SOPHIST

Notationselemente der UML 2

Details Diagramme

Die Diagramme der UML 2

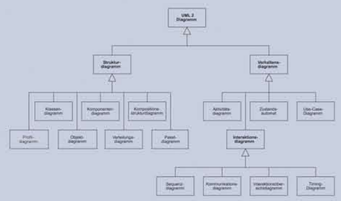
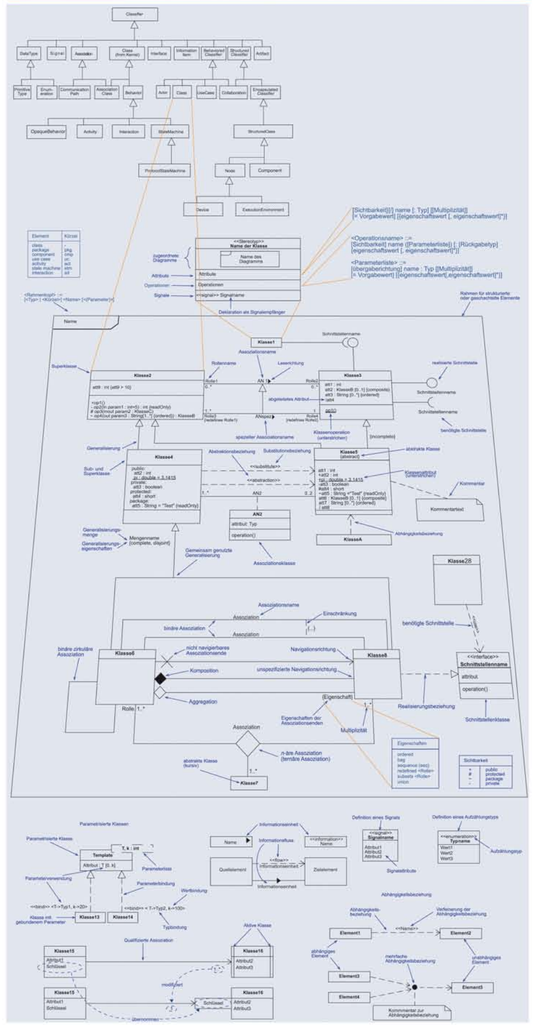
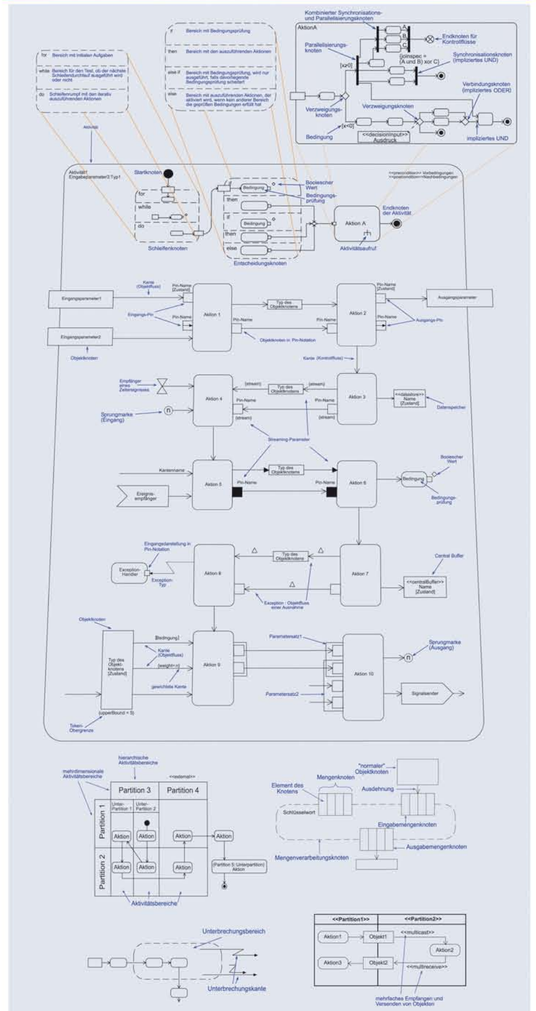


