Software Engineering

Anforderungsanalyse zur Entwicklung eines SW-Systems zur Unterstützung der Einführung von Gleitarbeitszeit

vorgelegt von

Tom Graupner
Markus Klemm
Leonard Hecker

Inhaltsverzeichnis

1 Einführung

Das Unternehmen EKS[1] evaluiert aktuell die Umstellung ihres Arbeitszeitmodells zur Gleitzeit. Die Erfassung und Auswertung der Arbeitszeit soll dabei durch ein Software-System unterstützt werden. Die vorliegende Anforderungsanalyse beschäftigt sich zunächst mit den Rahmenbedingung und den Funktionen, die vom System übernommen werden sollen. Neben der Zusammenfassung aller funktionalen Anforderungen und der Struktur der Eigangs- und Ausgangsdaten, enthält diese Analyse verschiedene Anwendungsfalldiagramme[2], sowie ein Entity Relationship Model, welches die Speicherung der Daten veranschaulicht.

1 Abkürzung für ENTWICKLUNG VON KUNDENSPEZIFISCHER SOFTWARE

2 Als Abkürzung wird im folgenden AWD verwendet. Daran angelehnt ist die Abkürzung AWF für einen Anwendungsfall

2 Dokumentation der Anforderungen

Anforderungen an ein Software-Produkt werden im Allgemeinen zunächst in funktionale und nicht-funktionale Anforderungen unterteilt. Erstere decken dabei die Fähigkeiten und die Beschaffenheiten ab, die der Benutzer der Software zur Problemlösung oder zur Erreichung seines Zieles benötigt. Nichtfunktionale Anforderungen unterteilen sich weiterhin in Rahmenbedingungen und Qualitätsanforderungen.

2.1 Funktionale Anforderungen

Die folgende Auflistung enthält die groben Funktionen, die vom Software-System erfüllt werden sollen. Bei einigen handelt es sich dabei um *abstrakte Funktionen*, welche sich im weiteren Verlauf der Analyse feiner aufgliedern werden.

- **Anwesenheit erfassen** ≪ *abstrakt* ≫
- **Urlaub planen - Mitarbeiter** ≪ *abstrakt* ≫
- **Urlaub verwalten - Abteilungsleiter** ≪ *abstrakt* ≫
- **Krankheitsdaten erfassen**
- **Anwesenheit auswerten**
- **Zeitauswertung für Abteilungsleiter** ≪ *abstrakt* ≫

2.1.1 Tabellarischer überblick

Die folgenden Tabellen fassen nun alle voneinander unabhängigen funktionalen Anforderungen an das Software-System zusammen. Im Rahmen der Anforderungsanalyse verwendet man für unabhängige funktionale Anforderungen ebenfalls den Begriff *essentielle Funktionen*.

Funktion	**Eingangsdaten**	**Ausgangsdaten**	**Bemerkungen**	**abstrakter AWD**
Betreten	MA-ID und Uhrzeit	Zutritt und Speicherung der Zeit, alternativ Zutrittsverweigerung	Bei einer ungültigen MA-ID kann der Zutritt verweigert werden	**Anwesenheit erfassen**
Verlassen	MA-ID und Uhrzeit	Verlassen und Speicherung der Zeit, alternative Fehlermeldung		
Wachdienst informieren	Mitarbeiterliste	Detaillierte Information an den Wachdienst	Der Wachdienst wird stündlich darüber informiert, welche Mitarbeiter sich im Gebäude befinden	

Urlaub beantragen	Urlaubswunsch	Urlaubsantrag	Urlaub wird unter Verwendung der eigenen MA-ID beim jeweiligen Abteilungsleiter beantragt	**Urlaub planen, Mitarbeiter**
Urlaubsinformationen anzeigen	Wunsch nach Urlaubsinformationen	Informationen zu Urlaubsterminen, Beantragungsstatus, verbrauchten und verbleibenden Urlaubstagen		
Urlaubsantrag stornieren	Storno-Wunsche	Storno-Bestätigung mit Aktualisierung der Urlaubsdaten	Mitarbeiter kann offene, abgelehnte und genehmigte (noch nicht angetretene) Urlaubsanträge stornieren	
Urlaubsvorschlag annehmen	Urlaubsvorschlag des Abteilungsleiter	Aktualisierung der Urlaubsinformationen	Abteilungsleiter können Mitarbeiter ihrer Abt. Vorschläge unterbreiten	
Urlaubsvorschlag stornieren	Urlaubsvorschlag des Abteilungsleiter und Storno-Wunsch	Stornierungsmitteilung und Aktualisierung der Urlaubsinformationen	Abteilungsleiter können Mitarbeitern ihrer Abt. Vorschläge unterbreiten	

