
IREB Certified Professional for Requirements Engineering - Requirements Modeling, Advanced Level -

Lehrplan

Version 1.0
Ausgabe vom 1. März 2011

Nutzungsbedingungen:

1. Jede Einzelperson und jeder Seminaranbieter darf den Lehrplan als Grundlage für Seminare verwenden, sofern die Inhaber der Urheberrechte als Quelle und Besitzer des Urheberrechts anerkannt und benannt werden. Des Weiteren darf der Lehrplan zu Werbezwecken nur mit Einwilligung des IREB e.V. verwendet werden.

2. Jede Einzelperson oder Gruppe von Einzelpersonen darf den Lehrplan als Grundlage für Artikel, Bücher oder andere abgeleitete Veröffentlichungen verwenden, sofern die Autoren und das IREB e.V. als Quelle und Besitzer des Urheberrechts genannt werden.

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Die Verwertung ist – soweit sie nicht ausdrücklich durch das Urheberrechtsgesetz (UrhG) gestattet ist – nur mit Zustimmung der Berechtigten zulässig. Dies gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmung, Einspeicherung und Verarbeitung in elektronischen Systemen und öffentliche Zugänglichmachung.

Danksagung

Dieser Lehrplan wurde erstellt von Manfred Ferken, Peter Hruschka, Stefan Queins und Thorsten Weyer.

Allen sei für das ehrenamtliche Engagement gedankt.

Urheberrecht © 2011 des Lehrplans IREB Certified Professional for Requirements Engineering Requirements Modeling , Advanced Level, besitzen die aufgeführten Autoren. Die Rechte sind übertragen auf das IREB International Requirements Engineering Board e.V.

Vorwort

Dieses Modul ist Basis für die Weiterbildung von Requirements Engineers, Business Analysts, Systemanalytiker und alle anderen Projektbeteiligten, die gewillt sind, mit Modellen im Requirements Engineering zu arbeiten. Ziel ist es, Wissen zu vermitteln, wie man effektiv mit Modellen umgeht und sie jeweils zielsicher und pragmatisch einsetzen kann.

Zweck des Dokumentes

Dieser Lehrplan definiert die fortgeschrittenen Stufe (Advanced Level) des Moduls Modeling des Certified Professional for Requirements Engineering Zertifikats des International Requirements Engineering Board (IREB). Der Lehrplan dient den Ausbildungsanbietern als Grundlage für die Erstellung ihrer Kursunterlagen. Die Lernenden können sich anhand des Lehrplans auf die Prüfung vorbereiten.

Inhalt des Lehrplans

Inhaltsabgrenzung

In dieser fortgeschrittenen Stufe wird das Thema „Requirements modellbasiert ermitteln und dokumentieren“ vertieft. Der Schwerpunkt der Ausbildung liegt dabei auf der Modellierung von Informationsstrukturen, Funktionen, Verhalten, sowie der Szenariomodellierung im Requirements Engineering. Für die Vertiefung der anderen Lehreinheiten im Foundation Level des „Certified Professional for Requirements Engineering“ (wie z. B. „Anforderungen ermitteln“, „Anforderungen verwalten“ und „Anforderungen prüfen und abstimmen“) sind eigene Module des Advanced Levels veröffentlicht oder in Planung.

Die folgende Tabelle gibt einen Überblick über das Modul „Requirements Modeling“

	Thema	Erläuterung	Zeitbedarf
1	Grundlagen der Modellierung im Requirements Engineering	Was ist ein Modell? Was sind die Vorteile der Modellierung im RE? Wie kann man die Qualität von Modellen im RE beurteilen?	0,5 Stunden
2	Informationsmodellierung	Wie werden im RE Informationsstrukturen (z. B. Begriffssysteme der Fachterminologie, Datenstrukturen) modelliert?	5,5 Stunden
3	Funktions- und Verhaltensmodellierung	Wie werden Use-Cases (Anwendungsfälle) im RE modelliert? Wie werden datenflussorientierte Funktionsmodelle, Ablaufmodelle und Zustandsmodelle im RE konstruiert und ergänzend genutzt?	7 Stunden
4	Szenarienmodellierung	Wie werden Szenarien im RE durch Modelle dokumentiert? Wie können Szenariomodelle im RE weiter genutzt werden?	3 Stunden
5	Zusammenspiel von Modellen	Wie kann eine modellübergreifende Qualitätssicherung durchgeführt werden? Wie sind die Modelle des RE in den Entwicklungsprozess eingebettet? Wie ist der Zusammenhang zu einer Anforderungsspezifikation?	2 Stunden

Detailierungsgrad

Der Lehrplan wurde auf der Grundlage des Foundation Levels entwickelt und behandelt die Themen Informationsmodellierung, Ablaufmodellierung, Zustandsmodellierung und die Modellierung von Szenarien in wesentlich detaillierterer Form; d.h. es werden die Grundlagen des Foundation Levels vertieft und ergänzende Aspekte vermittelt, die Expertenwissen und Best Practices beinhalten.

Lernziele / Kognitive Stufen des Wissens

Jeder Abschnitt dieses Lehrplans ist einer kognitiven Stufe zugeordnet. Eine höhere Stufe umfasst die niedrigeren Stufen. In den Formulierungen der Lernziele werden für die Stufe K1 das Verb „kennen“ und für die Stufe K2 die Verben „können und anwenden“ stellvertretend für die nachfolgend aufgelisteten Verben der gleichen Stufe verwendet.

- **K1 (kennen):** aufzählen, bezeichnen, erkennen, nennen, wiedergeben
- **K2 (können und anwenden):** analysieren, anwenden, ausführen, begründen, beschreiben, beurteilen, darstellen, entwerfen, entwickeln, ergänzen, erklären, erläutern, ermitteln, formulieren, identifizieren, interpretieren, schlussfolgern, übertragen, unterscheiden, vergleichen, verstehen, vorschlagen, zusammenfassen



Alle Begriffe aus dem Glossar sind zu kennen (K1), auch wenn sie in den Lernzielen nicht explizit genannt sind.

Im Lehrplan wird die Abkürzung RE für Requirements Engineering verwendet.

