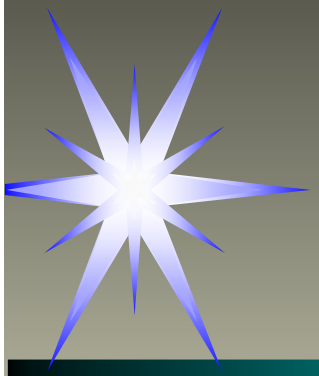


V-Modell mit UML

Max Kleiner

Open or programming for change

- The Unified Modeling Language [UML95] is a third-generation object-oriented modeling language for specifying, visualizing, and documenting the artifacts of an object-oriented system under development. It fuses the concepts of different notations.

- Es reicht aus, den Produkttypen Diagramme zuzuordnen. Die Produktflüsse des V-Modells halten dann fest, in welchen Aktivitäten und Phasen ein Diagrammtyp zu gebrauchen sei.

Agenda Architect's Day

- Die Zuordnung der UML Diagramme zu den Phasen
- Die 8 Basis Diagramme der UML 2 und deren Durchgängigkeit in der Fallstudie BROKER
- Praktisches Template im “Strukturbeispiel“

Einspruch

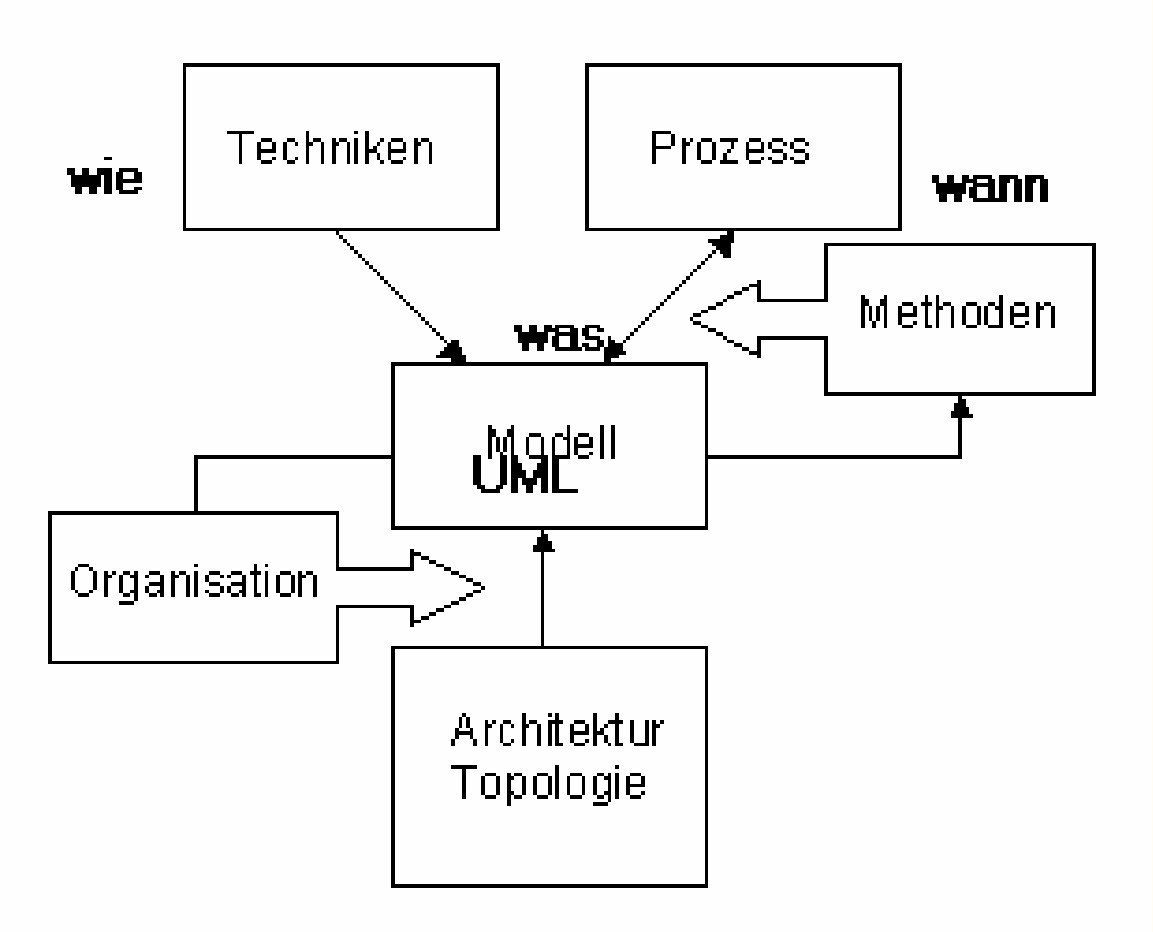
- “The process of defining the requirements is iterative, which means that e.g. if a Use Case is drawn and the according Activity is not what the customer expects, then we go back to the Use Case and correct it”.
  - GUI Design and DB Modellierung ist nicht Teil der UML !
  - Die gemachten Aussagen bezüglich Phasen und Submodell SE sind von der Abstraktionsebene für einige Vorgehensmodelle gültig.
  - Die Zuordnung der UML zu den Phasen suggeriert eine lineare Abfolge, soll aber nur den Abbildungscharakter verdeutlichen und lässt sich nach wie vor iterieren.

Handlungsbedarf

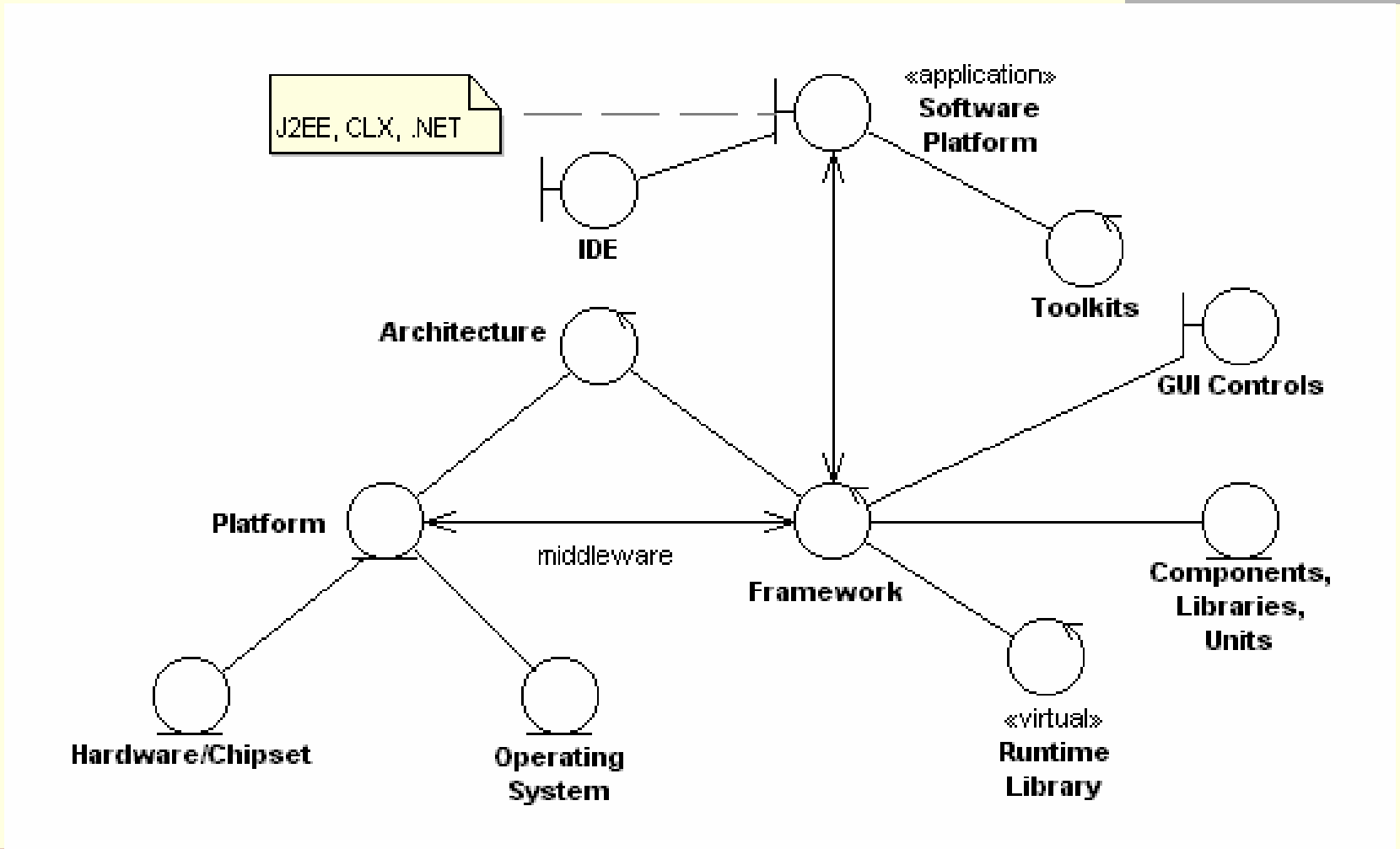
- Die bekannten Vorgehensmodelle geben in der Regel vor, was getan und erstellt werden muss (Aktivitäten – Produkte)
- Sie machen wenig Aussagen dazu, wie dies geschehen soll.
- Konkret bspw.: Die Beschreibung der Geschäfts-Prozesse in der Phase Analyse erfolgt in Form des Activity Diagrams.
- Remember: Model->Code->System (MCS)

Prozess und Architektur?