Funktion	**Eingangsdaten**	**Ausgangsdaten**	**Bemerkungen**	**abstrakter AWD**
Urlaubsantrag genehmigen	Urlaubsantrag eines MA	Aktualisierung der Urlaubsdaten und Bestätigung	Abteilungsleiter müssen Anträge ihrer Mitarbeiter genehmigen	**Urlaub verwalten, Abt.-Leiter**
Urlaubsantrag ablehnen	Urlaubsantrag eines MA	Aktualisierung der Urlaubsdaten und Absage	Abteilungsleiter können Anträge ihrer Mitarbeiter ablehnen	
Vorschlag unterbreiten	Urlaubsvorschlag des Abteilungsleiters	Urlaubsvorschlag an Mitarbeiter und Aktualisierung der Urlaubsinformationen	Abteilungsleiter können Mitarbeitern Urlaubsvorschläge unterbreiten	
Urlaubsinformationen eines Mitarbeiters anzeigen	Wunsch des Abteilungsleiters nach Urlaubsinformationen eines Mitarbeiters	Detaillierte Informationen zur Abteilung	Abteilungsleiter können sich zur Entscheidungsunterstützung die Urlaubsinformationen eines Mitarbeiters anzeigen lassen	

Krankmeldung erfassen	Krankenschein eines Mitarbeiters	Aktualisierung der Urlaubsinformationen	Sachbearbeiter (HR) erfasst Krankmeldungen von Mitarbeitern und betroffene Urlaubsinformationen werden sofort aktualisiert	

Anwesenheit auswerten	Anwesenheitsinformationen eines Mitarbeiters	Detaillierte Arbeitszeitauswertung des Mitarbeiters	Die Auswertung wird wöchentlich automatisch erstellt und dem Mitarbeiter per Email zugesandt	

Funktion	Eingangsdaten	Ausgangsdaten	Bemerkungen	abstrakter AWD
Gesamtbilanz anfordern	Wunsch nach Gesamtbilanz	Gesamtbilanz enthält detaillierte Informationen zur Arbeitszeitauswertung der Abteilung	Die Kennzahlen sind absolut und prozentual angegeben und betreffen einen beliebigen, abgelaufenen Zeitraum	**Zeitauswertung für Abt.-Leiter**
Urlaubszeitbilanz anfordern	Wunsch nach Urlaubsbilanz	Urlaubszeitbilanz enteält beantragte Urlaubstage der Abteilung in einem vorausschauenden Zeitraum	Anträge werden absolut und prozentual bezogen auf die Gesamtarbeitszeit dargestellt	
Anwesenheitsliste anfordern	Wunsch nach Anwesenheitsliste	Liste enthält alle momentan anwesenden Mitarbeiter der eigenen Abteilung		

2.1.2 Struktur der Eingangs- und Ausgangedaten

Jede essentielle Funktion besitzt definierte Eingangs- und Ausgangsdaten. Die folgende Auflistung stellt die Struktur der entsprechenden Daten aller zuvor genannten essentiellen Funktionen dar.

Funktion	Struktur der Eingangsdaten	Struktur der Ausgangsdaten
Betreten	+ MA-ID + Uhrzeit	Bestätigung des Zutritts und Datensatz :={MA-ID, Uhrzeit}, alternativ Fehlermeldung
Verlassen	+ MA-ID + Uhrzeit	Bestätigung des Verlassen und Datensatz :={MA-ID, Uhrzeit}, alternativ Fehlermeldung
Wachdienst informieren	Stündlicher Trigger zum Auslösen der Benachrichtigung	Detaillierte Mitarbeiterliste mit den Spalten {MA-ID, Nachname, Vorname, Büro}