Lehrplanaufbau

Der Lehrplan besteht aus fünf Hauptkapiteln. Ein Kapitel umfasst eine Lehreinheit (LE). Jeder Haupttitel eines Kapitels beinhaltet die kognitive Stufe des Kapitels, das ist die höchste Stufe der Teilkapitel. Weiterhin werden die Unterrichtszeiten genannt, welche in einem Kurs mindestens für dieses Kapitel aufgewendet werden sollten. Wichtige Begriffe des Kapitels, die im Glossar definiert sind, sind am Anfang des Kapitels aufgelistet.

Beispiel: LE 1 Modelle und Modellierung (K1)
Dauer: 0,5 Stunden
Begriffe: Stakeholder

Lernziele:
LZ 1.1 Begriffsabgrenzungen im Bereich der Modellierung, die Vor- und Nachteile von (grafischer) Modellierung gegenüber natürlichsprachlichen Beschreibungen kennen. (K1)
LE 1.1 Grundlagen der Modellierung

Das Beispiel zeigt, dass in Kapitel 1 Lernziele der Stufe K1 enthalten sind und 30 Minuten für das Lehren des Materials in diesem Kapitel vorgesehen sind. Die Begriffe Modell und grafisches Modell bzw. deren Definition aus dem Glossar müssen gekannt werden.

Im ersten Unterkapitel geht es bei der Lehreinheit 1.1 um Grundlagen der Modellierung. Begriffsabgrenzungen im Bereich der Modellierung sowie die Vor- und Nachteile von (grafischer) Modellierung gegenüber natürlich-sprachlichen Beschreibungen sind zu kennen

Die Prüfung

Auf diesem Lehrplan basiert die Prüfung für das Zertifikat des Requirements Modeling, Advanced Level.



Eine Prüfungsfrage kann Stoff aus mehreren Kapiteln des Lehrplans abfragen. Alle Abschnitte dieses Lehrplans können geprüft werden.

Das Format der Prüfung ist Multiple Choice sowie eine bewertete Hausaufgabe, Details regelt die Prüfungsordnung.

Prüfungen können unmittelbar im Anschluss an einen Kurs, aber auch unabhängig davon (z. B. in einem Prüfzentrum) abgelegt werden. Die vom IREB lizenzierten Zertifizierungsstellen sind auf der Webseite aufgelistet: <http://certified-re.de>.

Inhalt

Nutzungsbedingungen.....	1
Danksagung.....	2
Vorwort.....	2
Zweck des Dokumentes.....	2
Inhalt des Lehrplans.....	2
Inhaltsabgrenzung.....	2
Detaillierungsgrad.....	3
Lernziele / Kognitive Stufen des Wissens.....	4
Lehrplanaufbau.....	4
Die Prüfung.....	5
LE 1 Modell und Modellierung (K1).....	7
LE 1.1 Grundlagen der Modellierung (K1).....	7
LE 1.2 Überblick über die Einsatzmöglichkeiten verschiedener grafischer Modellierungssprachen sowie deren Stärken und Schwächen (K1).....	8
LE 1.3 Kriterien für die Beurteilung der Qualität von Modellen (K1).....	8
LE 2 Informationsmodellierung (K2).....	9
LE 2.1 Auffrischung des Foundation Levels (K2).....	10
LE 2.2 Modellierung von Fachklassen, Attributen und Datentypen (K2).....	10
LE 2.3 Modellierung von einfachen Beziehungen (K2).....	11
LE 2.4 Generalisierung und Spezialisierung (K2).....	11
LE 2.5 Evolution der Informationsmodelle (K2).....	12
LE 2.6 Bewertung von Informationsmodellen (K2).....	12
LE 3 Funktions- und Verhaltensmodellierung (K2).....	13
LE 3.1 Funktions- und Verhaltensmodelle: Auffrischung (K2).....	14
LE 3.2 Use-Case-Modelle (K2).....	14
LE 3.3 Funktionsmodellierung (K2).....	14
LE 3.4 Verhaltensmodellierung (K2).....	15
LE 3.5 Das Zusammenspiel von Funktionen und Verhalten (K2).....	15
LE 4 Szenarienmodellierung (K2).....	17
LE 5 Umgang mit Modellen und deren Einbettung inden Gesamtprozess (K2).....	19
Glossar.....	21
Literatur.....	24

LE 1 Modell und Modellierung (K1)

Dauer: 0,5 Stunde
Begriffe: Modell, grafisches Modell

Lernziele:

- LZ 1.1 Begriffsabgrenzungen im Bereich der Modellierung, die Vor- und Nachteile von (grafischer) Modellierung gegenüber natürlich-sprachlichen Beschreibungen kennen. (K1)
- LZ 1.2 Einen Überblick über die Einsatzmöglichkeiten verschiedener grafischer Modellierungssprachen sowie deren Stärken und Schwächen kennen. (K1)
- LZ 1.3 Möglichkeiten zur Beurteilung der Qualität von Modellen kennen. (K1)

Voraussetzung: Grundkenntnisse über Kommunikation zwischen Menschen. [Wiederholung aus dem Foundation Level. Wie kann man kommunizieren? Über (Umgangs-) Sprache oder über (formalisierte) Modelle.]

In dieser Lehreinheit werden die Grundlagen der Modellierung im Requirements Engineering betrachtet. Hierzu wird der Modellbegriff und die Frage, wieso man modellieren sollte, genauer betrachtet. Darüber hinaus wird in dieser Lehreinheit ein Überblick über die im RE eingesetzten Arten von Modellen gegeben und die Stärken und Schwächen der einzelnen Modelle für die Verwendung im RE diskutiert. Zum Abschluss dieser Lehreinheit werden die Grundlagen zur Beurteilung der Qualität von Modellen im RE betrachtet.

Literatur: Kap 6 „Anforderungen modellbasiert dokumentieren“ aus [Pohl und Rupp]

LE 1.1 Grundlagen der Modellierung (K1)

Dauer: 10 Minuten

Ein Modell ist eine Abbildung eines zugrunde liegenden Realitätsausschnitts, die zweckgerichtet von bestimmten Aspekten der Realität abstrahiert und dadurch die Fähigkeit besitzt, nur solche Informationen über die zugrunde liegende Realität abzubilden, die für den jeweiligen Verwendungszweck von Bedeutung sind. Ein Modell setzt sich aus vielen Modellelementen zusammen, die nichts anderes als Ausprägungen von Notationselementen darstellen. Diese Notationselemente mit ihrer Syntax und Semantik bilden den Sprachumfang der Modellierungssprache. Grafische Modelle werden mittels Diagrammen repräsentiert, in denen die Modellelemente dargestellt werden, die gemeinsam einen spezifischen Aspekt abbilden.