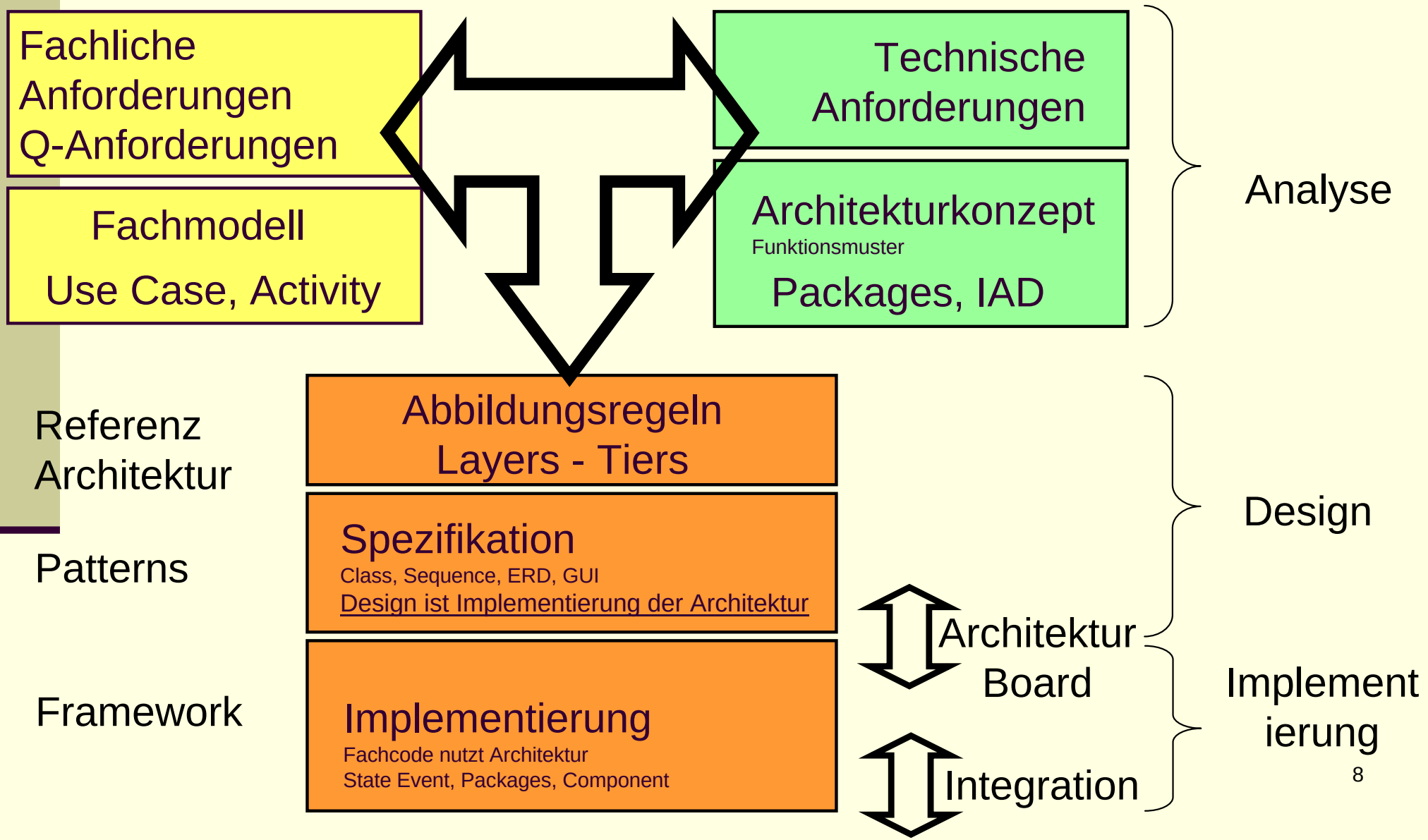
- Die Integration der Diagrammtypen beschreibt WIE man zum WAS innerhalb der Phasen kommt!



Ein IT Architekturschema



Rahmenbedingung

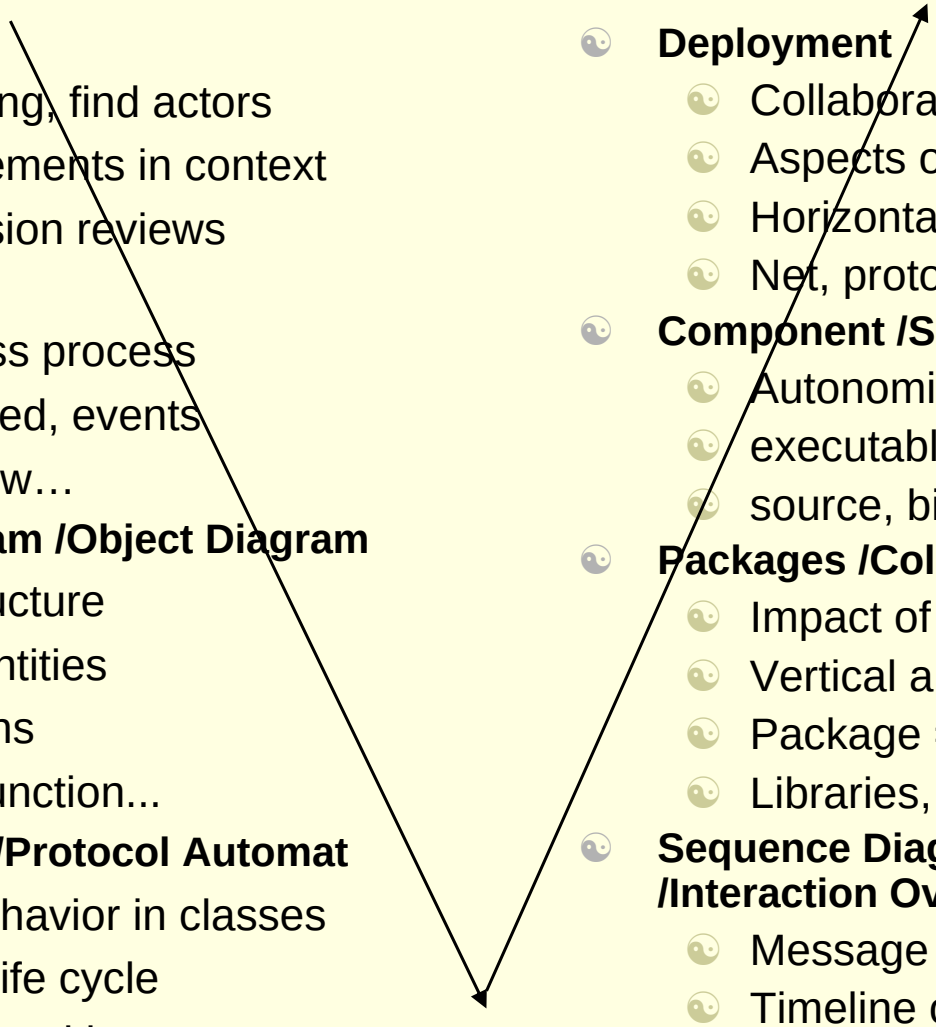


Produkttypen mit V

Phase	UML	Instance
Analysis	Use Case	Requirements
Analysis	Activity	Business Units
Analysis/Design	Class Diagram	Objects and Types
Design	State Event	State Values
Implement	Sequence	Scenarios
Implement	Packages	Versions
Integration	Component	Files
Integration	Deployment	Devices

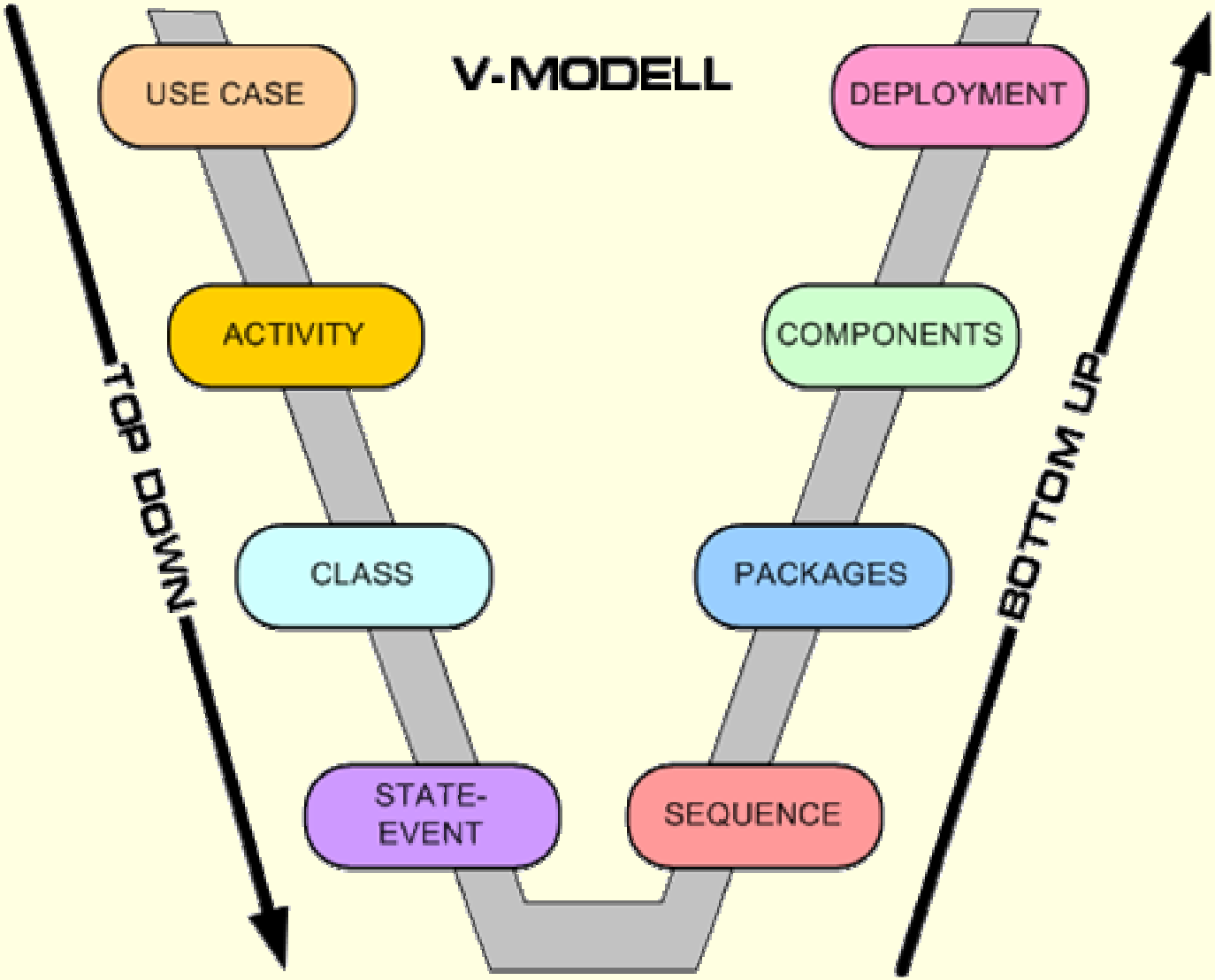
UML - V Zuordnung

- **Use Case**
  - Initializing, find actors
  - Requirements in context
  - Discussion reviews
- **Activity**
  - Business process
  - Paralleled, events
  - Workflow…
- **Class Diagram /Object Diagram**
  - find structure
  - OOP-Entities
  - Relations
  - Utility function...
- **State Event /Protocol Automat**
  - dyn. Behavior in classes
  - Object life cycle
  - State transitions

