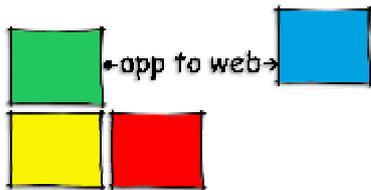




Fraunhofer Institut
Experimentelles
Software Engineering

Öffnung der Anwendung ORBIS-Dienstplan für Internet-Dienste durch Domänenanalyse



Autoren:
Isabel John,
Dirk Muthig,
Peter Sody,
Enno Tolzmann

IESE-Report Nr. 092.01/D
Version 1.0
Dezember 2001

Eine Publikation des Fraunhofer IESE

Das Fraunhofer IESE ist ein Institut der Fraunhofer-Gesellschaft.

Das Institut überträgt innovative Software-Entwicklungstechniken, -Methoden und -Werkzeuge in die industrielle Praxis. Es hilft Unternehmen, bedarfsgerechte Software-Kompetenzen aufzubauen und eine wettbewerbsfähige Marktposition zu erlangen.

Das Fraunhofer IESE steht unter der Leitung von
Prof. Dr. Dieter Rombach
Sauerwiesen 6
67661 Kaiserslautern

Abstract

Dieser Report dokumentiert die Projektergebnisse einer Domänenanalyse bei der Firma SIEDA, die vom Fraunhofer Institut Experimentelles Software Engineering unterstützt wurde. Im Rahmen des Projekts Application2Web wurde gemeinsam vom Fraunhofer IESE und der Sieda GmbH eine Domäne identifiziert, die hohes Verbesserungspotential bei der Realisierung von Web-Diensten im Rahmen des Produkts Orbis-Dienstplan hat. Diese Domäne wurde analysiert, Gemeinsamkeiten und Unterschiede in möglichen Anwendungen identifiziert und die Migrationsmöglichkeiten von der alten zur neuen Realisierung untersucht.

Schlagworte: **Produktlinien, Domänenanalyse, Redesign, Migration, E-Business**

Inhaltsverzeichnis

1	Einleitung	1
1.1	Kontext	1
1.1.1	Das Verbundprojekt Application2Web	1
1.1.2	SIEDA	2
1.1.3	Fraunhofer Institut Experimentelles Software Engineering	3
1.2	Ziele	4
1.3	Domänenanalyse	4
1.4	Domänenidentifikation	5
1.5	Dokumentstruktur	6
2	Anwendungskontext ORBIS-Dienstplan	8
2.1	Kontext	8
2.2	Anwendungsfälle	9
2.2.1	Berechnung des Samstagszuschlags im Regeldienst	9
2.2.2	Berechnung von Überstundenzuschlägen	10
2.2.3	Bericht „Dienstleistungsübersicht“	11
2.2.4	Bericht „Kontenübersicht“	11
2.2.5	Bericht „Wechselschichtzulage“	12
2.3	Komponentenstruktur	12
2.3.1	Realisierung im aktuellen System	13
2.4	Realisierung im Web	14
3	Berichtskomponente	16
3.1	Kontext	16
3.2	Funktionale Anforderungen	16
3.3	Berichte	17
3.3.1	Bericht „Dienstleistungsübersicht“	17
3.3.2	Bericht „Berechnungsregeln“	18
3.3.3	Bericht „Wechselschichtzulage“	20
3.3.4	Bericht „Umbuchungen“	21
3.3.5	Bericht „Konto“	23
3.4	Realisierung	24
4	Dienst- und Mitarbeitermanager	25
5	Abrechnungsmodul	27
	Literatur	29

Anhang	30
A. Glossar	30
B. Notation	34
A.1 Darstellung der Anforderungen	34
A.2 Datenformat	34

1 Einleitung

Dieser Report dokumentiert die Projektergebnisse einer Domänenanalyse bei der Firma SIEDA, die vom Fraunhofer Institut Experimentelles Software Engineering unterstützt wurde. In dieser Einleitung werden zunächst der Kontext in dem die Domänenanalyse durchgeführt wurde, die Ziele der beteiligten Partner sowie allgemeine Informationen zur Durchführung vorgestellt.

1.1 Kontext

Die mit diesem Report dokumentierte Domänenanalyse wurde im Kontext des Forschungsprojektes APPLICATION2WEB¹ durchgeführt, welches im Abschnitt 1.1.1 allgemein beschrieben wird. An der Domänenanalyse beteiligt waren die Firma SIEDA (siehe Abschnitt 1.1.2) sowie als Forschungspartner das Fraunhofer Institut Experimentelles Software Engineering (siehe Abschnitt 1.1.3). Beide Partner sind in Kaiserslautern ansässig.

1.1.1 Das Verbundprojekt Application2Web

Der Markt für Web-Hosting und Softwareportale hat sich zwischen 1997 und 1998 auf 951 Millionen Dollar weltweit verdoppelt und wird nach einer Forrester-Studie bis zum Jahr 2003 weiter bis auf 14,6 Milliarden Dollar anwachsen. Bereits 30 Unternehmen mieten inzwischen Oracle-Software über das Internet und auch andere Großunternehmen wie Microsoft und Sun planen, ihre Office-Pakete über das Internet zu vermieten. Damit deutsche Unternehmen den Anschluss in diesem wichtigen Wachstumsmarkt nicht verlieren, müssen Investitionen in Forschung und Entwicklung getätigt werden.

Diese Investitionen bergen jedoch ein Risiko: Zwar gibt es für einzelne Bereiche wie den elektronischen Handel oder Online-Banking schon fertige Lösungen, doch lassen sich diese Lösungen nicht auf andere Anwendungsgebiete übertragen und an die Bedürfnisse kleinerer Unternehmen anpassen. Es geht vielmehr darum, schon bestehende Software so zu modifizieren, dass sie im Internet verfügbar gemacht werden kann. Bestehende Lösungen leisten dies nicht oder nur unzureichend.

Eine Investition in die richtigen Technologien zum Zugang zu diesen Markt verschafft einen erheblichen Wettbewerbsvorteil: Die Nutzung des Internet wird

¹ Projekt Application2Web, BmWi Fördernr 16IN0048, Laufzeit: 10/00 – 12/02

ständig kostengünstiger bei steigenden Bandbreiten. Kunden von Anwendungen und Dienstleistungen werden also in Zukunft häufiger in Erwägung ziehen, Anwendungen zu mieten anstatt sie zu kaufen (beispielsweise um die Total-Costs-of-Ownership zu reduzieren). Das Web macht die Anwendungen zudem weitgehend unabhängig von Hardware-Plattformen, Betriebssystemen sowie Nutzungszeit und -ort. Dadurch erhöht sich die Flexibilität und Interoperabilität der Anwendungen wodurch neue Anwendungsgebiete erschlossen werden.

Das Ziel des BmWi geförderten Projektes APPLICATION2WEB¹ (kurz: app2web) ist es daher, insbesondere kleinen und mittleren deutschen Unternehmen zu ermöglichen, ihre Anwendungen und Dienstleistungen effektiv und kostengünstig über das Web verfügbar zu machen. Um dies zu erreichen setzt sich APPLICATION2WEB als technische Ziele die Integration der Komponententechnologie mit Software-Reengineering-Verfahren und die (Weiter-) Entwicklung von entsprechenden Prozessen, Softwarearchitekturen, Methoden und Werkzeugen.

Projektergebnisse von Application2Web sind neue Prozesse, Methoden und Werkzeuge zur komponenten-basierten Konstruktion von browserbasierten Anwendungen. Durch die Entwicklung von Prototypen nach bereits einem Jahr wird das hohe Umsetzungspotential der Technologien nachgewiesen. Das Konsortium mit den Forschungseinrichtungen Fraunhofer IESE und FZI Forschungszentrum Informatik und den neun involvierten Unternehmen (SIEDA, MARKET MAKER, Viva Tech, CAS, PTV, ABB, Debis Systemhaus, Bauer&Partner, Sage KHK) ist bestens auf diese Aufgaben vorbereitet.

Im Rahmen des Projekts Application2Web wurde bei der SIEDA GmbH, Kaiserslautern in Zusammenarbeit mit dem Fraunhofer IESE eine Domänenanalyse mit anschließendem Grobdesign einer webfähigen Komponente durchgeführt. Der vorliegende Report enthält die Ergebnisse dieser Domänenanalyse und beschreibt das Design der ausgewählten Komponenten.

1.1.2 SIEDA

Die Firma SIEDA - Systemhaus für intelligente EDV-Anwendungen GmbH wurde 1993 gegründet und beschäftigt zur Zeit vierzehn Mitarbeiter. Die SIEDA GmbH ist ein Spezialist für die Entwicklung von "intelligenter" Software zur Lösung komplexer Optimierungsprobleme schwerpunktmässig bei der Personaleinsatzplanung.

ORBIS Dienstplan ist eine Personaleinsatzplanung mit Einsatzschwerpunkt im Öffentlichen Dienst. Im Gegensatz zu den meisten am Markt verfügbaren Produkten ist sie auf die Planung von Mehrschichtmodellen (z.B. mit Diensten rund um die Uhr) spezialisiert. Hierbei unterstützt die Software nicht nur die üblichen rollierenden Arbeitszeitsysteme, sondern ermöglicht eine vollständige

Automatisierung der Dienstplanung, indem eine intelligente Planungskomponente für diese Aufgabe eingesetzt wird. Neben der eigentlichen Dienstplanung werden auch die geleisteten Dienste über das System erfasst und anschließend die Daten für die Lohn- und Gehaltsabrechnung unter Berücksichtigung der tariflichen Bestimmungen (u. and. BAT, BMT-G, Beamte) ermittelt.