Im Gegensatz zur Dokumentation von Informationen in natürlicher Sprache besitzen Modelle im RE in der Regel eine grafische Notation, zugehörige definierte Abstraktionsmechanismen und einen höheren Formalisierungsgrad, wodurch die Zweckmäßigkeit der Modelle, im Vergleich zum Einsatz natürlicher Sprache, in der Regel erheblich erhöht wird.

LE 1.2 Überblick über die Einsatzmöglichkeiten verschiedener grafischer Modellierungssprachen sowie deren Stärken und Schwächen (K1)

Dauer: 10 Minuten

Im Requirements Engineering kommen verschiedenste grafische Modellierungssprachen zum Einsatz. Die Eignung einer Modellart im RE hängt dabei immer vom jeweiligen Verwendungszweck ab. Zur Modellierung von Informationsstrukturen im RE werden z.B. Klassendiagramme der UML verwendet. Zur Modellierung von Funktionen kommen z.B. Datenflussmodelle der Strukturierten Analyse zum Einsatz sowie Statecharts zur zustandsorientierten Verhaltensmodellierung. Im Rahmen dieses Lernziels wird ein Überblick über die verschiedenen Modellarten im RE gegeben und deren Stärken und Schwächen erläutert.

LE 1.3 Kriterien für die Beurteilung der Qualität von Modellen (K1)

Dauer: 10 Minuten

Für die Verwendung grafischer Modellierungssprachen ist es notwendig, die Qualität von erstellten Modellen beurteilen zu können. Ein einfaches Schema ermöglicht eine Beurteilung der Qualität von Modellen anhand der drei Ebenen der Semiotik von Modellen (siehe CPRE FL). So wird die syntaktische Qualität eines Modells danach beurteilt, inwieweit die syntaktischen Regeln der gewählten Modellierungssprache eingehalten wurden. Die semantische Qualität eines grafischen Modells wird danach beurteilt, inwieweit die in dem Modell dokumentierten Informationen im Hinblick auf die zugrunde liegende Realität gültig sind. Die pragmatische Qualität eines Modells bestimmt sich darüber, inwieweit das Modell für den Verwendungszweck auch tatsächlich geeignet ist.

Alle drei Qualitäten können durch spezifische Qualitätskriterien geprüft werden.

LE 2 Informationsmodellierung (K2)

Dauer: 5,5 Stunden (davon mindestens 2 Stunden Übungen)
Begriffe: Informationsmodell, Datenmodell, Domäne, Fachbegriff (Fachklassen und Datentypen) (Verwandte Begriffe: Entität, Entitätsklasse), Glossar, Klassendiagramm, Attribut, Beziehung (Assoziation), Generalisierung und Spezialisierung

Lernziele:

- LZ 2.1.1 Für Fachbegriffe und Beziehungen einschätzen können, was besser in Modellen spezifiziert werden kann und wofür eine natürlich sprachliche Dokumentation besser geeignet ist. (K2)
- LZ 2.2.1 Heuristiken zum Finden von Fachklassen können und anwenden (K2)
- LZ 2.2.2 Heuristiken zum Finden von Attributen und deren Datentypen können und anwenden (K2)
- LZ 2.3.1 Heuristiken zum Finden von einfachen Beziehungen können und anwenden (K2)
- LZ 2.4.1 Heuristiken zum Finden von Generalisierungen und Spezialisierungen können und anwenden (K2)
- LZ 2.5.1 Aus einem vorliegenden Modell weitere Fachbegriffe und Beziehungen ableiten können und anwenden (K2)
- LZ 2.6.1 Fehler und Schwachstellen in Klassendiagrammen finden können (K2)

Voraussetzung: Grundkenntnisse der UML-Syntax für Klassendiagramme, Verständnis von Multiplizitäten, Kenntnis der Konzepte Aggregation, Komposition und Generalisierung, Klassen und Objekte unterscheiden können.

Für das präzise Verständnis einer Anwendungsdomäne sind deren fachliche Begriffe und die Beziehungen zwischen diesen von fundamentaler Bedeutung. Ein gutes Glossar mit Definitionen der fachlichen Begriffe wird in manchen Fällen für ein Anforderungsdokument ausreichen. Tieferes Verständnis der Domäne wird jedoch durch Ordnung aller fachlichen Begriffe in Form eines Klassenmodells erreicht. Das Klassenmodell repräsentiert die statischen Informationsstrukturen der Domäne, die für das gewünschte System wichtig sind. Es besteht aus dem Klassendiagramm und den dazugehörigen Definitionen von Entitäten, Attributen und Beziehungen. Das Glossar wird also nicht nur (grafisch) strukturiert, sondern auch um Beziehungen angereichert.

Das Lesen eines vorliegenden Klassendiagramms ist verhältnismäßig leicht, das korrekte Erstellen eines solchen Modells ist demgegenüber wesentlich schwieriger. Da die Modellierung von Informationsstrukturen im RE darüber hinaus von großer Bedeutung ist, widmet sich diese Lehreinheit vollständig der Art und Weise, wie die Modellierung von Informationsstrukturen im RE eingesetzt werden kann. Ein wesentliches Ziel dieser Lehreinheit ist das Erlernen und Beherrschen von Praktiken zum Aufbau des Klassenmodells. Unter Nutzung unterschiedlicher Quellen (wie Umgangssprache, alten Systemdokumenten, anderen Modellen, Reports und Bildschirmmasken, ...) helfen sie beim Aufbau eines präzisen Informationsmodells, das möglichst einfach gepflegt und langfristig verwendet werden kann.

LE 2.1 Auffrischung des Foundation Levels (K2)

Dauer: 15 Minuten

Wiederholung der Grundelemente aus dem Foundation Level: Wie modelliert man Fachklassen, Attribute und Beziehungen (in der UML-Notation).

