Vorlage Systemspezifikation

glossary:licence:partial$licence.adoc

Dieser Lizenztext bezieht sich auf die Vorlage. In Dokumenten, die auf der Vorlage basieren, ist dieser Text zu löschen.

Allgemeine Hinweise zur Dokumentvorlage

Diese Dokumentvorlage enthält die Gliederung für eine V-Modell-konforme Gesamtsystemspezifikation (Pflichtenheft) eines Softwaresystems (Software-Entwicklungsprojekt) mitsamt Ausfüllhinweisen und Beispielen in den jeweiligen Kapiteln. Die Hinweise und Beispiele (auch dieser Abschnitt) sind in der Spezifikation für ein (Software-)System später zu entfernen. Sie dienen nur als Hilfestellung für die Erstellung des Dokuments.

Nach V-Modell wird zwischen einem Lasten- und Pflichtenheft unterschieden. Das Lastenheft soll nach V-Modell alle fachlichen Anforderungen an das zu entwickelnde System enthalten und wird vom Auftraggeber als Dokument sowie ggf. (in Zusammenarbeit mit dem Auftragnehmer) als Anforderungsliste erstellt. Das Pflichtenheft konkretisiert diese Anforderungen und fügt bei Bedarf neue Anforderungen hinzu. Ziel des Pflichtenhefts ist es, alle funktionalen und nicht-funktionalen Anforderungen an das neu zu entwickelnde System zu beschreiben. Es wird vom Auftragnehmer in Zusammenarbeit mit dem Auftraggeber erstellt.

Die Gliederungen von Lasten- und Pflichtenheft unterscheiden sich laut Dokumentvorlage aus dem V-Modell nur minimal. Das Lastenheft wird in der Praxis aber entweder weniger formal notiert als das hier beschriebene Pflichtenheft oder in Form einer Anforderungsliste[1] erstellt. Im Lastenheft notiert der Kunde seine Anforderungen an die neu zu erstellende Anwendung oder an die Änderungen daran.

Das Pflichtenheft für ein neu zu erstellendes System soll der hier beschriebenen Spezifikationsmethodik genügen. Hierbei kann es abhängig von der Funktionalität sinnvoll sein, von den hier aufgestellten Regeln abzuweichen, um die Spezifikation verständlicher und kompakter zu gestalten. Dies ist mit dem Kunden im Vorhinein abzustimmen.

Für Änderungen an bestehenden Systemen ist die hier beschriebene Methodik hingegen mit Augenmaß anzuwenden. Für kleine Änderungen kann ein informeller Text zur geplanten Änderung genügen, der vom Kunden abgenommen wird. Für größere Änderungen kann es Sinn machen, Teile der hier beschriebenen Methodik zu nutzen oder bestehende Dokumente nach dieser Methodik anzupassen und die Änderungen zu markieren.

Zur Ablösung eines Altsystems durch ein neu erstelltes System muss die Datenmigration, eine Inbetriebnahmeplanung und ein Prozess zur Schulung der Mitarbeiter beschrieben werden. Diese Punkte werden hier nicht dokumentiert.

Sowohl Systemspezifikationen neu erstellter Anwendungen als auch Änderungen an bestehenden Anwendungen werden nach Inbetriebnahme in eine zentrale „Masterspezifikation“ integriert. Zu diesem Zeitpunkt müssen auch informell beschriebene bzw. informell abgesprochene Änderungen in die Systemspezifikation der jeweiligen Anwendung nach dieser Spezifikationsmethodik

zurückfließen.

Die Gliederungen in den Dokumentvorlagen des V-Modells XT sind für die Spezifikation von Softwaresystemen nicht konkret genug, um auszuschließen, dass wesentliche Anforderungen vergessen werden. Auf Basis der Erfahrungen in bisherigen Projekten wurde die vorliegende Dokumentvorlage erstellt, die eine Konkretisierung der Vorlagen aus dem V-Modell XT ist. Das Mapping dieser Gliederung mit den Dokumentvorlagen des V-Modells XT zeigt die nachfolgende Abbildung.

V-Modell XT	Spezifikationsmethodik
1. Einleitung	1. Allgemeines
2. Ausgangssituation und Zielsetzung	2. Projektgrundlagen
3. Funktionale Anforderungen	3. Geschäftsprozesse
4. Nichtfunktionale Anforderungen	4. Anwendungskomponenten
5. Risikoakzeptanz	5. Datenmodell
6. Lebenszyklusanalyse und Gesamtsystemarchitektur	6. Benutzerschnittstelle
7. Schnittstellenübersicht	7. Druckausgaben
8. Lieferumfang	8. Nachbarschnittstellen
9. Abnahmekriterien	9. Nichtfunktionale Anforderungen
10. Anforderungsverfolgung zu den Anforderungen (Lastenheft)	10. Querschnittskonzepte
11. Anforderungsverfolgung	11. Betriebliche Anforderungen
12. Abkürzungsverzeichnis	12. Offene Punkte
13. Literaturverzeichnis	13. Anhang
14. Abbildungsverzeichnis	

Figure 1. Abbildung der Spezifikationsmethodik auf das V-Modell XT

Für die Bezeichnung von Elementen der Systemspezifikation wird folgende Konvention verwendet, um den Typ des Elements kenntlich zu machen:

<*Elementtypkürzel*><Name_des_Elements>_

Der Typ des Elements wird in gekürzter Form dem Elementnamen vorangestellt. Der Elementname wird mit Unterstrichen statt mit Leerzeichen notiert (z.B. DIA_Personalien_ändern). Durch diese Schreibweise wird die spätere Durchsuchbarkeit und eindeutige Identifikation der Elemente gewährleistet.

Jedes Element wird im Spezifikationsdokument in einem Kapitel mit einer Überschrift im folgenden Format notiert:

<*Elementtyp*> <*Elementtypkürzel*><Name_des_Elements>_

Zum Beispiel „Dialog DIA_Personalien_ändern“.

Die Gliederung der Überschriften soll dem Muster aus diesem Dokument entsprechen. Die Inhalte der jeweiligen Kapitel werden nachfolgend beschrieben.

Für die Erstellung der UML-Diagramme, die in diesem Dokument als Beispiele aufgeführt sind, wurde das Werkzeug „Enterprise Architect“ der Firma SparxSystems verwendet. Abschnitte in dieser Darstellung erklären, wie das jeweils davor abgebildete Diagramm im Enterprise Architect zu erstellen ist.

Anstelle des Enterprise Architect kann auch jedes andere Modellierungswerkzeug eingesetzt werden, dass den Sprachumfang von UML in hinreichender Breite abdeckt.

Falls Sie den Enterprise Architect einsetzen, können Sie Kapitel 14 generelle Hinweise entnehmen, wie dieser zu verwenden ist. **Bitte dieses Kapitel zuerst lesen!** Wichtig sind insbesondere die zu verwendenden UML-Profile, die zu verwendende Paket-Struktur und die Tipps zum Umgang.

Allgemeines

Zusammenfassung (Management-Summary)

Unter diesem Kapitel soll eine Zusammenfassung des Dokuments auf wenigen Seiten erfolgen.

Beispiel: Dieses Dokument enthält Ausfüllhinweise für die Erarbeitung von Systemspezifikationen nach dem V-Modell XT in einer für den Kunden angepassten Fassung. Es handelt sich um Empfehlungen, die im Einzelfall in Absprache mit dem Kunden abgewandelt werden können. Abweichungen, insbesondere in der Dokumentstruktur sind jedoch in Kapitel 1.2 anzugeben und mit einer Begründung zu versehen.

Leseanleitung

Die Leseanleitung enthält Hilfen für den Leserkreis. Inhalte sind zum Beispiel

- *Aufbau der Dokumentation (Aufteilung in Dokumente, interne Struktur)*
- *Beschreibung der verwendeten Methodik*
- *Empfehlungen für verschiedene Lesergruppen*
- *Leseanleitung zu verwendeten Notationen (z.B. UML 2.0).* == Projektgrundlagen

Ziele und Rahmenbedingungen

Die zentralen Ziele und Rahmenbedingungen bilden das einleitende Kapitel einer Spezifikation. Damit wird ein inhaltlicher Einstieg in die Spezifikation gegeben. Es werden die folgenden Themen behandelt:

- *Zweck und Ziele des Systems bzw. des Projektes*
- *Ausgrenzungen bezüglich des Projektumfangs*
- *Projektbeteiligte und Nutzer des Systems*

- *Vorgaben und Rahmenbedingungen des Projektes (fachlich, technologisch, organisatorisch, politisch, gesetzlich, etc.)*
- *Kurzer Abriss der Projekthistorie*
- *Bedeutende Projektrisiken*

Fachlicher Architekturüberblick

Der Architekturüberblick beschreibt die Einordnung der zu entwickelnden Anwendung in die Anwendungslandschaft des Auftraggebers aus fachlicher Sicht. Er identifiziert die Nachbarsysteme und stellt deren Schnittstellen dar. Die Anwendung wird grob in Anwendungskomponenten gegliedert, wobei sich diese Gliederung hier ausschließlich an der Fachlichkeit orientiert.

