
# Was ist ein Modell? (Und wenn ja, wieviele?)

Version 1.01

*Dieses Dokument wurde verfasst von …*
Dr. Jürgen Pitschke, BCS-Dr. Jürgen Pitschke, www.enterprise-design.eu

**Inhaltsverzeichnis**

# 1 Was ist ein Modell?

Die einfachsten Begriffe bereiten oft die größten Probleme. Wenn wir über visuelle Modellierung mit BPMN, UML, SysML und anderen Notationen sprechen, benutzen wir das Wort „Modell" sehr häufig. Bei unkritischer Betrachtung und manchmal unterstützt durch verwendete Werkzeuge wird der Begriff „Diagramm" synonym verwendet. Dabei wird aber Form und Inhalt vermischt. Wichtig ist der Inhalt. Oft kann derselbe Inhalt in verschiedenen Formen dargestellt werden – ein Modell kann durch verschiedene Diagrammtypen dargestellt werden.

Das vorliegende Dokument ist durch Fragen meiner Studenten motiviert. Das Verständnis für den Begriff Modell ist wichtig. Man erkennt leicht, wann dieses Verständnis nicht da ist. Dann finden sich Arbeitspakete mit Bezeichnungen wie „Klassendiagramm entwickeln" im Projektplan. Ist das Modellverständnis dagegen vorhanden, würde an gleicher Stelle das Arbeitspaket vielleicht mit „Informationsmodell Endanwenderportal erstellen" bezeichnet. Die Diskussion erfolgt im Zusammenhang von ***visuellen Modellen***, d.h. der grafischen Darstellung von Prozessen, Systemen, Anforderungen usw. mit Hilfe von Standardnotationen wie BPMN, UML, SysML und anderen.

## 1.1 Der Begriff „Modell"

Welche Bedeutung hat der Begriff Modell? Wiktionary gibt im Englischen 7 verschiedene Bedeutungen an:

"**model** (*plural* **models**)
1. A person who serves as a subject for artwork or fashion, usually in the medium of photography but also for painting or drawing.
   *The beautiful model had her face on the cover of almost every fashion magazine imaginable.*
2. A miniature representation of a physical object.
   *The boy played with a model of a World War II fighter plane.*
3. A simplified representation (usually mathematical) used to explain the workings of a real world system or event.
   *The computer weather model did not correctly predict the path of the hurricane.*
4. A style, type, or design.
   *He decided to buy the turbo engine model of the sports car.*
5. The structural design of a complex system.
   *The team developed a sound business model.*
6. A praiseworthy example to be copied, with or without modifications
   *British parliamentary democracy was seen as a model for other countries to follow*
7. (*logic*) An interpretation function which assigns a truth value to each atomic proposition."[1]

In der deutschen Version von Wiktionary reicht es immerhin für drei Bedeutungen:

**„Modell**
1. pragmatisch (auf relevante Eigenschaften) verkürzte (verkleinerte) Abbildung
2. Muster, das vervielfältigt wird
3. Person, die z.B. einem Künstler als Vorlage, Modell dient"[2]

Der Brockhaus kennt ebenfalls 7 Bedeutungen des Begriffs „Modell".

Wenn wir über visuelle Modellierung sprechen sind die Definitionen 3, 5 und 6 der englischen Version von Interesse. Auch Definition 2 und 7 können von Belang sein.

[1] http://en.wiktionary.org/wiki/model, 31.08.2009
[2] http://de.wiktionary.org/wiki/Modell, 31.08.2009

## 1.2 Modell („Einfaches Modell")

Unter einem Modell wollen wir im Weiteren immer eine Darstellung im Sinne der Definition 3 verstehen. Ein Modell ist eine vereinfachte Darstellung eines uns interessierenden Gegenstandes. Ein Modell entsteht durch Abstraktion – wir ignorieren bestimmte Eigenschaften des zu untersuchenden Gegenstandes, um eine bestimmte Eigenschaft hervorzuheben. Um ein solches Modell von kompositen Modellen (siehe Abschnitt 1.3) abzugrenzen, verwenden wir auch den Begriff „einfaches Modell"

