

Free your work.





Spezifikation und Generierung von graphischen Benutzeroberflächen

Das Projekt

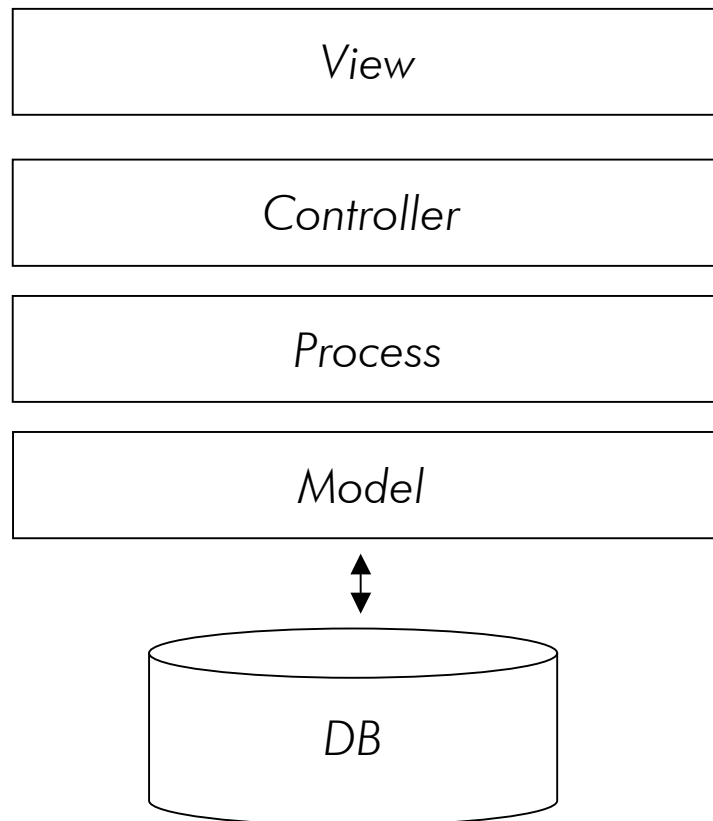
Entwicklung einer „Mobile Workforce“
Lösung

Mobile Erfassung von Messwerten



Die Architektur

Architektur der Anwendung

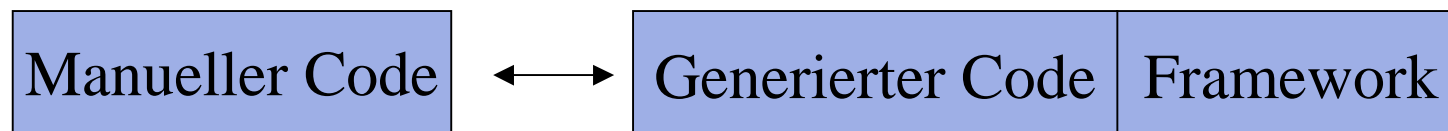


- Traditionelle MVC Architektur
- Unitech PDA
- Windows CE
- .NET compact Framework
- C#
- SQL Server CE

- Eigenentwickeltes Framework
- Codegenerierung aus UML Model

Architektur auf Generierungsverfahren optimieren

- Beispiel: Persistenzschicht
 - Extrem einfach (8 Klassen)
 - Extrem performant
 - Erfüllt nur die wichtigsten Anforderungen (und nur diese)
 - Nicht für andere Projekte wiederverwendbar
 - Komplexität wird in das Generierungsverfahren verschoben



Das Problem

Automatisierte Erstellung der Benutzeroberfläche

- Bisher nur universitäre Ansätze
- oder Erzeugung von Oberflächen Prototypen

- Aber was macht die Erstellung einer Oberfläche eigentlich so aufwendig?

Warum sind Oberflächen so „teuer“?

AUFGABE 0123456789/00
ABGESCHL.

