

Martin Glinz Harald Gall

Software Engineering

Wintersemester 2005/06

Kapitel 3

Modelle



Universität Zürich
Institut für Informatik

3.1 Modelle in der Informatik

3.2 Software Engineering und Modelle

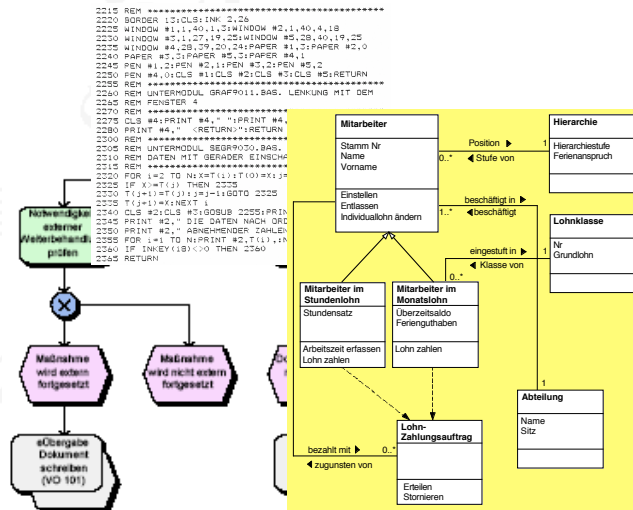
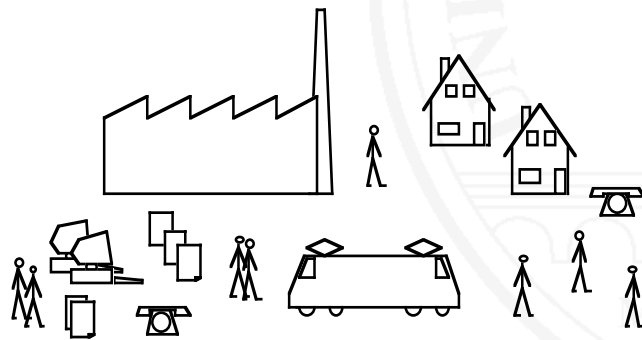
3.3 Modellbildung



Modelle in der Informatik

Modell (model) [in der Informatik] – 1. Konkretes oder gedankliches Abbild eines vorhandenen Gebildes. 2. Konkretes oder gedankliches Vorbild für ein zu schaffendes Gebilde

Original (original) – Das abgebildete oder zu schaffende Gebilde



Charakteristika

- **Nicht wertneutral**
 - Abbild oder Vorbild in der Wahrnehmung der beteiligten Personen für einen bestimmten Verwendungszweck
- **Größtmögliche Ähnlichkeit** zwischen Original und Modell **kein Ziel**
 - Bewusste Abstraktion und Gestaltung des Modells
- **Validierung erforderlich**
 - Alle relevanten Eigenschaften des Originals adäquat und vollständig auf Eigenschaften des Modells abgebildet?
- ⇒ Die Vorstellung «Modell = Ausschnitt der Realität» greift zu kurz («naiver Realismus»)
- ⇒ Modelle sind **Abbildung** und **Konstruktion** der Realität

3.1 Modelle in der Informatik

3.2 Software Engineering und Modelle

3.3 Modellbildung



Modelle im Software Engineering

- **Modellierung gehört zum Fundament des Software Engineerings**
 - Software ist vielfach (immer?) selbst ein Modell
 - Anforderungen sind Modelle der Problemstellung
 - Architekturen und Entwürfe sind Modelle der Lösung
 - Testvorschriften sind Modelle des korrekten Funktionierens des Codes
 - usw.
- ⇒ „Die Artefakte der Software-Entwicklung sind Modelle“ (Jürgen Ebert)
- ⇒ Wer Software entwickelt oder pflegt, braucht **solides Wissen und Können in Modellierung**

Der Stoff aus dem Teil Modellierung der Vorlesung Informatik II [http://www.ifi.unizh.ch/req/courses/inf_II] wird in dieser Vorlesung vorausgesetzt. In diesem Kapitel werden nur einige Kernpunkte repetiert.