Ein gutes einfaches Modell stellt genau eine Sicht auf den zu untersuchenden Gegenstand dar. Das Vermischen von Sichten in einem Modell führt immer zu unverständlichen und nicht pflegbaren Modellen. Wenn wir uns z.B. mit der Modellierung von Geschäftsprozessen befassen, so soll das Modell entweder den Ist-Zustand oder den Soll-Zustand des Prozesses beschreiben, nicht aber beides in einem Modell. Wenn wir Prozessdekomposition betreiben, so soll ein Modell genau eine Detailstufe darstellen, aber nicht Makroprozess und Aufgabenebene vermischen. Bei der Betrachtung von Geschäftsprozessen spielen verschiedene Aspekte eine Rolle, z.B. Steuerfluss innerhalb eines Prozesses, Informationsaustausch zwischen Prozessbeteiligten. Diese Aspekte sollen getrennt dargestellt werden. Dem trägt z.B. die Version 2.0 der BPMN Rechnung. Die neuen Diagrammtypen gestatten die Trennung von verschiedenen Sichten auf einen Geschäftsprozess besser als die vorangegangene Version.

Aus diesen Überlegungen lassen sich Grundprinzipien für die Modellierung ableiten:
- Ein gutes (einfaches) Modell stellt genau einen Aspekt des zu betrachtenden Gegenstandes dar.
- Unabhängige Konzepte werden voneinander getrennt betrachtet. Z.B. Trennung von Geschäftsprozess und Geschäftsregeln.
- Stabile Konzepte werden von instabilen Konzepten getrennt dargestellt, z.B. Rollen in einem Geschäftsprozess getrennt von der Organisationsstruktur.

## 1.3 Komposite Modelle

Natürlich reicht die Darstellung einer einzelnen Sicht (ein einfaches Modell) für die Beschreibung eines Anwendungsbereichs niemals aus. Eine für unseren jeweiligen Zweck ausreichende Beschreibung des „uns interessierenden Gegenstandes" wird daher immer eine Kombination verschiedener einfacher Modelle beinhalten. Solche „Zusammengesetzte Modelle" nennen wir komposite Modelle. Sie werden solche kompositen Modelle bereits kennen. Bezeichnungen sind z.B. Unternehmensmodell oder Systemmodell. Der Begriff „komposites Modell" entspricht der Definition 5 aus Wiktionary.

Unser Vorgehensmodell nutzt das Zachman-Framework als Architekturmuster. Daher ordnen wir einfache und komposite Modelle den Perspektiven, Sichten und Zellen des Zachman-Frameworks zu. Ein einfaches Modell ist immer einer einzelnen Zelle und einer einzelnen Detailsicht zugeordnet. Abschnitt 1.4 diskutiert das weiter.

Beispiele für komposite Modelle:
- Unternehmensmodell: umfasst Modelle für alle Perspektiven und Sichten des Zachman-Frameworks
- Business Model (Geschäftsmodell): umfasst alle einfachen Modelle der Perspektive „Business Concepts"
- Systemmodell: umfasst alle einfachen Modelle der Perspektiven „System Logic" und „Technology Physics"
- Modell eines „Service"[3]: Kombination von Einzelmodellen der Perspektiven „System Logic" und „Technology Physics"

[3] Service im Sinne einer Service Oriented Architecture (SOA)

Komposite Modelle setzen sich immer aus mehreren einfachen Modellen zusammen. Wir benötigen daher Möglichkeiten, die Zusammenhänge zwischen Modellen (einfachen und kompositen) zu beschreiben. Ein Modell als Verfeinerung eines anderen Modells, die Abbildung von Ist- auf Soll-Modell oder die Zusammenfassung von Modellen, die zusammen das „Business Model" etablieren. Auch diese Aspekte werden häufig durch visuelle Modelle beschrieben – eine „Prozesslandkarte" beschreibt Zusammenhänge zwischen einzelnen Prozessmodellen, ein „Übersichtsmodell" zeigt die Zusammenhänge zwischen verschiedenen Sichten auf einen Gegenstand.