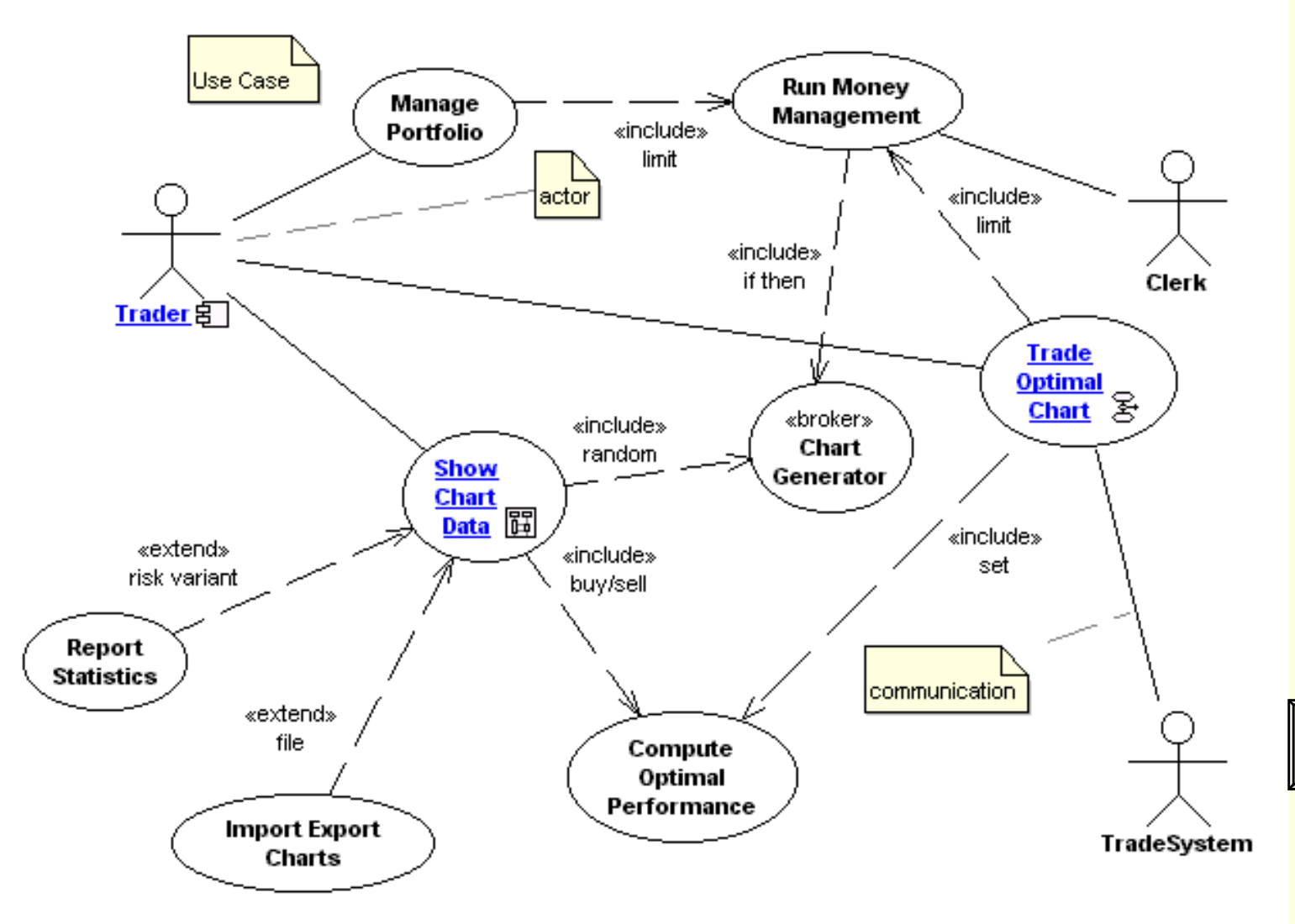


- **Deployment**
  - Collaboration of components
  - Aspects of deployment
  - Horizontal System architecture
  - Net, protocols and topologies
- **Component /Subsystem /Composition**
  - Autonomic interfaces
  - executable groups
  - source, binary, executable
- **Packages /Collaboration Patterns**
  - Impact of changes
  - Vertical architecture
  - Package =unit as a module
  - Libraries, patterns
- **Sequence Diagram /Communication /Interaction Overview /Time Diagram**
  - Message flow of objects
  - Timeline of call stack
  - Dynamic call cascades