LG AQ PL PP1 PP2

Lg#: 99-999-9999/9

Straße.: 0123456789012345 XXXXX
67890123456789

PLZ: 99999

Ort: 012345678901234567890123

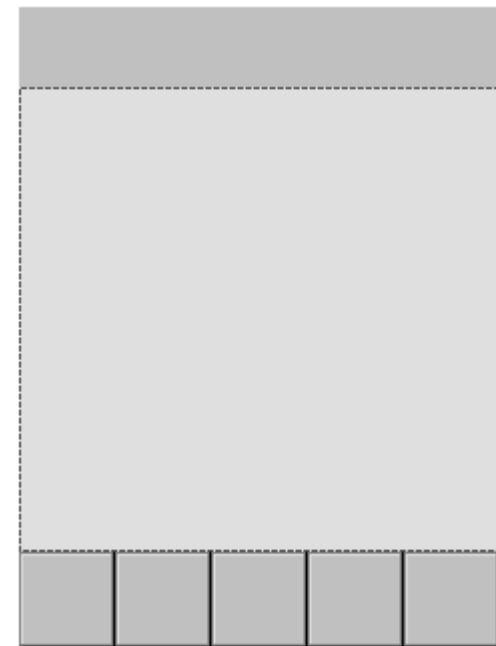
Auftrag: 012345678901234567890123
456789012345678901234567
890123456789012345678901
23456789

- Layout sehr schnell erstellt
- Dynamisierung des Dialogs aufwendig!
 - Kopieren der Daten aus den Entitäten auf die Oberfläche (Konvertierung)
 - Übernahme von Daten-Änderungen in die Entitäten
 - Dynamisches Ein-/Ausblenden und Aktivieren/Deaktivieren von Elementen

Die Lösung

Einheitliches Dialog Design

- Einheitlicher Aufbau
- Definierte Verwendung von wenigen „Controls“
- Navigationskonzept
 - Für nicht-modale Dialoge
 - Für modale Dialoge
- Transaktionskonzept
- Fokussierungskonzept



Idee: Trennung der Dialogspezifikation

- Visuelle Spezifikation mit GUI-Builder
 - Alle statischen Eigenschaften je Sprache
 - Welche Oberflächenelemente? Wo? Wie groß?
 - Welche statischen Texte?
 - Farben und andere dekorative Eigenschaften
- Verhaltensspezifikation mit UML
 - Alle dynamischen Eigenschaft unabhängig von der Sprache
 - Welche Daten werden dargestellt/editiert?
 - Unter welcher Bedingung editierbar?
 - Unter welcher Bedingung aktiviert?

