

Automatic XML Schema Generation From UML

A. F. Griesser, Ph.D.

Copyright 2007 by
Prometheus Computing LLC
www.PrometheusComputing.com

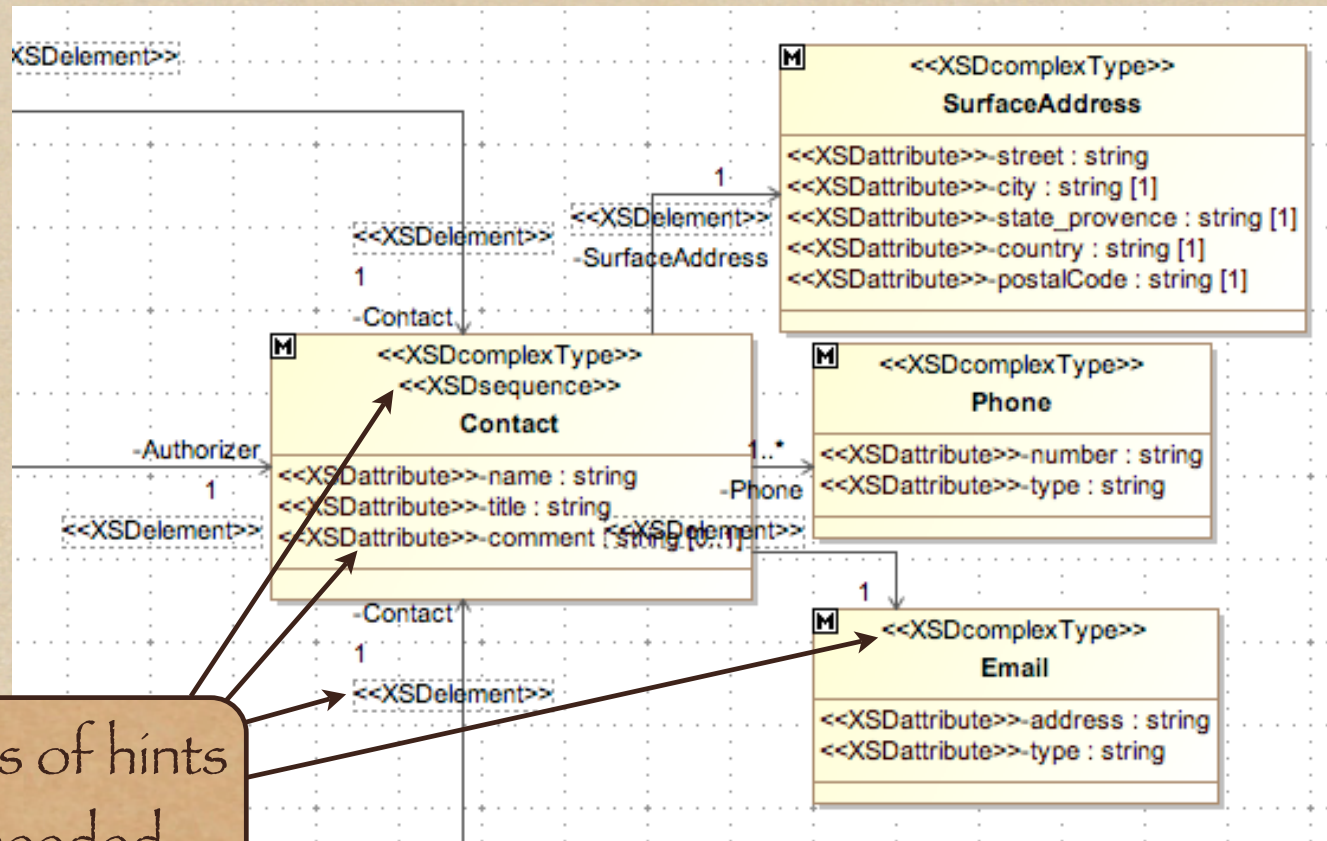
Why Automate?

- ◆ Consistency
- ◆ Productivity
- ◆ Time to market
- ◆ You've got better things to do than make schemas by hand

Existing Automation

- ◆ Highly customizable
- ◆ Difficult to configure & use
- ◆ Poor support for inheritance, interfaces
- ◆ You end up modeling the Schema...
instead of the domain

Modeling the Schema



Lots of hints
needed

Requirements

- ◆ MagicDraw... Tested on:
 - ◆ Version 14, 12.5
 - ◆ Mac (OS X), Ubuntu Linux
 - ◆ Personal Edition, Enterprise Edition
 - ◆ (but not all combinations of above)
- ◆ Expected to work on Windows