SIEDA beabsichtigt, die Ergebnisse des Projektes app2web auf zwei Arten umzusetzen: Zum einen werden Teile der Applikationen über einen Web-Browser zugänglich gemacht. Dadurch wird die Kundenzufriedenheit erhöht und die führende Stellung auf dem Gebiet der Dienstplanungs-Systeme gestärkt. Zum zweiten werden Domänenmodelle eingesetzt, um die Entwicklung von Produkten für neue Märkte zu vereinfachen und zu beschleunigen. Um dieses Ziel zu erreichen, soll im Rahmen des Projektes entsprechendes Know-how in die Entwicklungspraxis einfließen. Diese Maßnahmen sollen SIEDA dazu dienen, ihren Marktanteil im Bereich der Dienstplanungs-Systeme in den nächsten fünf Jahren signifikant zu erhöhen.

1.1.3 Fraunhofer Institut Experimentelles Software Engineering

Das Fraunhofer Institut Experimentelles Software Engineering (IESE) hat den Auftrag Ergebnisse aus der Grundlagenforschung zu sondieren, für den industriellen Einsatz aufzubereiten und letztendlich dauerhaft in die industrielle Praxis zu transferieren und zu etablieren. Experimentelles Software Engineering steht dabei für folgendes Vorgehen. Zunächst werden in Abhängigkeit von der spezifischen Umgebung eines industriellen Partners Technologien auszuwählen, die in dieser Umgebung optimalen Nutzen (d.h., Verbesserungen) bringen können. Danach werden die ausgewählten Technologien in Pilotprojekten beim Industriepartner eingeführt. Diese Einführung wird durch ein Messprogramm begleitet, das eine objektive Überprüfung des möglichen bzw. erbrachten Nutzens der eingeführten Technologien nach dem Pilotprojekt erlaubt.

Im Rahmen der durchgeführten Domänenanalyse soll demzufolge die Eignung einer Domänenanalyse für die effektive Öffnung bestehender Anwendungen für Internet-Dienste ermittelt werden. Als initiale Fallstudien dient dazu die Domänenanalyse bei SIEDA. Die Auswertung der dabei gewonnenen Erfahrungen sollen helfen die Eignung des Vorgehens für zukünftige, ähnlich gelagerte Transferprojekte abschätzen zu können. Gleichzeitig wird durch das Projekt die Zusammenarbeit mit den bestehenden Projektpartnern vertieft, wodurch der Aufbau längerdauernder Beziehungen zu diesen Organisationen unterstützt wird. Neben den positiven Auswirkungen auf die Zusammenarbeit mit der Industrie trägt das Projekt insbesondere durch die Fallstudien wertvolle Bestandteile zu bereits begonnenen bzw. geplanten Dissertationsarbeiten bei.

1.2 Ziele

Zusätzlich zu den allgemeinen Zielen des Projekts gab es bei der Domänenanalyse bei SIEDA noch spezielle Ziele. Die folgenden Ziele werden bei Domänenanalyse und Design verfolgt:

- Auswahl und Redesign einer Komponente, die wichtig für die Webfähigkeit des Systems ist
- Durch die Integration eines Webinterface wird das System zukunftsfähig, die Wettbewerbsfähigkeit der SIEDA GmbH wird erhöht, die Kundenzufriedenheit wird gesteigert und neue Märkte können erschlossen werden.
- Reduktion der Komplexität des Systems
- Verbesserung der Usability
- Erhöhung der Flexibilität der ausgewählten Komponente und bessere Konfigurierbarkeit des Systems bzw. der Komponente
- Verbesserung der Wartbarkeit
- Erhöhung der Kohäsion und Verringerung der Kopplung der ausgewählten Komponente

Es muss eine Komponente zur Analyse ausgewählt werden, die diese Ziele optimal unterstützt. Dazu muss die Komponente nach außen sichtbar (Webfähigkeit), nicht zu komplex und nicht zu sehr im System verteilt sein. Mit dieser Komponente kann dann eine Domänenanalyse durchgeführt werden.

1.3 Domänenanalyse

Domänenanalyse bedeutet, Anforderungen an Systeme in einer bestimmten Anwendungsdomäne systematisch zu erfassen. Dabei gilt es Gemeinsamkeiten, d.h. immer wieder auftretende Anforderungen, als auch Variabilitäten, d.h. Anforderungen, die nicht von allen Systemen erfüllt werden müssen, zu identifizieren und integriert zu dokumentieren. Das integrierte Dokument enthält dann ein Domänenmodell, das die Grundlage für das Design und die Implementierung von domänenweit wiederverwendbaren Teilsystemen oder Komponenten darstellt.

Um die Anforderungen der angebotenen Web-Funktionen an die dahinterstehende Anwendung zu erfassen und die Unterschiede zur bereits bestehenden Anwendung zu modellieren, wurde eine Domänenanalyse durchgeführt. Dabei wurde die am IESE entwickelte Methode PuLSE-CDA (CDA steht für Customizable Domain Analysis) [BMW99] eingesetzt. Diese Domänenanalyse ist ein Teil

des PuLSE™ Frameworks (Product Line Software Engineering)² [BFK+99], das den gesamten Lebenszyklus einer Software-Produktlinie abdeckt.

Die Anpassung von CDA bedeutet, dass zunächst die benutzten Modelle kontextbezogen ausgewählt wurden (ein Beispiel für eine Anpassung findet sich in [ABB+01] und [ABM00]). Auf dieser Grundlage wurden dann die Dienste beschrieben, die von einer Domänen oder von den die Domäne realisierenden Komponenten gefordert werden. Diese Dienste können sowohl für bestehende als auch für zukünftige Systeme verhältnismäßig einfach dargestellt werden und beschreiben die gemeinsamen und unterschiedlichen Funktionen der Systeme aus der Sicht eines Domänenexperten.

Ziel dieses App2web-Arbeitspaketes ist es, Gemeinsamkeiten und Unterschiede zwischen den verschiedenen Produkten der Anwendungsdomänen für webbasierte Anwendungen zu erfassen. Diese werden verwendet, um die verschiedenen Produkte der Domäne zu beschreiben, sie von Produkten anderer Domänen abzugrenzen und ihre zukünftige Entwicklung abzuschätzen.

1.4 Domänenidentifikation

Vor der eigentlichen Domänenanalyse muss eine geeignete Domäne identifiziert werden. Die auszuwählende Domäne soll möglichst viele der oben genannten Ziele optimal unterstützen. Sie muss also ein hohes Verbesserungspotential bzgl. Wartbarkeit, Usability, Kopplung und Kohäsion aufweisen, und vor allem eine enge Verbindung zu der geplanten Web-Schnittstelle besitzen.

Folgende Domänen wurden als mögliche Kandidaten für eine Domänenanalyse angesprochen:

- Workflow Komponente
Es soll eine neue Komponente erstellt werden, die Workflow-Aspekte bei der Dienstplanung wie z.B. Springer zwischen Abteilungen, variable Dauer des Dienstplans, allgemeiner Workflow bei der Bearbeitung des Dienstplans oder Koordination zwischen Abteilungen enthält. Momentan ist die Abarbeitung abteilungsorientiert und monatsbasiert, d .h. es gibt keine variable Struktur.
- Planungskomponente
Planerische Aspekte sollen in dieser Komponente besser realisiert werden, es soll z.B. eine feinere Granularität der Planung, wie eine stundenweise Planung, die Berücksichtigung von verteilten Standorten usw. berücksichtigen. Die Betrachtung von planerischen Aspekten kann auf mehreren Ebenen geschehen, es kann zuerst eine abteilungsübergreifende Grobplanung durchgeführt werden, dann wird erst eine mitar-

² PuLSE™ is a registered trademark of Fraunhofer IESE. For more information see www.iese.fhg.de/PuLSE

beiterbezogene Feinplanung gemacht. Momentan findet die Grobplanung der Dienstpläne manuell statt, nur die Feinplanung wird im System automatisch abgewickelt. Die neue Planungskomponente würde es ermöglichen, beide Ebenen der Planung im System abzudecken.

- **Abrechnungskomponente**
In dieser Komponente sollen abrechnungsrelevante Informationen besser aufgearbeitet werden. Bestehende Lohnarten sollen für Kunden modifiziert oder neue Lohnarten angelegt werden können. Hierzu gehört auch, dass Vorrangbeziehungen zwischen Lohnarten und die zu berücksichtigenden Steueraspekte explizit in der Komponente realisiert und für den Benutzer dargestellt werden. Die Abrechnung soll für den Benutzer erleichtert werden, indem Abrechnungskonten geführt werden, auf denen Umbuchungen vorgenommen werden können. Buchungen können schließlich in Berichten für unterschiedliche Benutzergruppen aufbereitet werden, so dass die einzelnen Abrechnungsschritte nachvollziehbar sind.

Jede der beschriebenen Komponenten ist bereits teilweise im System realisiert, existiert jedoch nicht als explizite Komponente mit definierten Schnittstellen sondern die Funktionalität ist über das System verteilt. Bei jeder dieser Komponenten wäre also die existierende Funktionalität im Code zu identifizieren und herauszulösen, um anschließend die alte und neue Funktionalität in einer Komponente mit definierten Schnittstellen zu realisieren.