## 1.4 Modelle und das Zachman-Framework

Das Zachman-Framework ist ein Architekturmuster, dargestellt durch eine Matrix. Es definiert verschiedene Perspektiven (Zeilen) und verschiedene Sichten in jeder Perspektive (Spalten). Gegenstand des Frameworks ist dabei das gesamte Unternehmen. Abbildung 1 zeigt das Framework.

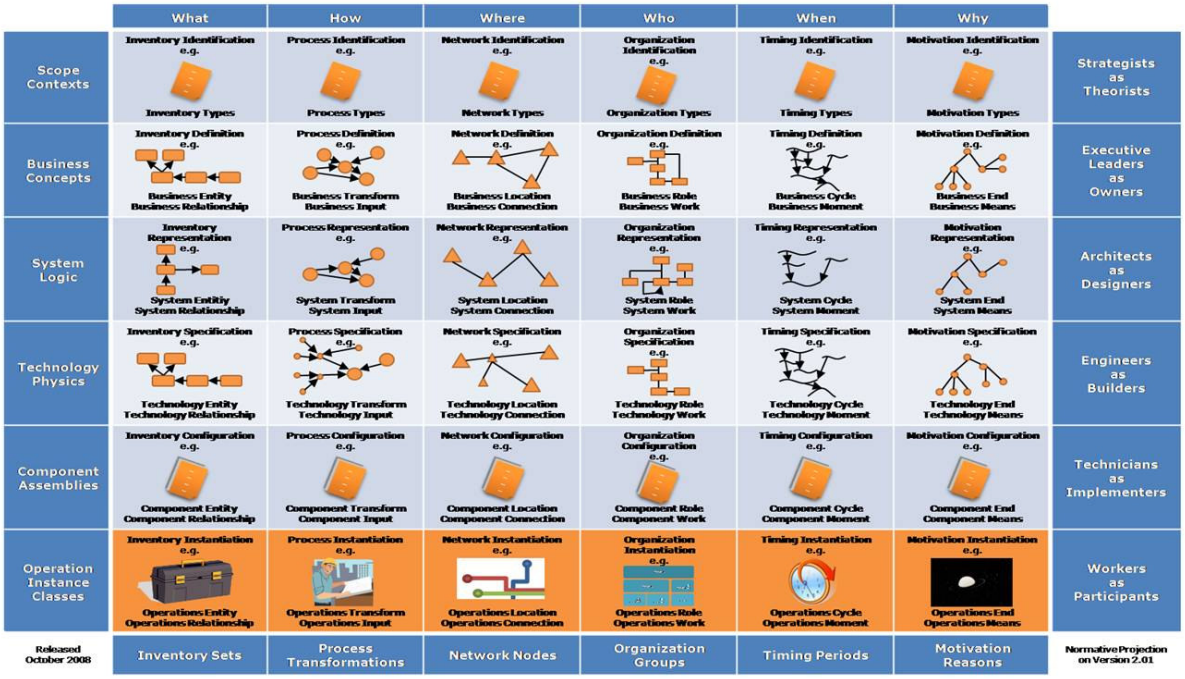


**Abbildung 1: Zachman™-Framework[4], Quelle: www.zachmaninternational.com**

Ein häufiges Missverständnis über das Zachman-Framework ist, dass die Zeilen von oben nach unten mehr Detail enthalten. Das ist falsch. Jede Zeile definiert eine neue Perspektive. Das Anwachsen von Detail geschieht innerhalb einer Zelle, das Framework ist eigentlich 3-dimensional. Abbildung 2 versucht das zu verdeutlichen.