Folgende Methoden unterstützen das benutzerzentrierte RE:

LE 2.2 Modellierung von Fachklassen, Attributen und Datentypen (K2)

Dauer: 1,25 Stunden (davon 0,5 Stunde Übung)

Die Grundlage von Informationsmodellen sind die Fachbegriffe, wie sie in der Sprache der Stakeholder verwendet werden. In diesem Abschnitt lernen Sie erste, grundlegende Heuristiken zum Auffinden dieser Fachbegriffe. Allgemein bekannt ist die Suche nach fachlich relevanten Substantiven, die jedoch eine ziemlich mageren Ausbeute an relevanten Fachklassen liefert: Nur 15% der Substantive in umgangssprachlichen Texten sind relevante Fachklassen. Deshalb müssen weitere Heuristiken an dieser Stelle eingesetzt werden, wie zum Beispiel die Werteheuristik, die festlegt, dass ein Begriff, der einen Wert annimmt, eher als Attribut, denn als Klasse modelliert wird. Neben diesen beiden prominenten Vertretern häufig eingesetzter Praktiken existieren erfahrungsbasierte Regeln, die den Requirements-Engineer bei der Erstellung eines qualitativ hochwertigen Informationsmodells unterstützen.

Informationsmodelle ohne Untermauerung durch präzise Definitionen sind jedoch nur unzureichend. Deshalb sollten gefundene Fachbegriffe mit einer präzisen Bedeutung hinterlegt werden. Aus solchen Definition ergeben sich weitere Hinweise auf Verbesserung und Stabilisierung der gefundenen Fachbegriffe.

Heuristiken helfen auch sich bei der Modellierung nicht in endlosen Debatten zu verlieren, sondern zeitnah Entscheidungen zu treffen.

LE 2.3 Modellierung von einfachen Beziehungen (K2)

Dauer: 1,25 Stunden (davon 0,5 Stunde Übung)

Für Requirements Engineers sollte eine gründliche Analyse der Beziehungen zwischen den Klassen einen hohen Stellenwert einnehmen. In den Beziehungen spiegelt sich wichtiges Fachwissen wider. Im Gegensatz zu Verhaltensmodellen und funktionalen Modellen (wie State Charts, Aktivitätsdiagramme, etc.) zeigen Beziehungen im Informationsmodell grundsätzliche, fachliche Zusammenhänge in der zu beschreibenden Realität auf. Beziehungen sind daher die Basis für langfristig verwendbare Informationsmodelle.

Auch zum Finden von Beziehungen gibt es Praktiken. Die einfachste auf Basis der Umgangssprache ist die Suche nach Verben, die im Zusammenhang mit zwei relevanten Substantiven auftreten. Weiterhin geben isolierte Klassen in Diagrammen einen deutlichen Hinweis auf noch fehlende Beziehungen. Aber auch Fremdschlüssel aus Datenbankschemata oder die Analyse von Oberflächen oder Reports helfen beim Finden der Beziehungen zwischen Fachklassen.

Wie bei den Fachbegriffen werden die Beziehungen erst durch eine präzise Spezifikation aussagekräftig. Dazu müssen Namen für Assoziationen und Rollen sowie Multiplizitäten angegeben werden.

LE 2.4 Generalisierung und Spezialisierung (K2)

Dauer: 1,25 Stunden (davon 0,5 Stunde Übung)

Eine häufig auftretende, spezielle Beziehung ist die Generalisierung/ Spezialisierung, d.h. die Bildung von Ober- und Unterklassen.

Zur Entscheidung, wann es sich lohnt eine Fachklasse zu spezialisieren bzw. mehrere gefundene Fachklassen zu generalisieren, gibt es neben den beiden Grundsätzen zur Klassifizierung („Alle Eigenschaften einer Fachklasse sollten für alle ihre Fachobjekte relevant sein“ und „Jede Eigenschaft sollte redundanzfrei nur an einer Stelle definiert sein“) wiederum eine Reihe praktischer Tipps und Empfehlungen, wann sich Ober-/Unterklassenbildung lohnt und wann nicht.

Dazu zählt die Untersuchung der gesprochenen oder aufgeschriebenen Sprache. Formulierungen mit „entweder-oder“ oder „ist ein“ sowie Wörter wie zum Beispiel „Art“, „Typ“, „Kategorie“ deuten auf eine Generalisierung oder Spezialisierung hin. Zum Finden von Spezialisierungen während der Überarbeitung eines Modells dient die Betrachtung der Attribute einer Fachklasse, in dem man die Frage stellt: Kann eine Spezialisierung entlang der Werte eines Attributs gebildet werden (zum Beispiel für das Attribut „Geschlecht“ bei einer Klasse „Person“)?

Tieferes Verständnis bezüglich Spezialisierung erreicht man durch Untersuchung detaillierter Eigenschaften der entstehenden Unterklassen in Bezug auf deren Disjunktheit (disjoint/overlapping) bzw. Vollständigkeit der Unterklassenbildung (complete/incomplete).

LE 2.5 Evolution der Informationsmodelle (K2)

Dauer: 1,25 Stunden (davon 0,5 Stunde Übung)

Die statisch-strukturellen Informationsmodelle sind meist die stabilste Art von Requirements-Modellen. Trotzdem entwickeln sie sich langsam, evolutionär weiter. Dieser Block behandelt Themen wie: Ab wann sollte ein Attribut als eigenständige Entität herausgelöst werden? Wie kann man durch ähnliche Attribute von Entitäten neue Entitäten und neue Beziehungen entdecken? Wann sollte man Beziehungsattribute modellieren und ab wann sollte man solche „Assoziativentitäten“ in eigenständige Entitäten auflösen?

Die Grenze zwischen Analyse und Design in diesem Bereich ist fließend. Requirements Engineers sollten wissen, wie weit sie bei bestimmten Vorgehensmodellen im Gesamtprojekt mit Informationsmodellen gehen können und sollen.

LE 2.6 Bewertung von Informationsmodellen (K2)

Dauer: 15 Minuten

In dieser Zusammenfassung werden die Kriterien zur Bewertung von Informationsmodellen vorgestellt, um ein gutes von einem weniger guten Informationsmodell zu unterscheiden?