Die Rolle der Modellierung im Software Engineering

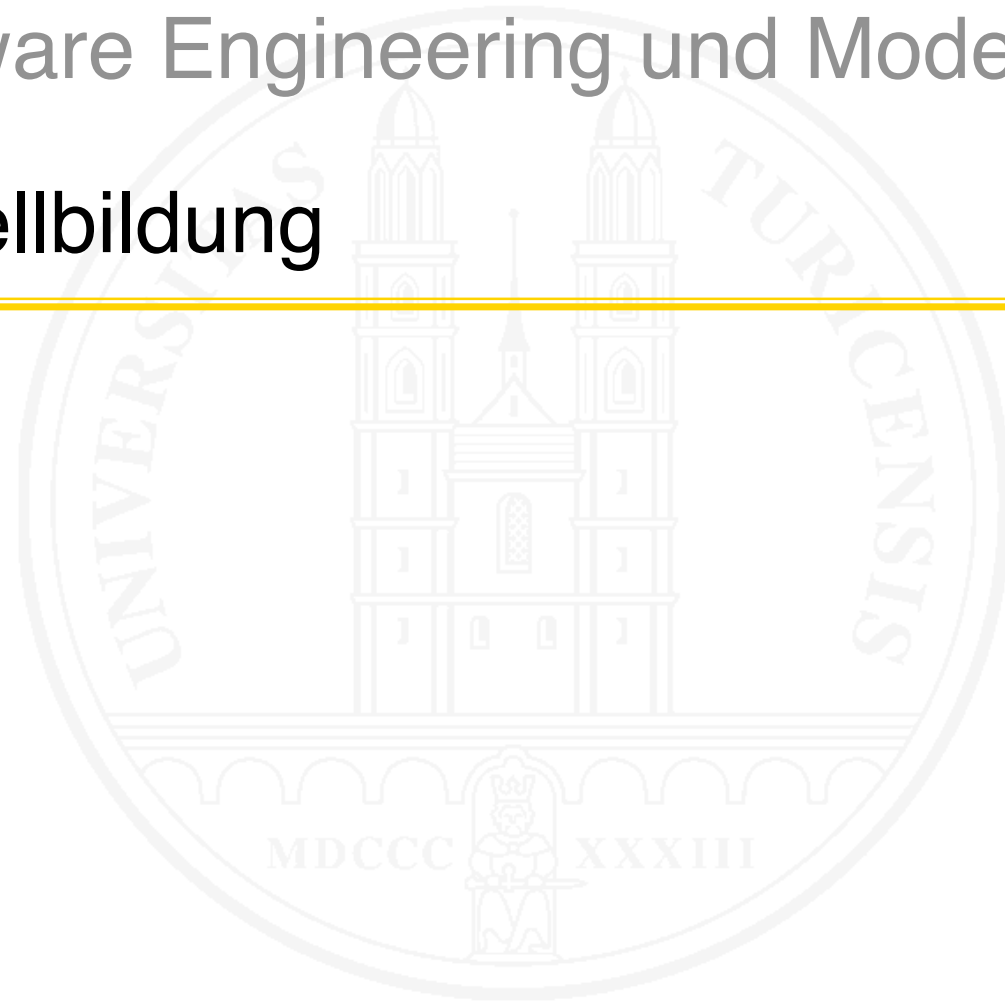
- Modellierung **liefert**
 - ein grundlegendes Verständnis der Natur von Software-Artefakten
 - eine Reihe grundlegender Sprachen und Methoden zur Problembeschreibung und -analyse
 - grundlegendes Problemlösungswissen
 - konzeptionelle Grundlagen für den Bau von Software-Werkzeugen

- Modellierung **liefert nicht**
 - das notwendige domänenspezifische Wissen
 - das notwendige Informatik-Fachwissen (Algorithmen, Datenstrukturen, ...)
 - das notwendige Prozesswissen (Arbeitstechniken und Verfahren)
 - das notwendige Führungswissen