Die Entscheidung für die zu realisierende Komponente fiel auf die Abrechnungskomponente, da bei dieser Komponente zum einen die Kopplung innerhalb des Systems nicht zu groß ist (d.h. der Code ist auf eine beherrschbare Menge von Stellen verteilt) und zum anderen die Abrechnungskomponente Kern der zu realisierenden Web-Komponente ist.

1.5 Dokumentstruktur

Die Struktur des Reports entspricht den Subdomänen (oder Teilsystemen/-komponenten), die aufgrund ihrer Beziehung zur ausgewählten Domäne (Abrechnung) im Rahmen der Domänenanalyse betrachtet und analysiert wurden. Abbildung 1 zeigt die Domänenhierarchie und verweist auf die Nummern der Kapitel in denen jede Teildomäne in diesem Report betrachtet wird. Bei den unbenannten Domänen 6-9 handelt es sich um interne Domänen, die nicht zur allgemeinen Veröffentlichung freigegeben wurden³.

³ In Ausnahmefälle kann eine Zugangsberechtigung für den internen Bericht gewährt werden. Wende Sie sich zu diesem Zweck bitte an einen der Autoren.

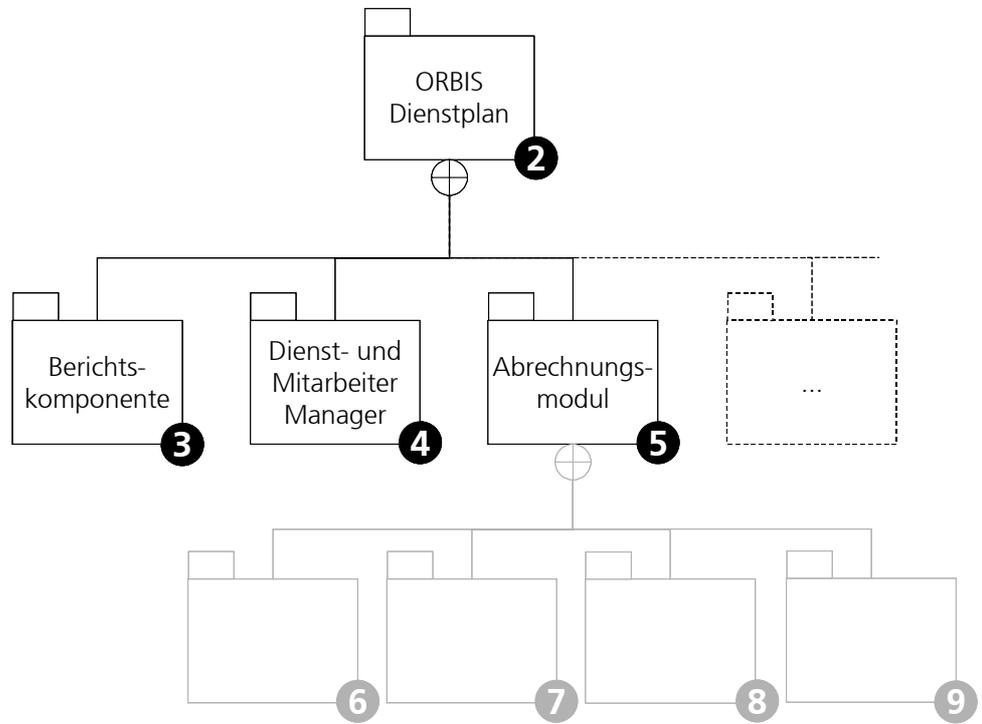


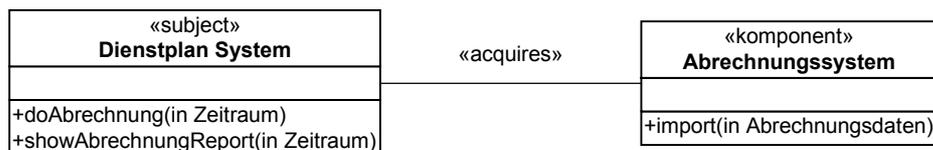
Abbildung 1 Zuordnung von Subdomänen zu Kapiteln

2 Anwendungskontext ORBIS-Dienstplan

In diesem Kapitel wird das Gesamtsystem ORBIS-Dienstplan von einem externen Standpunkt aus betrachtet, um Funktionalitäten in der Abrechnungsdomäne anhand von Benutzungsszenarien zu ermitteln.

2.1 Kontext

ORBIS-Dienstplan ist eine Software zur Personaleinsatzplanung. Als integrierter Bestandteil ermöglicht ORBIS-Dienstplan die Berechnung der un stetigen Bezugsanteile (Zuschläge und Zulagen) zur Automatisierung der Abrechnung. Das System setzt ein externes Abrechnungssystem voraus.



In ORBIS-Dienstplan Version 2.0 sind unterschiedliche Lohnartengruppen für Angestellte und Beamte jeweils mit und ohne Anwendung des Einkommenssteuergesetzes hinterlegt. Der Kunde kann basierend auf den hinterlegten Lohnartengruppen eigene definieren, indem er hinterlegte Lohnartengruppe zusammen mit den enthaltenen Lohnarten kopiert. Für die weitere Bearbeitung der selbstdefinierten Lohnartengruppe stehen dem Kunden nun folgende Möglichkeiten zur Verfügung:

Die Verwendung einer enthaltenen Lohnart kann für die Abrechnung und das Berichtswesen ein- oder ausgeschaltet werden. Mehrere Lohnarten können in der Abrechnung oder im Berichtswesen zu einer Lohnarten zusammengefasst werden.

Die Auswertung von Lohnarten erfolgt mitarbeiterbezogen. Dafür werden Mitarbeiter über Ihre Tarifgruppe (siehe [Sch99]) Lohnartengruppen eindeutig zugeordnet. Erfolgt die Auswertung für die Monatsabrechnung, so werden die Daten in einer Exportdatei ausgegeben, wobei für jede Lohnart sämtliche Positionen aufsummiert werden. Bei der alternativen Auswertung in Berichten können auch die Einzelpositionen der Lohnart mit den jeweils gültigen Kostenstellen ausgegeben werden.

Neben der Abrechnung werden auch Dienste- und Ausfallstatistiken in ORBIS-Dienstplan über die Auswertung von Lohnarten erzeugt.

2.2 Anwendungsfälle

In diesem Abschnitt werden Anwendungsfälle beschrieben, die zu der gewählten Domäne Abrechnung gerechnet werden. Dazu gehören Anwendungsfälle, die geleistete Dienste eines Mitarbeiters den verschiedenen Lohnarten zuordnen (Dienste mit Zuschlägen werden dabei als eigene Lohnart angesehen), als auch Anwendungsfälle zur Erstellung von Berichten, die Dienstplan- und Abrechnungsbezogene Daten enthalten. Letztere greifen auf Anwendungsfälle des ersten Typs zurück. Die hier beschriebenen Anwendungsfälle dienen als illustrative Beispiele der grundsätzlichen Funktionalitäten der Anwendungsdomäne, wobei kein Wert auf Vollständigkeit gelegt wurde.

2.2.1 Berechnung des Samstagszuschlags im Regeldienst

Über die Tarifgruppe des Mitarbeiters wird unter anderem das Konto ‚Samstagszuschlag im Regeldienst‘ ermittelt. Diesem Konto ist eine entsprechende Lohnart mit den Attributen Diensttyp = Regeldienst und Einheit = Stunden zugeordnet. Die Berechnungsregel lautet: Arbeit an Samstagen in der Zeit von 13h00 bis 20h00.

Für den Mitarbeiter wird nun aus allen Diensten im Zeitraum die Menge aller Zeitintervalle ermittelt, in denen Regeldienst geleistet wurde, da der Diensttyp = Regeldienst.

Zeitintervalle von vorrangigen Lohnarten, wie z.B. Feiertagsarbeit, werden aus dieser Menge entfernt.

Von dieser Menge werden nun laut Berechnungsregel alle Zeitintervalle entfernt, die nicht Samstags in der Zeit zwischen 13h00 und 20h00 liegen.

Diese Menge ist die Ergebnismenge.

Für Detaillierte Berichte werden die einzelnen Zeitintervalle und deren Dauer ausgegeben, für externe Abrechnungssysteme die Summe der Dauer der einzelnen Zeitintervalle übergeben.

2.2.2 Berechnung von Überstundenzuschlägen

Das Ziel ist die Ermittlung aller Zeitintervalle, die durch Überstunden entstanden sind und nicht innerhalb der laufenden Woche durch Freizeit ausgeglichen wurden.

Über die Tarifgruppe des Mitarbeiters wird unter anderem das Konto ‚Überstundenzuschlag‘ ermittelt. Diesem Konto ist eine entsprechende Lohnart mit den Attributen Diensttyp = Regeldienst und Einheit = Stunden zugeordnet. Ergebnisse dieses Kontos kommen durch Umbuchungen anderer Konten zustande. Es wird die Umbuchungsregel ‚Überstundenzuschlag‘ angewendet.

Die Umbuchungsregel ‚Überstundenzuschlag‘ nimmt Umbuchungen von Zeitintervalle aus den Konten ‚Mehrarbeit‘(positive Umbuchung) und ‚Freizeitausgleich‘(negative Umbuchung) vor.

Die Frist ist dabei jeweils Umbuchung innerhalb der Woche (von Montag bis Sonntag).