Funktion	Struktur der Eingangsdaten	Struktur der Ausgangsdaten
Urlaub beantragen	Urlaubswunsch := {MA-ID, Liste: zu beantragende Urlaubstage}	Urlaubsantrag := {MA-ID, Nachname, Vorname, Liste: zu beantragende Urlaubstage}
Urlaubsinformationen anzeigen	Wunsch nach Urlaubsinformationen	Urlaubsinformationen := {verbrauchte Urlaubstage, verbleibende Urlaubstage, Liste: Urlaubstermine inkl. Status (offen, genehmigt, abgelehnt}
Urlaubsantrag stornieren	Urlaubsantrag s.o. (Status: offen / genehmigt und noch nicht angetreten).	Bestätigung der Stornierung und aktualisierte Urlaubsinformationen := {verbrauchte Urlaubstage, verbleibende Urlaubstage, Liste: Urlaubstermine inkl. Status (offen, genehmigt, abgelehnt}
Urlaubsvorschlag annehmen	Urlaubsvorschlag := {MA-ID, Liste: vorgeschlagener Urlaubstage }	Bestätigung des Urlaubsvorschlages und aktualisierte Urlaubsinformationen := {verbrauchte Urlaubstage, verbleibende Urlaubstage, Liste: Urlaubstermine inkl. Status (offen, genehmigt, abgelehnt}
Urlaubsvorschlag stornieren	Urlaubsvorschlag s.o.	Bestätigung des Urlaubsvorschlages und aktualisierte Urlaubsinformationen := {verbrauchte Urlaubstage, verbleibende Urlaubstage, Liste: Urlaubstermine inkl. Status (offen, genehmigt, abgelehnt}
Urlaubsantrag genehmigen	Urlaubsantrag s.o.	Bestätigung der Genehmigung und aktualisierte Urlaubsinformationen := {verbrauchte Urlaubstage, verbleibende Urlaubstage, Liste: Urlaubstermine inkl. Status (offen, genehmigt, abgelehnt}
Urlaubsantrag ablehnen	Urlaubsantrag s.o.	Bestätigung der Ablehnung und aktualisierte Urlaubsinformationen := {verbrauchte Urlaubstage, verbleibende Urlaubstage, Liste: Urlaubstermine inkl. Status (offen, genehmigt, abgelehnt}
Urlaubsvorschlag unterbreiten	Wunsch einen Vorschlag zu unterbreiten oder konkreter Urlaubsantrag s.o.	Urlaubsvorschlag := {MA-ID, Nachname, Vorname, Liste: vorgeschlagener Urlaubstage} und aktualisierte Urlaubsinformationen := {verbrauchte Urlaubstage, verbleibende Urlaubstage, Liste: Urlaubstermine inkl. Status (offen, genehmigt, abgelehnt}
Urlaubsinformationen eines Mitarbeiters anzeigen	Wunsch nach Urlaubsinformationen	Urlaubsinformationen := {verbrauchte Urlaubstage, verbleibende Urlaubstage, Liste: Urlaubstermine inkl. Status (offen, genehmigt, abgelehnt}

Funktion	Struktur der Eingangsdaten	Struktur der Ausgangsdaten
Krankmeldung erfassen	Krankenschein eines Mitarbeiters := {Nachname, Vorname, Geburtsdatum, Zeitraum der Bescheinigung}	Entsprechende Aktualisierung der Mitarbeiterdatensätze für Urlaubstage, Soll- und Ist-Arbeitszeit
Anwesenheit auswerten	Arbeitsstunden, Urlaubstage und Krankmeldungen des Mitarbeiters	Detaillierte Arbeitszeitauswertung :={Soll-Arbeitszeit, Ist-Arbeitszeit, Stand des Arbeitszeitkontos }
Gesamtbilanz anfordern	Wunsch nach Gesamtbilanz	Gesamtbilanz zu Abteilung :={Sollarbeitszeit, tatsächliche Arbeitsstunden, Urlaubstage, Krankheitstage, überstunden}
Urlaubsbilanz anfordern	Wunsch nach Urlaubsbilanz	Urlaubsbilanz der Abteilung :={beantragte Urlaubstage (absolut und prozentual bezogen auf Gesamtarbeitszeit)}
Anwesenheitsliste anfordern	Wunsch nach Anwesenheitsliste	Anwesenheitsliste der Abteilung :={MA-ID, Nachname, Vorname, Arbeitsplatz}

2.2 Qualitätsanforderungen

Nachdem weder interne, noch externe Qualitätsanforderungen explizit in den vorliegenden Rahmenbedingungen genannt sind, lautet die Aufgabe hier globale Anforderungen zu formulieren und eigene Gedanken zu entwickeln.