Der Architekturüberblick sollte knapp und klar gestaltet werden. Ein gutes Übersichtsbild zeigt die Einordnung der Anwendung in die bestehende Anwendungslandschaft und kann einen Eindruck von der Komplexität des Vorhabens aufzeigen.

Im Rahmen der Masterspezifikation wird ein Architekturüberblick der gesamten Anwendungslandschaft gepflegt, in die sich die Anwendung einordnet. Dieser ist nach Inbetriebnahme zu aktualisieren.

Geschäftsprozesse

Die Spezifikation der Geschäftsprozesse beschreibt die fachlichen Abläufe, in welche die zu erstellende Anwendung eingebettet ist. Sie dienen dazu, die Systemgrenzen festzulegen, sowie den Kontext der Systembenutzung zu identifizieren und zu verstehen. Ein Geschäftsprozess wird oft nicht durch eine einzelne Anwendung, sondern erst durch das Zusammenspiel verschiedener Anwendungen beim Kunden und den an seine Anwendungen angeschlossenen externen Anwendungen umgesetzt. Entsprechend kann es sinnvoll sein, das Geschäftsprozessmodell als eigenes Dokument unabhängig von der restlichen Systemspezifikation zu notieren.

Geschäftsprozessmodelle helfen dem Auftragnehmer, den geschäftlichen Kontext des Auftraggebers zu verstehen. Für den Auftraggeber sind sie eine gute Basis, um Chancen für Prozessoptimierungen zu erkennen. Dieses Verständnis ist besonders dann wichtig, wenn bei einem Ablauf Objekte (d.h. physische Dinge und Daten) zwischen Personen und Organisationseinheiten weitergegeben werden sollen und dieser Ablauf durch eine Anwendung unterstützt wird. Ist das Verständnis über diesen Ablauf bisher nicht vorhanden (im Team des Auftragnehmers und beim Auftraggeber), dient die Geschäftsprozessmodellierung dem Erkenntnisgewinn. Sind die Geschäftsprozesse bereits an anderer Stelle modelliert und beschrieben, wird hier diese Stelle referenziert.

Die wesentlichen Bestandteile der Spezifikation der Geschäftsprozesse sind Organigramm und Prozessbeschreibung. Ist- und Soll-Prozesse sollen getrennt beschrieben werden.

Die folgenden Fehler beim Einsatz der Geschäftsprozessmodellierung müssen vermieden werden:

Falscher Einsatzbereich

Für isolierte "Trivialanwendungen" (Datenpflege, Reporting) bringen Geschäftsprozessmodelle keine weiteren Erkenntnisse. Hier reicht es aus, die Schritte zu benennen und (später) als

Anwendungsfälle kurz zu beschreiben. Eine grafische Darstellung des Ablaufes in Form von Geschäftsprozessmodellen ist überflüssig.

Zu große Detaillierung

Die Abfolge von Eingaben und Masken an der Benutzeroberfläche sollte kein Inhalt des Geschäftsprozessmodells sein. Sie wird im Baustein Dialogspezifikation textuell oder als Interaktionsdiagramm beschrieben.

Überspezifikation

Beschreibung von explizit modellierten Sachverhalten in Beschreibungstexten oder umgekehrt. Ein häufiges Beispiel für Überspezifikation (und Verletzung der Unabhängigkeit von der Implementierung) sind Bezeichner für Aktivitäten, die die assoziierten Anwendungen enthalten.

Mangelnde Fokussierung

Fachbereiche können den Anspruch haben, sämtliche Stützprozesse detailliert zu modellieren. Dies bringt aber für die Analyse der Kernprozesse häufig keinen Mehrwert, sondern im Gegenteil, die Stützprozesse machen das Prozessmodell unübersichtlicher. Zum Beispiel lohnt es sich nicht, die Aktivitäten für Posteingang, Hauspost und Aktenhaltung zu modellieren – es sein denn, es geht um die Einführung eines Dokumenten-Management-Systems.

Die Modellierung von Geschäftsprozessen verfolgt immer ein konkretes Ziel. Dies ist meist die Feststellung von Optimierungsbedarfen bezüglich der Organisation der Abläufe oder der IT-Unterstützung. Je nach Ziel des Projekts muss im Vorfeld festgelegt werden, welche Elemente im Modell enthalten sein müssen und welche nicht für die Erreichung des Ziels notwendig sind. So ist für die Feststellung eines IT-Optimierungspotenzials die Modellierung von Anwendungen essenziell, weniger aber die Modellierung von Akteuren oder Wahrscheinlichkeiten für das Durchlaufen einzelner Zweige im Modell.

Die Abläufe innerhalb eines Geschäftsprozesses unterteilen sich in Geschäftsvorfälle, die wiederum aus einzelnen Aktivitäten bestehen. Daraus ergibt sich für die Beschreibung die folgende Struktur des Dokuments. Auch hier muss bei der Festlegung der Ziele für die Prozessmodellierung festgelegt werden, bis zu welchem Detaillierungsgrad die Modellierung sinnvoll ist. Z.B. kann es genügen, die Geschäftsprozesse und –vorfälle textuell und grafisch zu beschreiben, die Aktivitäten hingegen nur textuell in der Beschreibung der Geschäftsvorfälle zu definieren.

Die Geschäftsprozesse werden in der Masterspezifikation als eigenes Dokument gepflegt. Änderungen an den Geschäftsprozessen werden spätestens mit der Übernahme der Systemspezifikation in die Masterspezifikation nachgepflegt.

Gesamtverfahren <Name des Gesamtverfahrens>

Ein Kontextdiagramm zeigt alle Geschäftsprozesse mit beteiligten Organisationseinheiten und Anwendungen (UML-Aktivitätendiagramm).

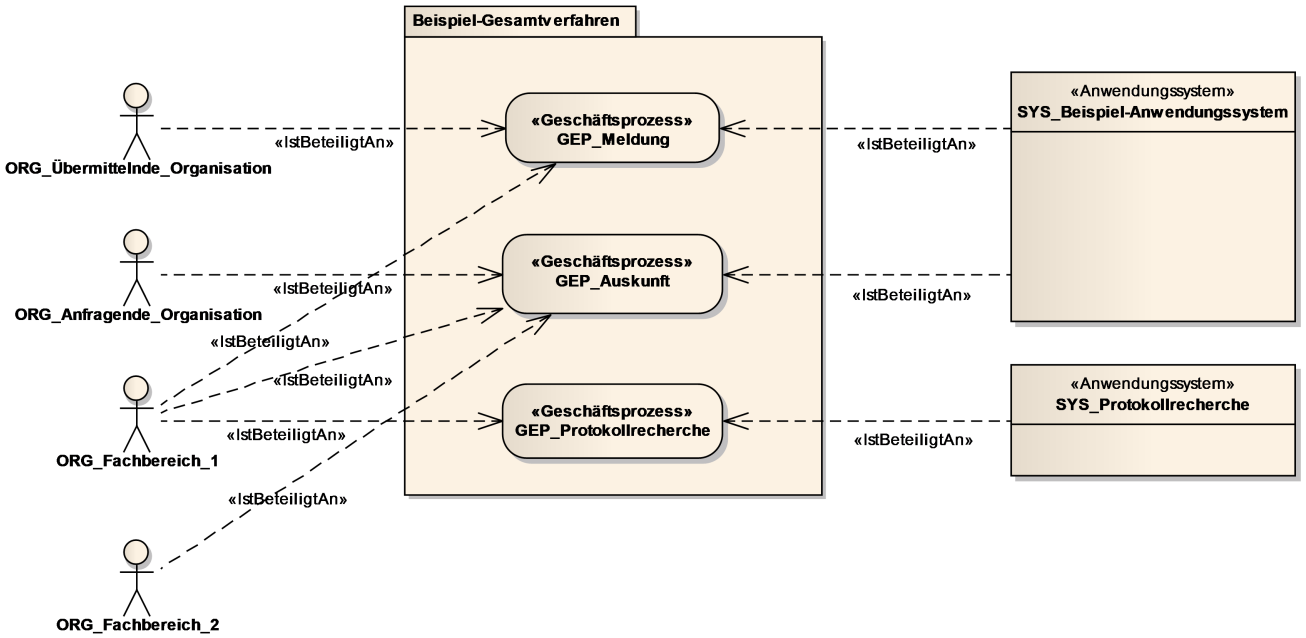


Figure 2. Gesamtverfahren <Name des Gesamtverfahrens>

Erstellung des Diagramms im Enterprise Architect

Man zieht Organisationseinheiten aus dem UML-Profil „Organisation“ in ein Analysis-Diagramm und benennt sie mit Präfix „ORG_“. Dann zieht man den Ordner des Gesamtverfahrens in das Diagramm und blendet seine Inhalte aus. Man zieht neue Geschäftsprozesse und Anwendungen aus dem UML-Profil „Prozesse“ in das Diagramm und benennt sie mit Präfix „GEP_“ bzw. „SYS_“. Schließlich nutzt man den Verbinder IstBeteiligtAn, um die Organisationseinheiten und die Systeme mit den Geschäftsprozessen zu verbinden.