Das Prinzipschema



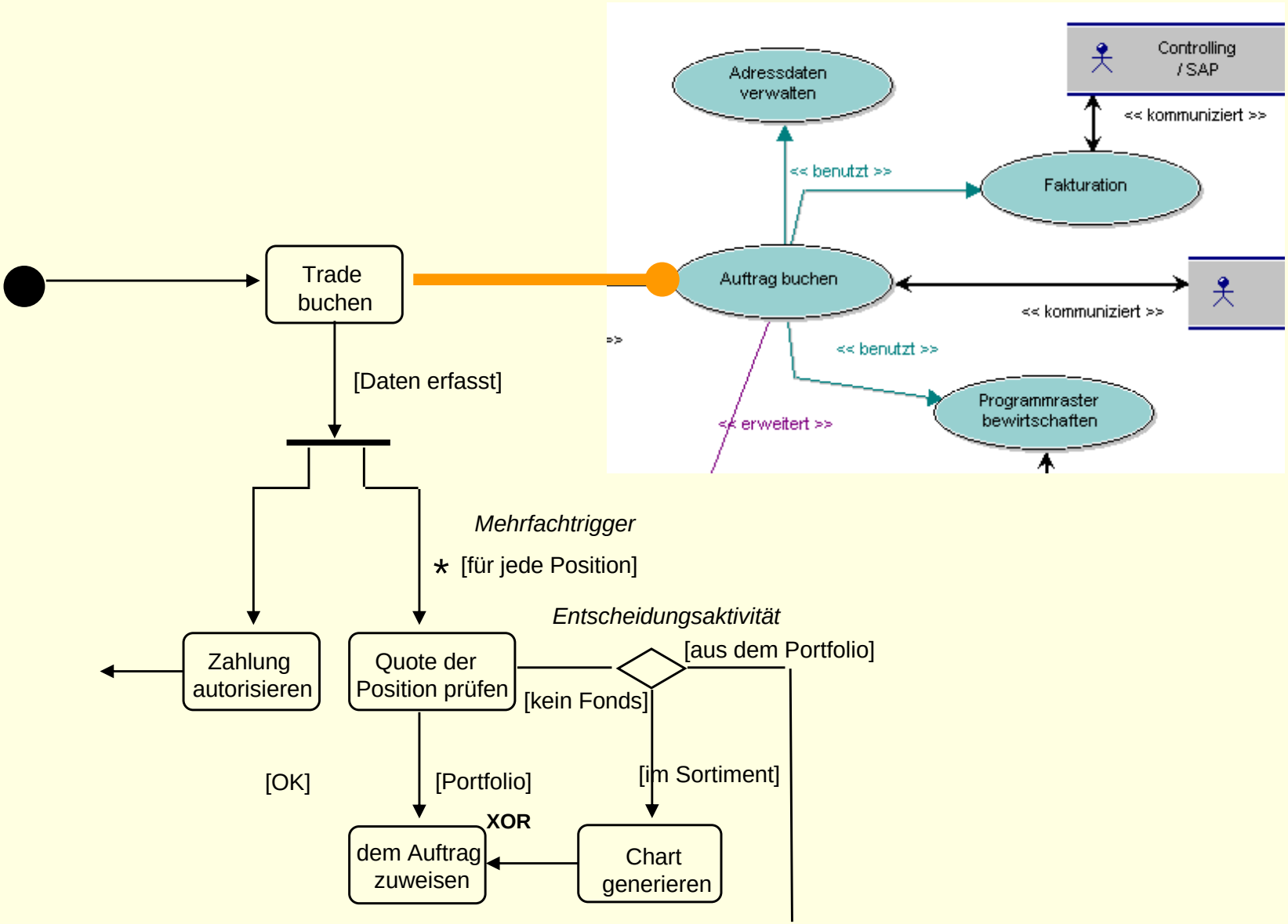
Fallstudie BROKER: Use Case



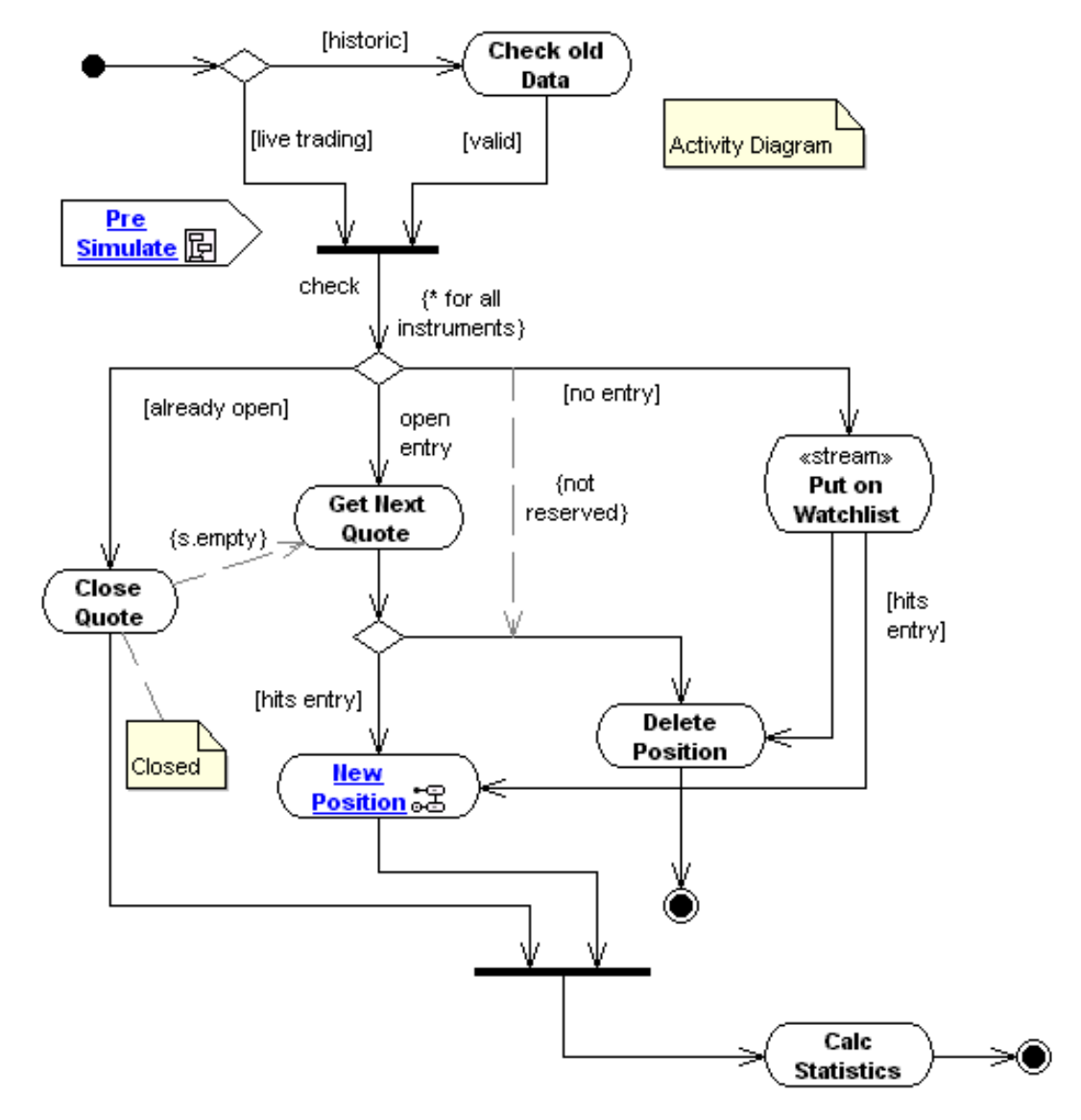
Analyse

Pflichtenheft

Relation Use Case - Activity

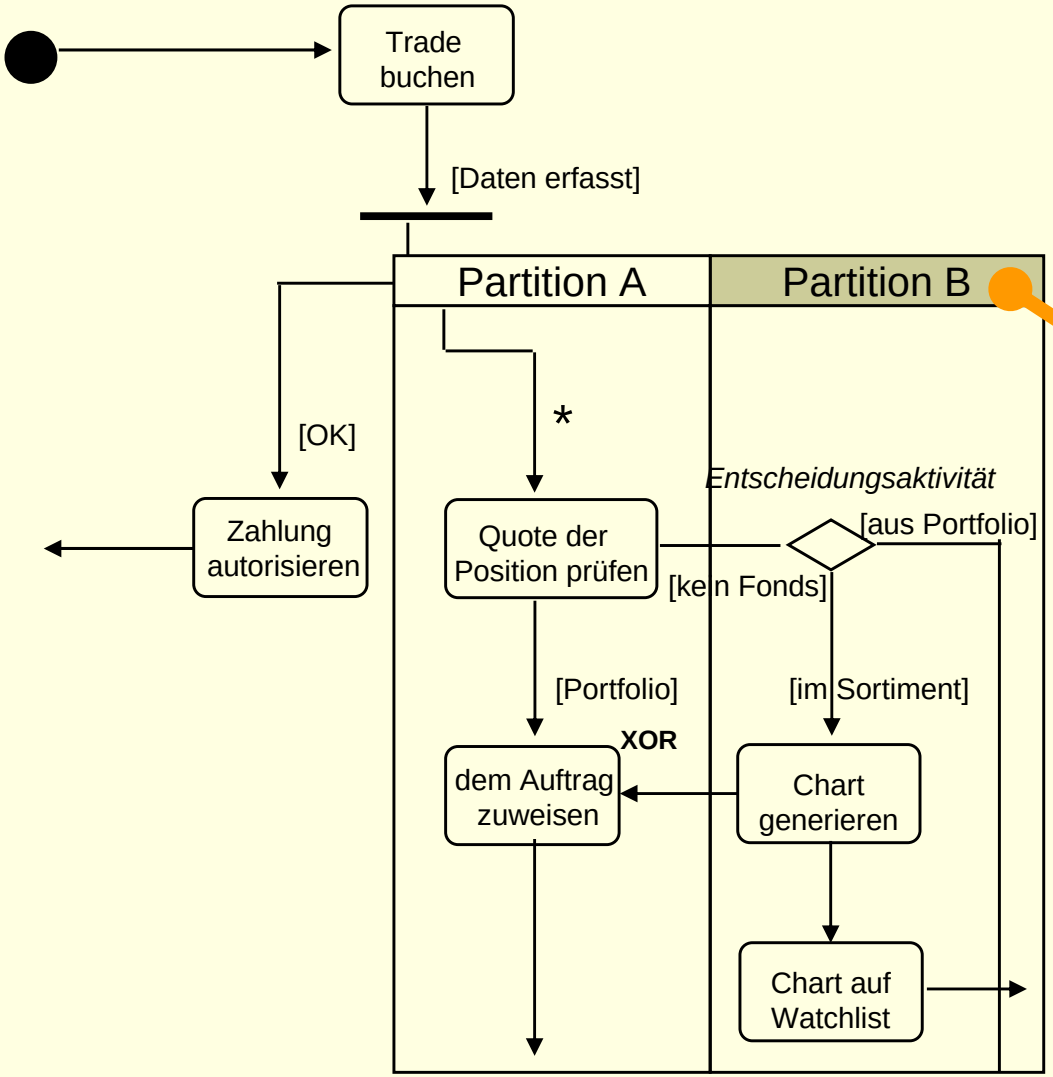


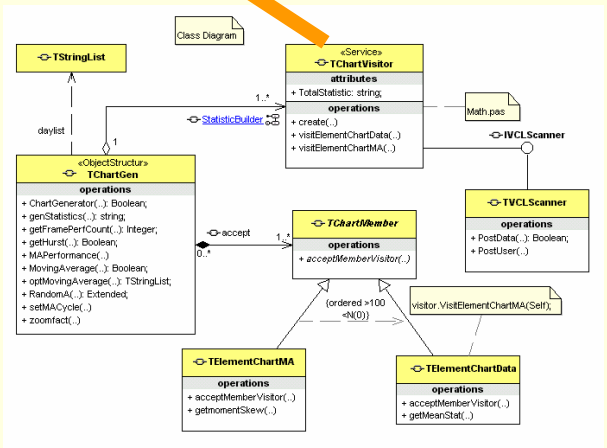