Die Umbuchungsregel lautet nun:

(1) $\text{BucheUm}(\text{Zeitraum}) :=$

$\text{for each Woche}(\text{Montag bis Sonntag}) \text{ of } \text{Zeitraum} \text{ do } \text{BucheUmWoche}(\text{Konto } \text{‚Mehrarbeit‘}(\text{Woche}), \text{ Konto } \text{‚Freizeitausgleich‘}(\text{Woche}))$

(2) $\text{BucheUmWoche}(\text{Konto } \text{‚Mehrarbeit‘}, \text{ Konto } \text{‚Freizeitausgleich‘}) :=$

$\text{Buche Konto } \text{‚Mehrarbeit‘} \text{ positiv auf Konto } \text{‚Überstundenzuschlag‘}$

$\text{Buche solange Konto } \text{‚Freizeitausgleich‘} \text{ negativ auf Konto } \text{‚Überstundenzuschlag‘} \text{ bis}$

$\text{entweder Konto } \text{‚Freizeitausgleich‘} = 0 \text{ oder Konto } \text{‚Überstundenzuschlag‘} = 0;$

(3) $\text{Konto } \text{‚Mehrarbeit‘}(\text{Woche}) :=$ Alle Zeitintervalle des Kontos ‚Mehrarbeit‘ innerhalb der angegebenen Woche. Dem Konto ‚Mehrarbeit‘ ist die Berechnungsregel ‚zuschlagspflichtige Mehrarbeit‘ zugeordnet. In diesem Konto befinden sich alle Zeitintervalle, die im Dienst mit ‚zuschlagspflichtig‘ gekennzeichnet sind.

(4) $\text{Konto } \text{‚Freizeitausgleich‘}(\text{Woche}) :=$ Alle Zeitintervalle des Kontos ‚Freizeitausgleich‘ innerhalb der angegebenen Woche. Dem Konto ‚Freizeitausgleich‘ ist die Berechnungsregel ‚Freizeitausgleich‘ zugeordnet. In diesem Konto befinden sich alle Zeitintervalle, die im Dienst mit dem Fehlzeitengrund ‚frei‘ gekennzeichnet sind.

2.2.3 Bericht „Dienstleistungsübersicht“

Ein Benutzer fordert für einen Mitarbeiter und einen Zeitraum eine Dienstleistungsübersicht an. Der Bericht liefert für jeden Tag innerhalb des Zeitraums folgende Informationen:

- Bezeichner des geleisteten Dienstes/ der Fehlzeit.
- Auf dem Zeitkonto angerechnete Arbeitszeit.
- Zusätzlich zum Dienst erfasste Daten wie Fehlzeiten, Mehrarbeit.
- Dem Tag zuzuordnende Zuschläge.

Weiterhin werden folgende Informationen geliefert:

- Zeitkonto und Urlaubskontenstände am Beginn und Ende des Zeitraums.
- Überstundenkonto mit einer Übersicht der im Zeitraum vorgenommenen Umbuchungen.

2.2.4 Bericht „Kontenübersicht“

Der Benutzer fordert für einen Mitarbeiter und einen Zeitraum eine Kontenübersicht an. Es wird eine Liste von Konten angezeigt. Die Liste enthält alle Konten, bei denen der Kontenstand am Beginn oder Ende des Zeitraums nicht Null war oder bei denen sich der Kontenstand durch eine Umbuchungsregel innerhalb des Zeitraums geändert hat.

Nach Auswahl eines Kontos aus der Liste werden folgende Informationen angezeigt:

- Kontostand am Beginn und Ende des Zeitraums.
- Alle Umbuchungsregeln, die das Konto betreffen, in der Reihenfolge der Ausführung. Für jede dieser Umbuchungsregel werden sowohl Quell- als auch Zielkonten sowie die umgebuchten Daten aufgeführt.
- Sofern sich der Kontostand durch eine Berechnungsregel geändert hat, werden Informationen über die Berechnungsregel sowie der Daten, die in das Konto eingeflossen sind, aufgeführt.
- Wählt der Benutzer ein Quell oder Zielkonto einer der aufgeführten Umbuchungsregeln an, so wird eine Kontoübersicht des ausgewählten Kontos angezeigt. Dadurch ist eine Navigation zwischen verschiedenen Konten, die über Umbuchungsregeln in Beziehung stehen, möglich.

2.2.5 Bericht „Wechselschichtzulage“

Ein Benutzer fordert für einen Mitarbeiter und einen Abrechnungszeitraum einen Bericht über die Wechselschichtzulage an.

Der Bericht liefert in chronologischer Reihenfolge die ausgeführten Berechnungsregeln. Es werden dabei nicht alle Berechnungsregeln aufgeführt, sondern nur Berechnungsregeln von vorher festgelegten Kategorien. Für jede Berechnungsregel werden folgende Informationen angezeigt:

- Bezeichnung der Berechnungsregel
- Ergebnis der Berechnungsregel
- Von der Berechnungsregel verwendete Berechnungsregeln, sofern diese von einer der vorher festgelegten Kategorien sind.
- Zusätzliche Erläuterungen des Ergebnisses

2.3 Komponentenstruktur

Das *Dienstplansystem* ORBIS arbeitet mit verschiedenen, externen Abrechnungssystemen zusammen. Das externe Abrechnungssystem wird von ORBIS mit den für die Abrechnung notwendigen Daten versorgt. Dabei benötigt das Abrechnungssystem pro Mitarbeiter und Abrechnungszeitraum die folgenden Attribute der für die Abrechnung relevanten Konten:

- Lohnart.
- Kostenstelle
- Entstehungsdatum bzw. Entstehungszeitraum
- Wert

Für unterschiedliche Kostenstellen bzw. Entstehungsdatum sind jeweils die zugeordneten Werte zu übergeben.

Intern lässt sich das Dienstplansystem in mehrere Teilkomponenten zerlegen (s. Diagramm). Entsprechend der gewählten Domäne steht das Abrechnungsmodul im Mittelpunkt dieses Dokuments. Die Berichtskomponente *ReportWriter* bereitet Abrechnungsdaten auf und bestimmt damit hauptsächlich die Spezifikation der Schnittstelle, die das Abrechnungsmodul zur Verfügung stellt. Aus diesem Grund wird zunächst die Berichtskomponente spezifiziert, um danach die Spezifikation des Abrechnungsmoduls abzuleiten.

Die Komponenten *Dienstmanager* und *Mitarbeitermanager* stellen lediglich einfache, ergänzende Daten zur Verfügung, wirken sich jedoch nicht signifikant

auf das Abrechnungsmodul aus. Der *Dienstmanager* liefert die geplanten und geleisteten Dienste eines Mitarbeiters für einen Zeitraum. Der *Mitarbeitermanager* ermittelt die Personaldaten der Mitarbeiter.

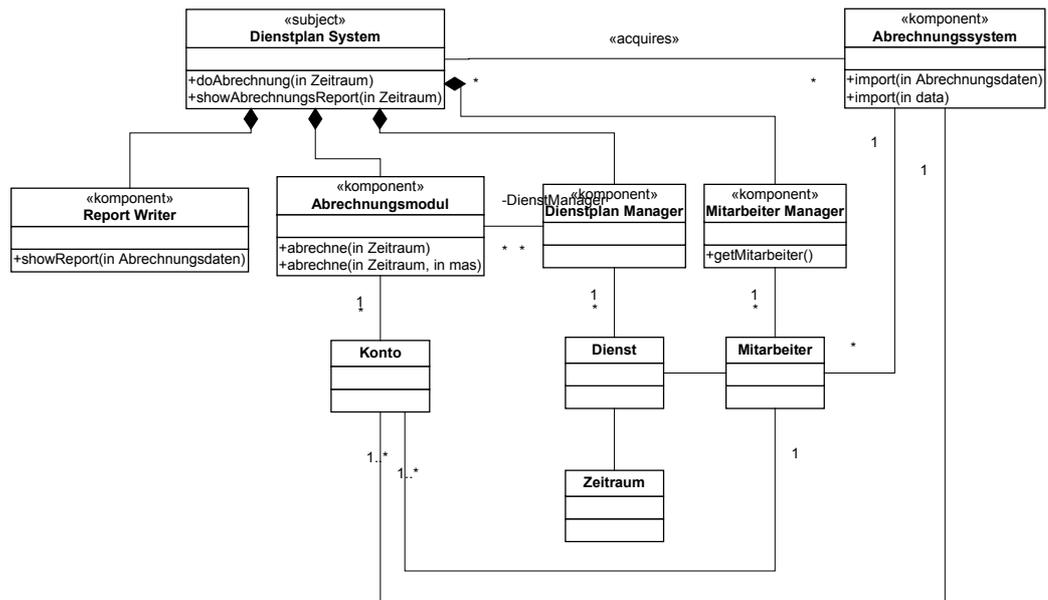


Abbildung 2 Komponentenstruktur ORBIS-Dienstplan

2.3.1 Realisierung im aktuellen System

Dienstmanager

Die Komponente Dienstmanager wird im wesentlichen in der Bibliothek Container realisiert. Hier werden durch die Klasse BerechnungTACacheWrapper Daten für die Berechnung zur Verfügung gestellt. Die von BerechnungTACacheWrapper abgeleiteten Klassen ZeitkontoBerechnungTransakt und UrlaubsBerechnungTransakt liefern Daten, die jeweils nur für die Berechnung der Zeitkonten bzw. Urlaubskonten benötigt werden.

Berechnungsregeln

Die Berechnungsregeln des Datentyps Stunden und Tage, die direkt in eine Lohnart fließen, befinden sich in der Bibliothek Repkon in der Klasse KontoReport. Die Berechnungsregeln verwenden dabei den Dienstmanager. Die Vorrangregelung von Konten ist explizit in den Berechnungsregeln hinterlegt und kann in wenigen Fällen über Eigenschaftszuordnungen verändert werden.