[4] John Zachman, The Zachman Framework For Enterprise Architecture: Primer for Enterprise Engineering and Manufacturing, Zachman International, 2006, Electronic Book

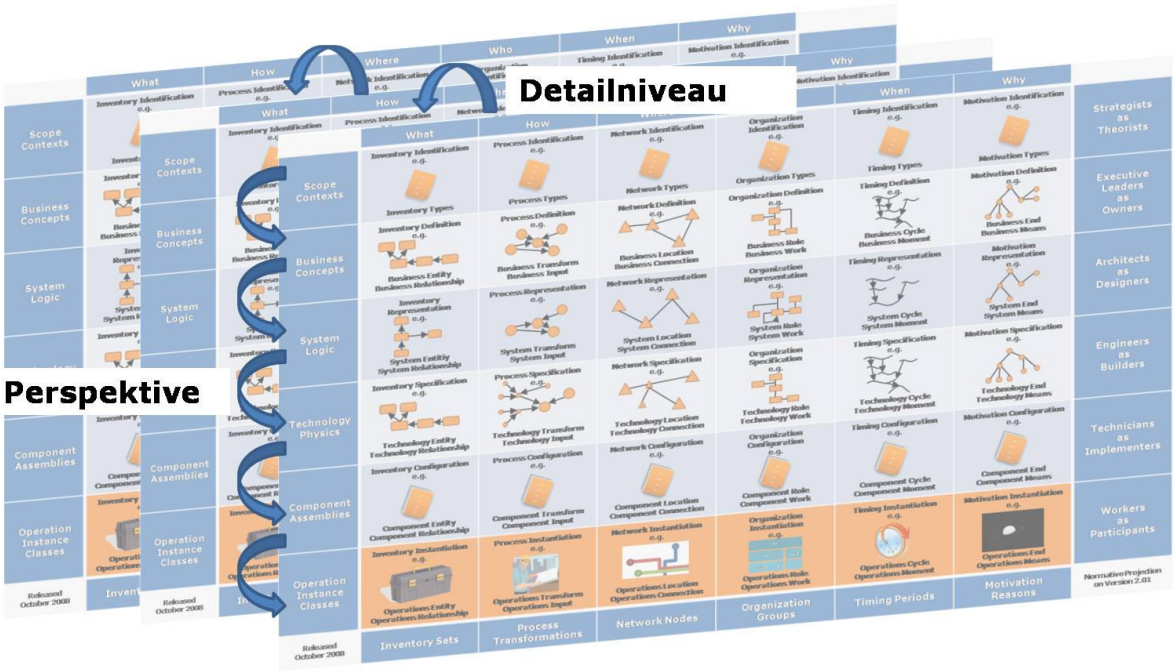


**Abbildung 2: Zachman™-Framework – Perspektiven und Detailtiefe**

Zwischen den Perspektiven bestehen Zusammenhänge. Die Perspektive „Business Concepts" (Zeile 2) definiert Anforderungen und Voraussetzung für die Perspektive "System Logic" (Zeile 3).

Sprechen wir von einem Unternehmensmodell, dann sprechen wir von einem kompositen Modell, das alle Einzelmodelle in allen Perspektiven, Sichten und Detailstufen beinhaltet.
Das „Business Model" beinhaltet alle Einzelmodelle der Perspektive „Business Concepts". Der im deutschen häufig benutzte Begriff Fachmodell beinhaltet Einzelmodelle aus den Perspektiven „Scope" und „Business Concepts". Ein „Systemmodell" (für ein IT-System) beinhaltet Einzelmodelle aus den Perspektiven „System Logic" und „Technology Physics".

Tabelle 1 ordnet häufig benutzte Bezeichnungen für Modelltypen und Artefakte den einzelnen Zellen des Zachman-Frameworks zu. Die Abbildung gibt dabei lediglich eine Idee, welche Modelltypen typischerweise für die jeweilige Sicht genutzt werden. Für eine tiefere Betrachtung ist eine exakte Definition des Inhalts des jeweiligen Modelltyps notwendig. Tabelle 1 zeigt ausschließlich einfache Modelle. Tabelle 1 beinhaltet Modelle und Artefakte aus verschiedenen Detailstufen.