Activity



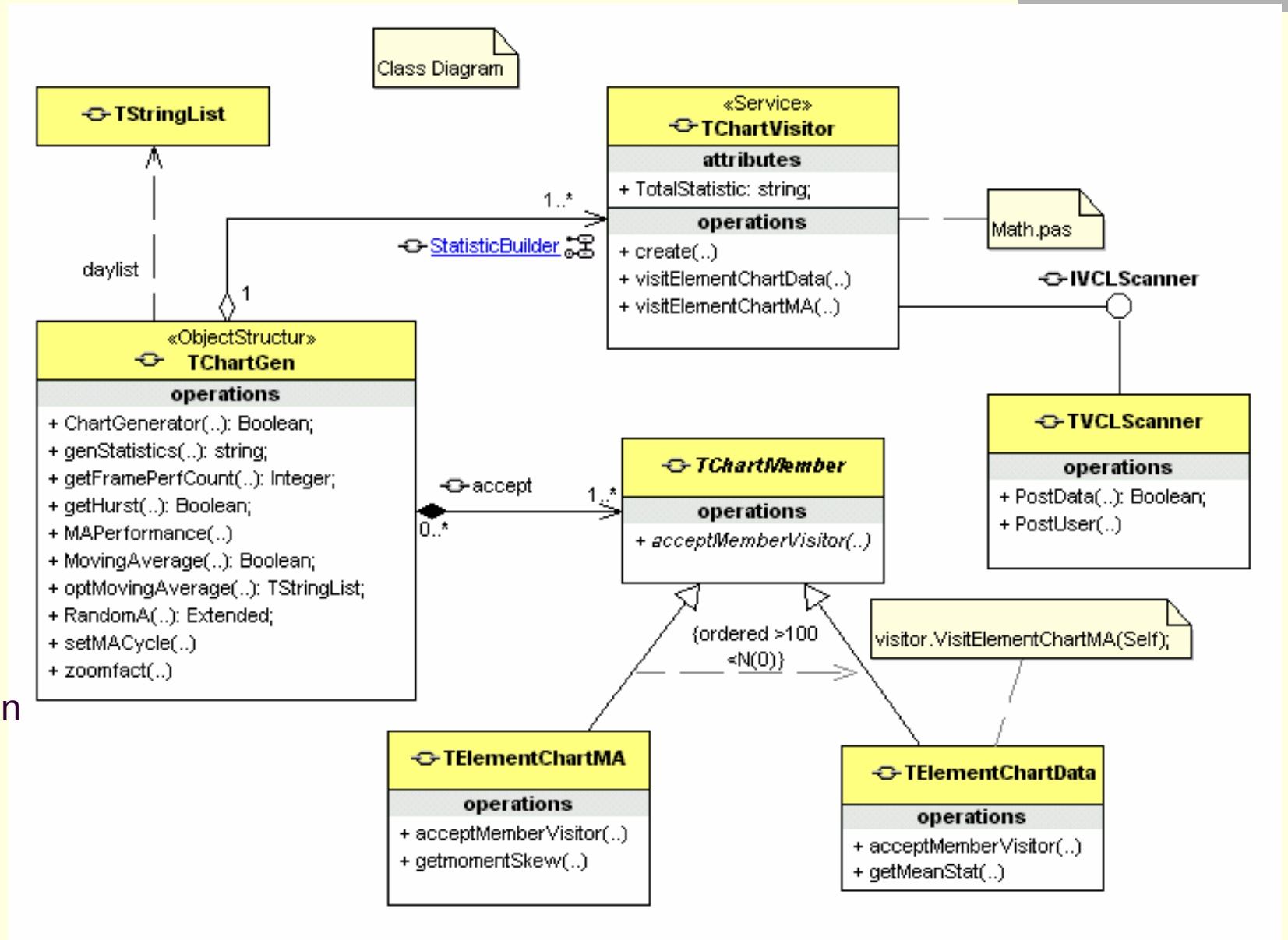
Analyse

Relation Activity - Class





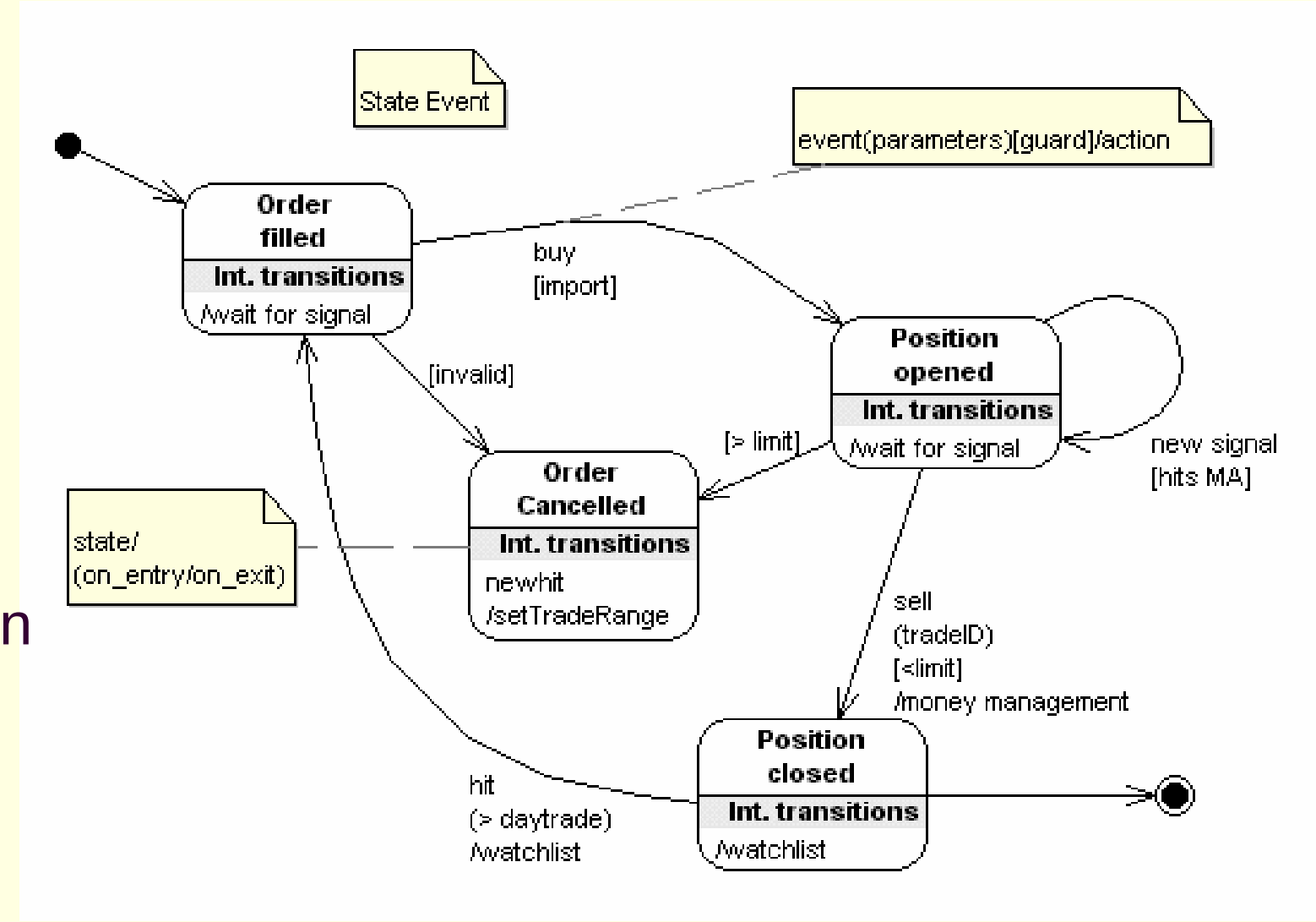
Class Diagram



Design

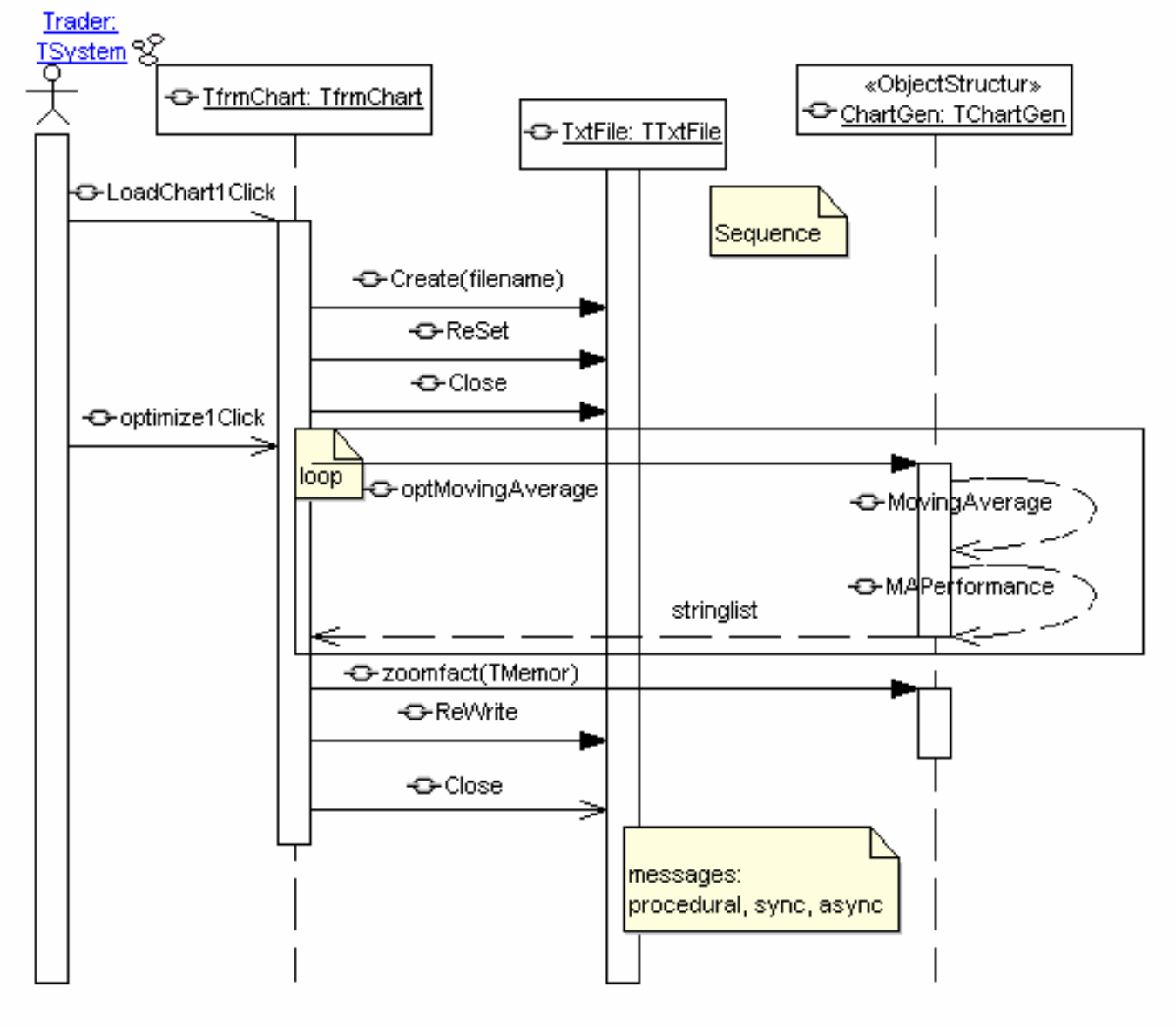
Spezifikation

State Event of a Class



Design

Sequence Diagram