Diagramme	Beschreibung / Frage beantwortet das Diagramm	Eckpunkte
Klassendiagramm	Aus welchen Klassen besteht mein System und wie stehen diese untereinander in Beziehung	Beschreibt die statische Struktur des zu entwerfenden oder abzubildenden Systems. Enthält alle relavanten Strukturzusammenhänge und Datentypen. Bildet die Brücke zu den dynamischen Diagrammen. Normalerweise unvercichtbar.
Paketdiagramm	Wie kann ich mein Modell so schneiden, dass ich den Überblick bewahre?	Organisiert das Systemmodell in größere Einheiten durch logische Zusammenfassung von Modellelementen. Modellierung von Abhängigkeiten und Inklusion möglich.
Objektdiagramm	Welche innere Struktur besitzt mein System zu einem bestimmten Zeitpunkt (zur Laufzeit) Klassendiagrammschnappschuss?	Zeigt Objekte und deren Attributbelegungen zu einem bestimmten Zeitpunkt. Wird nur beispielhaft zur Veranschaulichung verwendet. Detailniveau wie im Klassendiagramm. Sehr gute Darstellung von Mengenverhältnissen.
Kompositionsstrukturdiagramm	Wie sieht das Innenleben einer Klasse, einer Komponente, eines Systemteils aus?	Ideal für die Top-down-Modellierung des Systems (Guru-Teil-Hierarchien). Mittleres Detailniveau, zeigt Teile eines „Gesamtelements“ und deren Mengenverhältnisse Präzise Modellierung der Teile-Beziehungen über spezielle Schnittstellen (Ports) möglich.
Komponentendiagramm	Wie werden meine Klassen zu wieder verwendbaren, verwaltbaren Komponenten zusammengefasst und wie stehen diese miteinander in Beziehung?	Zeigt Organisation und Abhängigkeiten einzelner technischer Systemkomponenten. Modellierung angebotener und benötigter Schnittstellen möglich.
Verteilungsdiagramm	Wie sieht das Einsatzumfeld (Hardware, Server, Datenbanken, ...) des Systems aus? Wie werden die Komponenten zur Laufzeit wohin verteilt?	Zeigt das Laufzeitumfeld des Systems mit den „greifbaren“ System-teilen (meist Hardware). Darstellung von „Softwaresvern“ möglich. Hohes Abstraktionsniveau, kaum Notationselemente.
Use-Case-Diagramm	Was leistet mein System für seine Umwelt (Nachbarsysteme, Stakeholder)?	Präsentiert die Äußersicht auf das System. Geeignet zur Kontextabgrenzung. Hohes Abstraktionsniveau, einfache Notationsmittel.
Aktivitätsdiagramm	Wie läuft ein bestimmter Ablauf/orientierter Prozess oder ein Algorithmus ab?	Sehr detaillierte Visualisierung von Abläufen mit Bedingungen, Schleifen, Verzweigungen. Parallelisierung und Synchronisation möglich. Darstellung von Daten- und Kontrollflüssen.
Zustandsautomat	Welche Zustände kann ein Objekt, eine Schnittstelle, ein Use-Case, ... , bei welchen Ereignissen annehmen?	Präzise Abbildung eines Zustandsmodells mit Zuständen, Ereignissen, Nebenläufigkeiten, Bedingungen, Ein- und Ausstiffaktionen. Schachtelung möglich.
Sequenzdiagramm	Wer tauscht mit wem welche Informationen in welcher Reihenfolge aus?	Stellt detailliert den Informationsaustausch zwischen Kommunikationspartnern dar. Sehr präzise Darstellung der zeitlichen Abfolge auch mit Nebenläufigkeiten. Schachtelung und Flusssteuerung (Bedingungen, Schleifen, Verzweigungen) möglich.
Kommunikationsdiagramm	Wer kommuniziert mit wem? Wer „arbeitet“ im System zusammen?	Stellt den Informationsaustausch zwischen Kommunikationspartnern dar. Übersicht steht im Vordergrund (Details und zeitliche Abfolge weniger wichtig).
Timing-Diagramm	Wann befinden sich verschiedene Interaktionspartner in welchem Zustand?	Visualisiert das exakte zeitliche Verhalten von Klassen, Schnittstellen. Geeignet für die Detailbetrachtungen, bei denen es überaus wichtig ist, dass ein Ereignis zum richtigen Zeitpunkt eintritt.
Interaktionsübersichts-diagramm	Wann läuft welche Interaktion ab?	Verbindet Interaktionsdiagramme (Sequenz-, Kommunikations- und Timing-Diagramme) auf TOP-Level-Ebene. Hohes Abstraktionsniveau. Leseansatz für Interaktionsdiagramme.