Geschäftsprozess GEP_<Bezeichnung des Geschäftsprozesses>

Ein Geschäftsprozess ist eine funktions- und stellenübergreifende Folge von Arbeitsschritten zur Erreichung eines geplanten Arbeitsergebnisses in einer Organisation (Unternehmen, Behörde, etc.). Er dient direkt oder indirekt zur Erzeugung einer Leistung für einen Kunden oder den Markt. Ein Geschäftsprozess kann sich aus Aufgaben im Sinn von elementaren Tätigkeiten (Aktivitäten) zusammensetzen. Diese Aktivitäten werden zu Geschäftsvorfällen zusammengefasst, welche die Bestandteile des Geschäftsprozesses bilden.

Als zu verwendende Namenskonvention beginnen alle Geschäftsprozessnamen mit dem Präfix „GEP_“, gefolgt von einem Substantiv und ggf. einem Unterstrich und einem Verb. Bei der Wahl des Namens ist darauf zu achten, dass dieser kurz und prägnant ist.

Der Geschäftsprozess wird textuell beschrieben, wobei das Ziel, die beteiligten Organisationseinheiten und der Ablauf benannt werden. Der Ablauf wird zusätzlich grafisch in einem Geschäftsprozessdiagramm (englisch Business Process Diagram - BPD) in der Geschäftsprozessmodellierungsnotation (englisch Business Process Modeling Notation - BPMN) dargestellt.

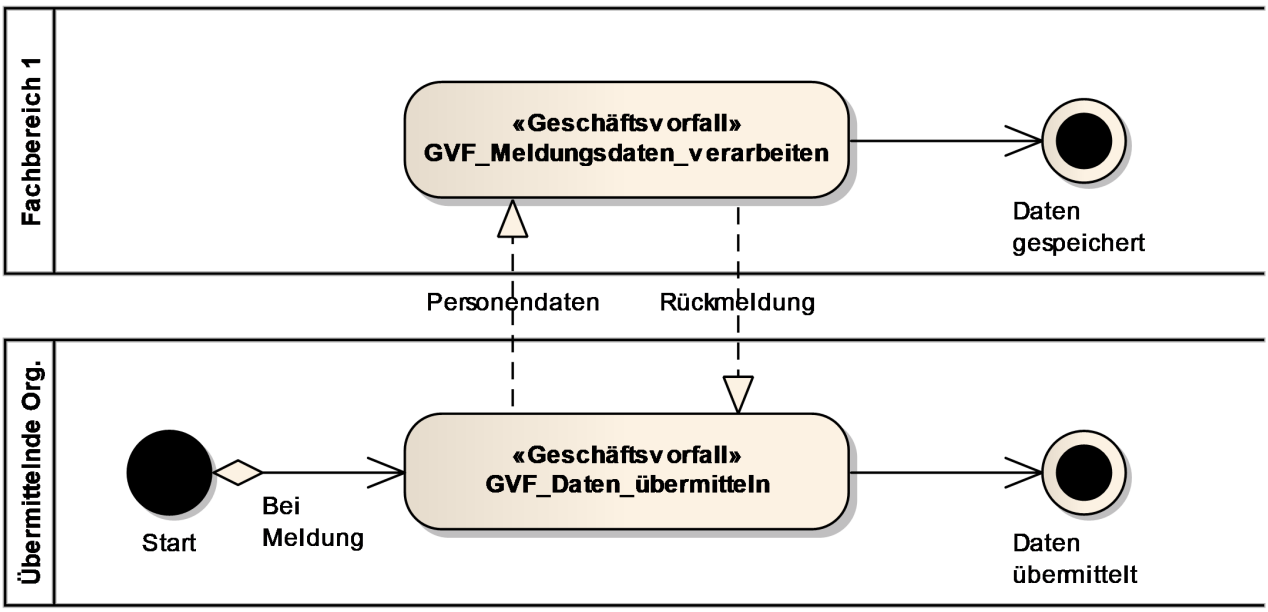


Figure 3. Geschäftsprozess: GEP_<Bezeichnung des Geschäftsprozesses>

Erstellung des Diagramms im Enterprise Architect

Die am Geschäftsprozess beteiligten Organisationseinheiten zieht man als „Pools“ aus dem UML-Profil „Prozesse“. Die Geschäftsvorfälle, Start-, Ende- und Verzweigungselemente stammen aus demselben UML-Profil. Zur Darstellung des Ablaufs der Geschäftsvorfälle je Organisationseinheit wird die Verbindung „Ablauf“ genutzt. Der Verbinder „Nachricht“ wird genutzt, um zu zeigen, welche Nachrichten zwischen den Organisationseinheiten ausgetauscht werden. Falls ein Geschäftsvorfall nur unter bestimmten Bedingungen ausgeführt wird, wird dies durch die Auswahl von „Source Role > Aggregation > Shared“ und der Angabe eines Namens in dessen Eigenschaften gezeigt. Bei Bedarf können zusätzlich Verzweigungselemente genutzt werden.

Geschäftsvorfall: GVF_<Bezeichnung des Geschäftsvorfalls>

Ein Geschäftsvorfall ist die Bündelung elementarer Tätigkeiten (Aktivitäten) innerhalb eines Geschäftsprozesses, die durch ein Ereignis ausgelöst und dann lückenlos innerhalb einer Organisationseinheit bearbeitet werden.

Als zu verwendende Namenskonvention beginnen alle Geschäftsvorfallnamen mit dem Präfix „GVF_“, gefolgt von einem Substantiv, einem Unterstrich und einem Verb. Bei der Wahl des Namens ist darauf zu achten, dass dieser kurz und prägnant ist.

Der Geschäftsvorfall wird textuell beschrieben, wobei das Ziel, die beteiligten Organisationseinheiten und der Ablauf benannt werden. Der Ablauf wird zusätzlich grafisch in einem Geschäftsprozessdiagramm in der Geschäftsprozessmodellierungsnotation dargestellt.

Es werden nur diejenigen Geschäftsvorfälle beschrieben, die mindestens teilweise vom Auftraggeber durchgeführt werden.

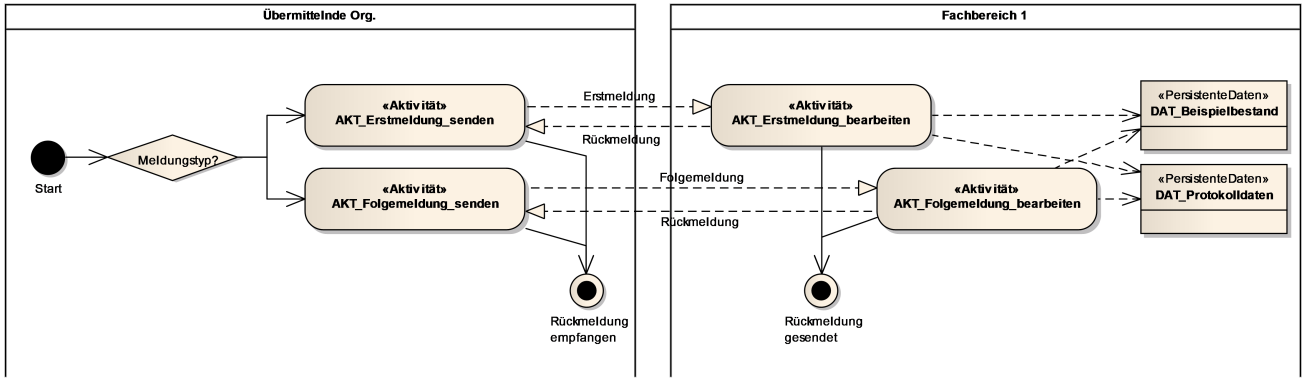


Figure 4. Geschäftsvorfall GVF_Meldung_durchführen

Erstellung des Diagramms im Enterprise Architect

Man baut das Diagramm analog zum Geschäftsprozess-Diagramm auf, fügt aber Aktivitäten statt Geschäftsvorfälle hinzu. Die Pools stellt man vertikal dar über ihr Kontextmenü mit „Advanced > Vertical Partition“. Persistente Daten und Dokumente fügt man aus dem UML-Profil „Prozesse“ hinzu.