Die Vorteile und Nachteile von Informationsmodellen, ihre Stärken und Schwächen werden zusammenfassend beurteilt. Da Informationsmodelle für komplexe Anwendungsbereiche nicht immer leicht verständlich für alle Stakeholder sind, kann das Informationsmodell Grundlage für ein Glossar als alternative Darstellung zu Klassendiagrammen sein.

LE 3 Funktions- und Verhaltensmodellierung (K2)

Dauer: 7 Stunden (davon 3 ¼ Stunden Übung)
Begriffe: Use-Case, Aktivität, Aktion, Aktivitätsdiagramm, Datenflussdiagramm, Zustand, Ereignis, Statechart

Lernziele:

- LZ 3.2.1 Use-Cases finden, Use-Case-Modellierung können und anwenden (K2)
- LZ 3.3.1 Sequenzielle Abläufe mit Aktivitäts- und Datenflussdiagrammen beschreiben können (K2)
- LZ 3.3.2 Datenflussmodelle und Ablaufmodelle jeweils zum richtigen Zweck einsetzen können und anwenden (K2)
- LZ 3.3.3 Kriterien für die Abstraktion von Funktionen kennen und sinnvoll einsetzen können und anwenden (K2)
- LZ 3.4.1 Reaktive Abläufe mit Statecharts modellieren können (K2)
- LZ 3.4.2 Komplexe Zustandsmodelle mit Hierarchisierung, Parallelität und Historie einsetzen können (K2)
- LZ 3.5.1 Die Unterscheidung von Funktionen und Zuständen gezielt nutzen können und anwenden. (K2)

Voraussetzung: Kenntnisse über Scope und Kontextbildung von Systemen, Kenntnis der UML-Syntax für Use-Case-Diagramme, Aktivitätsdiagramme und Zustandsmodelle (gemäß Foundation Level).

Programm = Daten + Algorithmen. Mit dieser markanten Feststellung hat Nikolaus Wirth einen komplexen Sachverhalt einprägsam zusammengefasst. Nach der Datenmodellierung im letzten Kapitel stehen in diesem Teil die Algorithmen im Vordergrund: das gewünschte oder geforderte Ablaufverhalten des Systems. Dieses kann mit diversen grafischen Modellen präzisiert werden. Bereits im Lehrplan des Foundation Levels ist die Zerlegung eines Systems in Use-Cases (Anwendungsfälle) ausführlich behandelt worden. Dieser Abschnitt widmet sich schwerpunktmäßig drei Themen:

- eher lineare funktionale Abläufe durch Aktivitätsdiagramme oder Datenflussdiagramme zu präzisieren,
- eher asynchrone Abläufe durch Statecharts zu spezifizieren und
- zu lernen, wann eher die eine Form von Diagrammen und wann die andere zum Einsatz kommen sollte.

Außerdem wird die Komplexitätsbeherrschung durch Hierarchisierung dieser Modelle betrachtet. Sie lernen, wie Sie auch komplexeste Abläufe durch sinnvolle Kombination von Funktions- und Zustandsmodellen überschaubar und verständlich spezifizieren können.

LE 3.1 Funktions- und Verhaltensmodelle: Auffrischung (K2)

Dauer: 15 Minuten

Aus dem Foundation-Level Lehrplan werden die Bedeutung der System- und Kontextabgrenzung (Scope-Festlegung), sowie die Syntax für Use-Cases (Anwendungsfälle) wiederholt. Sie frisken im Überblick die möglichen präziseren Beschreibungen von Abläufen und Verhalten auf, z. B. durch Aktivitätsdiagramme, Datenflussdiagramme, Zustandsmodelle, Ereignis-Prozess-Ketten, Petrinetze, Programmablaufpläne – von denen im Folgenden nur auf Aktivitätsdiagramme, Datenflussdiagramme und Statecharts genauer eingegangen wird.

LE 3.2 Use-Case-Modelle (K2)

Dauer: 1 Stunde (davon 0,5 Stunde Übung)

Use-Cases (Anwendungsfälle) stellen die größte Modellierung von Funktionen innerhalb des festgelegten Scopes dar. Im Mittelpunkt dieses Abschnitts steht ein tieferes Verständnis darüber, wie man Use-Cases systematisch identifiziert und zu welchem Zweck und in welchen Situationen Use-Case-Modelle konstruiert werden sollten. Zur Identifikation von Use-Cases werden typische Systematiken vorgestellt, wie sie in der Praxis häufig zum Einsatz kommen:

- die Konzentration auf externe Auslöser oder Zeitauslöser
- die Durchgängigkeit der Abläufe von Systemgrenze zu Systemgrenze

Sie lernen darüber hinaus, wie eine geeignete Granularität für Use Cases festgelegt wird und wie Use-Cases eines Systems sinnvoll voneinander abgegrenzt werden können. Zudem lernen Sie den gezielten Einsatz von „Include“-Beziehungen und „Extend“-Beziehungen, um damit die Verständlichkeit von Use-Case-Diagrammen zu erhöhen. Sie lernen darüber hinaus auch den zweckmäßigen Einsatz von Generalisierungsbeziehungen zwischen Use Cases und zwischen Akteuren im Systemkontext, um in Use-Case-Diagrammen über die gemeinsamen Eigenschaften von Use Cases oder Akteuren zu abstrahieren. Sie lernen auch die Strukturierung des Use-Case-Modells durch fachliche Komponenten und die Analyse und Dokumentation der Beziehung zwischen Use Cases und den umgebenden Geschäftsprozessen.

LE 3.3 Funktionsmodellierung (K2)

Dauer: 1,75 Stunden (davon 0,75 Stunde Übung)

In diesem Kapitel lernen Sie, Use-Cases entweder ablauforientiert (mittels Aktivitätsdiagrammen), andererseits datenflussorientiert (mittels Datenflussdiagrammen) zu modellieren. Dabei steht nicht die Syntax der beiden Diagrammarten im Vordergrund, sondern das grundsätzliche Denken in Datenflüssen oder das Denken in Steuerflüssen. Sie lernen und üben die Erstellung von detaillierten Funktionsmodellen auf verschiedenen Abstraktionsebenen.