| | What | How | Where | Who | When | Why |
|---|---|---|---|---|---|---|
| Scope Contexts | Liste der Fachkonzepte | Prozess-landkarte | | Stakeholder-modell | | Projektauftrag Vision Problem-Statement |
| Business Concepts | Faktenmodell Regelmodell | Prozessmodell Aktivitäten-modell | Liste der Geschäftsorte | Organigramm Org.struktur Rollenmodell (Business) | Zeitmodell | Geschäfts-anforderungen |
| System Logic | Informations-Modell Logisches Datenmodell | Funktions-modell Workflow-modell Programm-design | Verteilungs-modell (logisch) Komponenten-modell | Rollenmodell (System) | Zeitmodell | Design-anforderungen |
| Technology Physics | Physisches Datenmodell | Impl.-modell | Verteilungs-modell | | | System-anforderungen |
| Component Assemblies | Datenbank-struktur | Programmcode | | | | |
| Operations Instance Classes | | | | | | |

Tabelle 1: Einfache Modelle und Artefakte im Zachman-Framework

In letzter Zeit wird in Publikationen der Begriff „Capability" häufiger benutzt. Unterschieden werden „Business Capabilities" und „System Capabilities". Diese sind jeweils Zusammenfassungen von Einzelmodellen meist der Spalten „What" und „How" der jeweiligen Perspektive.

## 1.5 Modell als Muster (Pattern)

In der Wiktionary-Definition für den Begriff „Model" wird unter Nummer 6 eine weitere interessante Definition gegeben: „A praiseworthy example to be copied, with or without modifications." Für diese Art von Modell wird im Umfeld der Unternehmensmodellierung allgemein der Begriff „Pattern" (Muster) verwendet.

Die Wiktionary-Definition gibt das gewollte Verständnis sehr gut wieder. Es handelt sich dabei um eine Vorlage, die wiederverwendet wird. Dabei kann das Muster angepasst, geändert oder erweitert werden. In Systementwurf und –implementation kennen wir Architekturmuster für die Architektur eines IT-Systems, z.B. das Model-View-Controler-Muster. Solche wiederkehrenden Muster gibt es aber in allen Perspektiven und Sichten des Zachman-Frameworks und auch für komposite Modelle. Für die Darstellung von Geschäftsprozessen wäre ein Muster „Wiedervorlage" denkbar, das einen bestimmten generischen Ausschnitt eines Geschäftsprozesses beschreibt.

# 2 Modelle und Kommunikation

Ein gutes Modell dient einem bestimmten Zweck – und am besten nur einem!

Modelle in den Perspektiven „System Logic" und „Technology Physics" können dazu genutzt werden, Quellcode für Programme oder für Workflow-Engines (automatisch) zu erzeugen. Z.B. lässt sich aus UML-Klassen-Diagrammen Java-Quelltext erzeugen, aus ER-Diagrammen Datenbankstrukturen oder aus BPMN-Diagrammen BPEL-Code. Damit dies gelingt, muss das Modell verschiedenen formalen Anforderungen entsprechen. Die Diskussion um Modelle und den Gebrauch von Standardnotationen scheint häufig von solchen Erwägungen getrieben zu sein. Ein Blick in die Praxis zeigt jedoch oft ein anderes Bild. Scott Ambler stellte in einer Umfrage fest, dass die Mehrheit der Projektteams Modelle *nicht* für die automatische Systemgenerierung nutzt, sondern für die Dokumentation von System, Design, Anforderungen, usw[5]. Das gilt häufig auch für Modelle in anderen Perspektiven. Interessant ist dabei auch die Unterscheidung der Teams nach „agilen" oder „traditionellen" Vorgehensweisen in der Auswertung. Scott Amblers Befragung richtete sich an Software-Entwickler.

Ob die Stichprobe repräsentativ ist, sei dahin gestellt. Meine Erfahrung zeigt jedoch ein ähnliches Verteilungsmuster bei meinen Kunden. Wenn wir die Frage im Bereich Geschäftsprozessmodellierung stellen, dürften die Prozentzahlen für automatisch generierte Systemkomponenten noch geringer sein. Damit soll aber in keinem Fall gesagt werden, dass die (automatische) Generierung von Systemkomponenten (der Übergang von Zeile 3 zu Zeile 4) unwichtig ist. Sie wird in Zukunft sicher an Bedeutung gewinnen.

