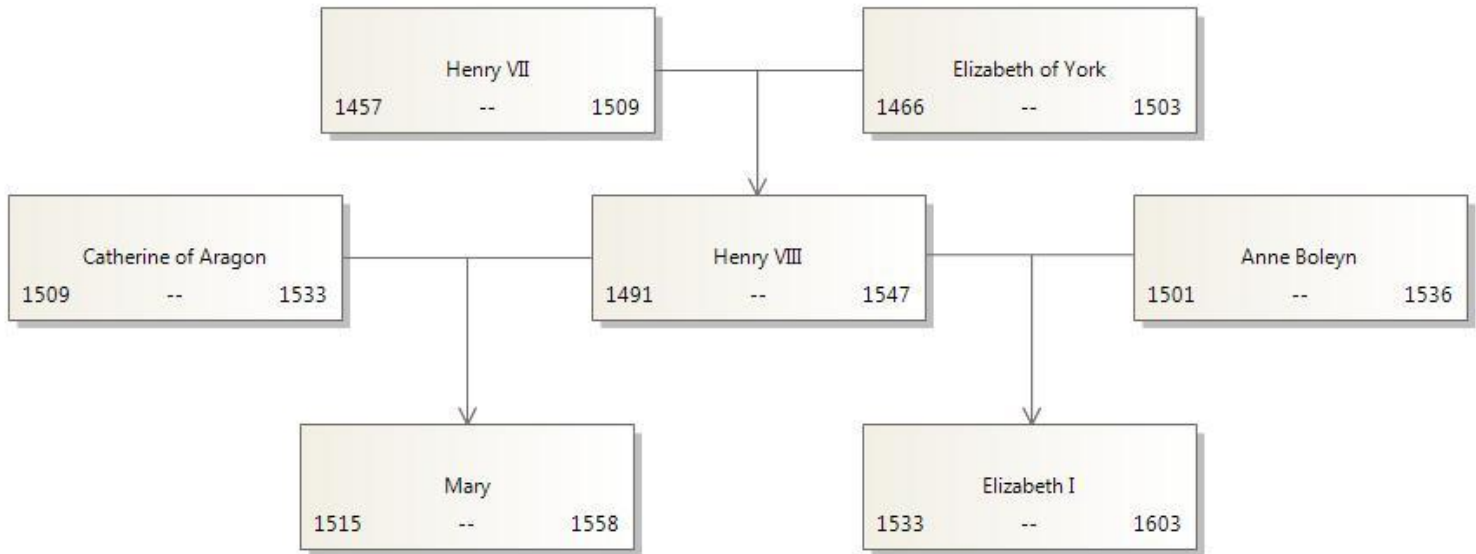


History in a nutshell

- 2004: Initial discussion about DSL
- 2005: DSL Tools added to Visual Studio
- 2006: Domain-specific
Development
- 2010: Renamed to VMSDK in VS2010

What is this SDK all about?

- Creating models



What is this SDK all about?

- Creating models
- Creating forms and GUI's

The screenshot displays a software application interface. The main window, titled 'Sample.farm*', shows a 'Farm' form with the following fields:

- Is Organic:
- Name: Fabrikam Farm
- Size: 270

Below the form, there are two sections: 'Animals' and 'Fields'. The 'Animals' section has buttons for 'New Goat' and 'New Sheep', and a table listing animals:

Name
Sheep1
Sheep25
Goat1
Goat2

The 'Fields' section has a button for 'New Field' and a table listing fields:

Name	Size
Field1	0
Field2	25

On the right side, the 'FarmDslExplorer' window shows a tree view of the farm's structure:

- Fabrikam Farm (Farm)
 - Animals
 - Goat1 (Goat)
 - Goat2 (Goat)
 - Sheep1 (Sheep)
 - Sheep25 (Sheep)
 - Fields
 - Field1 (Field)
 - Field2 (Field)

Below the tree view, the 'Properties' window shows the details for the selected 'Fabrikam Farm' object:

Property	Value
Is Organic	False
Name	Fabrikam Farm
Size	270

What is this SDK all about?