Wie komplex muss eine Funktion sein, damit sich eine Zerlegung der Funktion lohnt? Wann und nach welchen Kriterien sollte man Funktionen bündeln oder aufteilen? Wie ist der Zusammenhang zwischen der Beschreibung einer Funktion und Black-Box-Anforderungen, die testbar sein sollen. Wann sollten Funktionen weiter grafisch zerlegt werden und wann ist eher eine textliche Spezifikation sinnvoll? Wie vermeidet man Redundanz und Widersprüche bei hierarchisch zerlegten Datenflussdiagrammen (Balancing-Regeln)?

Sie lernen zwei praktische Vorgehensweisen: einerseits komplexe Prozesse durch Funktionsmodelle zu zerlegen und top-down zu verfeinern. Andererseits aber auch, eine Vielzahl von gewünschten Funktionen, die für ein System in unterschiedlicher Granularität gefordert werden, bottom-up in Form von Funktionsmodellen zu bringen. Beides hilft, Lücken und offene Punkte bezüglich Abläufen aufzudecken und durch gezielte Nachfragen zu schließen.

LE 3.4 Verhaltensmodellierung (K2)

Dauer: 2 Stunden (davon 1 Stunde Übung)

Im Lehrplan des Foundation Level sind Statecharts bereits thematisiert worden. Diese basieren auf den Arbeiten von David Harel, der die endlichen Automaten um zusätzliche Modellierungskonzepte erweitert hat. Die daraus resultierenden erweiterten endlichen Automaten haben als Statecharts inzwischen Eingang in populäre Methoden und Notationen gefunden, u.a. auch in die UML.

In diesem Kapitel lernen Sie, komplexes Systemverhalten mittels Statecharts auszudrücken. Wie kann man durch Schachtelung die unterschiedlichen fachlichen Ereignisse priorisieren, um Prozesse entweder zu unterbrechen oder fortzuführen? Wie kann man bei „Störungen“ oder Unterbrechungen wieder dort weitermachen, wo man unterbrochen wurde?

Sie lernen auch, mit Hilfe fortgeschrittener Modellierungskonzepte die Komplexität der Statecharts für die Beschreibung paralleler Prozesse und deren Synchronisation zu verringern.

Sie lernen und üben, jedes gewünschte Verhalten auch in komplexen Prozessen, transparent und präzise auszudrücken.

LE 3.5 Das Zusammenspiel von Funktionen und Verhalten (K2)

Dauer: 2 Stunden (davon 1 Stunde Übung)

Typischerweise sind in komplexen Systemen die Abläufe nie nur linear oder nur zustandsgesteuert. Nachdem Sie in den letzten beiden Lehreinheiten gängige Modellierungssprachen zur Funktionsmodellierung und Verhaltensmodellierung kennengelernt haben, werden in dieser Lehreinheit die Auswahl einer geeigneten Modellart sowie die Zusammenhänge zwischen der Funktions- und Verhaltensmodellierung betrachtet.

Zunächst wird betrachtet, zu welchem Zweck und in welchen Situationen Verhaltensmodelle in Form von Statecharts eingesetzt werden sollten und wann sich demgegenüber eher der Einsatz von Funktionsmodellen, wie z. B. Datenflussmodellen oder Aktivitätsmodellen der UML anbietet. Sie lernen unter anderem wie Zustandsautomaten (z. B. Statecharts) zur zustandsbasierten Beschreibung des Lebenszyklus von Fachklassen eingesetzt werden können. Sie lernen, wie Verhaltensmodelle zur zustandsbasierten Dokumentation der Beziehung zwischen dem System und dem Systemkontext verwendet werden können, um das reaktive Verhalten des betrachteten Systems gegenüber dem Systemkontext zu dokumentieren.

In Bezug auf die Funktionsmodellierung lernen sie, wie sie Datenflussdiagramme einsetzen, um Datenabhängigkeiten zu dokumentieren, u.a. auch Datenabhängigkeiten zwischen Funktionen bzw. Use Cases des betrachteten Systems. Darüber hinaus lernen sie, wie Aktivitätsdiagramme der UML dazu eingesetzt werden, Abläufe zu dokumentieren, wie z. B. die möglichen Abläufe eines Use Cases.

In dieser Lehreinheit lernen Sie auch, wie die Modelle der Verhaltens- und Funktionsperspektive zusammenhängen und welche Integrationspunkte es zwischen Verhaltens- und Funktionsmodellen gibt. Beispielsweise sind die in der Funktionsmodellierung betrachteten Funktionen in Verhaltensmodellen an Zustände und/oder an Zustandsübergänge gebunden, wodurch dokumentiert wird, in welchen Situationen eine Funktion vom betrachteten System ausgeführt wird und wann nicht. Sie lernen zudem auch, wie Sie die Vor- und Nachbedingungen von Use-Cases durch Zustände in Verhaltensmodellen dokumentieren und wie Sie verschiedene Use-Cases durch die Verwendung von Zustandsmodellen integrieren können.

LE 4 Szenarienmodellierung (K2)

Dauer: 3 Stunden (davon mindesten 1 Stunde Übung)
Begriffe: Szenario, Sequenzdiagramm, Kommunikationsdiagramm, Interaktionen

Lernziele:

- LZ 4.1 Beispielhafte Abläufe zur Gewinnung und Absicherung von Anforderungen modellieren können. (K2)
- LZ 4.2 Mit Sequenzdiagrammen Abläufe auf verschiedenen Abstraktions Ebenen transparent machen können. (K2)
- LZ 4.3 Mit Sequenzdiagrammen die Brücke zwischen Analyse, Design und Test schlagen können. (K2)
- LZ 4.4 Kommunikationsdiagramme als alternatives Ausdrucksmittel für Szenarien kennen. (K1)

Keine Voraussetzungen (dieses Thema wurde im Foundation Level nicht behandelt).

Requirements Methoden der 80er-Jahre bestanden auf Konsistenz, Vollständigkeit und Widerspruchsfreiheit. Ab den 90er-Jahren wurde die Kraft beispielhafter Abläufe aufgewertet. Nach dem Motto „Ein gutes Beispiel ist oft hilfreicher als eine schlechte Abstraktion“ erlaubten die Analysemodelle die Verwendung von Szenarien im Sinne beispielhafter Abläufe. Als Einstiegspunkt für die Ermittlung und Prüfung von Anforderungen fällt es Stakeholdern in den meisten Fällen wesentlich leichter, zunächst konkrete Beispiele zu betrachten und nicht direkt einen möglichst vollständigen Überblick über einen Sachverhalt zu geben oder diesen beurteilen zu müssen. Der Einsatz von Szenarien hat dabei allerdings immer zum Ziel, auf Grundlage der in Szenarien dokumentierten Beispiele letztlich zu einem möglichst guten Verständnis der wesentlichen Abläufe zu gelangen.