Zusammenhang zwischen Daten und Dialog

ANSPRECHPARTNER
15-995-1700/2

Person Adresse

Typ: Hausmeister

Vorn.: 01234567890123456789012

Nachn.: 01234567890123456789012

Tel.: 01234567890123456789

Mobil: 01234567890123456789

Fax: 01234567890123456789

E-Mail: 01234567890123456789012

Ansprechpartner

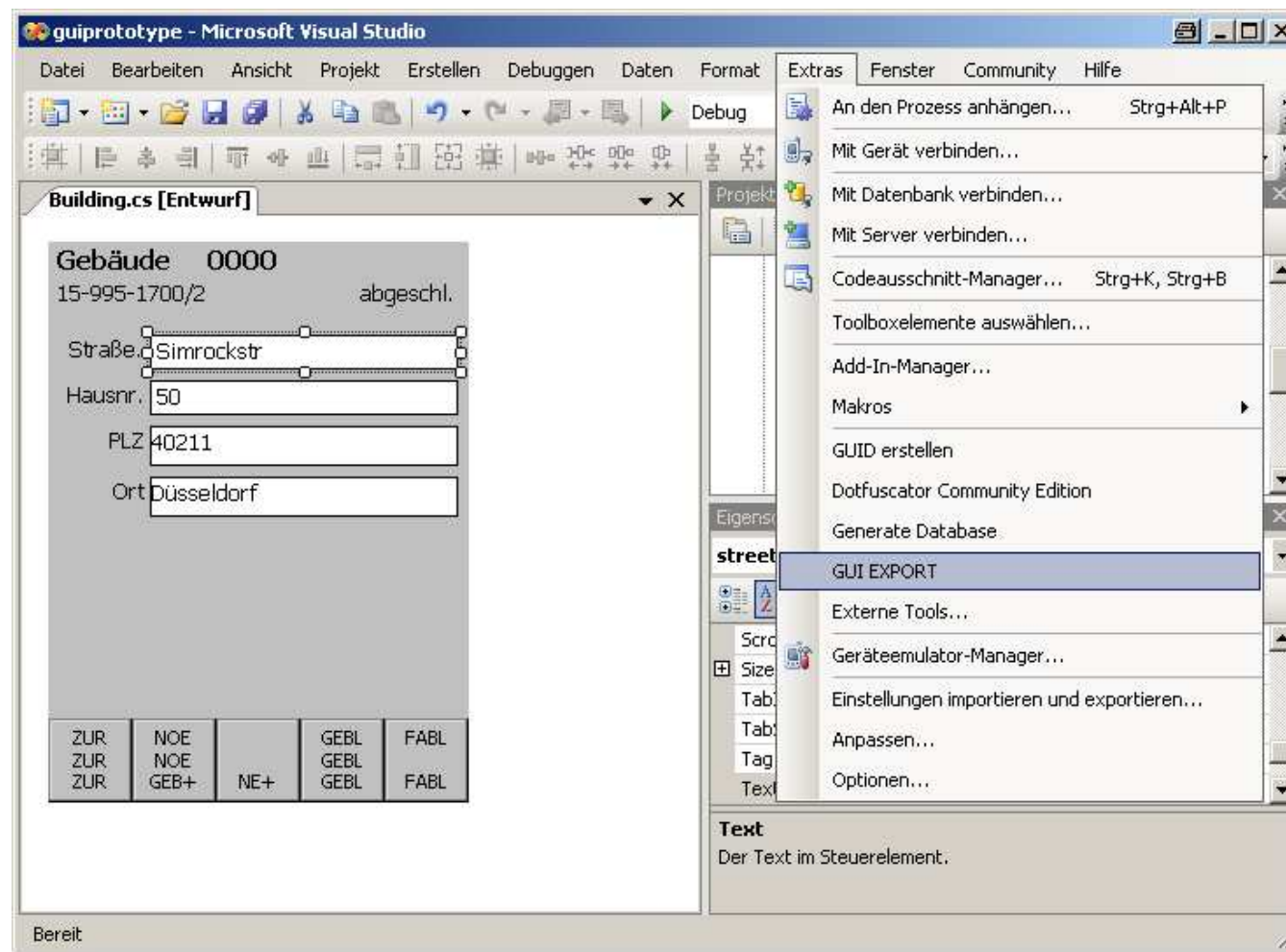
```
<<key>>-typ : AnsprechpartnerTyp [1]
-nachname : Zeichenkette<35> [0..1]
-vorname : Zeichenkette<35> [0..1]
-telefon : Zeichenkette<20> [0..1]
-mobiltelefon : Zeichenkette<20> [0..1]
-fax : Zeichenkette<20> [0..1]
-email : Zeichenkette<120> [0..1]
-strasse : Zeichenkette<40> [0..1]
-hausnummer : Zeichenkette<10> [0..1]
-plz : Zeichenkette<10> [0..1]
-ort : Zeichenkette<35> [0..1]
```