Geschäftsvorfall	
Kurzbeschreibung	*Zwei bis drei Sätze zur Beschreibung des Ziels, des Ablaufs und der beteiligten Organisationseinheiten (Rolle) des Geschäftsvorfalls.*
Vorbedingungen/ auslösendes Ereignis	*Die Vorbedingungen beschreiben den Zustand aller relevanten und nichttrivialen Voraussetzungen, die erfüllt sein müssen, damit der Geschäftsvorfall durchgeführt werden kann. Sie werden als Punkte-Liste beschrieben. Jeder Punkt beschreibt ein Set an Vorbedingungen, welche vollständig gelten müssen.* *Auslöser für den Geschäftsvorfall sind Ereignisse wie auslösende Handlungen anderer Akteure oder Zeitpunkte.*
Nachbedingungen/ Ergebnisse	*Beschreibung des erwarteten Zustandes nach Ausführung des Geschäftsvorfalls. Es kann hier auch mehrere Nachbedingungen geben, wenn es alternative Zustände nach der Ausführung gibt, z.B. Erfolg oder Fehler.*

Aktivität AKT_<Bezeichnung der Aktivität>

Eine Aktivität ist eine Tätigkeit, die einen elementaren, logischen Schritt innerhalb eines Geschäftsvorfalls bildet. Sie wird unterbrechungsfrei von einem Akteur ausgeführt. Eine Aktivität kann sowohl manuell als auch teilweise oder vollständig automatisiert (Computer-unterstützt) ablaufen (z.B. „Meldung prüfen“).

Als zu verwendende Namenskonvention beginnen alle Aktivitätennamen mit dem Präfix „AKT_“, gefolgt von einem Substantiv, einem Unterstrich und einem Verb.

Die Aktivität wird gemäß der nachfolgenden Tabelle textuell beschrieben.

Es werden nur diejenigen Aktivitäten beschrieben, die vom Auftraggeber durchgeführt werden.

Aktivität	
Organisationseinheit	*Beteiligte Organisationseinheit oder Rolle.*
Kurzbeschreibung	*Zwei bis drei Sätze zur Beschreibung des Ziels und des Ablaufs der Aktivität.*
Vorbedingungen/ auslösendes Ereignis	*Die Vorbedingungen beschreiben den Zustand aller relevanten und nichttrivialen Voraussetzungen, die erfüllt sein müssen, damit die Aktivität durchgeführt werden kann. Sie werden als Punkte-Liste beschrieben. Jeder Punkt beschreibt einen Satz an Vorbedingungen, der vollständig gelten muss.* *Auslöser für die Durchführung der Aktivität sind Ereignisse wie auslösende Handlungen anderer Akteure oder Zeitpunkte.*
Nachbedingungen/ Ergebnisse	*Beschreibung des erwarteten Zustandes nach Ausführung der Aktivität. Wenn möglich Verweise auf erzeugte persistente Daten oder Dokumente. Es kann hier auch mehrere Nachbedingungen geben, wenn es alternative Zustände nach der Ausführung gibt, z.B. Erfolg oder Fehler.*
Automatisierungsgrad	*Inwieweit wird die Aktivität durch Anwendungen unterstützt? Mögliche Ausprägungen sind „vollautomatisiert“, „teilautomatisiert“ und „manuell“.*
Beteiligte Systeme	*Beteiligte Anwendungen, wenn die Aktivität nicht manuell durchgeführt wird.*
Verwendete Anwendungsfälle	*Hier werden alle Anwendungsfälle als Spiegelstrichaufzählung aufgelistet, die die Aktivität umsetzen. Bei Beteiligung mehrerer Anwendungen werden die Anwendungsfälle den Systemen zugeordnet.*

Aktivität AKT_<Bezeichnung der Aktivität>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Aktivitäten des Geschäftsvorfalls zu beschreiben sind.

Geschäftsvorfall GVF_<Bezeichnung des Geschäftsvorfalls>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Geschäftsvorfälle und dazu gehörende Aktivitäten des Geschäftsprozesses zu beschreiben sind.

Geschäftsprozess GEP_<Bezeichnung des Geschäftsprozesses>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Geschäftsprozesse und dazu gehörende Geschäftsvorfälle und Aktivitäten zu beschreiben sind.

Dokumente

Ein Dokument ist ein in Papierform oder elektronisch vorliegendes Schriftstück, das innerhalb der Geschäftsprozesse des Gesamtverfahrens genutzt oder erstellt wird.

Dokument DOK_<Bezeichnung des Dokuments>

In diesem Abschnitt wird ein Dokument des Gesamtverfahrens beschrieben.

Dokument DOK_<Bezeichnung des Dokuments>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Dokumente des Gesamtverfahrens zu beschreiben sind.

Persistente Datenhaltung

Die verschiedenen Datenbestände des Gesamtverfahrens enthalten die Daten, die zur Ausführung der Geschäftsprozesse dauerhaft gespeichert werden. Eine Trennung der Daten in verschiedene Datenbestände kann durch die Aufteilung in verschiedene Organisationseinheiten oder unterschiedliche Zwecke der Datenhaltung begründet sein. Hier sollte neben der Beschreibung jedes Datenbestands auch eine grobe Mengenabschätzung der Größe des Datenbestands notiert werden.

Persistente Daten DAT_<Bezeichnung des Datenbestands>

In diesem Abschnitt wird ein Datenbestand des Gesamtverfahrens beschrieben.

Persistente Daten DAT_<Bezeichnung des Datenbestands>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere persistente Daten des Gesamtverfahrens zu beschreiben sind.

Organisationseinheiten

Verschiedene Organisationseinheiten nehmen verschiedene Rollen im Ablauf eines Geschäftsprozesses wahr. Aktivitäten eines Geschäftsprozesses können von den beteiligten Organisationseinheiten manuell (d.h. durch Personen), automatisiert (d.h. im Auftrag der Organisationseinheiten durch Anwendungen) oder teilautomatisiert durchgeführt werden. Dieselbe Organisationseinheit kann in verschiedenen Gesamtverfahren auftreten. Hier werden diese Organisationseinheiten mit ihren Aufgaben im Verfahren sowie ihrer Rolle im Geschäftsprozess beschrieben.

Ein Organigramm (UML-Komponentendiagramm) und eine textuelle Beschreibung geben einen Überblick über die beteiligten Organisationseinheiten (Rollen).