Mit den UML-Sequenzdiagrammen und verwandten Modellarten (wie z. B. Kommunikationsdiagramme, Interaktionsübersichtdiagramme) hat sich eine geeignete Notation etabliert, um beispielhafte Abläufe ausdrücken zu können. Während das Ziel bei der Funktionsmodellierung mit Use Cases und Aktivitätsdiagrammen in der Vollständigkeit der Beschreibung von Abläufen liegt, stehen bei der Szenario-modellierung durch Sequenzdiagramme die wesentlichen beispielhaften Interaktionsfolgen im Vordergrund. Somit lassen sich z. B. Schnittstellen zwischen der Umwelt und dem System besser modellieren und analysieren. Sequenzdiagramme beantworten die Frage: Wer tauscht mit wem welche Informationen in welcher Reihenfolge aus? Das Sequenzdiagramm zeigt die logische Abfolge für diese Kommunikation.

Sie lernen

- geeignete Beispiele für Abläufe auszuwählen, so dass ein tieferes Verständnis der Zusammenhänge gewonnen wird. (Normalfall, Sonderfälle, Stressszenarien, ...)
- Zusammenhänge auf verschiedenen Abstraktionsebenen auszudrücken (Kommunikation der Umwelt mit dem Gesamtsystem, Kommunikation wesentlicher fachlicher Systemteile)

LE 5 Umgang mit Modellen und deren Einbettung in den Gesamtprozess (K2)

Dauer: 2 Stunden (davon mindestens 1 Stunde Übung)

Lernziele:

- LZ 5.1 Die verschiedenen Modelle zur gegenseitigen Qualitätssicherung nutzen können (K2)
- LZ 5.2 Modelle geschickt strukturieren und managen können (K2)
- LZ 5.3 Modellbasiertes Requirements Engineering in den Gesamtentwicklungsprozess einbetten können (K2)
- LZ 5.4 Modelle als alternative Form von Anforderungen kennen. (K1)

Voraussetzungen: Qualitätskriterien für Anforderungsdokumente und Anforderungen kennen, Requirements Management (vgl. Foundation Level)

Die einzelnen Modellarten betonen jeweils einen Aspekt und nutzen dafür teilweise dieselben Modellelemente als Integrationspunkte, die für eine Qualitätssicherung genutzt werden können. Als Beispiel: Die Vollständigkeit der Integrationspunkte zwischen den Modellarten ermöglicht eine Qualitätssicherung bzgl. der Vollständigkeit der betrachteten Modelle.

Diese Überlappung von Modellarten (mit anderen Worten: die absichtlich in den Modellen enthaltene Redundanz) kann auch gezielt zur Konfliktidentifikation und zur Konfliktauflösung genutzt werden. Die Überlappung hilft demnach bei der Prüfung und Abstimmung der Anforderungen. Der wiederholte, häufige Wechsel der Sicht und zugehöriger Modellarten hilft dem Requirements Engineer, nicht perspektivenblind zu werden und somit nicht nur Fehler zu vermeiden, sondern auch den Analyseprozess zu beschleunigen.

Wenn Sie mit einer Art zu modellieren nicht mehr genügend neue Erkenntnisse über die Anforderungen erlangen, sollten Sie bewusst die Sicht wechseln, um den Analyseprozess zu beschleunigen oder die Kommunikation mit Ihren Stakeholdern effizienter zu gestalten.

Eine besondere Herausforderung bildet das Verwalten von Modellen, da in diesen Modellen sehr viele, zum Teil implizite Abhängigkeiten bestehen. Sie lernen in dieser Lehreinheit, welche Auswirkungen daraus für z. B. die Versionierung oder die Inkrementbildung folgen und wie Sie damit umgehen können. Weiterhin muss für das Management der Modelle betrachtet werden, wie Attribute einer Anforderung (Zustand, Risiko etc.) sinnvoll in der Modellierung eingesetzt werden können.

Im Projekt muss geklärt werden, welchen Stellenwert Modelle im Rahmen einer Spezifikation einnehmen sollen und können, bzw. wie deren rechtliche Relevanz definiert ist. In engem Zusammenhang damit steht die Verwendung der Anforderungsmodelle im Rahmen einer Anforderungsspezifikation. Hier müssen unter anderem die juristische Relevanz und die eventuell daraus folgenden, zusätzlichen Schritte betrachtet werden. Müssen die Modelle zum Beispiel in natürlichsprachliche Anforderungen übersetzt werden, und worin besteht dann der Mehrwert einer modellbasierten Anforderungsanalyse?

Glossar

Aktion: die kleinste Einheit einer ausführbaren Funktion im Modell oder im System (vgl. Aktivität)

Aktivität: eine (evtl. parametrisierte) größere Funktion, die in weitere Aktivitäten und schließlich in Aktionen (vgl. Aktion) zerlegt werden kann

Aktivitätsdiagramm: die UML-Notation für die grafische Darstellung von Abläufen

Assoziation: der UML-Begriff für einfache Beziehung (siehe diese) zwischen einer oder mehreren Fachklassen (siehe diese)

Beziehung: ein für das betrachtete System interessanter Zusammenhang von Klassen. Zu den Arten von Beziehungen zählen einfache Beziehungen (Assoziationen), Teile-Ganze-Beziehungen und die Generalisierungs-/Spezialisierungsbeziehung.

Datenmodell: umgangssprachliches Synonym für Informationsmodell (siehe dieses).

Datenflussdiagramm: Ein Datenflussdiagramm dokumentiert die Datenabhängigkeiten zwischen Funktionen des betrachteten Systems, in dem es dokumentiert, welche Ein- und Ausgabedaten spezifische Funktionen (bzw. Use-Cases) des Systems haben und welche Eingabedaten durch welche Funktionen des Systems in welche Ausgabedaten umgesetzt werden. Datenflussdiagramme sind u.a. Bestandteil der verschiedenen Ansätze zur Strukturierten Systemanalyse, wie z. B. SA von DeMarco oder Weinberg, SA/RT oder auch des Ansatzes der Essentiellen Systemanalyse nach McMenamin und Palmer.