Ein allgemeiner Punkt herausragender Bedeutung ist beispielsweise *Datensicherheit und Integrität*. Aufgrund der Sensibilität der zu verarbeitenden Daten und der mit ihnen verbundenen Business-Prozesse (e.g. Buchhaltung) ist unbedingt dafür zu sorgen, dass jegliche Daten *zugriffssicher, redundant* und unter *definierten Integritätsbestimmungen* gespeichert und verarbeitet werden.

Für die spätere Erweiterung oder Wartung der Software ist es außerdem von großer Bedeutung, alle Funktionen und Komponenten des Systems lückenlos zu dokumentieren.

Geht man etwas ins Detail und betrachtet die essentiellen Funktionen, so gibt es Punkte an denen die Benutzerfreundlichkeit deutlich verbessert werden kann. Empfohlen wären unter anderem *Interaktionen mit der Software zu bestätigen*. Gemeint ist damit, dem Benutzer Rückmeldung zu erfolgreich oder nicht erfolgreich abgeschlossenen Interaktionen zu geben.

Weitere vorstellbare Qualitätsanforderungen werden nach Bedarf mit dem Auftraggeber abgesprochen.

2.3 Rahmenbedingungen

Als abschließender Punkt der schriftlichen Formulierung der Anforderungen werden die Rahmenbedingungen festgehalten. Hierbei unterscheidet man zwischen *technologischen*, *rechtlichen* und *organisatorischen Rahmenbedingungen*.

Zu den *technischen Rahmenbedingungen* gehört dabei, dass das System Zugriff auf den betriebsinternen Jahreskalender benötigt. Dies ist notwendig, um Feiertage und Betriebsruhetage automatisch in die Bilanz der Arbeitszeitkonten einbeziehen zu können. Weiterhin sollen Urlaubstage und Krankmeldungen unmittelbar in die Bilanz einfließen.

Wichtigster Teil der *rechtlichen Rahmenbedingungen* ist zweifelsohne das Thema Datensicherheit. Die Vollständigkeit und Integrität der personenbezogenen Daten muss zu jedem Zeitpunkt gewährleistet sein. Dies ist notwendig um Rechtssicherheit zu schaffen, für den Arbeitgeber und den Arbeitnehmer.

Die *organisatorischen Rahmenbedingungen* beinhalten vor allem Details zu den Arbeitszeitmodellen im Unternehmen. So besitzt ein Standard-Arbeitstag 8 Stunden und eine Arbeitswoche dementsprechenden 40 Stunden. Das Arbeitszeitkonto eines jeden Mitarbeiters wird dabei vom Beginn des Arbeitsverhältnisses an kumulativ geführt.

3 Kontextdiagramm

Nach der ausführlichen Formulierung der Anforderungen folgt nun die Modellierung des SW-Systems. In einer ersten Abstraktion zeigt Abbildung 3.1 das entsprechenden *Kontextdiagramm*. Es zeigt des System und dessen Schnittstellen zur Umwelt, sowie die Beziehungen zwischen den Benutzern.