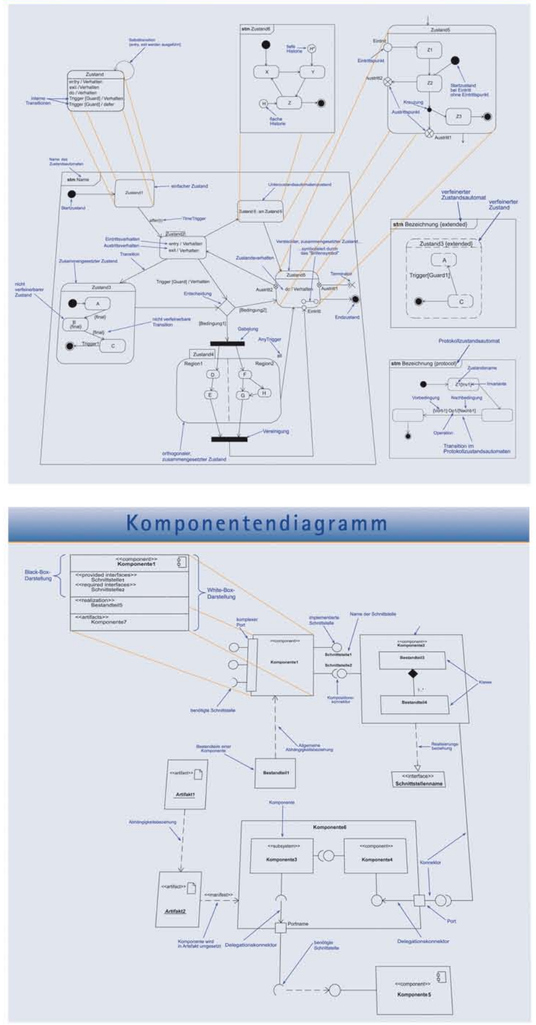
Klassendiagramm



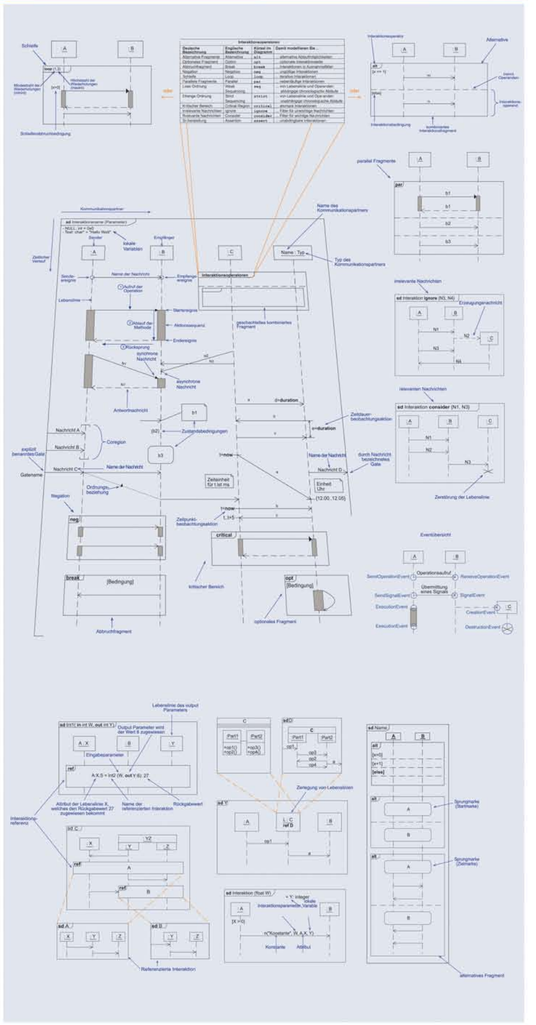
Aktivitätsdiagramm



Zustandsautomat

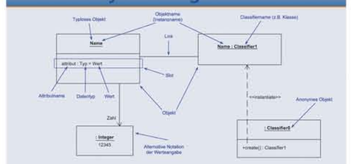


Sequenzdiagramm

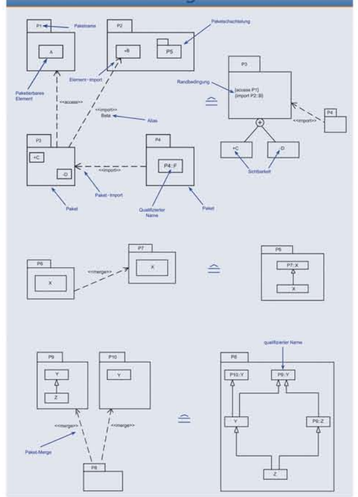


Komponentendiagramm

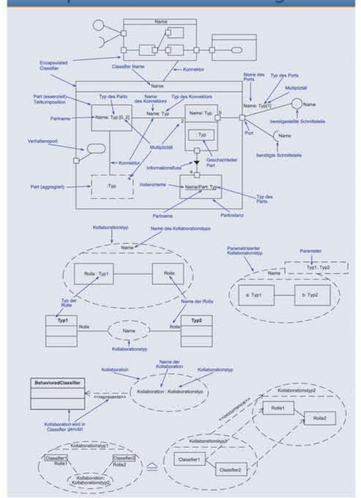
Objektdiagramm



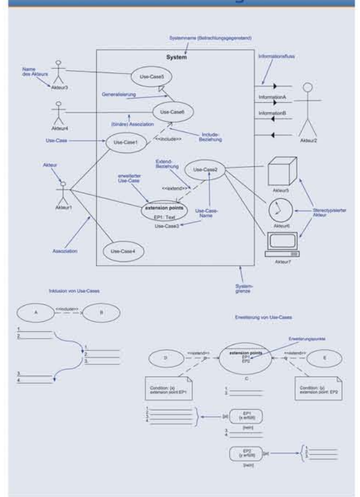
Paketdiagramm



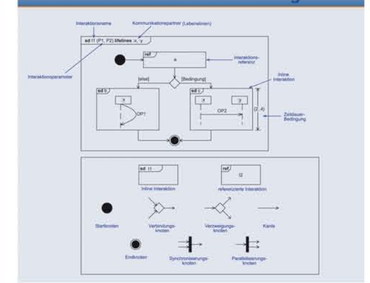
Kompositionsstrukturdiagramm



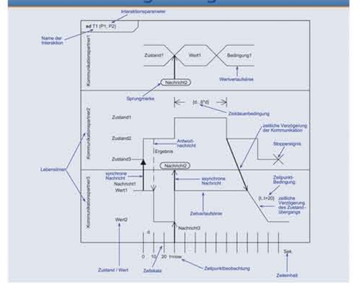
Use-Case-Diagramm



Interaktionsübersichtsdiagramm

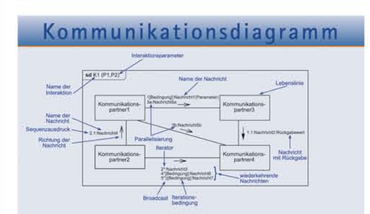


Timing-Diagramm

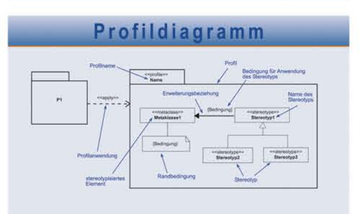


Verteilungsdiagramm

Kommunikationsdiagramm



Profildiagramm



Formale Diagramme sind wichtig im harten Geschäft um funktionierende Systeme. Und wir kennen uns richtig gut damit aus!

“Als Beteiligter am Standardisierungsprozess ist “UML 2 glasklar” für mich das beste derzeit verfügbare Buch am deutsch- und englischsprachigen Markt zu diesem Thema.”
Dr. Marko Bogner, Gründer der Gentleware AG

ISBN: 978-3446430570
Autoren: Chris Rupp, Stefan Queins, die SOPHISTen

Die Website zum Buch: **www.uml-glasklar.com**

Wir können auch anders...