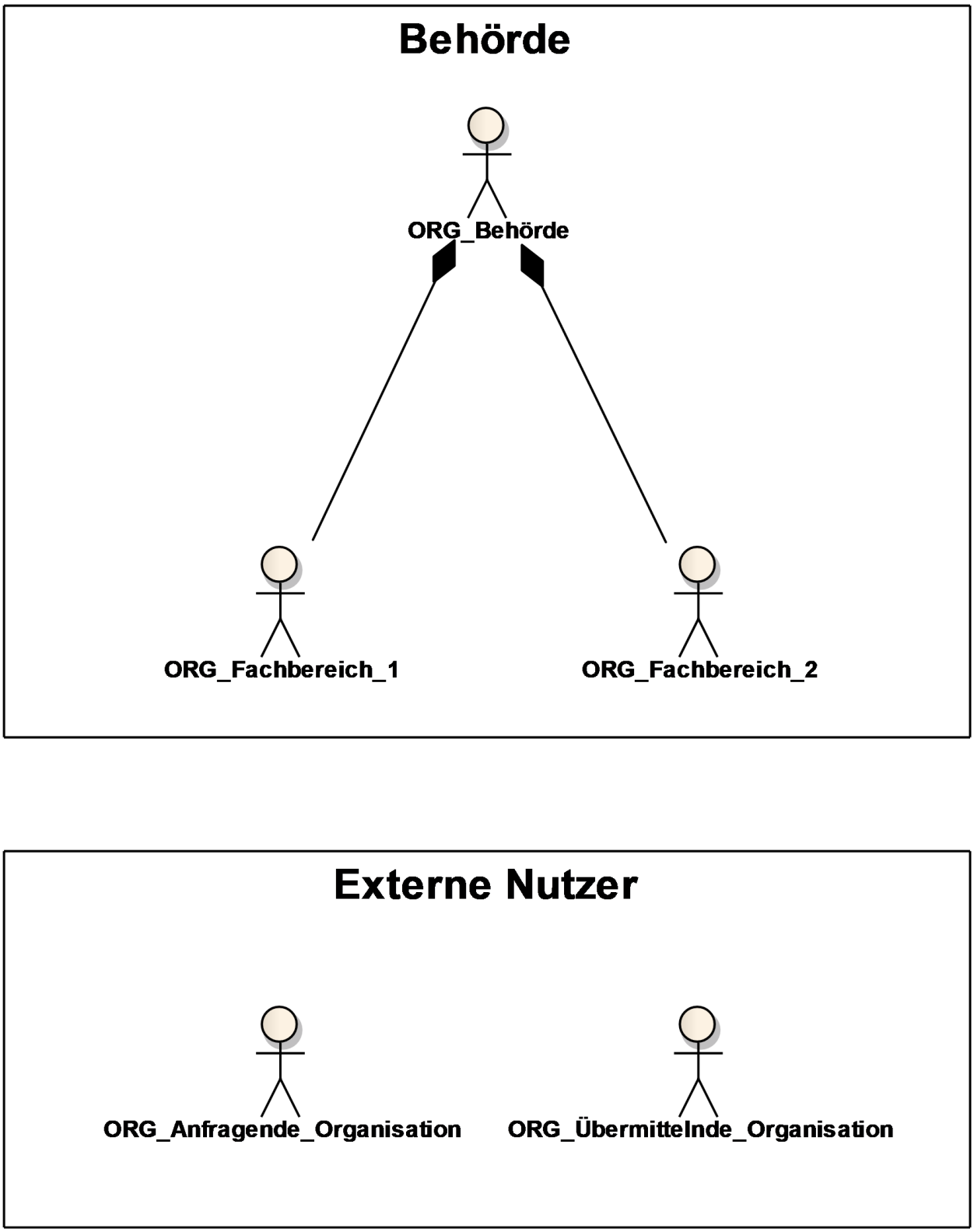


Figure 5. Organigramm von <abc>

Erstellung des Diagramms im Enterprise Architect

Man fügt alle Organisationseinheiten aus dem UML-Profil „Organisation“ in ein Use Case Diagramm ein und ordnet sie an. Zusammengehörige Organisationseinheiten erhalten eine gemeinsame Boundary mit passendem Namen und Schriftart Arial Bold 14. Man nutzt die

Verbinder aus demselben UML-Profil, um Organisationsstrukturen darzustellen.

Organisationseinheit ORG_<Bezeichnung der Organisationseinheit>

In diesem Abschnitt wird eine Organisationseinheit mit ihren Aufgaben im Verfahren sowie ihrer Rolle im Geschäftsprozess beschrieben.

Organisationseinheit ORG_<Bezeichnung der Organisationseinheit>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Organisationseinheiten zu beschreiben sind.

Anwendungen

Anwendungen unterstützen teilautomatisierte und automatisierte Aktivitäten. Anwendungen werden in fachliche Domänen unterteilt, die z.B. nach den Zuständigkeiten der Fachbereiche geordnet sein können. In diesem Abschnitt werden die Domänen textuell erklärt und grafisch die Anwendungen in die Domänen eingeordnet.

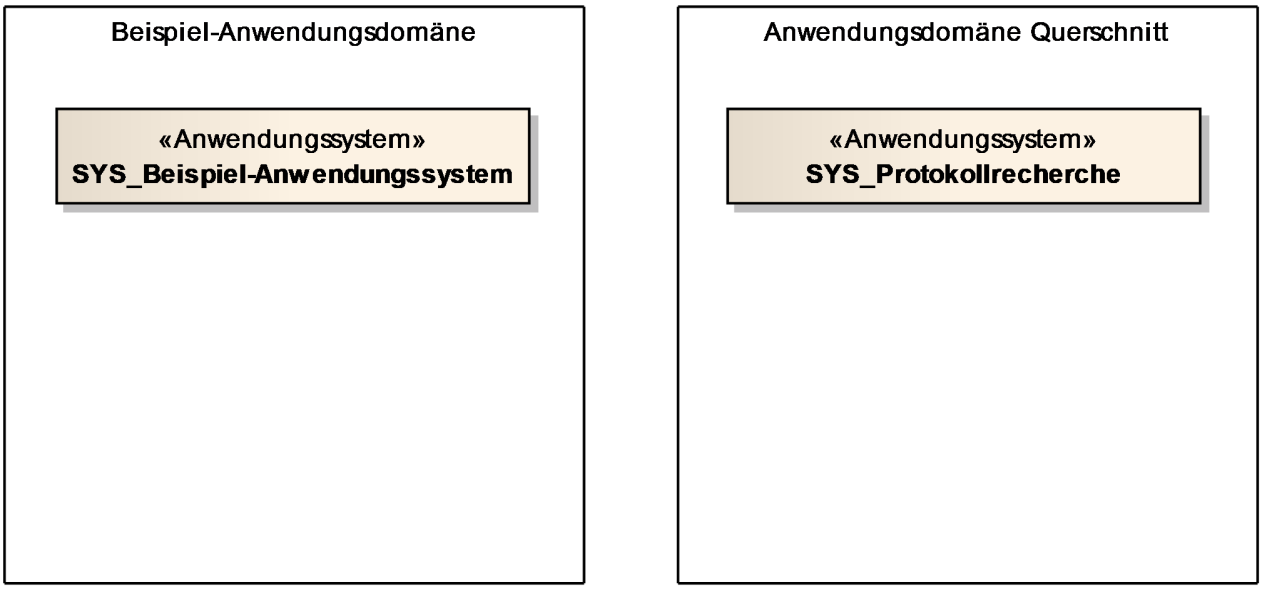


Figure 6. Anwendungen und ihre Domänenzugehörigkeit

Erstellung des Diagramms im Enterprise Architect

Alle Anwendungen werden in ein Use Case Diagramm gezogen. Dann werden sie nach Domänen geordnet und jede Domäne wird als Boundary mit Namen eingezeichnet.

Anwendung SYS_<Bezeichnung der Anwendung>

In diesem Abschnitt wird eine Anwendung beschrieben. Seine fachliche Zielsetzung wird beschrieben.

Anwendung SYS_<Bezeichnung der Anwendung>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Anwendungen zu beschreiben sind.

Anwendungskomponenten

In einer Systemspezifikation wird eine fachliche Anwendung spezifiziert. Die fachlichen Abläufe der Anwendung werden in Form von Anwendungsfällen beschrieben. Die Anwendungsfälle werden zu Anwendungskomponenten gruppiert und in entsprechenden Abschnitten unterschieden. Vorab erfolgt hier ein Überblick über alle Anwendungskomponenten in einem UML-Komponentendiagramm mit einer kurzen textlichen Erläuterung, welche Abläufe von den gezeigten Komponenten unterstützt werden.

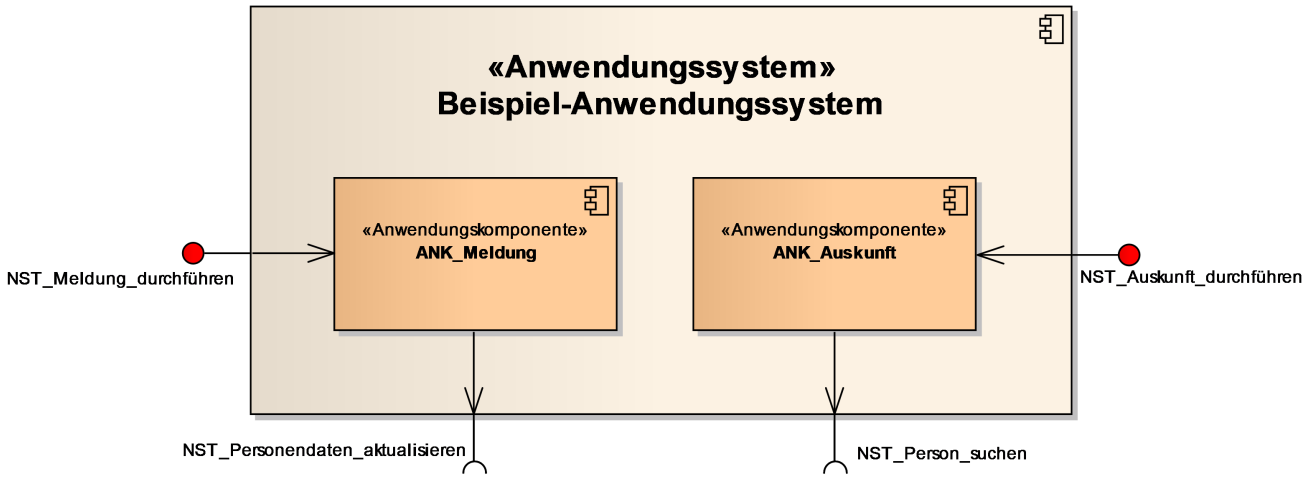


Figure 7. Übersicht über die Anwendungskomponenten

Erstellung des Diagramms im Enterprise Architect

Die Anwendung wird mit dem UML-Profil-Element „Anwendungskomponente“ dargestellt. Hierzu wird der Stereotyp von „Anwendungskomponente“ auf „Anwendung“ geändert. Im Diagramm wird die Schriftgröße auf Arial Bold 14 geändert. Die Anwendung wird mit dem Präfix „SYS_“ benannt.

Die angebotenen Schnittstellen werden als „Provided Interface“ dargestellt, die genutzten als „Required Interface“. Hierzu wählt man im Kontextmenü des Anwendung-Elements „Embedded Elements > Add Provided Interface“ bzw. „Embedded Elements > Add Required Interface“ aus. Die angebotenen Schnittstellen kann man farblich markieren: Solche, die ausschließlich innerhalb der Anwendungslandschaft des Kunden genutzt werden, sind rot markiert. Solche, die auch oder nur von externen Anwendungen aufgerufen werden, sind grün markiert.