Zusammenhang zwischen Datentypen und Controlls

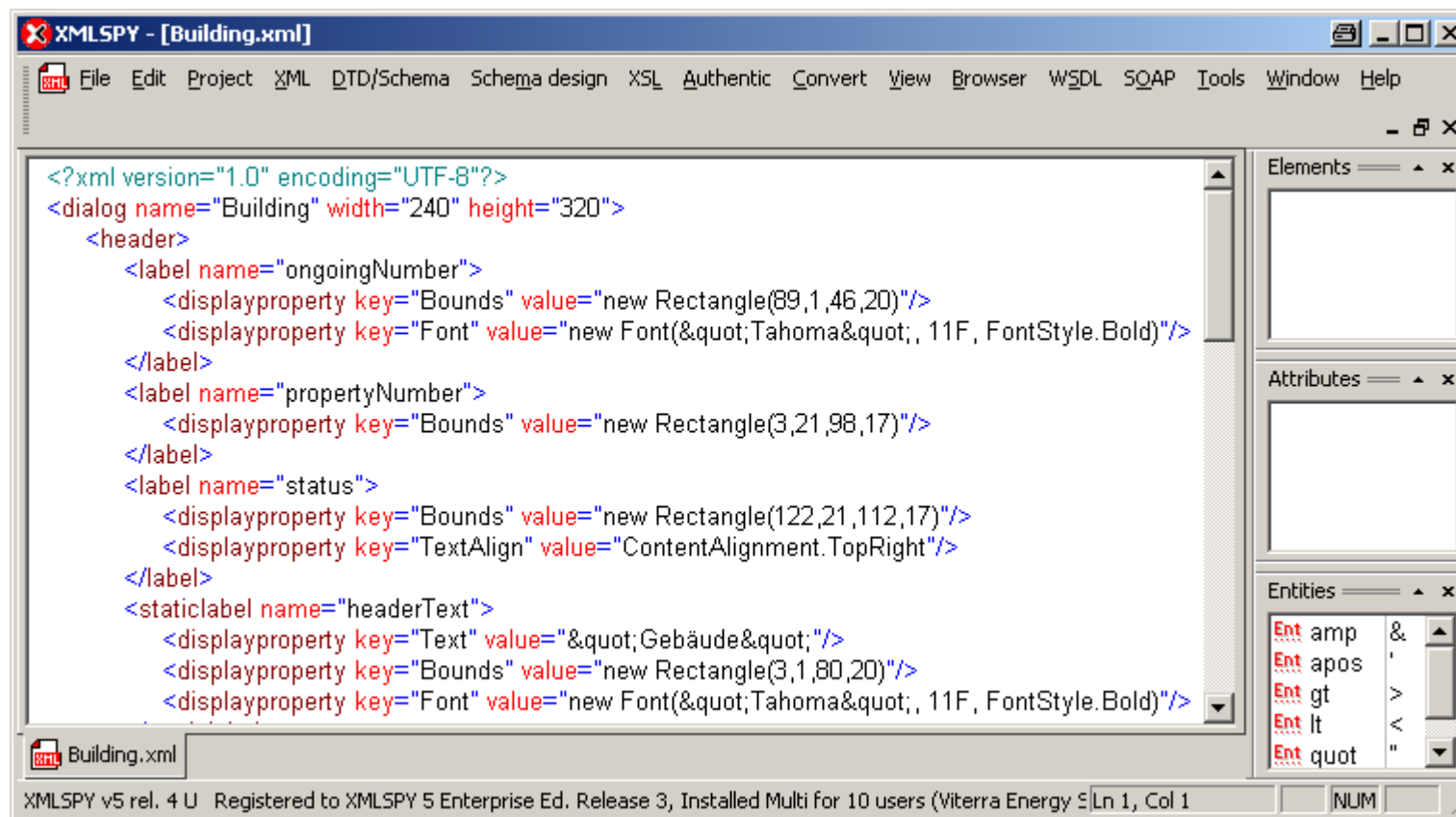
	Zeichenkette	Zahl	Nachkommazahl	Datum	Jahr	WocheJahr	Zeit	DatumZeit	Bild	Aufzählung	BO
Text											
Button											
Textfeld											
Checkbox											
Radio-Gruppe											
Tabelle											
Tabellen-Spalte											
Combobox											
Bild											
Zeichenbereich											