Was ist also der Hauptzweck der meisten Modelle, die in Projekten erstellt werden? Die Antworten von Anwendern darauf sind vielfältig. Das reicht von „Systemdokumentation", „Systemdesign", „Prozessdokumentation", „Prozessanalyse", „Prozessoptimierung", „Beschreibung von Anforderungen" bis hin zu „Erstellen von Arbeitsanweisungen". Hinterfragen wir diese Aspekte stellen wir fest, dass sich hinter all dem ein wichtiger Aspekt verbirgt:

***Modelle dienen der Kommunikation!***

Modelle in der Perspektive „Business Concepts" dienen zuerst der Kommunikation zwischen Fachanwendern („Business People"). Sie dienen auch der Kommunikation zwischen Fachanwender und IT-Spezialist. Für die Perspektive „Business Concepts" wird sich das auch nicht ändern – hier ist Kommunikation der Hauptzweck, um Analyse, Optimierung, Anforderungsdefinition zu ermöglichen. Das erklärt auch die extensive Nutzung von Modellen in „agilen Teams". Solche Teams haben offenbar einen hohen Bedarf an Kommunikation.

Wenn wir akzeptieren, dass Modelle zumindest in bestimmten Perspektiven zuerst der Kommunikation dienen, relativieren sich verschiedene Diskussionen sehr schnell. Damit Kommunikation gelingt, muss der „Empfänger" oder „Leser" das Modell verstehen. Damit tritt das Kriterium „formale Korrektheit" hinter das Kriterium „Verständlichkeit" zurück. Aber natürlich ist auch hier formale Korrektheit wichtig. Die Kommunikation soll nicht nur in verständlicher Form erfolgen, sondern auch Missverständnisse vermeiden und eine Umsetzung in ein streng formales Modell erlauben. Wenn wir eine Abwägung zwischen „Verständlichkeit" und „formaler Korrektheit" treffen müssen, ist Verständlichkeit wichtiger. Gegebenenfalls kommentieren wir das Modellelement, um eine spätere Formalisierung zu unterstützen.

Auch die Frage, welche Modellelemente aus den Standardnotationen wie UML und BPMN ein Modellierer einsetzt, ändert plötzlich die Richtung: Es ist nicht wichtig, jedes Element

[5] Results from Scott Ambler's July 2008 Modeling and Documentation Survey posted at www.agilemodeling.com/surveys/

einzusetzen und auch die letzte Möglichkeit der Notation zu nutzen. Das macht nur Sinn, wenn der Empfänger des Modells in der Lage ist, diese Elemente ebenfalls zu verstehen und im Sinne der Kommunikation zu nutzen. Wichtig sind hier ein „Styleguide", Namenskonventionen, Legenden zu den Modellen und auch die Beschreibung von Modellelementen, die den Leser des Modells beim Verständnis unterstützen.

Auch in Bezug auf benutzte Werkzeuge stellen sich andere Fragen: Welche Funktionen existieren zur Unterstützung von „Kommunikation"? Ist es einfach, verschiedene Ausgaben zu erzeugen? Das Spektrum reicht von publizierten Modellen im Intra-/Internet, Erstellung von Arbeitsanweisungen, Erstellen von Auswertungen, Standardreports bis hin zu formellen und rechtlich bindenden Dokumenten wie einer Ausschreibung. Ist es einfach Informationen im System zu erfassen, z.B. während eines Workshops? Werden verschiedene Formen und Medien der Ein-/Ausgabe unterstützt?

# 3 Ausblick

Die in Kapitel 1 und 2 entwickelten Ideen sind weiterzuführen:

- Welche einfachen Modelle sind zu betrachten?
- Welchen Inhalt haben diese Modelle? Wie sind sie in das Zachman-Framework einzuordnen?
- Welche Notation ist zur Darstellung des jeweiligen Inhalts geeignet?
- Wie sind einfache Modelle verknüpft?
- Welche kompositen Modelle sind zu betrachten?
- Welche Qualitätskriterien gelten für ein Modell, das der Kommunikation dient?

Auch die Frage der unterstützenden Werkzeuge verdient eine tiefere Betrachtung.
Weitere White Paper werden diesen Fragen gewidmet sein.