Angebotene Schnittstellen verknüpft man anschließend mit den Anwendungskomponenten, deren Anwendungsfälle aufgrund der Schnittstellen-Nutzung aufgerufen werden. Hierzu nutzt man die Verknüpfung „composite > delegate“ aus der UML-Toolbox des EA. Genutzte Schnittstellen kann man mit den nutzenden Anwendungskomponenten verbinden. Für

Schnittstellen, die querschnittlich von allen Anwendungskomponenten genutzt werden, ist dies nicht notwendig. Mit rechter Maustaste auf den Text "delegate" und anschließend „hide label“ versteckt man die Anzeige des Stereotypen.

Akteure

Das Kapitel enthält als Überblick ein Diagramm aller Akteure und deren Abhängigkeiten.

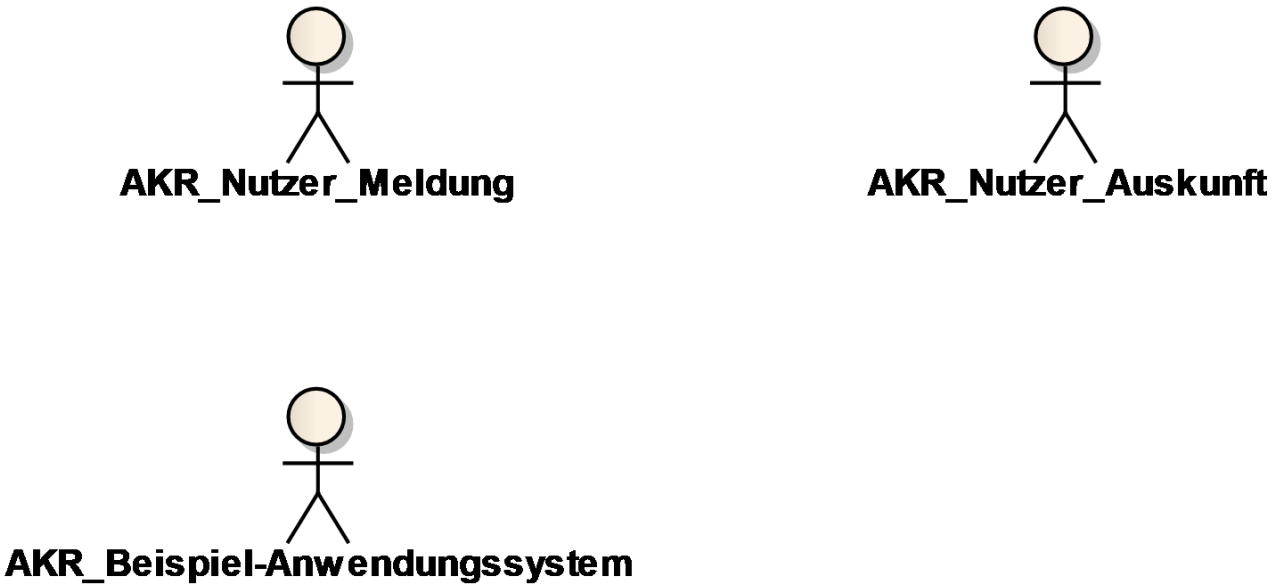


Figure 8. Übersicht Akteure

Zusätzlich werden in einer Liste alle Akteure benannt und kurz in Stichworten beschrieben. Die Namen der Akteure sind so eng wie möglich angelehnt an die Bezeichnungen, die in den Fachbereichen etabliert sind. Oft ist es zweckmäßig, auch die Beziehungen zwischen den Akteuren zu verdeutlichen. Die Liste der Akteure muss vollständig sein, allerdings nur für die zu spezifizierende Anwendung.

Als Akteure können auftreten:

- *Benutzer der Anwendung, geordnet nach ihrer Organisationszugehörigkeit oder Rolle,*
- *andere Anwendungen, die angebotene Schnittstellen der Anwendung nutzen,*
- *die Anwendung selbst, wenn es seine Batches oder andere automatisierte Prozesse ausführt.*

Man sollte in der Wahl der Akteure menschliche Benutzer, welche die Anwendung indirekt nutzen, bevorzugen gegenüber anderen, vorgeschalteten Anwendungen.

Namen von Akteuren beginnen mit dem Präfix „AKR“ gefolgt vom Klarnamen mit Unterstrichen statt Leerzeichen._

Table 1. Akteure von <System>

Name	Ziel
Name des Akteurs	*Ziel des Akteurs*

Anwendungskomponente ANK_<Bezeichnung der Anwendungskomponente>

Am Anfang jeder Anwendungskomponentenbeschreibung steht ein Überblick über die Anwendungsfälle. Er besteht aus der Liste der Anwendungsfälle und einem Anwendungsfalldiagramm.

Das Anwendungsfalldiagramm ist ein grafischer Überblick über die Anwendungsfälle. Es bietet oft einen schnelleren Überblick als die Liste der Anwendungsfälle. Zur übersichtlichen Darstellung kann ein Anwendungsfalldiagramm ausschließlich mit den Akteuren verwendet werden.

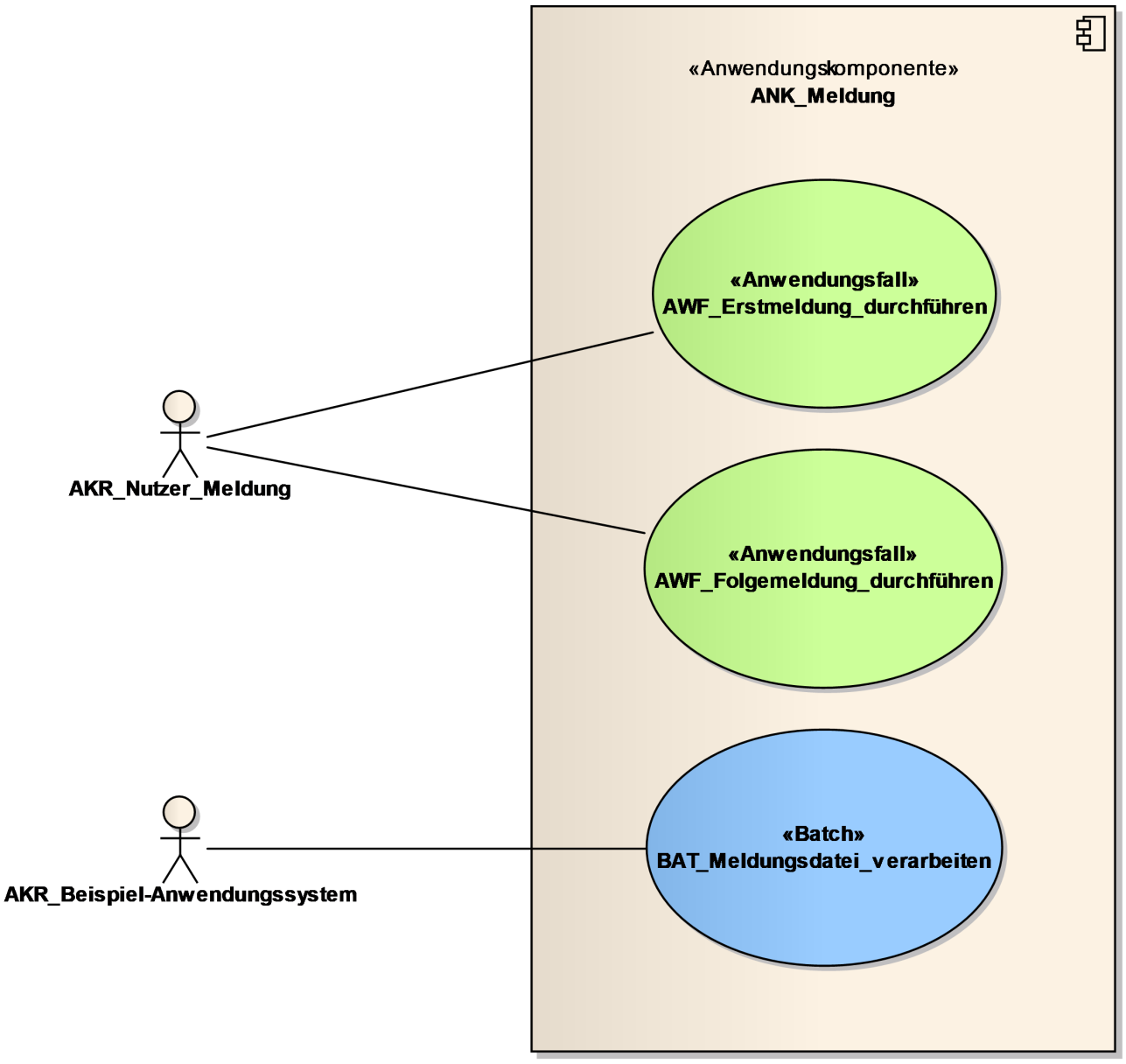


Figure 9. ANK_Meldung

Erstellung des Diagramms im Enterprise Architect

Die Klammer um alle Anwendungsfälle und Batches bildet die Anwendungskomponente. Deren Element zieht man aus dem UML-Profil „Anwendungsfälle“ ins Diagramm. Die Akteure ordnet man außerhalb der Komponente an und verknüpft sie über den Verbinder „FührtAnwendungsfallAus“ mit den Anwendungsfällen. Die Stereotypen an den Verbindern blendet man aus.