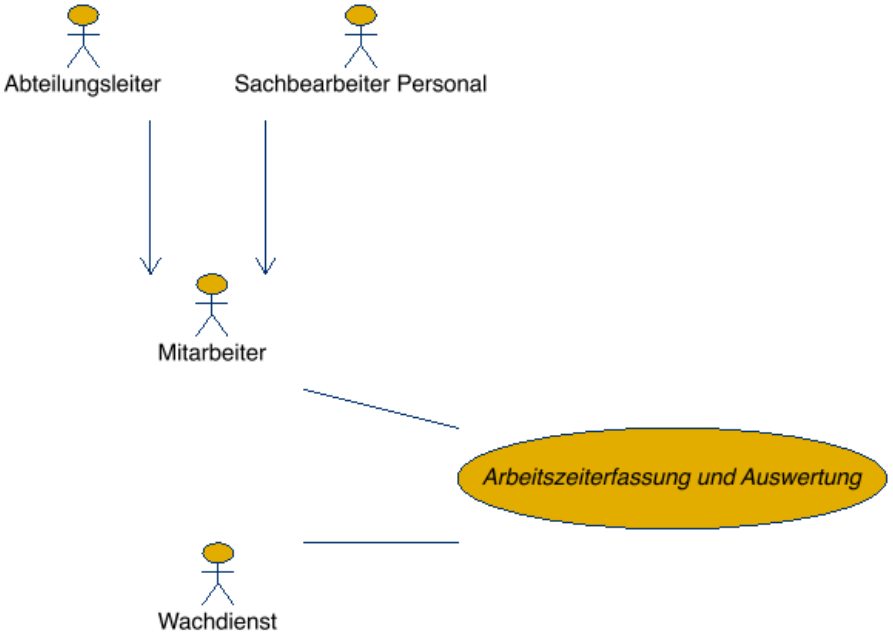

Abbildung 3.1: Kontextdiagramm zum SW-System *Arbeitszeiterfassung und Auswertung*. Der Akteur Mitarbeiter generalisiert die Akteure Abteilungsleiter und Sachbearbeiter Personal

4 Anwendungsfalldiagramme

In einer weiteren Abstraktion werden die einzelnen Funktionen und ihre Kommunikationsbeziehungen zu den verschiedenen Akteuren dargestellt.

4.1 AWD der groben Funktionalität

Abbildung 4.1 zeigt die oberste Abstraktionsebene der Anwendungsfalldiagramme. Die enthaltenen *abstrakten Funktionen* Kapseln dabei mehrere verwandte Anwendungsfälle.

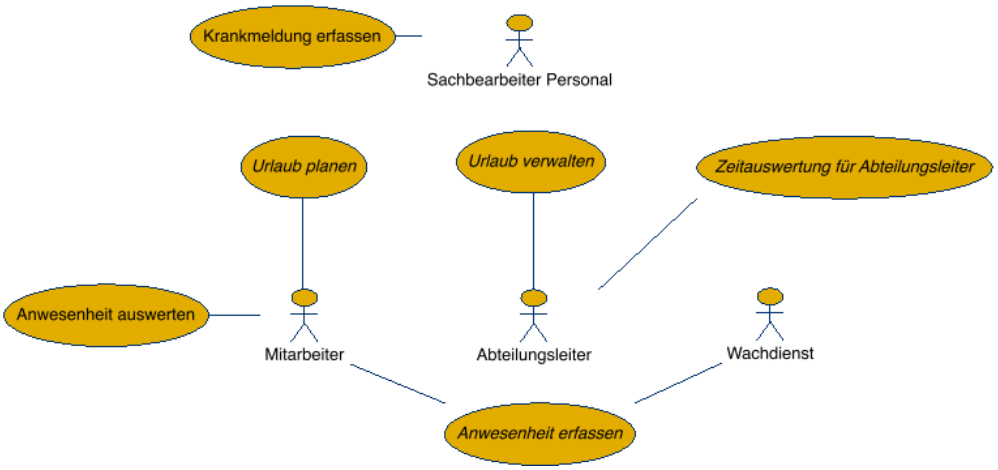

Abbildung 4.1: Anwendungsfalldiagramm zur groben übersicht.

4.2 AWD der Funktionalität *Urlaub planen*

Beispielhaft wird nun die abstrakte Funktion *Urlaub planen* näher betrachtet. Abbildung 4.2 zeigt das entsprechende AWD. Ein Mitarbeiter besitzt die Möglichkeiten Urlaub zu beantragen oder zu stornieren. Er kann weiterhin Urlaubsvorschläge seines Abteilungsleiters annehmen oder ebenfalls stornieren und seine persönlichen Urlaubsinformationen anzeigen.