Der Modellierungsprozess

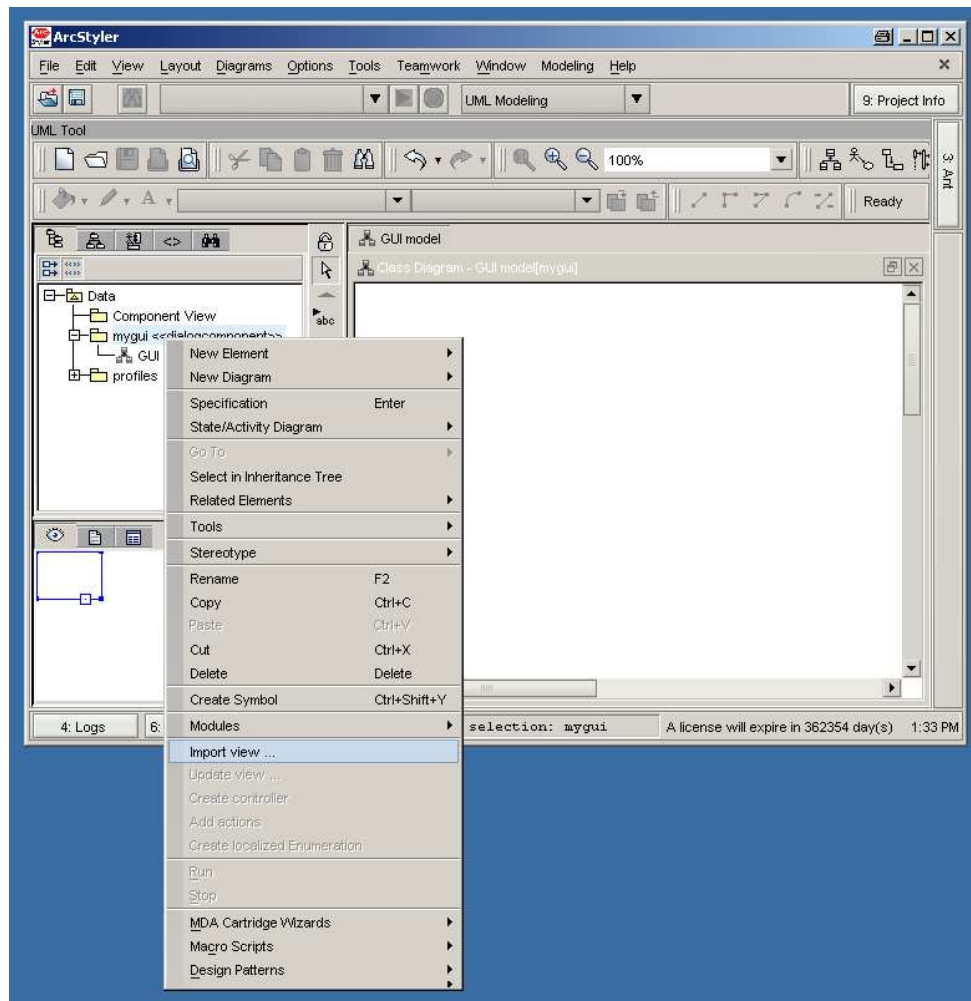
1. GUI Modellierung + XML Export



1.1 Beispiel für XML Datei

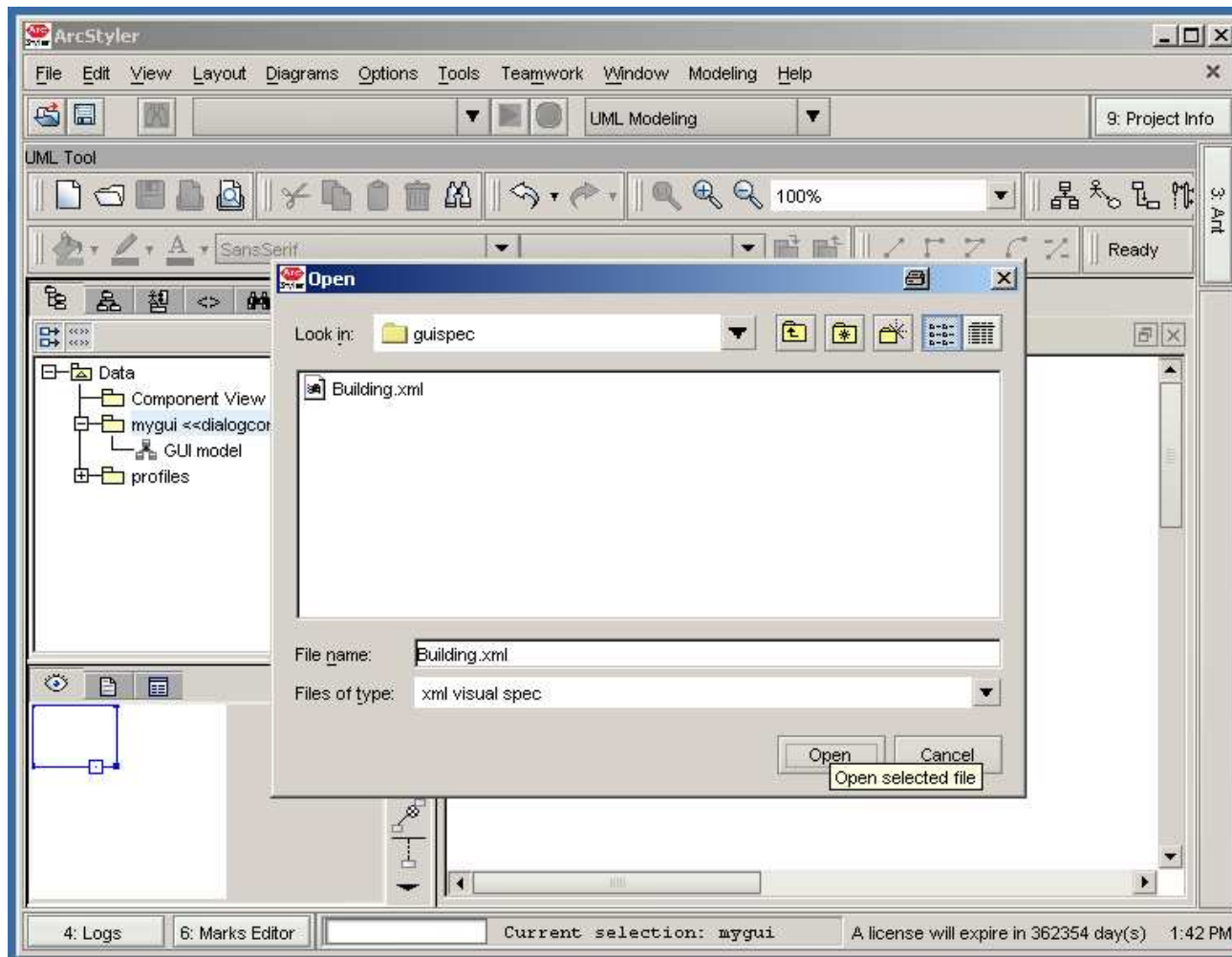


2. Import in das UML Werkzeug

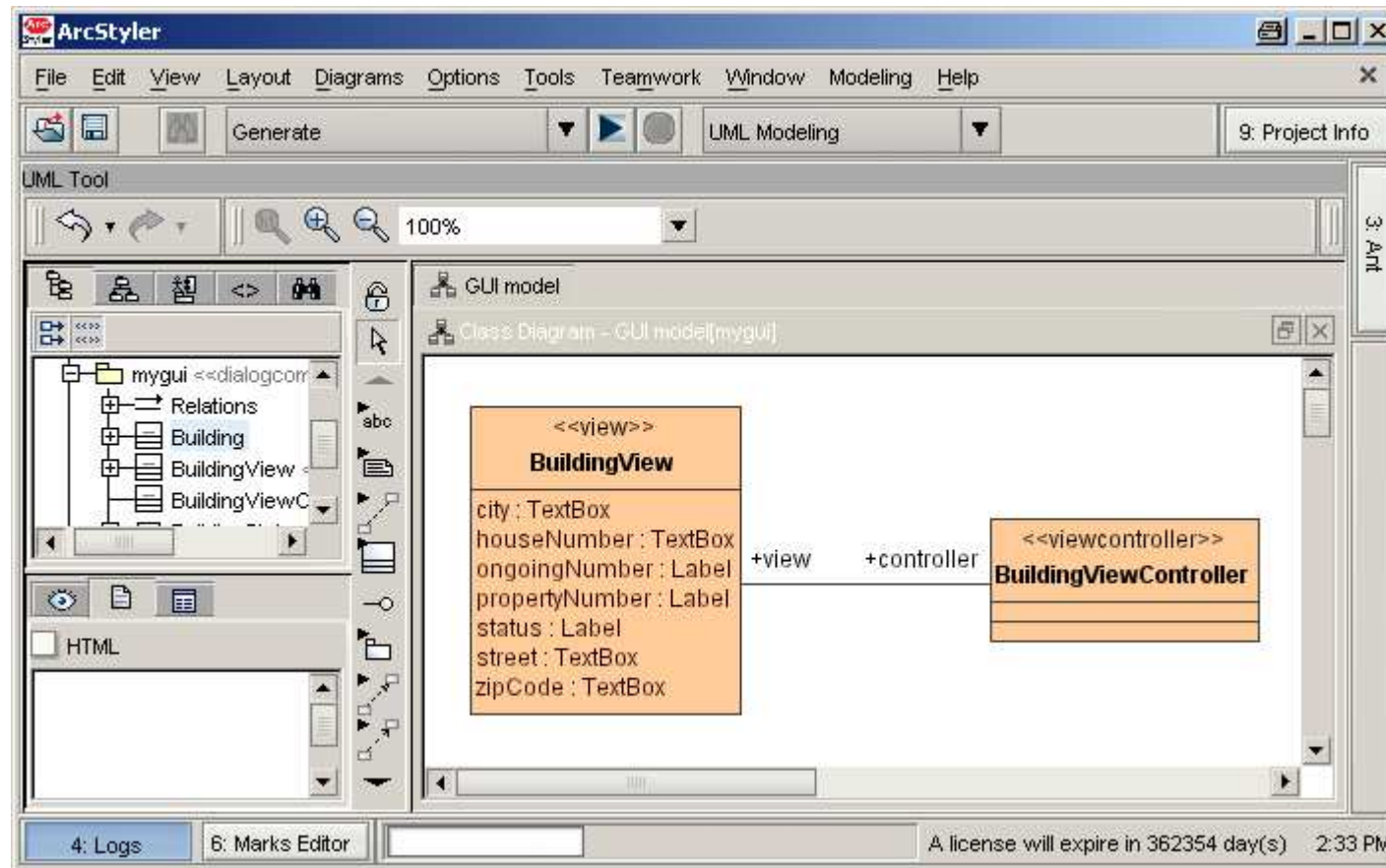


- Packet auswählen
- „Import view“ im Kontextmenü

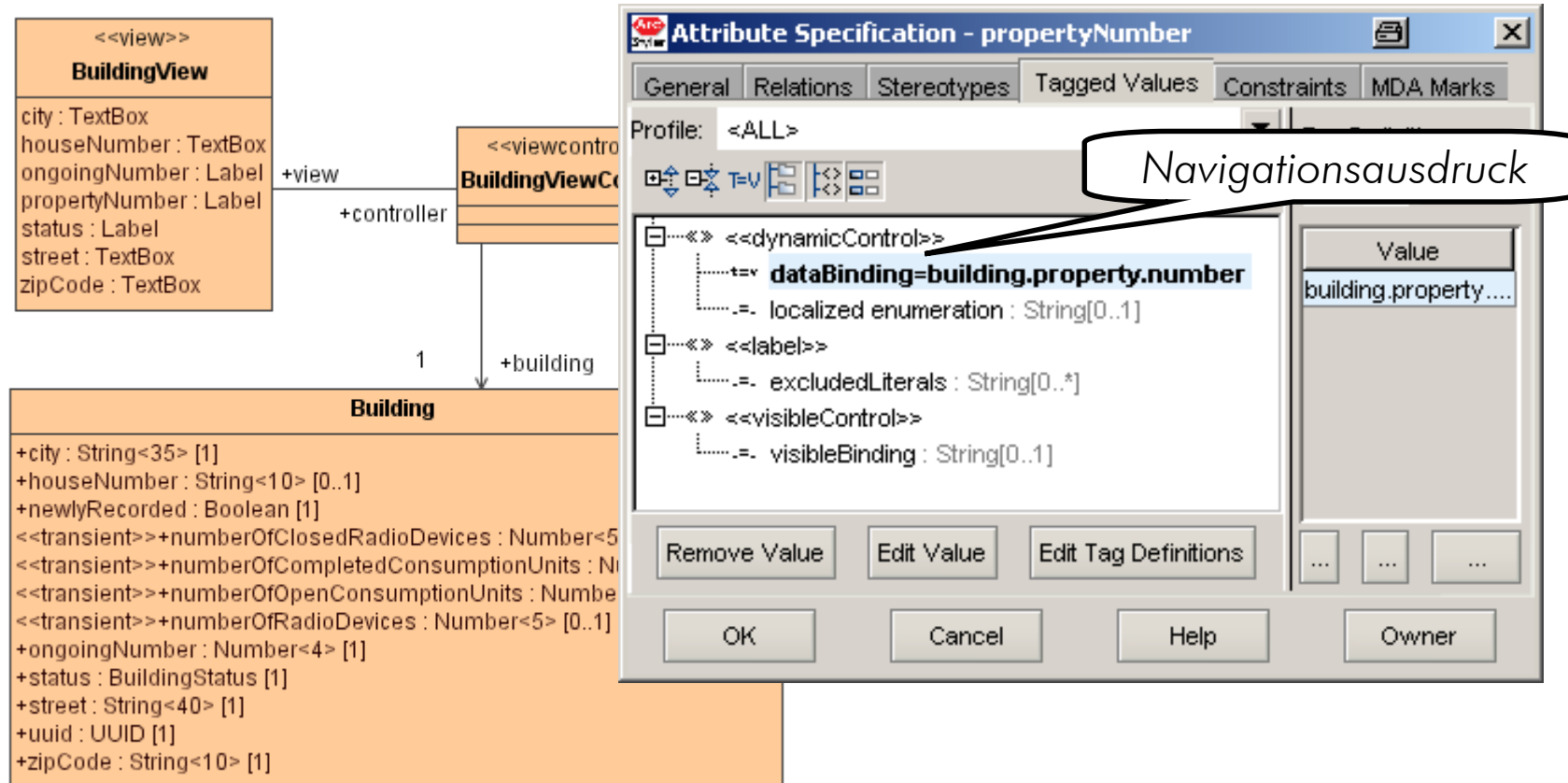
2.1 XML Datei auswählen