- Creating models
- Creating forms and GUI's
- **Generate textual artifacts**

```
<#@ template inherits="Microsoft.VisualStudio.TextTemplating.VSHost.ModelingTextTransformation" debug="true" #>
<#@ output extension=".txt" #>
<#@ FamilyTree processor="FamilyTreeDirectiveProcessor" requires="fileName='Sample.ftree'" #>

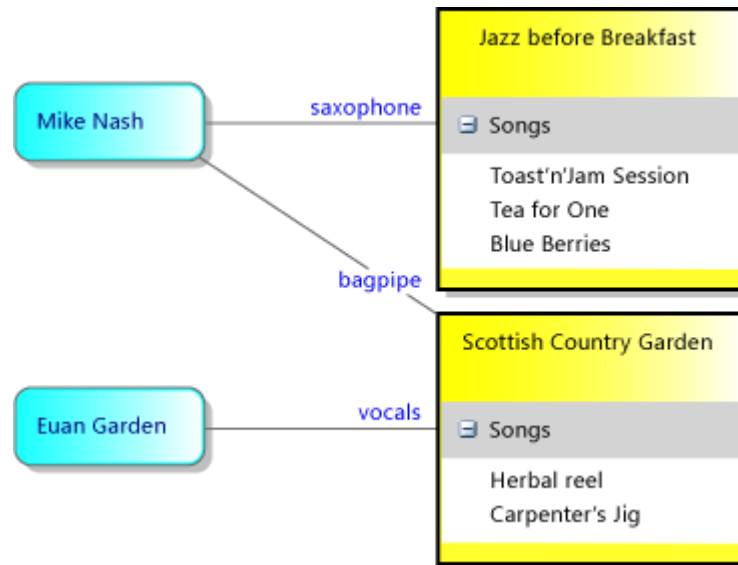
<# foreach (Person person in this.FamilyTreeModel.People) { #>
    <#= person.Name #><#if (person.Town != null) {#> of <#= person.Town.Name #> <#}#>

    <# foreach (Person child in person.Children) { #>
        <#= child.Name #>

<# } } #>
```

What is this SDK all about?

- Creating models
- Creating forms and GUI's
- Generate textual artifacts
- Feels very database centered



First Impressions

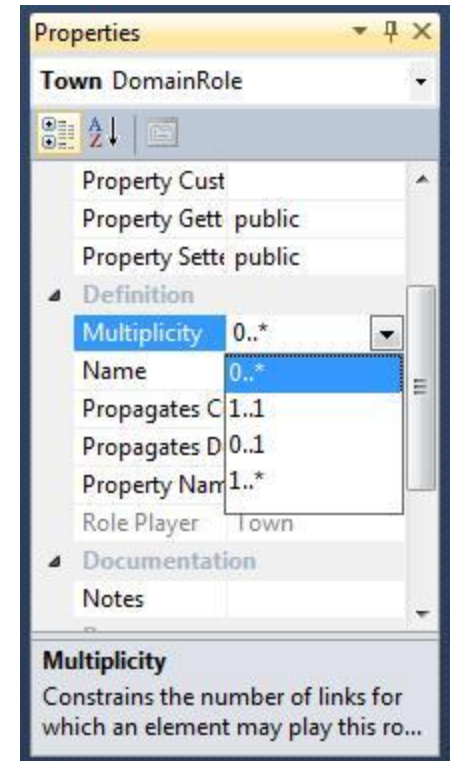
- Steep learning curve

First Impressions

- Steep learning curve
- Hard to find documentation

First Impressions

- Steep learning curve
- Hard to find documentation
- Cardinalities are limited



First Impressions

- Steep learning curve
- Hard to find documentation
- Cardinalities are limited
- **Support for transformations?**