Mehr über unsere Bücher erfahren Sie unter **www.sophist.de/publikationen**.

Und wenn es mal brennt...

... rufen Sie ein Sondereinsatzkommando der SOPHISTen

Machen Sie sich unser Wissen zu nutze! Buchen Sie SOPHIST Berater, um ...

- Management und Mitarbeiter gezielt über die UML 2 oder Requirements-Engineering zu informieren und zu motivieren.
- Mitarbeiter zu schulen, welche professionell mit Objektorientierung und Requirements-Engineering arbeiten sollen.
- Software- und Systementwicklung mit der UML 2 problemlos durchzuführen.
- professionell in die UML-Systementwicklung einzusteigen oder umzusteigen.
- von Anfang an die richtigen Schritte zu gehen, um später Aufwände und Kosten zu vermeiden.

Die Spezialisten von SOPHIST helfen Ihnen mit Know-how und Projekterfahrung zielgenau da, wo Sie es im Moment brauchen.

Rufen Sie uns an unter **+49 (0) 911 40 900-0** oder mailen Sie uns an **heureka@sophist.de**

SOPHIST GmbH
Vordere Cramergasse 13
90478 Nürnberg
Deutschland

Telefon: **+49 (0) 911 40 900-0**
Fax: **+49 (0) 911 40 900-99**
E-Mail: **heureka@sophist.de**

Möchten Sie mehr Informationen zu unseren Dienstleistungen und Produkten? Besuchen Sie uns unter **www.sophist.de**