Diagramm: Ein Diagramm einer Modellart ist die Repräsentation eines Teilbereichs des Modells der zu modellierenden Domäne, das durch eine Menge sachlogisch zusammenhängender Modellelemente dieser Modellart gebildet wird.

Domäne: Ein Fachgebiet, das Themenbereich bzw. Wissensgebiet einer Requirementsspezifikation ist; die Gesamtheit des Wissens innerhalb eines Fachbereichs.

Entität: Ist ein materielles oder immaterielles Objekt in der Domäne, der einem Modellbildungsprozess zugrunde liegt. Informationsmodelle, die die Struktur von Informationen dokumentieren, basieren auf der Klassifikation gleichartiger Entitäten zu Typen bzw. Klassen. Die Betrachtung von Entitäten geschieht auf der Instanzebene, d.h. der Modellebene M0.

Entitätsklasse: Eine Entitätsklasse abstrahiert über die Gemeinsamkeiten einer Menge von Entitäten der Instanzebene. Entitätsklassen bilden die zentralen Modellelemente der Modellebene M1 und werden z. B. in Entity-Relationship-Diagrammen modelliert.

Ereignis: Ein Ereignis kennzeichnet das Auftreten einer spezifischen Änderung von Eigenschaften des betrachteten Systems oder dessen Umgebung.

Fachbegriff: Ein Fachbegriff ist (meistens) ein Substantiv aus der Sprache des Fachbereichs oder des Anwendungsbereichs.

Fachdatentyp: Ein Fachdatentyp ist ein Fachbegriff, der im Informationsmodell als Datentyp eines Attributs einer Fachklasse verwendet wird.

Fachklasse: Eine Fachklasse ist ein Fachbegriff, der im Informationsmodell als Klasse modelliert ist.

Grafisches Modell: ein Modell, das als Notation Diagramme nutzt (im Gegensatz zu z. B. formelbasierten, mathematischen Modellen).

Generalisierung/Spezialisierung: eine spezielle Beziehung zwischen Modellelementen, bei denen das eine Modellelement eine Verallgemeinerung (Generalisierung) des anderen Modellelements (Spezialisierung) darstellt. Die Eigenschaften des allgemeineren Modellelements vererben sich auf das speziellere Modellelement.

Heuristik: aus dem Griechischen (Heureka: Ich habe es gefunden oder entdeckt): ein Tipp oder Trick von Modellierungsexperten, um Teile von Modellen zu finden, zu entdecken.

Informationsmodell: ein Modell einer Domäne, das deren Entitäten (bzw. Fachbegriffe) mit ihren Eigenschaften und Beziehungen abbildet.

Interaktion: bezeichnet das wechselseitige Aufeinandereinwirken von Modellelementen (z. B. Akteuren, Systemen und Systemteilen, Klassen, ...)

Klassendiagramm: eine grafische Darstellung von Informationen und Beziehungen zwischen diesen.

Kommunikationsdiagramm: ein UML-Ablaufdiagramm zur Darstellung von Szenarien, zur Modellierung eines beispielhaften Zusammenspiels von Objekten aus dem Informationsmodell.

Modell: Ein Modell ist ein abstrahierendes Abbild einer existierenden oder Vorbild für eine geplante Realität. Ein Modell einer Domäne in einer spezifischen Modellierungssprache setzt sich für gewöhnlich aus einer Menge von Diagrammen dieser Modellierungssprache zusammen.

Modellelement: Ein Modellelement einer Modellart ist ein Aspekt in der zugrunde liegenden Domäne, der für die gewählten Modellarten relevant ist, in dem Sinne, dass die jeweilige Modellart ein Notationselement besitzt, durch welches dieser

Aspekt in Diagrammen dieser Modellart repräsentiert werden kann.

Modellierungssprache: Eine Modellierungssprache, auch konzeptuelle Modellierungssprache, definiert die Syntax und Semantik für eine Klasse gleichartiger Modelle. Sie setzt sich aus einer Anzahl von Notationselementen zusammen.

Notationselement: Ein Notationselement einer Modellart ist ein Konzept mit spezifischer Semantik, welches in der Regel auch eine spezifische grafische Notation besitzt und zur Konstruktion von Diagrammen der jeweiligen Modellart verwendet werden kann.

Sequenzdiagramm: ein UML-Ablaufdiagramm zur Darstellung von Szenarien, bei denen zeitliche Reihenfolgen im Vordergrund stehen.

Spezialisierung: siehe Generalisierung.

Statechart: ein UML-Diagramm, in dem Zustände des Systems (oder eines Teilsystems) und die durch Ereignisse ausgelösten Übergänge im Vordergrund stehen.

Steuerungsklasse: eine Klasse im Informationsmodell, deren Hauptaufgabe das Steuern des Zusammenspiels anderer Klassen ist.

Szenario: Eine sequenzielle Folge von Interaktionen (oder Ereignissen) die beispielhaft das Erreichen eines bestimmten Ziels oder eines bestimmten Mehrwertes beschreibt.

Use-Case: ein Bündelung aller möglichen Szenarien, die eintreten können, wenn ein Akteur versucht, mit Hilfe des betrachteten Systems ein bestimmtes fachliches Ziel zu erreichen.

Zustand: Ein Zeitraum (zwischen zwei relevanten Ereignissen), in denen das System wartet oder bestimmte Aktivitäten ausführt.

Literatur

[Pohl und Rupp 2009] K. Pohl, C. Rupp: Basiswissen Requirements Engineering. dpunkt Verlag, Heidelberg, 2009

Das Kurzbuch zum Foundation Level, dessen Inhalt für das Advanced Level Requirements Modeling bekannt sein sollte.

[UML-Bücher] Die Unified Modeling Language ist derzeit die populärste Notation für Requirements Modelle. Es gibt zahlreiche Bücher zur Unified Modeling Language, die u.a. auch Modelle behandeln, die im Bereich des Requirements Engineering zum Einsatz kommen. Von einer expliziten Empfehlung wird hier abgesehen.