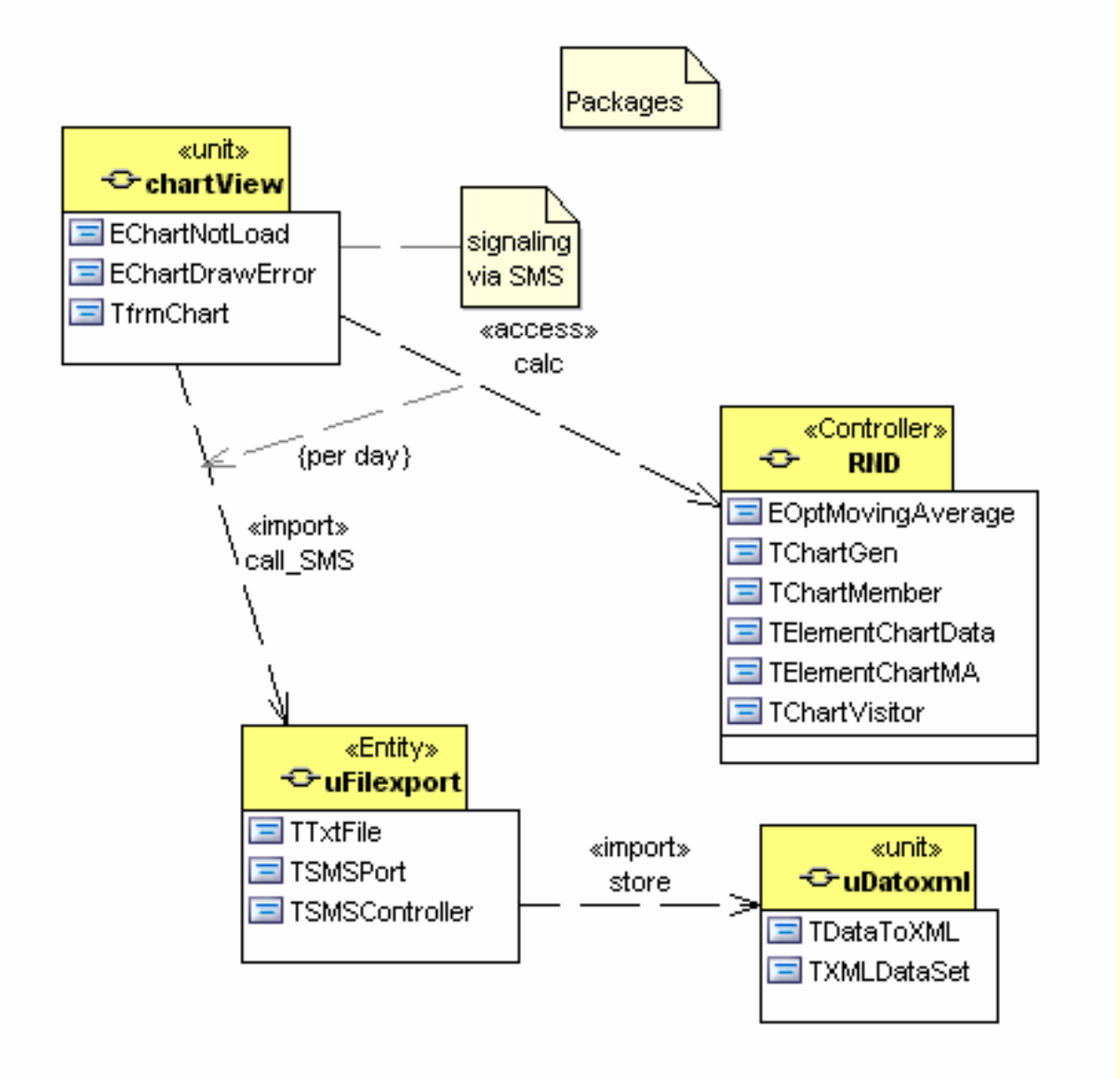


Implement

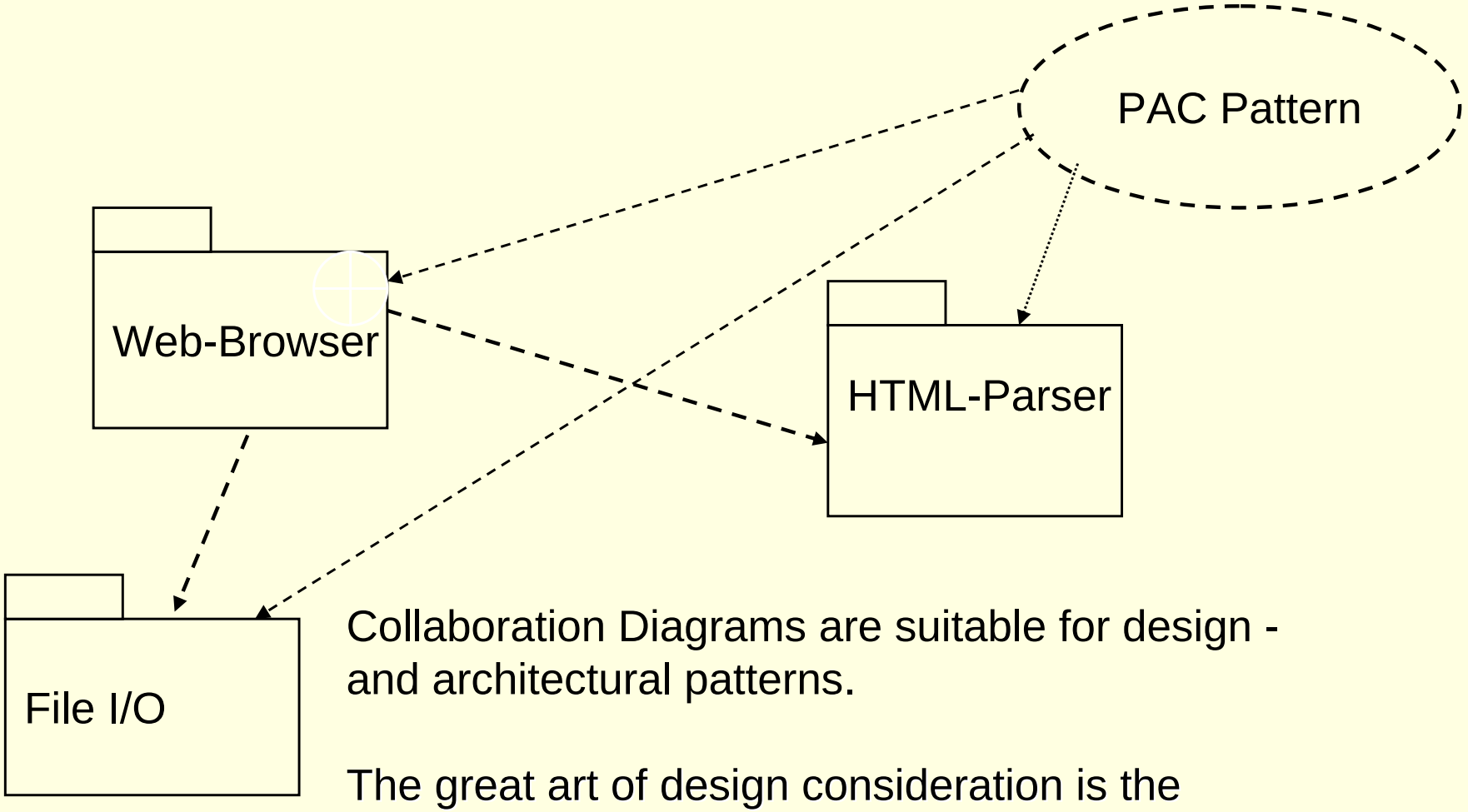
Systemhandbuch

Packages

Implement



Packages with Patterns

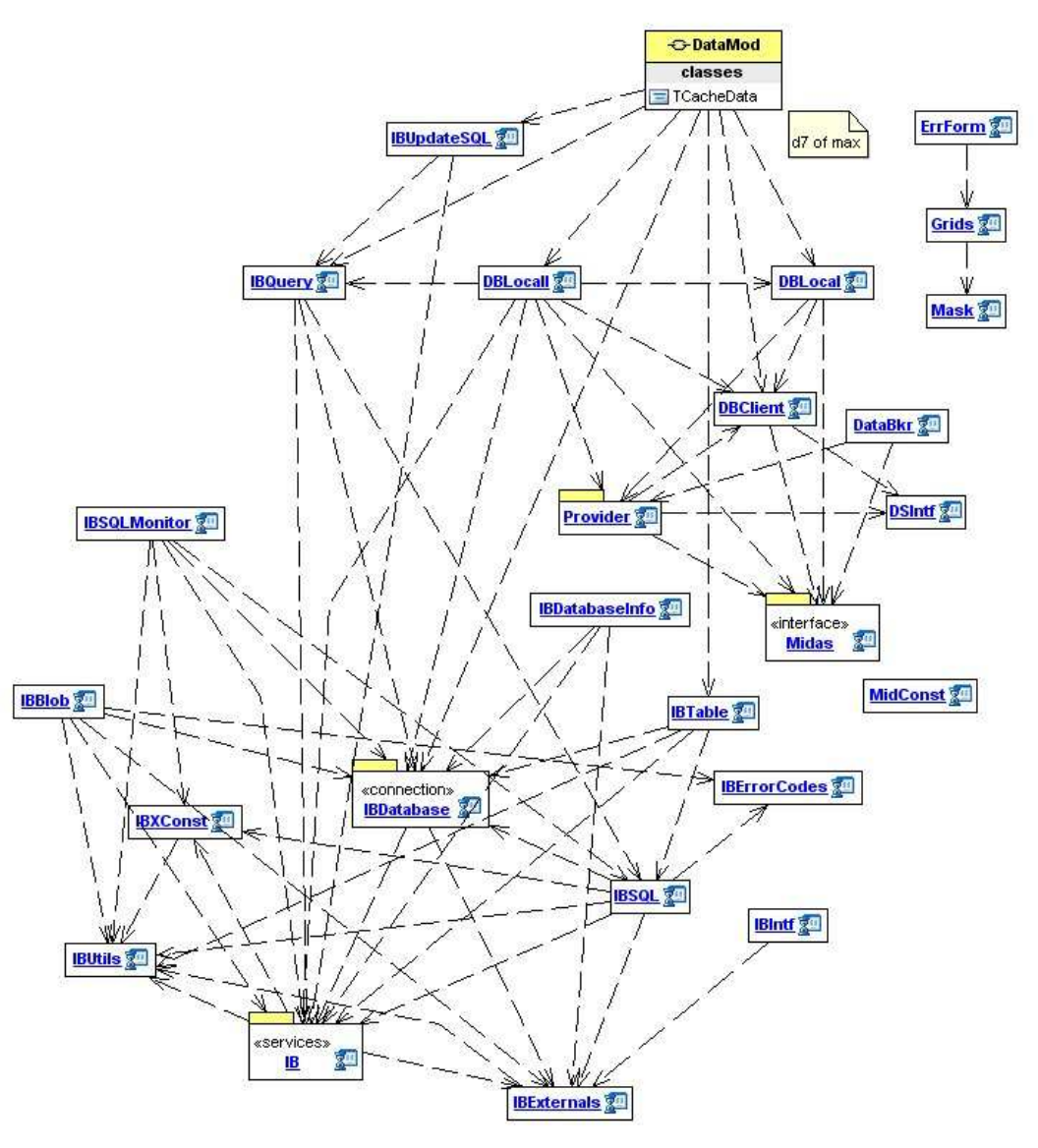


Collaboration Diagrams are suitable for design - and architectural patterns.