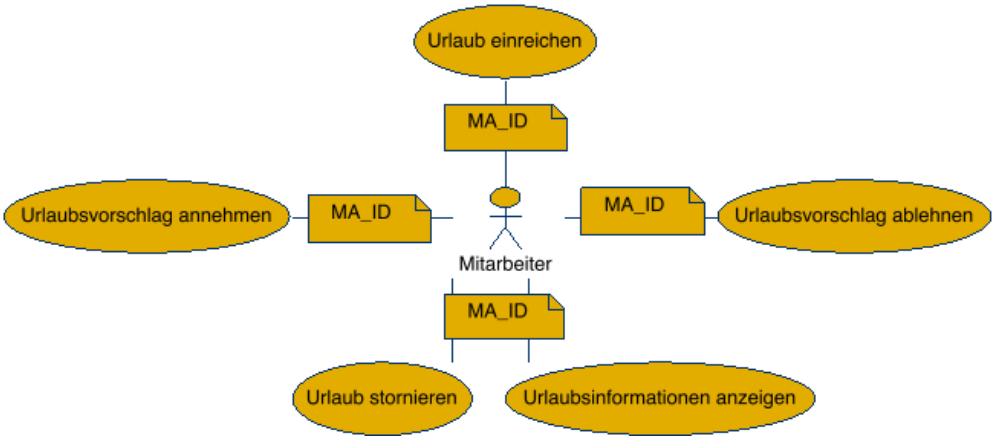

Abbildung 4.2: Anwendungsfalldiagramm zur abstrakten Funktion *Urlaub planen - Mitarbeiter*

4.3 Detaillierte Beschreibung der essentiellen Funktionalität *Urlaub beantragen*

Die essentielle Funktion *Urlaub beantragen* gehört zu den kleinsten von anderen unabhängigen Anwendungsfällen. Beschreiben kann man sie wie folgt:

Der Mitarbeiter verfasst seinen Urlaubsantrag, basierend auf seinen aktuellen Urlaubsinformationen. Der Antrag enthält die Daten MA-ID, Nachname, Vorname und eine Liste der zu beantragenden Urlaubstage.

Nach CHRIS RUPP beschreibt man den Anwendungsfall alternativ unter Zuhilfenahme von Schatzschablonen folgendermaßen:

SW-System *Arbeitszeit erfassen und auswerten* muss dem Mitarbeiter die Möglichkeit bieten Urlaub einzureichen.

Eine weitere Detaillierte Form der Modellierung bietet das Aktivitätsdiagramm. Es stellt die einzelnen Aktionen dar, die in einer Funktionalität gekapselt sind.
Im Fall der essentiellen Funktionalität *Urlaub beantragen* ist das Aktivitätsdiagramm trivial, dargestellt in Abbildung 4.3

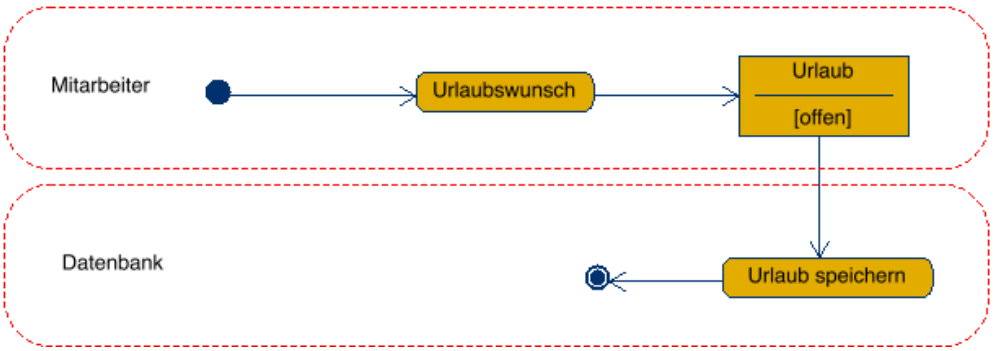

Abbildung 4.3: Aktivitätsdiagramm der essentiellen Funktion *Urlaub beantragen*.

5 Zustandsdiagramm eines Urlaubsantrages

Der Anwendungsfall *Urlaub planen* eines Mitarbeiters soll nun erneut detaillierter betrachtet werden. Dazu greifen wir das Objekt *Urlaubsantrag* heraus und beschreiben es in einem Zustandsdiagramm genauer. Dieses Diagramm enthält die verschiedenen Zustände einer Betrachtungseinheit und beschreibt gleichzeitig die übergänge zwischen den Zuständen.