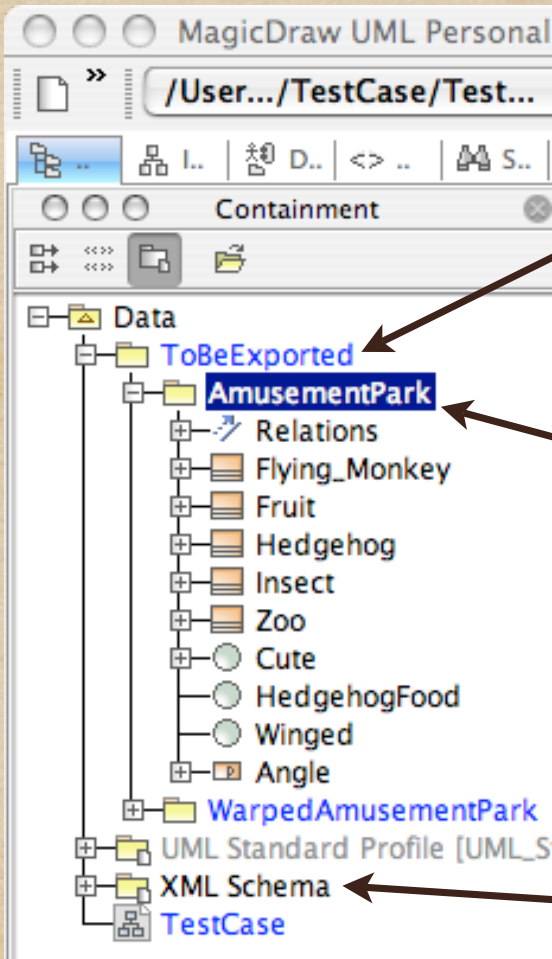
Installation

1. Create “SchemaGen” directory in your home directory
2. Move Zip file contents into MagicDraw Plugins folder
3. Restart MagicDraw

Use - Package Setup

- ◆ Add 'XML Schema' Profile to your project
- ◆ A package that contains:
 - ◆ no packages - becomes a Schema
 - ◆ other packages - used for organizing
- ◆ Add <<XML Schema>> stereotype to packages (to give it necessary tags)
- ◆ Edit the package's tags

Two Kinds of Package



◆ Contains packages

◆ Organizes schemas

◆ Contains no packages

◆ Corresponds to a schema

◆ XML Schema Profile

Use - Modeling

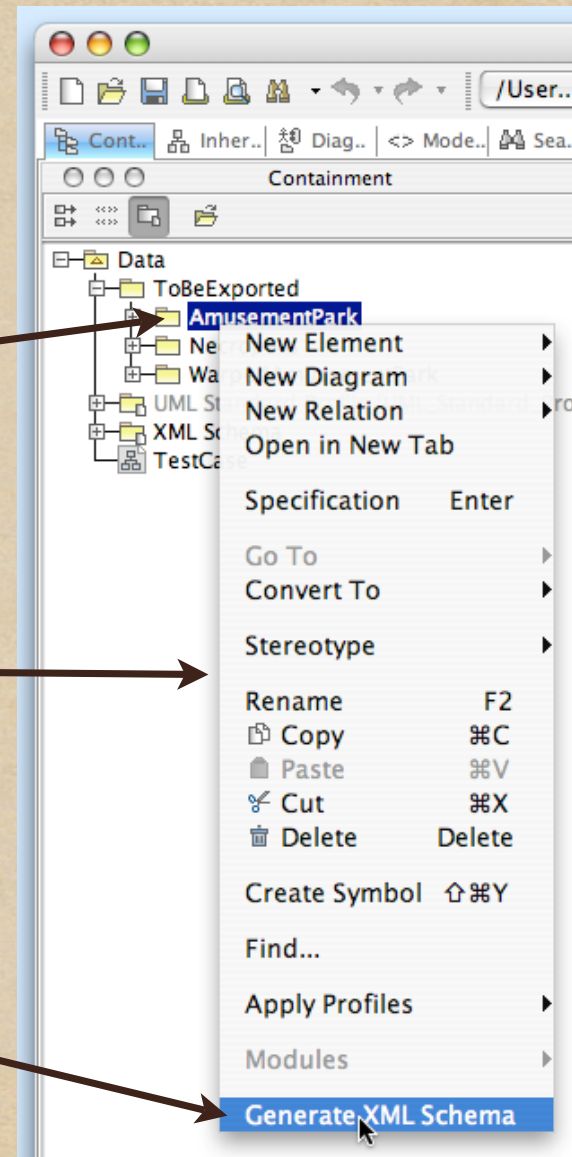
- ◆ Create classes, interfaces inside package
- ◆ Use 'xsd' types for primitives
- ◆ Associations must form a tree
- ◆ Add <<Root>> stereotype to Classes that correspond to root objects

Use - Generate Schema

- ◆ Select a package
- ◆ In containment tree, right click on package
- ◆ From context menu select
 “Generate XML Schema”
- ◆ Schema(s) appear in SchemaGen folder
 (overwrites without warning)

Generating the Schema

1. Select package in containment tree
2. Right click to get context menu
3. Select "Generate XML Schema"



Currently supported

- ◆ Classes (including abstract)
 - ◆ Single inheritance
- ◆ Interfaces, implementation
- ◆ Schemas importing other schemas
- ◆ Extension of classes/interfaces defined in imported schemas
- ◆ Enumerations, restrictions

Not Yet Supported

- ◆ Interface inheritance
- ◆ Graphs other than trees
- ◆ Bidirectional associations
- ◆ Qualifiers, Association Classes
- ◆ Ternary & higher associations
- ◆ Template Class

Possible Future Work

- ◆ Expanded UML Schema Profile
- ◆ Facet validation
- ◆ Alternative UML > XML Schema mappings
- ◆ Support for more UML features
- ◆ Suffixes on different types of elements
- ◆ Configurability, without the learning curve