Konten und Umbuchungsregeln

Bei der aktuellen Realisation erfolgt keine scharfe Trennung zwischen Konten und Umbuchungsregeln. Die Konten beinhalten auch die Umbuchungsregeln.

Die Bibliothek Berechnung stellt eine Grundfunktionalität für Konten und Umbuchungsregeln zur Verfügung. Dies schließt eine Protokollierung der Berechnung sowie die Persistenz von Konten ein. Das allgemeine Verhalten der Konten ist in der Klasse `Rechenschritt` hinterlegt.

Die Persistenz von Konten wird in der Klasse `TRechenschrittSpeichern` realisiert.

Die konkreten Konten und somit auch deren Umbuchungsregeln befinden sich in den Bibliotheken `Zeit für Zeitkonten` bzw. `Urlaub für die Urlaubskonten`. Die Konten verwenden keine Berechnungsregeln der Bibliothek `Repkon`.

Wechselschichtzulage

Die Berechnungsregel für die Wechselschichtzulage befindet sich in der Bibliothek `Zeit` und verwendet wie die `Zeitkonten` die Klasse `Rechenschritt`.

2.4 Realisierung im Web

Die Web Funktionalität kann bei ORBIS-Dienstplan in verschiedenen Ausprägungen und in verschiedenen Intensitäten integriert werden.

Die Basislösung ist eine Webschnittstelle, die einem erweiterten Nutzerkreis Daten zur Verfügung stellt. Die Mitarbeiter, deren Dienstpläne mit ORBIS-Dienstplan erstellt werden, können ihre Dienstzeiten und ihre Abrechnungsrelevanten Daten einsehen. Dies kann von der Abrechnungskomponente gesteuert werden. Die Ausgabe wird von einer modifizierten Berichtskomponente übernommen, die dann Abrechnungsdaten nicht wie im Ursprungssystem in eine Datei bzw. einen Report schreibt, sondern die Daten (beispielsweise) als HTML-Seite zur Verfügung stellt. Dies ist ein rein statischer Dienst mit minimaler Interaktion zwischen Client und Server.

Eine weitere Stufe würde vorsehen, dass diese Mitarbeiter über eine Webschnittstelle auch Änderungswünsche äußern können. Die reine Darstellung von Daten reicht hier also nicht mehr aus, es müssen Web-Formulare in die Ausgabe integriert werden und Interaktionsmöglichkeiten zwischen dem geplanten Mitarbeiter am Web-Client und den bisherigen Benutzern von ORBIS-Dienstplan geschaffen werden. Dies schließt dann die Beschäftigung mit Problematiken wie Verfügbarkeit der Daten, Durchführung von Transaktionen von Client zu Server und von Server zu Client, Aktualisierung der Daten usw. ein.

Weitet man die Interaktivität weiter aus, könnte es auch möglich sein, Planungen ganz über ein Webinterface stattfinden zu lassen, also nicht nur dem verplanten Benutzer Zugriff und Änderungsmöglichkeit auf seine Daten zu geben, sondern auch planende Benutzer weitgehend auszulagern. Dieses Szenario würde in letzter Konsequenz ein Application-Service-Provider ORBIS-Dienstplan sein, wo auf einem zentralen Server Planungen, Abrechnung... abgelegt werden können und nur noch über ein Webinterface geplant, abgerechnet und modifiziert wird.

Zur Realisierung der Basislösung eignet sich eine Analyse des Abrechnungsmoduls sehr gut. Die Daten, die dann über das Webinterface an den Mitarbeiter gehen, sind zum großen Teil Abrechnungsrelevante Daten und werden vom Abrechnungsmodul verwendet oder erzeugt. Die so verwendeten und erzeugten Daten werden in die Berichtskomponente übergeben, um verschiedene Berichte zu erzeugen. Eine neue Art von Berichten wären dann Web-Berichte. Um Web-Berichte erzeugen zu können, müssen die Daten und die Abrechnung sehr viel besser erläutert werden, da der Web-Interface Benutzer kein Dienstplansystem zur Hand hat, um eventuelle Fragen zu klären. Diese Erklärungen können nur in ausreichendem Maße erzeugt werden, wenn die Abrechnung transparenter gemacht werden kann. Dies wird durch die Domänenanalyse und das darauf folgende Redesign und Reimplementierung des Abrechnungsmoduls erreicht.

3 Berichtskomponente

3.1 Kontext

Über Berichte werden Daten ausgegeben, so dass diese in einer Ansicht vom Benutzer betrachtet, auf einem Drucker ausgegeben oder in ein Datenformat einer Standardsoftware(z.B. MS Excel) exportiert werden können.

Aufgabe der Berichtskomponente ist die Aufbereitung der Daten über eine Reihe von Services. Dieses erfolgt unabhängig von der Anwendungsarchitektur:

Bei ORBIS-Dienstplan, in der Version 2.0 eine klassische Client-Server Architektur, werden die Daten mit dem Berichtswerkzeug Seagate Crystal Report ausgegeben.

Das geplante Web Front-End wird die Daten mithilfe von Java Server Pages aufbereitet und als HTML-Seite in über einen Web-Browser darstellen. In beiden Fällen kommt die gleiche Berichtskomponente zum Einsatz.

In diesem Dokument werden nur Berichte im Kontext von Abrechnungsdaten betrachtet. Grundsätzlich müssen die im Bericht anzuzeigenden Daten während einer Abrechnung ermittelt werden.

3.2 Funktionale Anforderungen

Die in Berichten angezeigten Daten (Berichtselemente) können prinzipiell im Abrechnungsmodul oder in der Berichtskomponente vorgehalten werden. Ersteres wird favorisiert, um die interne Datenstruktur des Abrechnungsmoduls zu kapseln.

Je nach Bericht innerhalb der Abrechnung werden unterschiedliche Informationen benötigt. Dabei zeigt ein Bericht häufig nur eine Teilmenge der verfügbaren Daten an, d.h. es werden nur bestimmte Berichtselemente für den Bericht benötigt (F507). Zum Beispiel ist eine Vorauswahl des Benutzers (F501) für den Bericht Wechselschichtzulage möglich. Der Einsatz von Filtern auf den Berichtselementen kann hier die Ergebnismenge reduzieren, um den Datenstrom zwischen der Berichts- und der Abrechnungskomponenten möglichst gering zu halten.

Insgesamt müssen Zusammenhänge zwischen Berichtselementen beachtet und dargestellt werden. Dabei sind folgende Zusammenhänge von Interesse:

- Die chronologischer Reihenfolge der Ausführung der Berichtselemente (F502).
- Berichtselemente verwenden oder beeinflussen andere Berichtselemente (F503, F504).
- Berichtselemente gehören unterschiedlichen Typen oder Kategorien an.
- Berichtselemente beziehen sich auf unterschiedliche Datumswerte (F505).

Das Abrechnungsmodul soll den Zugriff auf Berichtselemente möglichst einfach für die Berichtskomponente gestalten. Deshalb werden hier verschiedene abrechnungsbezogene Berichte analysiert, um anschließend eine geeignete Schnittstelle des Abrechnungsmoduls zu spezifizieren.

Berichte, die auf Daten des Abrechnungsmoduls basieren, stellen im Wesentlichen die Informationen von Konten, Umbuchungsregeln und Berechnungsregeln dar. Ein Berichtselement enthält Informationen eines Kontos, einer Umbuchungsregel oder einer Berechnungsregel.

F501	Der Benutzer kann eine Vorauswahl die Menge bzw. Art der anzuzeigenden Daten festlegen.
F502	Die chronologische Reihenfolge der Ausführung der Berichtselemente kann dargestellt werden.
F503	Der Zusammenhang zwischen Umbuchungsregeln und Konten kann dargestellt werden.
F504	Der Zusammenhang zwischen Berechnungsregeln und Konten kann dargestellt werden.
F505	Eine datumsbezogene Darstellung der Berichtselemente ist möglich.
F506	Die Berichtskomponente darf keine Informationen über die konkrete Implementierung von Berichtselementen besitzen.
F507	Für jeden Typ von Berichtselementen können die notwendigen Informationen dargestellt werden.
F508	Über die Zusammenhänge der Berichtselemente kann navigiert werden.

3.3 Berichte

3.3.1 Bericht „Dienstleistungsübersicht“

Im Bericht Dienstleistungsübersicht werden im wesentlichen die Lohnarten tagbezogenen aufgeführt.

(1) ÜbersichtDienstleistung (Zeitraum, Mitarbeiter)

```

for each Tag in Zeitraum do
    AusgabeLohnartAnTag(Tag, Mitarbeiter)
(2) AusgabeLohnartAnTag (Tag, Mitarbeiter) :=
    Ausgabe(Dienst an Tag)
    for each Lohnart do
        if(Existiert Ergebnis der Lohnart an Tag)
            Ausgabe(Bezeichnung der Lohnart)
            Ausgabe(Ergebnis der Lohnart an Tag)
    
```

Die folgende Tabelle beschreibt die dabei verwendeten Datenstrukturen, die von der Methode *HoleErgebnisseLohnarten(Zeitraum, Mitarbeiter)* des Abrechnungsmoduls geliefert werden.