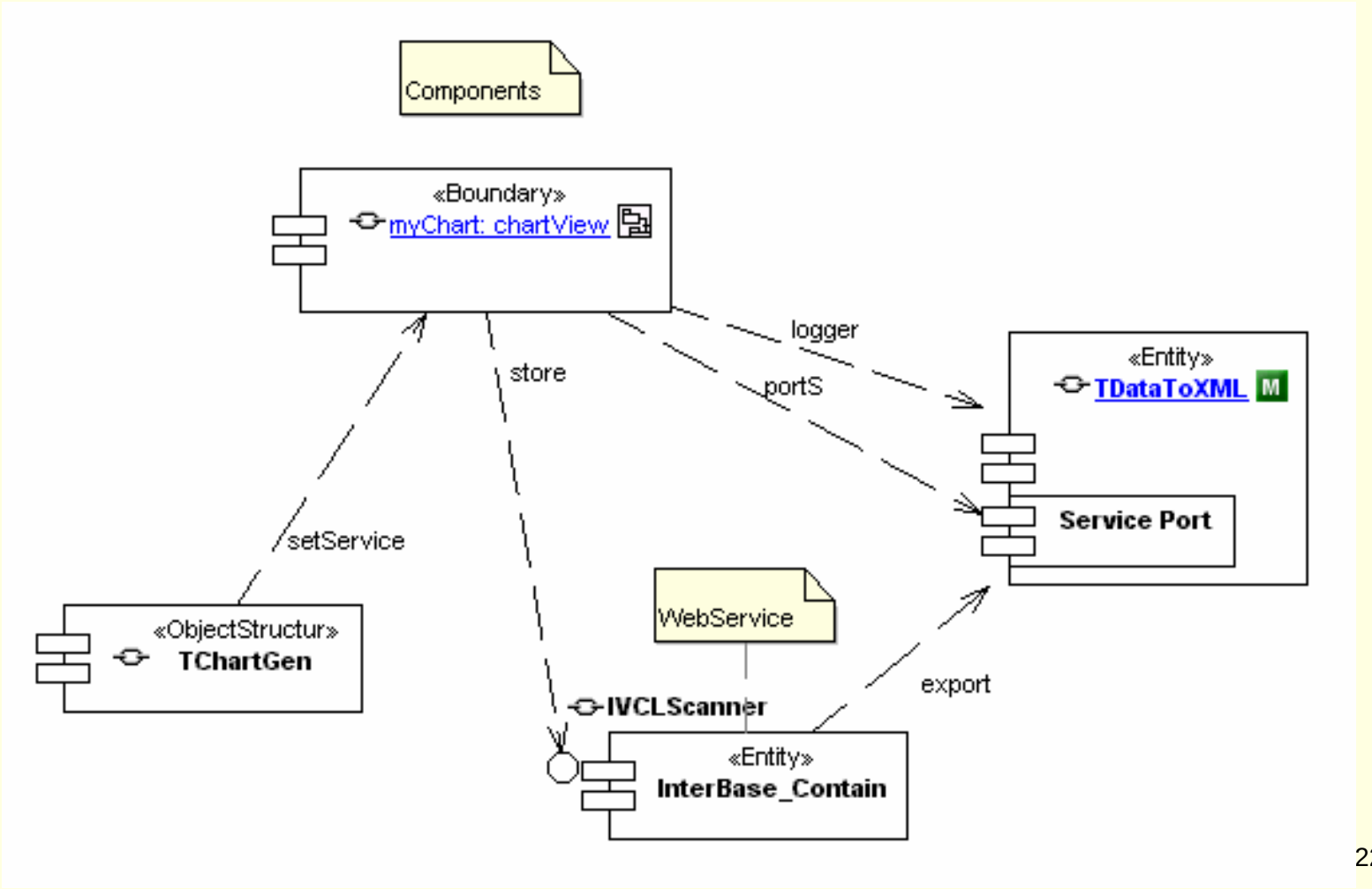
The great art of design consideration is the minimization of dependencies between packages which minimizes the impact of changes.