Die textuell aufgezählten Elemente sollen als Verweise dienen. Ihre Reihenfolge ist alphabetisch. Man muss sie in dieser und den nachfolgenden Stellen, und am besten auch im Text, mit Hyperlink auf die zugehörige Überschrift versehen. Hier könnte ein Word-Makro hilfreich sein.

Anwendungsfall AWF_<Bezeichnung des Anwendungsfalls>

Die Anwendungsfälle (engl. Use-Cases) spezifizieren die Funktionalität einer Anwendung. Ein Anwendungsfall beschreibt das Verhalten und die Interaktion einer Anwendung als Reaktion auf die zielgerichtete Anfrage oder Aktion eines Akteurs über eine Schnittstelle oder im Dialog. Anwendungsfälle enthalten eine textuelle Beschreibung der Außensicht der Anwendung in Form von Abläufen. Die Beschreibung erfolgt in der Sprache der Anwender.

Mögliche Fehlerquellen beim Schreiben von Anwendungsfällen sind:

Zu frühe Betrachtung von Sonderfällen

Es ist wichtig, dass der Standardablauf eines Anwendungsfalls deutlich erkennbar ist. Das kann man z.B. dadurch erreichen, dass man erst einmal nur diesen mit durchnummerierten Schritten beschreibt und in einem anschließenden Abschnitt "Alternative Abläufe" die Ausnahmen bezogen auf die einzelnen Schritte des Standardablaufs beschreibt.

Zu detaillierte Beschreibung der Anwendungsfälle

Insbesondere die detaillierte Beschreibung des Ablaufes der Dialoge. Anwendungsfälle sollten das fachliche Verhalten der Anwendung ohne konkrete Referenz auf die Benutzeroberfläche beschreiben.
Beispiel:
Nicht: „Der Anwender öffnet den Dialog … Er erfasst … Er betätigt die ok-Taste. Die Anwendung zeigt die Maske … mit …“
Sondern: „Der Anwender erfasst die Daten des Vertrages. Daraufhin berechnet die Anwendung die Raten und zeigt sie zur Kontrolle an“.

Die Dialoge werden in der Dialogspezifikation explizit beschrieben. Die Beschreibung der Anwendungsfälle sollte nicht mit der Spezifikation der Dialoge vermischt werden, da nicht unbedingt eine 1:1-Beziehung Anwendungsfall zu Dialog besteht._

Funktionale Zerlegung mit Anwendungsfällen

Beim Versuch einer möglichst redundanzfreien Beschreibung wird oft versucht, Anwendungsfälle funktional zu zerlegen. Die Anwendungsfälle, die dabei entstehen, sind im Allgemeinen zu klein und zu trivial, da der Gesamtzusammenhang verloren geht. Ein gutes Maß für die Granularität des Anwendungsfalles ist die Frage: „Bringt dieser Anwendungsfall ein für den Benutzer sichtbares

Ergebnis?“ Um umfangreiche oder mehrfach verwendete Funktionalität zu kapseln, sollten stattdessen Anwendungsfunktionen verwendet werden.

Mehrdeutig formulierte Anwendungsfälle

Bei der Beschreibung der Anwendungsfälle auf einen eindeutigen Satzbau achten. Das heißt:

- *Füllwörter wie eventuell, gegebenenfalls, und/oder vermeiden*
- *Passivsätze vermeiden*
- *Genaue Formulierungen benutzen: Wer tut wann was? Welche Hilfsmittel benutzt er dazu?*

```
Beispiel:
Nicht: Die Parameter für die Personensuche werden eingegeben und die
       zugehörige Person ermittelt. Gegebenenfalls müssen die
       Suchparameter erneut eingegeben werden.
Sondern: Standardablauf:
1. Der Benutzer erfasst die Parameter für die Personensuche und
   bestätigt die Eingabe.
2. Die Anwendung prüft die Eingabeparameter auf Vollständigkeit und
   Plausibilität.
3. Die Anwendung ermittelt die Treffermenge zu den Suchparametern.
4. Die Anwendung zeigt dem Benutzer die Treffermenge an.
1. Alternativer Ablauf:
  2a. Die Suchparameter sind nicht vollständig.
      Die Anwendung fordert den Benutzer auf, die Parameter zu ergänzen.
2. Alternativer Ablauf:
  2a. Die Suchparameter sind nicht plausibel.
      Die Anwendung fordert den Benutzer auf, die Parameter zu
      korrigieren.
```

Allgemein gültige Plausibilisierungen und Geschäftsregeln können im Datenmodell oder der Datentypbeschreibung hinterlegt werden. Sie müssen in der Anwendungsfallbeschreibung nicht berücksichtigt werden, was die Anwendungsfallbeschreibung kompakter macht. Bestimmte Plausibilisierungen sind auch am besten in der Dialog-Spezifikation (Abschnitt „Dialoge“) aufgehoben. Bei der Verteilung von Plausibilisierungen auf mehrere Spezifikationsteile müssen an zentraler Stelle Hinweise erfolgen und klare Kriterien genannt werden. Dafür bietet sich der Abschnitt „9.7 Querschnittskonzepte“ an.

Namen von Anwendungsfällen beginnen mit dem Präfix „AWF_“ gefolgt von einem Substantiv und einem Verb, z.B. „AWF_Visumantrag_prüfen“. Falls nötig kann noch ein Adjektiv vor das Substantiv gestellt werden. Der Titel ist ein eindeutiger Bezeichner des Anwendungsfalls. Er sollte so formuliert sein, dass er möglichst prägnant Hinweise auf Akteur und Ziel gibt.

Anwendungsfall	
Kurzbeschreibung	*Zusammenfassung des Ablaufs mit Ziel des Anwendungsfalls in wenigen Sätzen. Das Ziel ist die Absicht und der Grund, weshalb der Akteur den Anwendungsfall überhaupt anstößt.*

Anwendungsfall	
Akteure	*Rollen (von Personen), die den Anwendungsfall auslösen* *Namen von Rollen beginnen mit dem Präfix „AKR_“, gefolgt von einem Substantiv.*
Vorbedingungen/ auslösendes Ereignis	*Die Vorbedingungen beschreiben alle relevanten und nichttrivialen Voraussetzungen, die erfüllt sein müssen, damit der Anwendungsfall durchgeführt werden kann.* *Auslöser für die Durchführung des Anwendungsfalls sind Ereignisse wie auslösende Handlungen anderer Akteure oder zeitgesteuerte Aktivitäten.* *Da Vor- und Nachbedingungen alternativ gelten können, hat sich folgende Schreibweise bewährt: Als Aufzählung mit Bulletpoints werden die Alternativen genannt. Innerhalb eines Bulletpoints gelten alle Bedingungen gemeinsam. Wenn es nur „eine Alternative“ gibt, kann der Bulletpoint weggelassen werden.*
Nachbedingungen/ Ergebnisse	*Beschreibung des erwarteten Zustandes nach Ausführung des Anwendungsfalls. Wenn möglich Verweis auf erzeugte Daten (d.h. Referenz zum Datenmodell) und Liste der fachlichen Fehlersituationen mit Beschreibung.* *Die Nachbedingungen beschreiben den Zustand, wenn der Anwendungsfall abgeschlossen ist. Sie beziehen sich auf die Bedingungen, die in der Vorbedingung genannt sind.*