Datum	Format	Bemerkung
Dienstleistungs- übersicht	{Lohnartergebnis}	
Lohnartergebnis	Lohnart & {Ergebnis}	
Lohnart	Name & Datentyp & Persistent	
Name	String(20)	Eindeutiger Name.
Datentyp	enum(Stunden, Tage, Aufzählung)	Rückgabewert der Lohnart
Persistent	enum(ja, nein)	,ja' bei Lohnarten, die aus persistenten Konten hervorgehen
Ergebnis	Wert & Bezugstag	
Wert	Stunden Tage Aufzählung	Abhängig vom Datentyp
Bezugstag	Datum	
Stunden	Dauer	
Tage	Zahl	
Aufzählung	enum(Wechselschichtzulage, Schichtzulage1, Schichtzulage2, Schichtzulage3, Keine Schichtzulage, Nicht alle Daten vorhanden)	Aufzählungstyp der Wechselschichtzulage

3.3.2 Bericht „Berechnungsregeln“

Im Bericht Berechnungsregeln werden die angewandten Berechnungsregeln tagesbezogen ausgegeben.

```

(1) Übersicht Berechnungsregeln (Zeitraum, Mitarbeiter)
for each Tag in Zeitraum do
AusgabeBerechnungsregelAnTag(Tag, Mitarbeiter)
(2) AusgabeBerechnungsregelnAnTag (Tag, Mitarbeiter) :=
    Ausgabe(Dienst an Tag)
    for each Berechnungsregel do
if(Existiert Ergebnis der Berechnungsregel an Tag)
Ausgabe(Bezeichnung der Berechnungsregel)
Ausgabe(Ergebnis der Berechnungsregel an Tag)
if(Existieren vorrangig angewandte Berechnungsregeln)
Ausgabe(Bezeichnungen vorrangiger Berechnungsregeln)

```

Die folgende Tabelle beschreibt die dabei verwendeten Datenstrukturen, die von der Methode *HoleErgebnisseBerechnungsregeln(Zeitraum, Mitarbeiter)* des Abrechnungsmoduls geliefert werden.

Datum	Format	Bemerkung
Berechnungsregelübersicht	{Berechnungsregelergebnis}	Alle angewandten Berechnungsregeln
Berechnungsregelergebnis	Berechnungsregel & {Ergebnis}	Eine Berechnungsregel mit Ergebnissen
Berechnungsregel	Name & Datentyp & Vorrang	
Name	String(20)	Eindeutiger Name.
Datentyp	enum(Stunden, Tage, Aufzählung)	Rückgabewert der Berechnungsregel
Vorrang	{Name}	Die Namen vorrangiger Berechnungsregeln

Ergebnis	Wert & Bezugstag & Vorrang	Vorrangige Berechnungsregeln erscheinen nur dann, wenn die entsprechende Vorrangregelung angewandt wurde. Werden Ergebnisse, die aufgrund der Vorrangregelung Null sind, aufgeführt?
Wert	Stunden Tage Aufzählung	Abhängig vom Datentyp
Bezugstag	Datum	
Stunden	Dauer & Zeitraumliste	
Zeitraumliste	{Zeitraum}	
Zeitraum	Bezugstag & Von & Bis	Bezugstag ist der Tag, an dem der Zeitraum beginnt
Von	Zeit	Beginn des Zeitraums
Bis	Zeit	Ende des Zeitraums
Tage	Zahl	
Aufzählung	enum(Wechselschichtzulage, Schichtzulage1, Schichtzulage2, Schichtzulage3, Keine Schichtzulage, Nicht alle Daten vorhanden)	Aufzählungstyp der Wechselschichtzulage

3.3.3 Bericht „Wechselschichtzulage“

Die bei der Berechnung der Wechselschichtzulage durchgeführten Rechenschritte werden ausgegeben. Ein Rechenschritt kann dabei weitere Rechenschritte aufrufen.

(1) ÜbersichtWechselschichtzulage (Zeitraum, Mitarbeiter)

ListeRechenschrittypen := Ermittle aufgrund Berichtskonfiguration Typen von Rechenschritten, die ausgegeben werden sollen.

AusgabeRechenschritt(Rechenschritt(Wechselschicht, Zeitraum, Mitarbeiter))

(2) AusgabeRechenschritt (Rechenschritt) :=

Ausgabe(Bezeichnung des Rechenschritts)

Ausgabe(Ergebnis des Rechenschritts)

```

    Ausgabe(Erläuterungen des Rechenschritts, falls vorhanden)
  for each AbhängigerRechenschritt, der in Rechenschritt verwendet wurde do
  if(Typ des AbhängigerRechenschritt enthalten in ListeRechenschritttypen)
    AusgabeRechenschritt (AbhängigerRechenschritt)

```

3.3.4 Bericht „Umbuchungen“

Die Umbuchungen werden in der ihrer Ausführung entsprechenden Reihenfolge ausgegeben.

```
(1) UebersichtUmbuchungen(Zeitraum, Mitarbeiter)
```

Konten := Alle Konten, die als Quell oder Zielkonten in Umbuchungen(Zeitraum, Mitarbeiter) vorkommen

```
for each Konto in Konten do
```

```
  AusgabeKontostand(Beginn des Zeitraums)
```

```
    for each Umbuchung in Umbuchungen(Zeitraum, Mitarbeiter) do
```

```
      AusgabeUmbuchung(Umbuchung)
```

```
for each Konto in Konten do
```

```
  AusgabeKontostand(Ende des Zeitraums)
```

```
(2) AusgabeUmbuchung(Umbuchung) :=
```

```
  Ausgabe(Wert der Umbuchung)
```

```
  Ausgabe(Quellkonten der Umbuchung)
```

```
  Ausgabe(Zielkonten der Umbuchung)
```

```
  Ausgabe(Bedingung, die die Umbuchung auslöste, z.B. Frist)
```

Die folgenden Tabellen beschreiben die dabei verwendeten Datenstrukturen. Sie werden von den Methoden

HoleKontenMitUmbuchung(Zeitraum, Mitarbeiter) – liefert alle an Umbuchungen im Zeitraum beteiligte Konten

HoleKontenstand(Konto, Tag, Mitarbeiter) – liefert den Kontenstand am angegebenen Tag

HoleUmbuchungen(Kontenliste, Zeitraum, Mitarbeiter) – liefert, in chronologischer Reihenfolge, die im Zeitraum durchgeführten Umbuchungen, die Konten der Kontenliste verändert haben.

Kontenliste

Datum	Format	Bemerkung
Kontenliste	{Konto}	Liste von Konten
Konto	Name & Datentyp	
Name	String(20)	Eindeutiger Name.
Datentyp	enum(Stunden, Tage, Aufzählung)	Datentyp des Kontos

Kontenstand

Datum	Format	Bemerkung
Kontenstand	Wert & Einzelwerte	Der Wert ergibt sich aus der Summer der Einzelwerte
Wert	Stunden Tage Aufzählung	Abhängig vom Datentyp
Einzelwerte	{Einzelwert}	
Einzelwert	Bezugstag & Wert	
Bezugstag	Datum	
Stunden	Dauer	
Tage	Zahl	
Aufzählung	enum(Wechselschichtzulage, Schichtzulage1, Schichtzulage2, Schichtzulage3, Keine Schichtzulage, Nicht alle Daten vorhanden)	Aufzählungstyp der Wechselschichtzulage

Umbuchungsliste

Datum	Format	Bemerkung
Umbuchungs- liste	{Umbuchung}	Liste von Umbuchungen
Umbuchung	Name & Datentyp & Bezugstag & Wert & Quellkonten & Zielkon- ten & Bedingung	
Name	String(20)	Eindeutiger Name der Um- buchung
Datentyp	enum(Stunden, Tage)	Datentyp des Kontos
Bezugstag	Datum	Bezugstag
Wert	Stunden Tage	
Stunden	Dauer	
Tage	Zahl	
Quellkonten	Richtung & {Konto}+	
Zielkonten	Richtung & {Konto}+	
Richtung	enum(Zugang, Abgang)	Angabe, ob es sich um um eine Buchung auf das Konto oder vom Konto handelt.
Bedingung	Art & Beschreibung	
Art	enum(automatisch, manuell)	Art der Umbuchung
Beschreibung	String(50)	Beschreibung der Bedin- gung, die die Umbuchung auslöste bzw. Erläuterung der manuellen Umbuchung

3.3.5 Bericht „Konto“

Für ein Konto werden alle Änderungen innerhalb eines Zeitraums dargestellt.

(1) UebersichtKonto (Zeitraum, Mitarbeiter, Konto) :=

AusgabeKontostand(Beginn des Zeitraums)

AusgabeUmbuchungenInZeitraum (Konto, Zeitraum, Mitarbeiter)

AusgabeKontostand(Ende des Zeitraums)

(2) AusgabeUmbuchungenInZeitraum(Konto, Zeitraum, Mitarbeiter) :=

for each Umbuchung of Konto of Mitarbeiter im Zeitraum do

Ausgabe(Wert der Umbuchung auf/von Konto)

Ausgabe(„Zugang“ oder „Abgang“)

Ausgabe(Beteiligte Quell- und Zielkonten außer Konto)

Ausgabe(Bedingung, die die Umbuchung auslöste, z.B. Frist)

Die Datenstrukturen und Methoden aus dem Bericht „Umbuchungen“ können wiederverwendet werden. Der Parameter *Kontenliste* der Methode *HoleUmbuchungen* enthält dann das gewünschte Konto als einziges Element.

3.4 Realisierung

Zusammenhänge zwischen Berichtselementen lassen sich durch folgende Varianten realisieren:

- Das Abrechnungsmodul muss entweder einen Iterator zur Verfügung stellen, der über diese Zusammenhänge iterieren kann.
- Das Abrechnungsmodul muss die Daten in einer Form anbieten, sodass die oben aufgeführten Zusammenhänge alle dargestellt werden können.