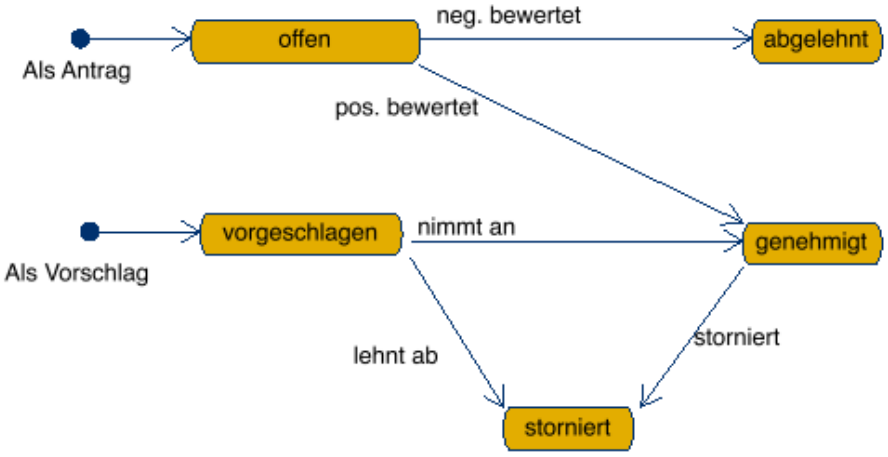

Abbildung 5.1: Zustandsdiagramm des Objekts *Urlaubsantrag*.

