

Author

Title

Sub/Work Title

University of Applied Sciences: Vorarlberg

Department of Computer Science
Professor / Supervisor

Dornbirn, 2014

Abstract

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Inhaltsverzeichnis

Abstract	iii
1 Einführung	1
1.1 Was ist Software?	1
1.2 Beobachtungen von Modellen	1
1.3 Probleme von Modellen	1
1.4 Wärs nicht schön wenn ...?	2
1.5 Sinnhaftigkeit	2
1.6 Modellkontext	2
1.7 Model Scope	2
1.8 Modelzweck	3
1.9 Modelvollständigkeit	3
1.10 Modelltypen	3
1.11 Metamodell-Architektur	4
2 Zusammenfassung	5

Listings

Abbildungsverzeichnis

1.1	Folien Bernd Wenzel	3
1.2	Folien Bernd Wenzel	4

Todo list

1 Einführung

1.1 Was ist Software?

Artifakte im Kontext einer Domäne (eine Beschreibung davon was wir machen) und im Kontext einer Technologie (eine Beschreibung davon wie wir etwas erreichen). Seit 1950 versucht man die Technologie zu abstrahieren um sich auf die jeweilige Domäne konzentrieren zu können (OPCodes, Speichermanagement, ...). Jede zusätzliche Abstraktionsschicht hat allerdings auch neue Probleme mit sich gebracht z.B. schlechtere Performanz, größerer Speicherbedarf, etc.

Mit der Zeit wurden diese Probleme beseitigt und auch die Hardware hat sich dementsprechend weiterentwickelt (Speichergröße, CPU-Leistung, ...).

1.2 Beobachtungen von Modellen

- auch Quellcode ist ein Model
- Graphische Modelle
 - Kontrollflussmodelle
 - Bachmanndiagramme
 - UML
 - ...

Modelle werden verwendet um Anforderungen zu Beschreiben, eine Situation zu erfassen oder inzwischen auch um Quellcode daraus zu generieren.

1.3 Probleme von Modellen

Oft erkennt man nicht den Nutzen eines Modells (oder auch den Grad der Vollständigkeit). Außerdem können Modelle nicht nur richtig oder falsch sein sondern auch irrelevant.

1.4 Wärs nicht schön wenn ...?

... es eine Umgebung gibt in welcher man

- aus einem Modell eine vollständige Software generieren könnte
- ein Model ausführen kann ohne dafür Code generiern zu müssen
- Plattformunabhängig ist und somit wiederverwendbar

1.5 Sinnhaftigkeit

- die Möglichkeit Modelle in Code umwandeln gibt es breits
- Transformation von abstrakten Code zu einem konkreten Model wird breits von Kompilern gemacht
- Parametrisierung von Modelltransformationen wird von STP und den meisten Kompilern unterstützt

1.6 Modellkontext

Klassifizierung von Artifakten anhand von Domänen

- IT Domäne
- Applikationsdomäne (Bank, Regierung, Kommerzielles)

1.7 Model Scope

- Strukturelles Modelle (Klassenmodell, Komponentemodell, ...)
- Verhaltens Modelle (Usecases)
- Installations Modelle

1.8 Modelzweck

- Illustrationsmodell (Verständlichkeit)
- Anforderungsmodell (Anforderungen an einen Zustand)
- Analysemodell (Repräsentation eines Zustands)
- Designmodell (Zustand den man haben möchte)

1.9 Modelvollständigkeit

Klassifizierung anhand des Grades an Vollständigkeit

- Unspezifisch
- Funktion-Vollständigkeit (z.B. alle Operationen in einem UML-Modell aber nicht zwangsläufig eine Implementierung)
- Ressourcen-Vollständigkeit (Vollständig im Sinne von Ressourcen die funktionale Anforderungen implementiert haben)

1.10 Modelltypen

Model Type	Scope	Context	Purpose	Completen.
Abstract Business Proc. Model	Applic. Domain	Process / Act.	Requirements	functionally complete
Concrete Business Model	Applic. Domain	Structure and/or Behaviour	Analysis or design	resource complete
Ideal PIM	IT Domain	Structure and/or Behaviour	Design	functionally complete
Ideal PSM	IT Domain	Structure and/or Behaviour	Design	resource complete

Abbildung 1.1: Folien Bernd Wenzel

1.11 Metamodell-Architektur

- Model - Formale Definition von Struktur, Verhalten und Einschränkungen für eine Gruppe von Objekten (in einem Moodell) gelten.
- Metamodell - Formale Definition von Struktur, Verhalten und Einschränkungen für eine Gruppe von Modellelementen.
- Metalevel - M0 (Daten), M1 (Modelle - Datenbanken), M2 (Metamodelle - Programmiersprache, UML), M3 (Metametamodelle - Selbstbeschreibend)

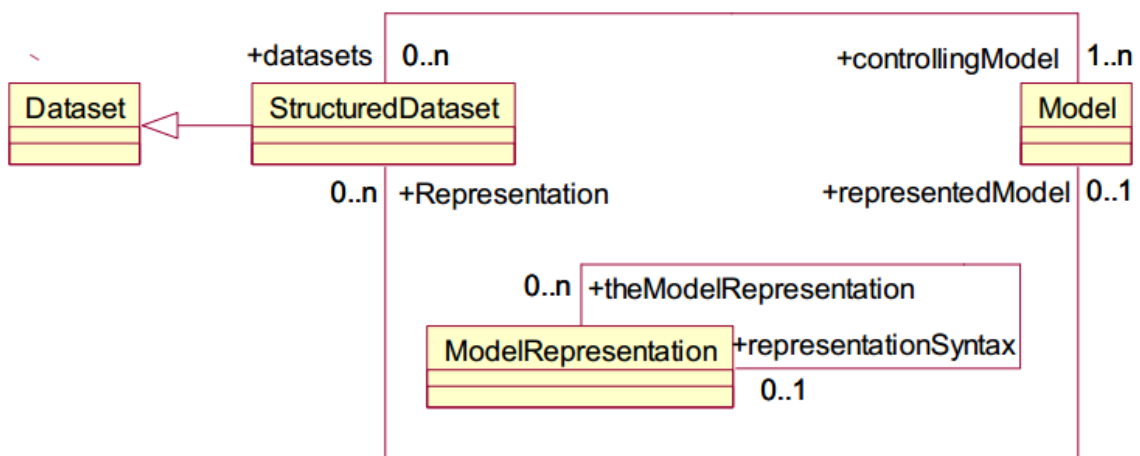


Abbildung 1.2: Folien Bernd Wenzel

2 Zusammenfassung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.