4 Dienst- und Mitarbeitermanager

Der Dienstmanager und der Mitarbeitermanager sind Komponenten, die Daten für das Abrechnungsmodul bereitstellen. Diese Komponenten sind nicht Teil dieser Anforderungserhebung. Deshalb wird hier nur die jeweilige Schnittstelle zur Abrechnungskomponente behandelt.

Der *Dienstmanager* liefert die Dienstplandaten eines Mitarbeiters für einen Zeitraum. Pro Mitarbeiter und Tag kann ein Dienst oder eine Fehlzeit zugeordnet sein. Dienste bestehen aus ein oder mehreren Eintrags-elementen, wobei die Eintrags-elemente aus einem Zeitraum, dem Diensttyp und abhängig vom Diensttyp aus weiteren Attributen bestehen. Die Zeiträume der Eintrags-elemente können dabei in die nächsten Tage hineinreichen (z.B. bei Nachtschichten). Der Dienstmanager ist auch in der Lage, nur Dienstplandaten eines bestimmten Diensttyps zu liefern.

Der Mitarbeitermanager liefert die Personaldaten eines Mitarbeiters. Relevant für das Abrechnungsmodul ist unter anderem die Tarifgruppe und die Rahmenschichtfolge des Mitarbeiters.

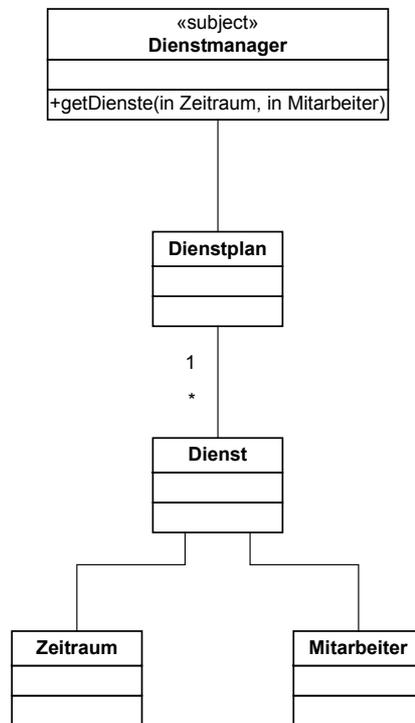


Abbildung 3

Klassendiagramm *Dienstmanager*

GetDienste ist die Hauptfunktion des Dienstmanagers.

Informal Description	Alle Dienste, die ein Mitarbeiter in einem vorgegebenen Zeitraums gearbeitet hat, werden ermittelt und zurückgegeben.
Receives	ma: Mitarbeiter z:Zeitraum
Changes	dienste: new Collection of Dienst
Assumes	Die gespeicherten Dienste sind tatsächlich von dem Mitarbeiter ma im Zeitraum z geleistet worden.
Result	Sämtliche Dienste, die ma in z arbeitete, sind ermittelt und zurückgegeben worden.

5 Abrechnungsmodul

Das Abrechnungsmodul beinhaltet die weiteren Komponenten Berechnungsregeln, Wechselschichtzulage, Konten und Umbuchungsregeln. Folgende Aufgaben leistet das Abrechnungsmodul:

- Das Führen von Konten.
- Die Ermittlung von Daten zur Übergabe an ein Abrechnungssystem.
- Die Ermittlung von Daten für die Berichtskomponente.

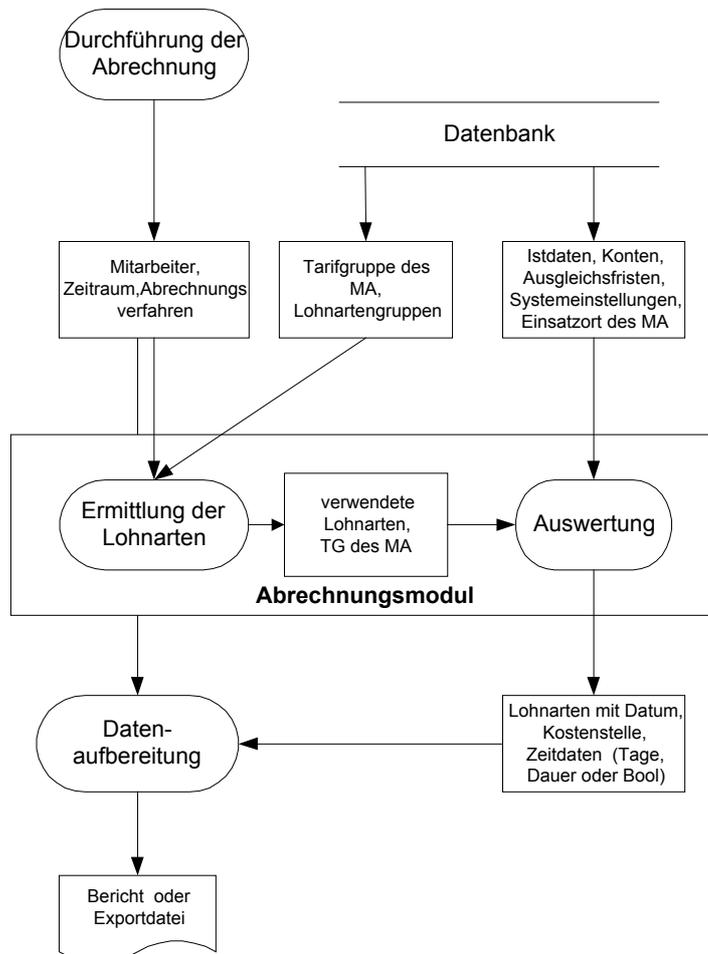


Abbildung 4 Datenflußdiagramm des Abrechnungsmoduls

Die Abrechnungskomponente benötigt für die Durchführung einer Abrechnung als zusätzliche Komponenten den Dienstmanager und den Mitarbeitermanager.

Die obige Abbildung zeigt den prinzipiellen Datenfluß der Abrechnung, wie sie in der neuen Komponente realisiert werden soll. Das Abrechnungsmodul ermittelt die abzurechnenden Lohnarten mit Hilfe von Konfigurationsdaten und Mitarbeiterdaten, welche in der Datenbank abgelegt sind. Diese Lohnarten werden abgerechnet und die Ergebnisse werden weiteren Komponenten zur Verfügung gestellt. Die Ergebnisse in Form einer Abrechnungsinformation (Lohnart mit Datum, Kostenstelle, Zeit..) werden dann aufbereitet und ausgegeben.

Literatur

- [ABB+01] C. Atkinson, J. Bayer, C. Bunse, O. Laitenberger, R. Laqua, E. Kamsties, D. Muthig, B. Paech, J. Wüst, and J. Zettel. Component-based Product Line Engineering with UML, Component Series, Addison-Wesley, 2001
- [ABM00] C. Atkinson, J. Bayer, and D. Muthig. Component-Based Product Line Development: The Kobra Approach, in P. Donohoe. Software Product Lines - Experience and Research Directions, Kluwer Academic Publishers, 2000
- [BFK+99] J. Bayer, O. Flege, P. Knauber, R. Laqua, D. Muthig, K. Schmid, T. Widen, and J.-M. DeBaud. PuLSE: A Methodology to Develop Software Product Lines, in Proceedings of the Fifth ACM SIGSOFT Symposium on Software Reusability (SSR'99), (Los Angeles, CA, USA), pp. 122-131, ACM, May 1999
- [BMW99] J. Bayer, D. Muthig, and T. Widen. Customizable Domain Analysis, in Krzysztof Czarnecki, Ulrich W. Eisenecker (Eds.): Generative and Component-Based Software-Engineering. First International Symposium, GCSE '99, Erfurt, Germany, September 1999, Revised Papers. Lecture Notes in Computer Science 1799. Springer-Verlag, Heidelberg, Germany, 2000
- [Sch99] Wolfgang Schelter. Das Tarifrecht der Angestellten in Krankenhäusern und Heimen (BAT/BAT-O), Courier Verlag, Frankfurt, 1999

Anhang

A. Glossar

- **Abrechnung**

Monatlich durchzuführende mitarbeiterbezogene Ermittlung von Bewegungsdaten (z.B. geleistete Nachtdienste), die an ein Abrechnungssystem zur Berechnung von Zuschlägen und Zulagen übergeben werden.

Abrechnungsdatensatz

Atomarer Datensatz, der vom Abrechnungsmodul erzeugt wird, um anschliessend für ein Abrechnungssystem oder einen Bericht weiter aufbereitet zu werden.

Abrechnungsfall

Abrechnung, die für einen Mitarbeiter und einen Abrechnungszeitraum durchgeführt werden soll.

- **Abrechnungssystem**

EDV-System das die Abrechnung der Löhne und Gehälter einschliesslich der unstetigen Bezugssysteme durchführt. Abrechnungssysteme werden an ORBIS-Dienstplan über eine Schnittstelle angebunden.

- **Ausfallstatistik**

Statistik zur Ermittlung der Ausfallzeiten / Fehlzeiten (Urlaub, Krankheitstage) innerhalb einer Organisation.

- **Ausgleichsfrist**

Zeitraum, der für den Ausgleich von zusätzlicher Arbeitsleistung (über die Sollarbeitszeit hinaus geleistete Arbeit, Arbeit durch Ruf- und Bereitschaftsdienste, Arbeit, die auf dem Zeitkonto erfasst wurde) zur Verfügung steht.