3. Erzeugte UML Klassen



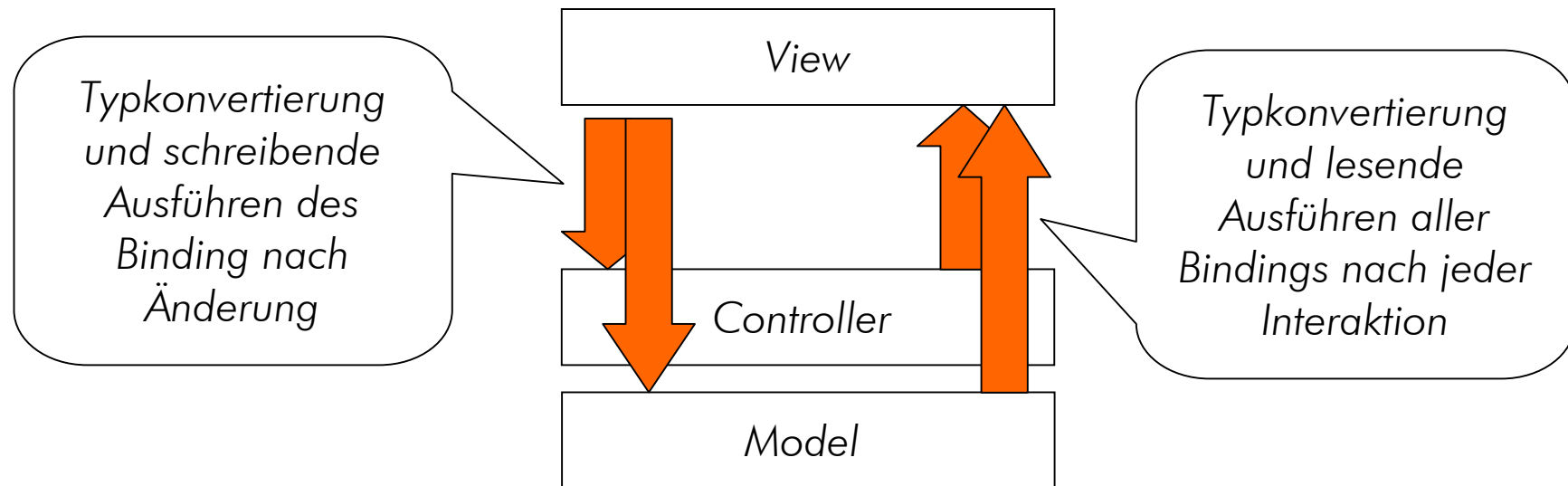
4. Verhaltensspezifikation



Dynamische Eigenschaften eines Controls

- „Data binding“: Navigationsausdruck zu einem darzustellenden/zu speichernden Datenwert
- „Write binding“: Alternativer Navigationsausdruck zum Schreiben
