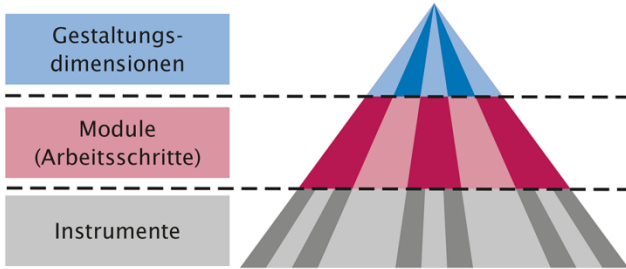


PROMIDIS Handlungsleitfaden



Instrument

Unified Modelling Language

Ziel

Der Begriff Modell beschreibt ein vereinfachtes Abbild der Wirklichkeit, dessen Ziel es ist, die Komplexität gegenüber der Realität auf relevante Attribute zu reduzieren. Das kann sinnvoll sein, um Sachverhalte auf einer vereinfachten abstrahierten Ebene darzustellen und um so die Verständlichkeit und den Überblick über einen komplexen Sachverhalt zu erhöhen.

Die Unified Modeling Language (UML) ist eine weitverbreitete Modellierungssprache, die sowohl von der Object Management Group (OMG) als auch von der International Organization for Standardization (ISO) in ISO-19505 standardisiert ist. UML ist sehr vielseitig und kann zur Modellierung unterschiedlichster Sachverhalte auf mehreren Ebenen verwendet werden.

Nutzbar bei den Modulen

IV. Produktivitätscontrolling

1. Prozesse auswählen und analysieren
2. Geschäftsdaten analysieren und visualisieren
4. Information Dashboard Design

Aufwand

Der Aufwand variiert stark und ist abhängig von der Komplexität, die mit dem Modell dargestellt werden soll. Ein Fahrrad in UML darzustellen benötigt 10 Minuten, ein Flugzeug möglicherweise Monate.

Vergleich

Vorteile

UML ist sehr vielseitig einsetzbar und mit mehreren Sprachelementen für die Modellierung von vielen unterschiedlichen Sachverhalten geeignet. Hinzu kommt, dass UML eine Kombination der Sprachelemente erlaubt, sodass sehr unterschiedliche Diagramme erstellt werden können, die perfekt auf den jeweiligen Sachverhalt zugeschnitten sind. Diese Flexibilität sowie die mehrfache Standardisierung

lassen UML zu einem der Hauptstandards für Modellierung werden.

Nachteile

Bei der Vielzahl von möglichen Modellierungsmethoden wird deutlich, dass nicht nur die richtige Modellierung, sondern auch die Wahl der richtigen Modellierungssprache ausschlaggebend für die Güte eines Modells ist.

Die Auswahl der Sprache ist die erste wichtige Entscheidung im Modellierungsprozess. Eine falsche Einschätzung birgt ein hohes Risiko, da die Wahl einer falschen Sprache dazu führen kann, dass wichtige Systemaspekte nicht erkannt und daraus resultierende Aktivitäten nicht ausgeführt werden.

Vorgehensweise

Die mit UML in der Version 2.0 erstellbaren Diagrammtypen lassen sich auf oberster Ebene in die zwei Gruppen der Strukturdiagramme und Verhaltensdiagramme einteilen.

Strukturdiagramme	Verhaltensdiagramme
Klassendiagramm	Aktivitätsdiagramm
Komponentendiagramm	Anwendungsfalldiagramm
Kompositionsstrukturdiagramm	Interaktionsübersichtsdiagramm
Objektdiagramm	Kommunikationsdiagramm
Paketdiagramm	Sequenzdiagramm
Profildiagramm	Zeitverlaufdiagramm
Verteilungsdiagramm	Zustandsdiagramm

Abb. 1: Diagrammtypen

Strukturdiagramme

Strukturdiagramme eignen sich zur Modellierung von statischen, zeitunabhängigen Elementen. Sie werden verwendet, um den Zustand bzw. den logischen Aufbau eines Sachverhalts darzustellen. Es gibt sieben unterschiedliche Arten von Strukturdiagrammen in UML.

Ein bekanntes ist das **UML-Klassendiagramm**, das die Beziehungen, Attribute und Funktionen von Softwareklassen in objektorientierten Programmiersprachen darstellt. Will man zum Beispiel eine Kundenbestellung modellieren, müssen drei Hauptklassen - Kunde, Bestellung und Artikel - dargestellt werden. Hinzu kommen passende Attribute wie etwa der Nachname des Kunden.

Das Zeichen vor dem Attribut steht für seine Sichtbarkeit (Abb.2). In diesem Fall deutet ein - an, dass das Attribut „private“ ist, sodass nur die Klasse selbst es sehen kann. Funktionen stehen im Klassendiagramm unterhalb der Trennlinie und sind durch zwei Klammern gekennzeichnet. Sie besitzen im Beispiel die Sichtbarkeit + für public, was so viel wie einen uneingeschränkten Zugriff bedeutet.

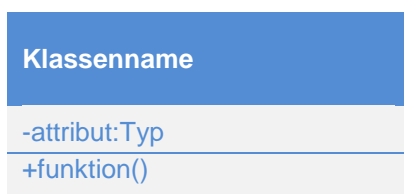


Abb. 2: Beziehungstypen in UML

Die Klassen Kunde, Bestellung und Artikel müssen durch Assoziationen verbunden werden, an denen die Multiplizitäten verzeichnet sind. Sie geben die mengenmäßige Relation der Objekte an. Zum Beispiel wird aus dem Diagramm deutlich, dass jede Bestellung genau einem Kunden zuzuordnen ist. Andersherum können einem Kunden jedoch mehrere Bestellungen zugeordnet werden.