Subsystem-Diagram

Architectural combinations and build patterns of packages



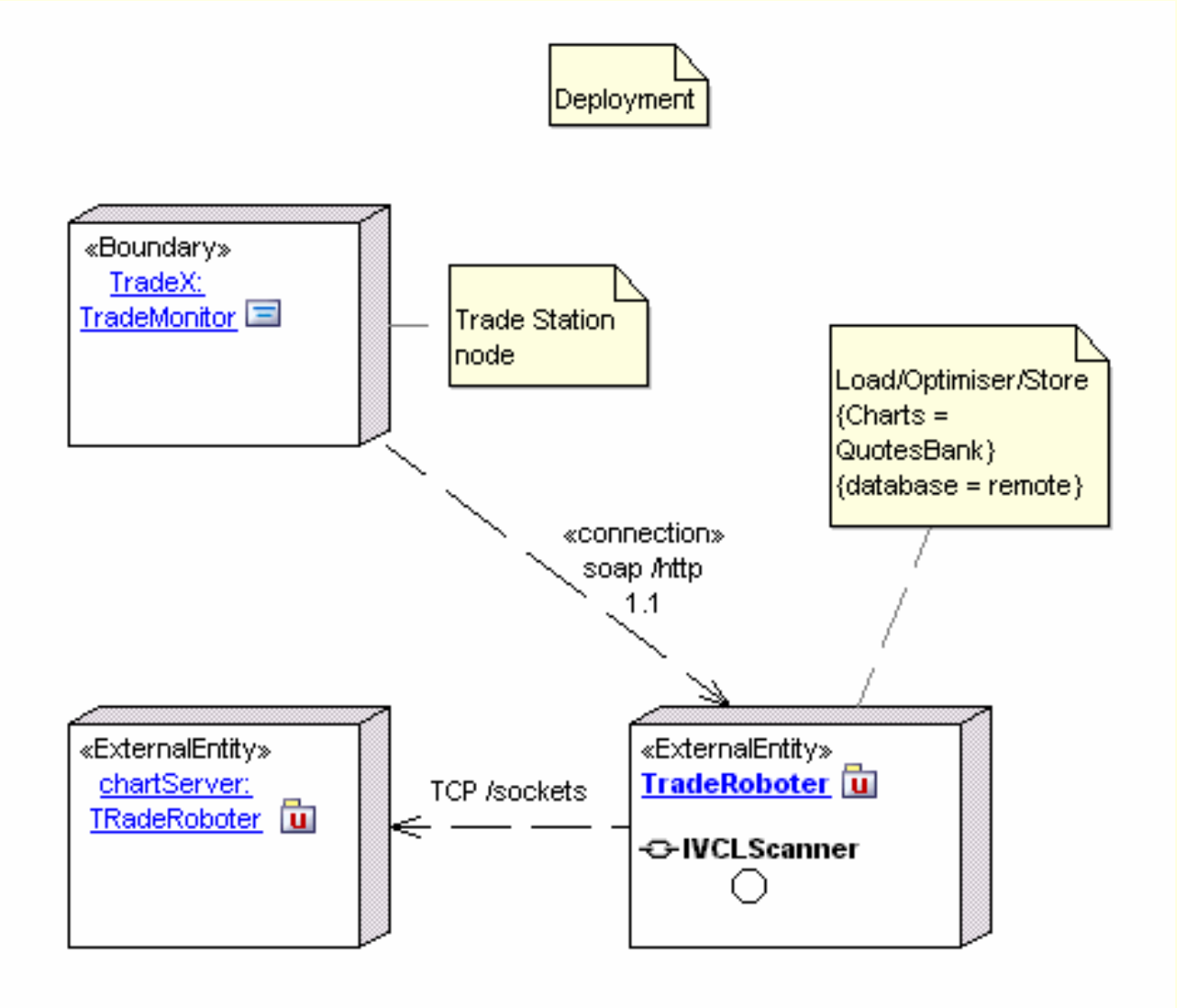
Component



Integration

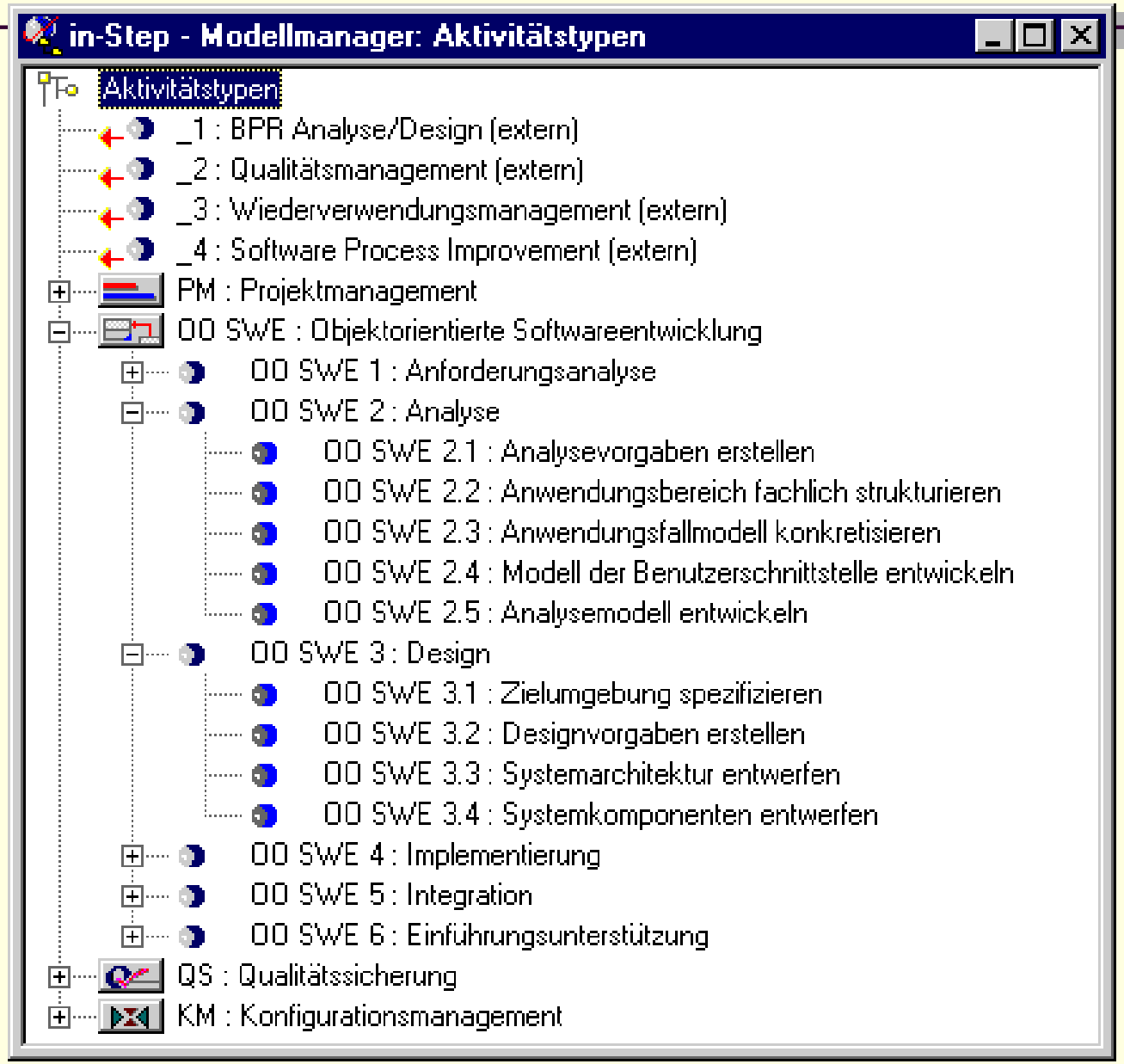
Betriebshandbuch

Deployment

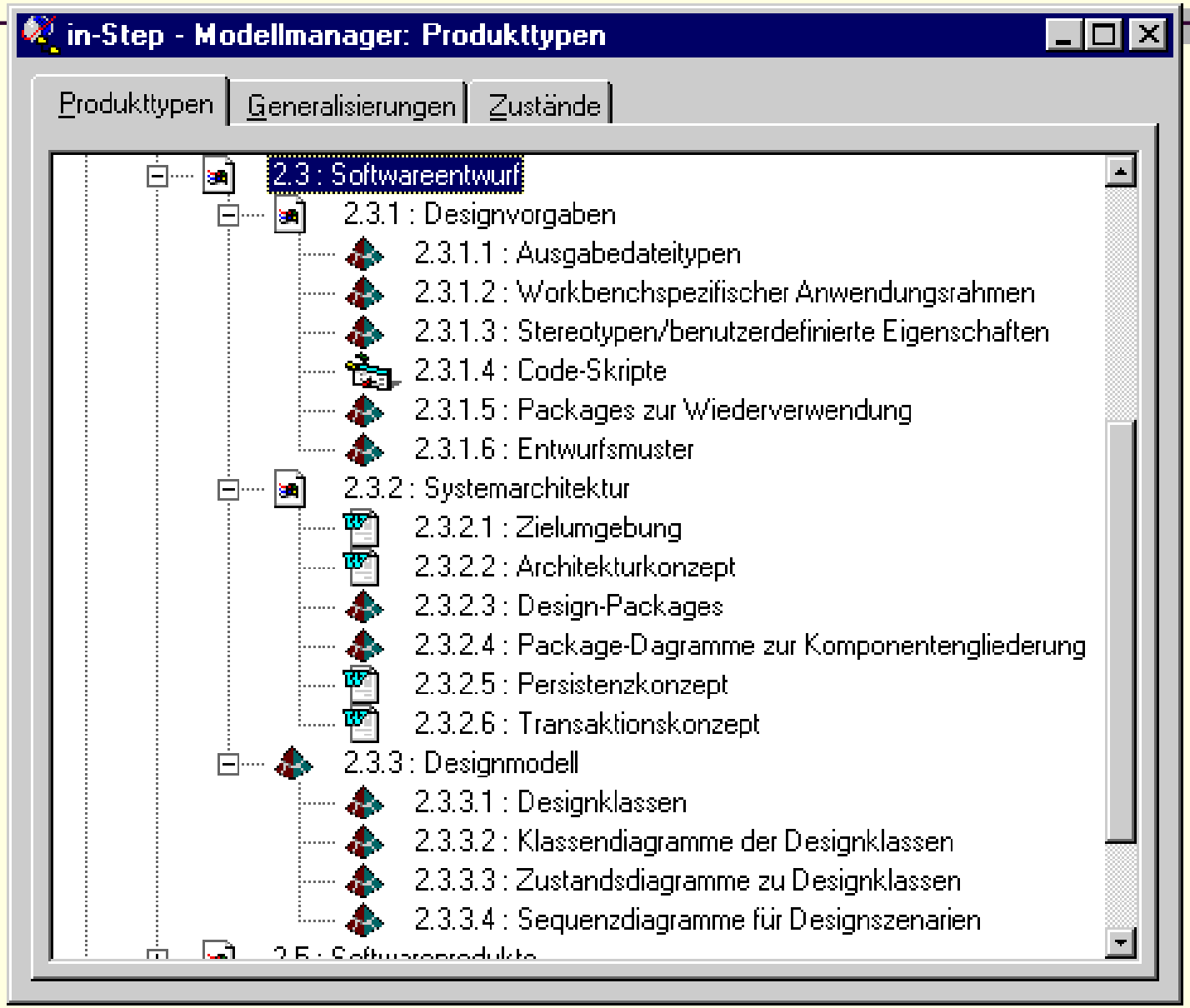


Integration

...Tools bearbeiten das Submodell SE mit einer Zuordnungstabelle...



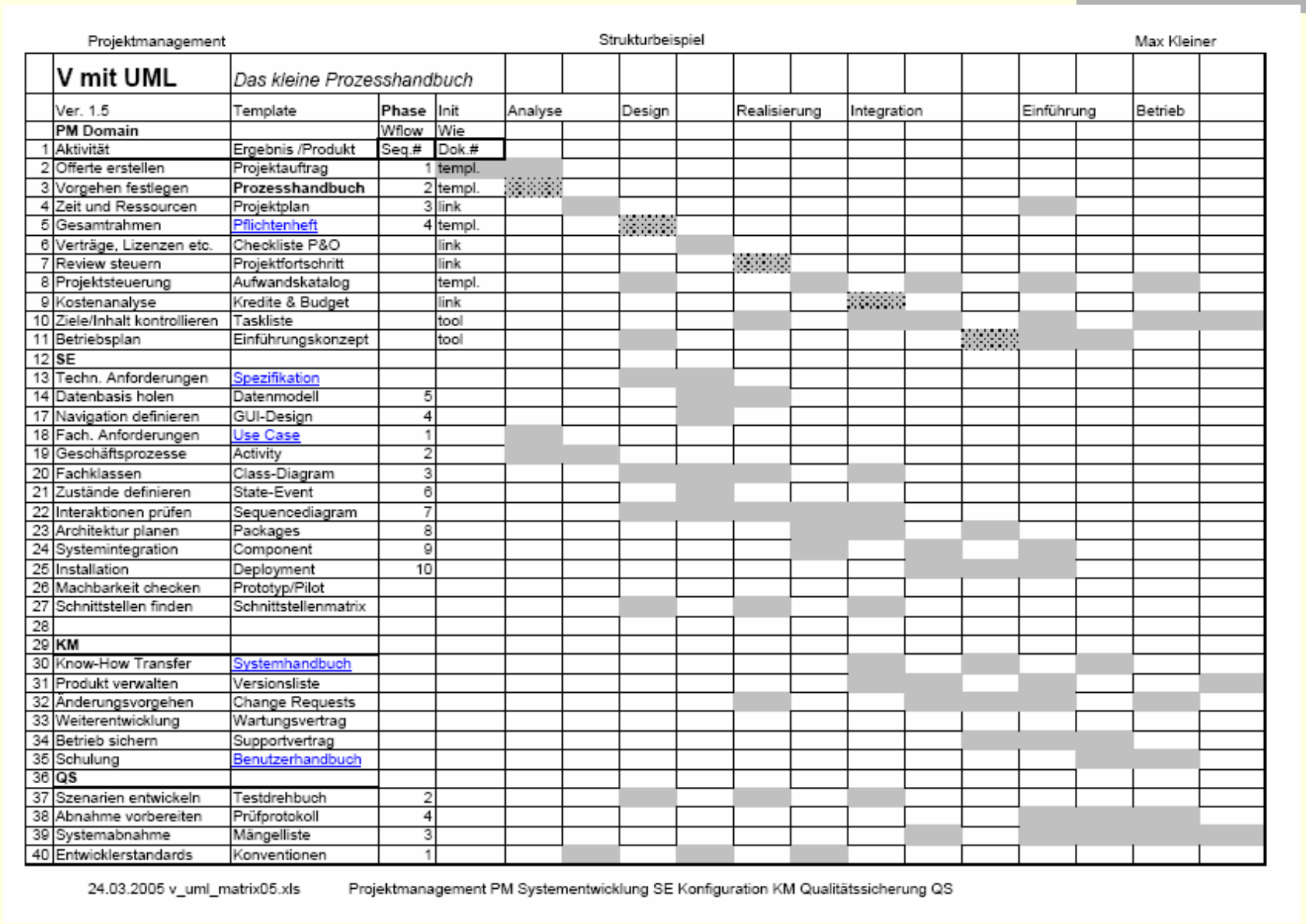
...oder Produkte - Orientiert



Konklusion

- Bei einer Zuordnung Diagrammtyp – Produkt als „Elementarmethode“ steht das „**wie**“ im Vordergrund.
- Das heißt, eine Elementarmethode ergänzt die Aussage darüber, **was** im Rahmen einer Aktivität erfolgen soll, durch die konkrete UML Notation als Ergebnis, **wie** es zu tun ist.

Strukturbeispiel

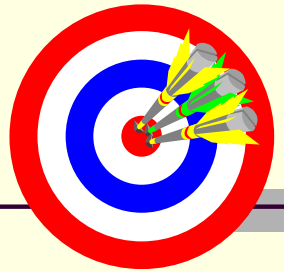


Projektmanagement — Strukturbeispiel — Max Kleiner

	V mit UML	*Das kleine Prozesshandbuch*			Analyse		Design		Realisierung		Integration		Einführung		Betrieb	
	Ver. 1.5	Template	**Phase**	Init												
	PM Domain		Wflow	Wie												
1	Aktivität	Ergebnis /Produkt	Seq.#	Dok.#												
2	Offerte erstellen	Projektauftrag	1	templ.												
3	Vorgehen festlegen	**Prozesshandbuch**	2	templ.												
4	Zeit und Ressourcen	Projektplan	3	link												
5	Gesamtrahmen	Pflichtenheft	4	templ.												
6	Verträge, Lizenzen etc.	Checkliste P&O		link												
7	Review steuern	Projektfortschritt		link												
8	Projektsteuerung	Aufwandskatalog		templ.												
9	Kostenanalyse	Kredite & Budget		link												
10	Ziele/Inhalt kontrollieren	Taskliste		tool												
11	Betriebsplan	Einführungskonzept		tool												
12	**SE**															
13	Techn. Anforderungen	Spezifikation														
14	Datenbasis holen	Datenmodell	5													
17	Navigation definieren	GUI-Design	4													
18	Fach. Anforderungen	Use Case	1													
19	Geschäftsprozesse	Activity	2													
20	Fachklassen	Class-Diagram	3													
21	Zustände definieren	State-Event	6													
22	Interaktionen prüfen	Sequencediagram	7													
23	Architektur planen	Packages	8													
24	Systemintegration	Component	9													
25	Installation	Deployment	10													
26	Machbarkeit checken	Prototyp/Pilot														
27	Schnittstellen finden	Schnittstellenmatrix														
28																
29	**KM**															
30	Know-How Transfer	Systemhandbuch														
31	Produkt verwalten	Versionsliste														
32	Änderungsvorgehen	Change Requests														
33	Weiterentwicklung	Wartungsvertrag														
34	Betrieb sichern	Supportvertrag														
35	Schulung	Benutzerhandbuch														
36	**QS**															
37	Szenarien entwickeln	Testdrehbuch	2													
38	Abnahme vorbereiten	Prüfprotokoll	4													
39	Systemabnahme	Mängelliste	3													
40	Entwicklerstandards	Konventionen	1													

24.03.2005 v_uml_matrix05.xls Projektmanagement PM Systementwicklung SE Konfiguration KM Qualitätssicherung QS

All the best!



- I‘m still confused but on a much higher level
- If you can‘t see it, you can‘t manage IT !
- max.kleiner.com