6 Entity Relationship Model

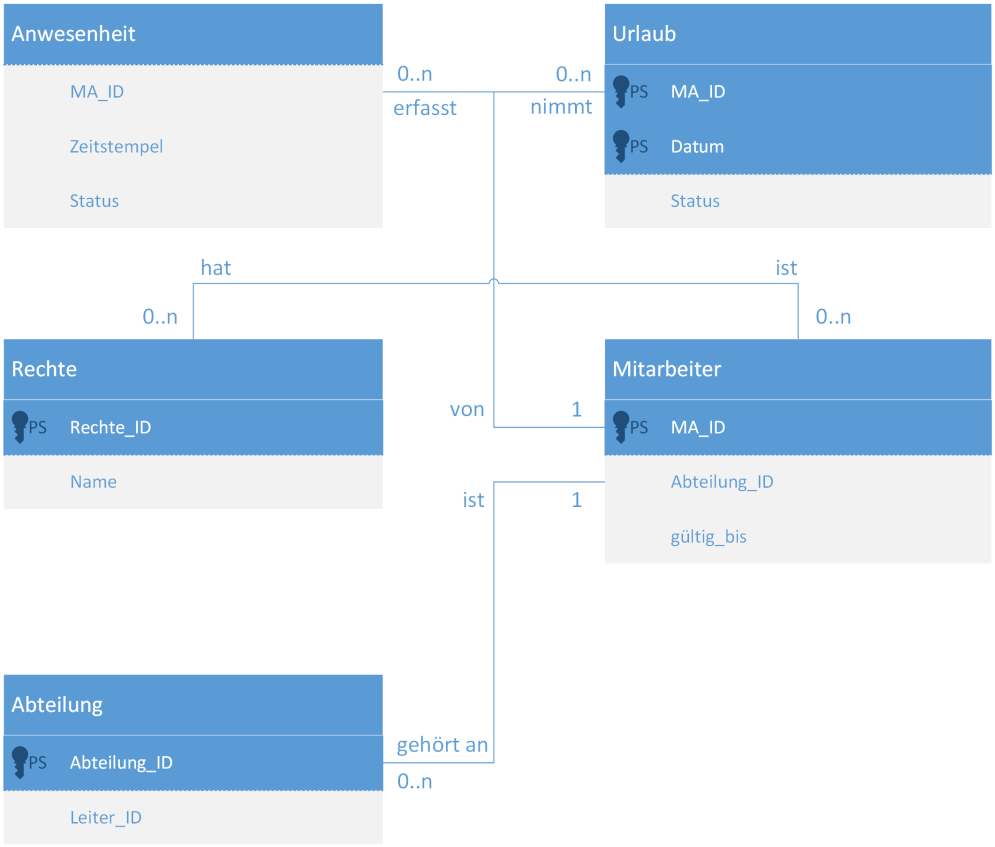

Abbildung 6.1: Entity Relationship Model aller essentiellen Funktionen

6.1 Detaillierte Beschreibung des Entity Relationship Model

Die Tabelle Mitarbeiter enthält alle *Mitarbeiter*, indiziert unter dem Primärschlüssel *MA_ID*. Außerdem gibt es ein Attribut *gültig_bis* um die Gültigkeit der Ausweise zu speichern. Die Tabelle könnte weitere Informationen enthalten, wie z.B. „Nachname“, „Vorname“, oder „Geburtsdatum “.
Jeder Mitarbeiter ist einer beliebigen Anzahl an Abteilungen zugehörig, weshalb eine Tabelle *Abteilung* existiert. Die Abteilungen der Firma sind unter dem Primärschlüssel *Abteilung_ID* indiziert. Jede Abteilung besitzt einen Abteilungsleiter, der durch den Fremdschlüssel *Leiter_ID*, welcher sich auf eine *MA_ID* bezieht, beschrieben wird.
Des Weiteren hat jeder Mitarbeiter eine beliebige Anzahl an Rechten, weshalb eine Tabelle *Rechte* existiert. In dieser sind diese unter dem Primärschlüssel *Rechte_ID* indiziert. Jedem Eintrag wird ein *Name* zugeordnet, um in der Software auf den jeweiligen Eintrag über den Namen zuzugreifen. Dabei würde die Software z.B. die Verknüpfung von Rechte_ ID auf Name umkehren und lokal zwischenspeichern, weshalb Rechte_ ID als Primärschlüssel dennoch sinnvoll ist.
Die Anwesenheitserfassung wird durch die Tabelle *Erfassung* ermöglicht. Unter der *MA_ID* und einem *Zeitstempel* wird der *Status* geloggt. *MA_ID* bezieht sich hierbei als Fremdschlüssel auf einen Mitarbeiter und *Status* enthält die Art des Eintrages, sprich „betreten “oder „verlassen “. Diese Tabelle enthält keinen Primärschlüssel, da MA_ ID zusammen mit Zeitstempel und notfalls mit Status zwar praktisch einen Primärschlüssel bilden würden, jedoch nicht theoretisch. Da ein Primärschlüssel außerdem keinen Nutzen in dieser Tabelle hätte und um auch die theoretische Korrektheit zu gewähren, existiert deshalb keiner. Stattdessen sollte sich ein regulärer Index auf MA_ ID und auf Zeitstempel für die Auswertung als nützlich erweisen.
Zuletzt existiert die Tabelle *Urlaub* um Urlaubsanträge zu erfassen. Diese werden hierbei unter dem zusammengesetzten Primärschlüssel aus *MA_ID*, einem Fremdschlüssel, welcher sich auf einen Mitarbeiter bezieht, und *Datum* des Urlaubstages. Möchte der Mitarbeiter mehrere Tage Urlaub nehmen so

müssen mehrere Einträge in der Tabelle erstellt werden. Die Alternative ist es ein „Startdatum “ und ein „Enddatum “ des Urlaubs zu speichern. Das hier vorgeschlagene Konzept ist jedoch der Alternative überlegen, da es zwar mehr Einträge benötigt, dies jedoch die Implementierung stark vereinfacht und etwaige Fehler verhindert. Außerdem existiert ein Attribut *Status* welcher den Status des Antrages enthält, sprich: „offen“, „abgelehnt “, oder „genehmigt “.

7 Glossar

Der abschließende Glossar soll es Personen aus verschiedenen Fachgebieten möglich machen, die vorliegende Anforderungsanalyse zu verstehen und mit ihr arbeiten zu können. Dazu werden wichtige Fachbegriffe aus dem Kontext des Software Engineering und der Anforderungsanalyse beschrieben.

Anwendungsfall Eine abstrakte Darstellung einer vom Software-System angebotenen Funktionalität (Aktivität). Er kapselt eine Menge von Aktionen, die sequentiell, bediengungsabhängig oder zyklisch abgearbeitet werden. Ein Anwendungsfall wird in Folge von Dateneingaben oder zeitlichen Ereignissen ausgelöst und führt in der Regel zu einem von außen sichtbarem Ergebnis.

Anwendungsfalldiagramm Das Anwendungsfalldiagramm, kurz AWD, stellt die funktionalen Anforderungen (Aktivitäten) aus Sicht des Anwenders dar. Diese Aktivitäten werden zu den Beteiligten aus dem Kontext (Akteuren) in Beziehung gesetzt.

Akteur Ein Akteur ist die abstrakte Darstellung einer externen Instanz, die mit dem System kommuniziert.

- abstrakte Funktion
- essentielle Funktion
- Entity Relationship Model
- Unified Modeling Language

- Mitarbeiter-ID
- t.b.c