Außerdem zeigt das Modell noch Unterschiede auf, z. B. mit den zwei Klassen Standard-Kunde und Prime-Kunde. Diese sind durch eine Vererbungsverknüpfung mit der Klasse Kunde verknüpft, sodass sie das Kunden-Objekt weiter spezifizieren und zwei Typen von Kunden einführen. Weitere in UML modellierbare Beziehungen, werden in Abbildung 2 dargestellt.

Die **Assoziation** beschreibt eine Beziehung zwischen zwei Klassen, für die Multiplizitäten angegeben werden können. Beispielsweise besitzt ein Kunde null bis unendlich Bestellungen.

Die **Aggregation** beschreibt, dass ein Objekt aus einem anderen besteht. Zum Beispiel besteht ein Gebäude aus mehreren Räumen.

Die **Komposition** stellt einen Spezialfall der Aggregation dar und besagt, dass die Teile nicht ohne das große Ganze existieren können.

Die **Generalisierung** drückt aus, dass eine Klasse die Spezialisierung einer anderen bildet und somit eine besondere Ausprägung dieser darstellt.

Realisierungen sind ähnlich, verwenden jedoch Schnittstellen, die sie weiter spezialisieren.

Verhaltensdiagramme

Neben den Strukturdiagrammen gibt es noch sieben Verhaltensdiagramme, die eine dynamische Sicht auf ein System erzielen und zur Darstellung von Prozessen, zeitlichen Abläufen oder Interaktionen zwischen prozessbeteiligten Einheiten geeignet sind.

Ein Diagramm, das den dynamischen Ablauf eines Prozesses darstellt ist zum Beispiel das **Aktivitätsdiagramm**. Ähnlich wie bei der ereignisgesteuerten Prozesskette (EPK) handelt es sich bei Aktivitätsdiagrammen um kontrollflussorientierte Modelle, die unterschiedliche Wege durch Verzweigungen und Zusammenführungen abbilden.

Wie in Abbildung 3 zu erkennen, hat auch die Notation Ähnlichkeit zur EPK. Das Aktivitätsdiagramm versucht dabei leicht verständlich und somit für eine breite Zielgruppe zugänglich zu sein.

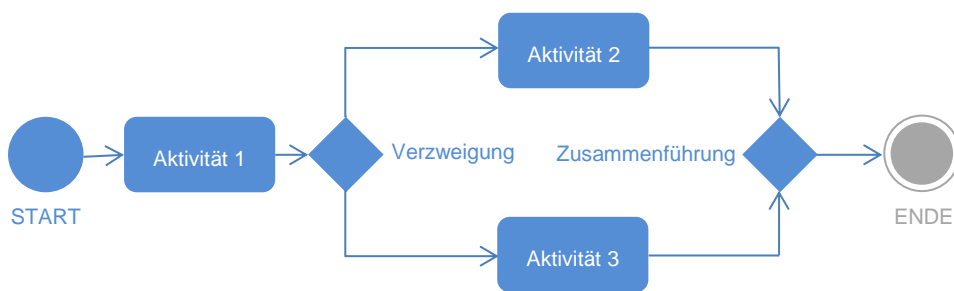


Abb. 3: Elemente des UML Aktivitätsdiagramms

Im Gegensatz zur EPK lassen Aktivitätsdiagramme jedoch mehr Freiräume, sodass sich mit ihrer Hilfe sehr schnell und einfach Diagramme erstellen lassen, die für die Analyse von Geschäftsvorfällen herangezogen werden können. Es ist ersichtlich, dass ähnliche Elemente, jedoch unterschiedliche Symbole zur Darstellung verwendet werden.

Ein weiteres Beispiel für ein Verhaltensdiagramm stellt das **Anwendungsfalldiagramm** dar. Es zeigt Fallbeispiele, die eine Dienstleistung bzw. eine Software erfüllen muss. Dazu werden alle an einer Dienstleistung beteiligten Personen dargestellt, die bestimmte Aktivitäten auf einem System ausführen. Durch Beziehungen wie „include“ oder „extend“ werden Abhängigkeiten modelliert, die verdeutlichen, aus welchen Teilkomponenten eine Aktivität besteht.

Weiterführende Informationen

- > Kecher, C. (2009): UML 2 – Das umfassende Handbuch, Galileo Computing, 3. Auflage, Bonn: Galileo Press
- > Becker, J.; Probandt, W.; Vering, O. (2012): Grundsätze ordnungsgemäßer Modellierung, Heidelberg: Springer Gabler Verlag
- > Becker, J.; Rosemann, M.; Schütte, R: Grundsätze ordnungsgemäßer Modellierung, Wirtschaftsinformatik, 37 (1995)5, S. 435-445

Impressum

Autor: Andreas Zolnowski; Universität Hamburg
 Redaktion: Alexander Sonntag, Beate Schlink;
 RKW Kompetenzzentrum
 November 2015

Diese Publikation wurde im Rahmen des Projektes **„Produktivitätsmanagement für industrielle Dienstleistungen stärken“ (PROMIDIS)** erstellt.