- „Enabled binding“: Navigationsausdruck zu einem Datenwert, der entscheidet, ob Element aktiviert/deaktiviert werden soll
- „Visible binding“: Navigationsausdruck zu einem Datenwert, der entscheidet, ob Element dargestellt werden soll oder nicht

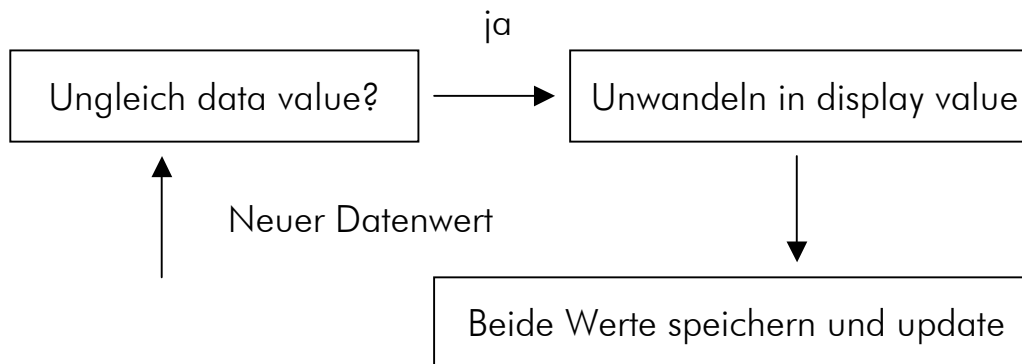
Interaktion Dialog/Controller



Double Buffered Binding

- Ersetzt das „Observer“-Muster zur Aktualisierung der Benutzeroberfläche

Datum: Display value: ‚31.12.2005‘
Data value: 31.12.2005



Das Resultat

Oberflächenanbindung zu 100%
automatisiert

Free your work.

Durch:

Constantin Szallies
constantin.szallies@liantis.com

Liantis GmbH & Co. KG
St.-Anton-Straße 69 - 71
47798 Krefeld
Fon: 0 21 51 / 931 86-60
Fax: 0 21 51 / 931 86-61
info@liantis.com
www.liantis.com