- **Berechnungsregel einer Lohnart**

Die Berechnungsregel definiert, wie die Bewegungsdaten der Lohnart aus den Dienstplandaten ermittelt werden. Berechnungsregel basieren im Gegensatz zu Umbuchungsregeln nicht auf den Kontenständen.

- **Bereitschaftsdienst**

Ein Bereitschaftsdienst verpflichtet den Angestellten, sich auf Anordnung des Arbeitgebers ausserhalb der regelmäßigen Arbeitszeit an einer vom Arbeitgeber bestimmten Stelle aufzuhalten, um im Bedarfsfall die Arbeit aufzunehmen (§15 Abs. 6a+b BAT).

- **Bereitschaftsdienststufe**

Je nach tatsächlich geleisteter Arbeit während eines Bereitschaftsdienstes wird die Zeit des Bereitschaftsdienstes wie folgt als Arbeitszeit gewertet (§15R 2a BAT, Nr. 6 zu § 15 Abs. 6a+b):

Stufe	Arbeitsleistung innerhalb des Bereitschaftsdienstes	Bewertung als Arbeitszeit
A	0 bis 10 %	15 %
B	>10 % und ≤25 %	25 %
C	>25 % und ≤40 %	40 %
D	>40 % und ≤49 %	55 %

- **Berichtswesen**

Teilsystem von ORBIS-Dienstplan, dessen Aufgabe im Erzeugen und Drucken von Berichten besteht.

- **Bewegungsdaten**

Oberbegriff für die Art der Daten, die vom Abrechnungsmodul einer Dienstplanung erzeugt werden. Bewegungsdaten werden an Abrechnungssysteme übergeben, um in Geldeinheiten bewertet zu werden.

- **Dienste**

Arbeitseinsatz eines Mitarbeiters, der durch den Zeitraum des Dienstes und die erfolgten Arbeitsunterbrechungen (z.B. Pausen) gekennzeichnet ist.

- **Dienstplandaten**

Die von einem Mitarbeiter geplanten bzw. tatsächlich erfolgten Dienste und Fehlzeiten. Die Dienstplandaten enthalten auch Attribute wie zuschlagspflichtige Mehrarbeit oder Freizeitausgleich bzw. Auszahlung von Feiertagsarbeit.

- **Dienststatistik**

Statistik zur Ermittlung der Dienstverteilung innerhalb einer Organisation.

- **Diensttyp**

Einteilung der Dienste in Kategorien nach Lage und Art des Dienstes. ORBIS-Dienstplan unterscheidet die Dienstypen Früh-, Tag-, Spät-, Nacht-, Bereitschafts- und Rufbereitschaftsdienst. Früh-, Tag-, Spät- und Nachtdienste werden auch als Regeldienste bezeichnet. Komplexe Dienste setzen sich aus einem Regeldienst und einem Ruf- oder Bereitschaftsdienst zusammen. Geteilte Dienste setzen sich aus mehreren Regeldiensten zusammen.

- **Einkommensteuergesetz**

Regelt die Besteuerung der Einkünfte von natürlichen und juristischen Personen.

- **Einsatzort**

Eindeutiger Aufenthaltsort des Mitarbeiters bezüglich der Dienstplanung.

- **Fehlzeiten**

Abwesenheitszeit eines Mitarbeiters. Fehlzeiten werden dabei durch einen Grund gekennzeichnet (z.B. Urlaub), der die Auswertung und Abrechnung der Fehlzeit eindeutig definiert.

- **Haustarif**

Tarifvertragliche Regelung, die individuell für eine bestimmte Organisation vereinbart wurde. Hautarife basieren im allgemeinen auf Standardtarifverträgen (z.B. dem BAT).

- **Konto**

Dient zur mitarbeiterbezogenen Erfassung von Vorgängen (Arbeitsleistungen, Fehlzeiten) einer Kategorie.

- **Kostenstelle**

Betriebsbereich oder -funktion, für den / die eine gesonderte Kostenerfassung erfolgt.

- **Lohnart**

Für die Monatsabrechnung werden neben Lohn bzw. Gehalt differenzierte Zeitzuschläge und Zulagen zu Lohnarten zusammengefasst.

- **Lohnartengruppe**

Menge von Lohnarten, die sich auf einen bestimmten Tarifvertrag bezieht.

- **Mitarbeiter**

Person, deren Stammdaten in ORBIS-Dienstplan erfasst sind und die innerhalb des Systems geplant werden kann.

- **Monatsabrechnung**

Monatliche wiederkehrende mitarbeiterbezogene Abrechnung der unstetigen Bezugsanteile (Zuschläge und Zulagen).

- **Organisation**

Struktur, die von ORBIS-Dienstplan verwaltet wird.

- **Persistentes Konto**

Konten, die vom Benutzer bearbeitet werden können, indem der Benutzer Buchungen auf dem Konto durchführt. Persistente Konten werden vom System gespeichert.

Charakteristisch für das persistente Konto ist, dass die Kontostände von zurückliegenden Perioden für die Durchführung von Berechnungen benötigt werden.

- **Personaleinsatzplanung**

Die periodische mitarbeiterbezogene Planung von Diensten und Fehlzeiten für einen definierten Zeitraum.

- **Tarifgruppe**

Regelungen eines Tarifvertrages werden innerhalb von ORBIS-Dienstplan durch eine Tarifgruppe definiert. Jeder Mitarbeiter ist einer Tarifgruppe eindeutig zugeordnet.

- **Umbuchungsregel**

Eine Umbuchungsregel definiert, wie Buchungen auf einer Anzahl von Konten auf ein anderes Konto unter Berücksichtigung der Ausgleichsfristen und geg. der Bewertungen umgebucht werden. Umbuchungsregel basieren hierbei auf den Kontenständen und nicht auf den Dienstplandaten.

-

- **Urlaubskonto**

Konto zur Erfassung der Fehlzeiten der Kategorie Urlaub in Tagen oder in Stunden.

- **Zeitkonto**

Konto zur Erfassung sämtlicher Arbeitsleistungen in Stunden.

- **Zulage**

Geldbetrag der zusätzlich zum Gehalt gezahlt wird, sofern eine bestimmte Anspruchsvoraussetzung erfüllt ist. (z.B. Wechselschicht- oder Schichtzulage sowie Monatszulagen)

- **Zuschlag**

Zusätzliche Leistung in Geld oder Zeit die neben dem Gehalt fällig ist, wobei sich die Höhe des Anspruchs aus einer Bewertung der geleisteten Arbeit ergibt (z.B. geleistete Nacharbeit).

B. Notation

A.1 Darstellung der Anforderungen

Einzelanforderungen werden im folgenden Text tabellarisch aufgeführt. Jede geforderte Funktion wird mit einem vorangestelltem "F" gefolgt von einer Nummer versehen, um diese eindeutig referenzieren zu können. Funktionen werden innerhalb eines Anforderungsabschnitts fortlaufend nummeriert.

Verfeinerung von Einzelanforderung sind möglich, wobei hierbei die Referenznummer um eine Unternummerierung erweitert wird (z.B. erhält die erste Verfeinerung der Funktion F101 die Nummer F101.1).

Funktionen, deren Realisierung für einen späteren Zeitpunkt geplant ist, erscheinen am Ende eines Anforderungsanschnitts und werden optischen von den früher zu realisierenden Anforderungen getrennt.

A.2 Datenformat

Alle benötigten Datensätze werden in Datenverzeichnissen unter Angabe ihres Datenformats genau spezifiziert. Datenformate werden als Ausdruck in einer Syntax, die der Erweiterten Backus-Naur-Form gleicht, angegeben. Die folgende Syntax wird verwendet:

A & B	Zusammengesetzter Ausdruck (Konkatenation)
A B	Exklusives Oder zweier Ausdrücke.
[A]	Ausdruck kann ignoriert werden.
{A}	Ausdruck kann beliebig oft wiederholt werden.
{A}+	Ausdruck erscheint mindestens einmal.
{A}n	Ausdruck erscheint genau n-mal.
{A}<=n	Ausdruck erscheint bis zu n-mal.
(A B) & C	Runde Klammern dienen zur Gliederung von Ausdrücken.
/Kom/	Bei /Kom/ handelt es sich um einen Kommentar.

Die folgenden Datentypen werden als Terminale verwendet, wobei die auftretenden Klammern jeweils optional sind:

integer	eine ganze Zahl.
string	Zeichenkette
1..10	Zahlenbereich von 1 bis 10.
enum(a , b)	Aufzählungstyp, dessen Elemente in Klammern angegeben werden.
Datum	zulässiges Datum (mit vierstelliger Jahreszahl).
Gültigkeit	Datum & Datum (Zeitraum)
Zeit	Zeitangabe in Stunden und Minuten
Dauer	Dauer in Stunden und Minuten

Dokumenten Information

Titel: Öffnung der Anwendung
ORBIS-Dienstplan für Internet-
Dienste durch Domänenanalyse

Datum: Dezember 2001
Report: IESE-092.01/D
Status: Final
Klassifikation: Public

Copyright 2001, Fraunhofer IESE.
Alle Rechte vorbehalten. Diese Veröffentlichung darf für kommerzielle Zwecke ohne vorherige schriftliche Erlaubnis des Herausgebers in keiner Weise, auch nicht auszugsweise, insbesondere elektronisch oder mechanisch, als Fotokopie oder als Aufnahme oder sonstwie vervielfältigt, gespeichert oder übertragen werden. Eine schriftliche Genehmigung ist nicht erforderlich für die Vervielfältigung oder Verteilung der Veröffentlichung von bzw. an Personen zu privaten Zwecken.