3.1 Modelle in der Informatik

3.2 Software Engineering und Modelle

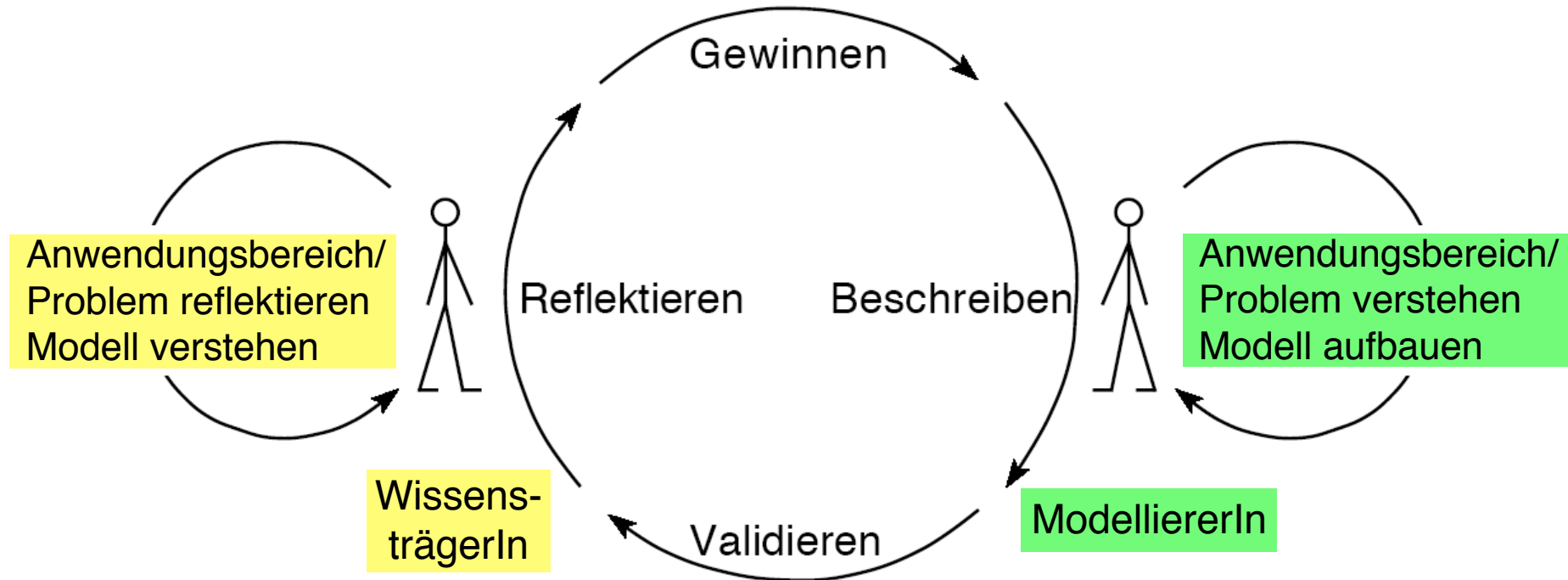
3.3 Modellbildung



Was ist Modellbildung

- **Modellbildung** – Prozess der Erstellung eines Modells
- Zwei Rollen:
 - **WissensträgerIn** – Person, welche das Wissen über den zu modellierenden Gegenstand bzw. Gegenstandsbereich (das Original im Sinn der Modelltheorie) hat
 - **ModelliererIn** – Person, welche ein Modell erstellt
- In jeder Rolle kann es mehrere reale Personen geben
- Eine Person kann beide Rollen gleichzeitig haben

Prinzipschema der Modellbildung



- Modellbildung ist ein **iterativer** Prozess
- Modellieren bedeutet immer auch **Reflektieren** über das Original – unabhängig ob dieses bereits existiert oder erst zu schaffen ist
- Modellbildung ist auch ein **Verstehens-** und **Konsensbildungsprozess**

Tätigkeiten in der Modellbildung

- **Reflektieren** – Überlegen und verstehen, was modelliert werden soll (Pragmatik des Modells, abzubildende/wegzulassende Merkmale, Umfang,...)
- **Gewinnen** – Informationen über das Original und die Intentionen der Wissensträger gewinnen (Diskutieren, lesen, fragen, rückfragen, suchen, analysieren, ...)
- **Beschreiben** – Gewonnene Informationen verstehen, ordnen, strukturieren, bewerten,... und mit geeigneten Mitteln beschreiben
- **Validieren** – Modelle (Zwischenergebnisse und fertiges Modell) durch Wissensträger überprüfen lassen: Ist es das, was sie wollen und brauchen?

Deskriptive und präskriptive Modellbildung

- **Deskriptive** Modellbildung
 - Modellierung eines **existierenden** Originals
oder
 - Modellierung eines **zukünftigen**, aber **nicht gestaltbaren** Originals

Beispiel: Komponentenstruktur eines im Einsatz befindlichen Informatiksystems

- **Präskriptive** Modellbildung
 - Modellierung eines zu **schaffenden**, **gestaltbaren** Originals

Beispiel: Anforderungsspezifikation für zu entwickelnde Software

Deskriptive und präskriptive Modellbildung – 2

- Deskriptive Modellbildung muss sich **streng an der Realität orientieren**
- Präskriptive Modellbildung darf **zukünftige Realität gestalten**
- Deskriptiv und präskriptiv sind Eigenschaften der **Modellbildung**, nicht der Modelle selbst:
dasselbe Modell kann **deskriptiv bezüglich eines Originals** und **präskriptiv bezüglich eines anderen Originals** sein

Beispiel:

Eine Anforderungsspezifikation ist *deskriptiv* bezüglich des Problems, das ein Kunde gelöst haben möchte, und *präskriptiv* gegenüber den zu erstellenden Entwürfen und Programmen

Literatur

Glinz, M. (2005). *Informatik Ila: Modellierung*. Folienskript, Universität Zürich.

http://www.ifi.unizh.ch/req/courses/inf_II/ (Folien herunterladbar im Abschnitt „Unterlagen“)

Ludewig, J. (2003). Models in Software Engineering – An Introduction. *Software and Systems Modeling* 2, 1.

Kent, W. (1978). *Data and Reality*. Amsterdam etc.: North-Holland.

Stachowiak, H. (1973). *Allgemeine Modelltheorie*. Wien: Springer.

Machen Sie sich mit dem Stoff des Teils Modellierung der Vorlesung Informatik II [http://www.ifi.unizh.ch/req/courses/inf_II/] vertraut.