Anwendungsfall	
Standardablauf	*Der Standardablauf ist der Ablauf von Aktionen der Akteure und der Anwendung, also die Interaktion zwischen Akteur und Anwendung, mit welchem der Akteur das Ziel erreicht. Aus dem Ablauf geht eindeutig hervor, was vom Anwender getan wird und was die Anwendung tut. Die Beschreibung des Ablaufs ist in der Regel ausführlicher als in der Kurzbeschreibung. Man muss aber darauf achten, dass sie nicht unnötig umfangreich wird und prägnant bleibt.* *Auch dialoglastige Anwendungsfälle beschreiben das fachliche Verhalten der Anwendung ohne konkrete Referenz auf die Benutzeroberfläche. Die Dialoge werden separat in der Dialogspezifikation beschrieben.* *Die einzelnen Schritte werden durchnummeriert.* *Falls die Beschreibung des Ablaufs bzw. einzelner Schritte zu komplex wird oder große Redundanzen zu anderen Anwendungsfällen entstehen, kann Funktionalität in Anwendungsfunktionen ausgelagert werden. Im Ablauf des Anwendungsfalls wird dann nur noch beschrieben, an welcher Stelle die Anwendungsfunktion angestoßen wird.* *Fachliche Spezial-Begriffe werden in der Ablauf-Beschreibung als bekannt vorausgesetzt. Die Definition der Begriffe erfolgt im Glossar.* *Falls der Anwendungsfall Zustandsänderungen auf Entitäten bewirkt, braucht nur die Änderung aus fachlicher Sicht genannt zu werden. Das formale und umfassende Zustandsmodell des Entitätstypen wird separat im Abschnitt „Fachliche Grundlagen“ beschrieben.* *Einfache Plausibilisierungen von Daten und die daraus resultierenden Fehlermeldungen gehören nicht zum Ablauf. Sie ergeben sich aus dem Datentyp und werden im Datenmodell beschrieben.*

Anwendungsfall	
Alternative Abläufe	*Alternative Abläufe, die in der Abfolge der Schritte wesentlich vom Standardablauf abweichen, können hier separat beschrieben werden. Die Trennung in Standardablauf und alternative Abläufe hilft, die Standardvariante einfach und übersichtlich zu halten.* *Verschiedene Alternativabläufe werden durch Zwischenüberschriften getrennt.* *Der Bezug zu den Schrittnummern im Standardablauf wird informell hergestellt. Z.B. „Anstelle von Schritt 3-5 selektiert der Benutzer …“ oder „Im Falle einer Zweitmeldung zeigt die Anwendung …“. Eine formale Zuordnung anhand einer Nummernsystematik wie z.B. „3b“ o.ä. würde die Lesbarkeit deutlich erschweren. In Alternativabläufen werden nur die abweichenden Schritte beschrieben. Es wird davon ausgegangen, dass alle nicht beschriebenen Schritte gleich dem Standardablauf sind.* *Kleinere Varianten, welche die Komplexität nur unwesentlich erhöhen (z.B. „sonst bricht das System die Verarbeitung mit einem sprechenden Fehler ab“), können in den Standardablauf mit eingearbeitet werden.* *Die Erweiterungen beschreiben alternative Abläufe des Standardablaufs. Falls eine Erweiterung zu komplex wird, sollte sie als eigener Anwendungsfall beschrieben werden.* *Typische alternative Abläufe sind Fehlerfälle. Fehlerfälle sind Abweichungen zum Standardablauf, die zu einem unerwünschten Verlauf oder gar zum Abbruch des Anwendungsfalls führen.*

*Das Anwendungsfalldiagramm zeigt den Anwendungsfall im Mittelpunkt. Zugehörige Dialoge, Entitäten, Nachbarschnittstellen und Nichtfunktionale Anforderungen sind um den Anwendungsfall angeordnet. Nichtfunktionale Anforderungen kann man hier darstellen, sofern sie spezifisch für den Anwendungsfall sind. Übergreifende Nichtfunktionale Anforderungen stellt man **nicht** bei jedem Anwendungsfall dar.*

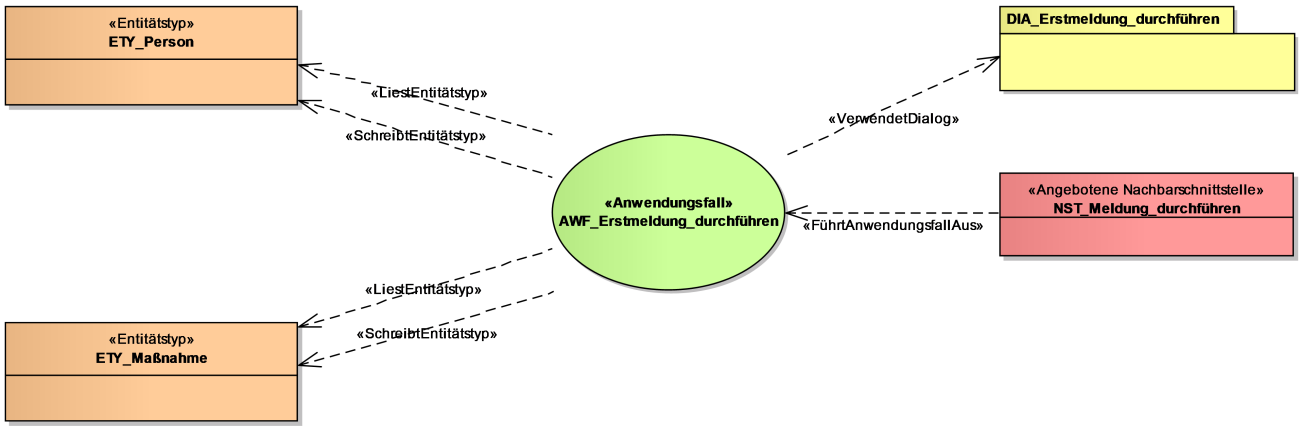


Figure 10. Anwendungsfall: AWF_Erstmeldung_durchführen

Erstellung des Diagramms im Enterprise Architect

Für die Verbindung zwischen dem Anwendungsfall und den zugehörigen Elementen nutzt man Stereotypen, die die Art der Verbindung beschreiben. Diese Verbinder stammen durchgängig aus dem UML-Profile für Anwendungsfälle.

Zur Darstellung von Dialogen zieht man den Ordner des Dialogs in das Diagramm, färbt ihn ein und blendet seine Inhalte aus.

Zur Verknüpfung der angebotenen Nachbarschnittstellen mit dem Anwendungsfall verwendet man den Verbinder „FührtAnwendungsfallAus“ aus dem UML-Profil „Nachbarschnittstellen“. Der gleichnamige Verbinder aus dem UML-Profil „Anwendungsfälle“ ist nur für die Verbindung zwischen Akteur und Anwendungsfall vorgesehen.

Zur Verknüpfung mit anderen, aufgerufenen Anwendungsfällen verwendet man den Verbinder „VerwendetAnwendungsfall“.

Schließlich blendet man alle Verbinder zwischen Elementen aus, außer sie verbinden die Elemente mit dem Anwendungsfall.

Für den Ablauf komplexer Anwendungsfälle kann man zusätzlich ein UML-Aktivitätendiagramm zeichnen. Das UML-Diagramm enthält die Abfolge der aufgerufenen Benutzeraktionen, Anwendungsfunktionen und deren Schnittstellenaufrufe. Dabei wird nur die oberste Ebene der Anwendungsfunktionen dargestellt. Wenn also eine Anwendungsfunktion selbst weitere aufruft, wird dies nicht dargestellt.

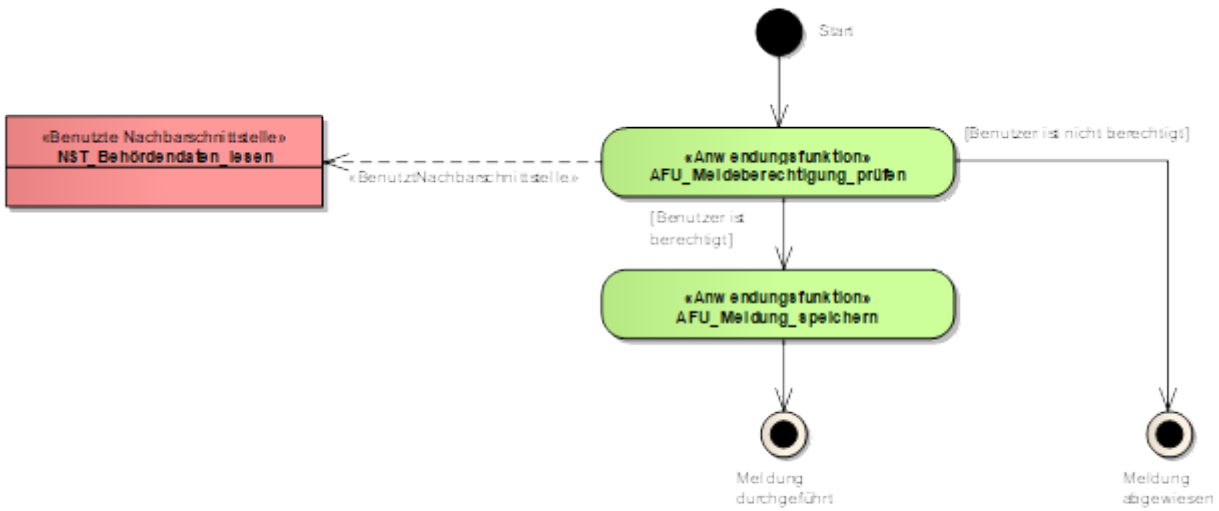


Figure 11. AWF_Erstmeldung_durchführen

Erstellung des Diagramms im Enterprise Architect

Start und Ende des Ablaufs sowie Verzweigungen erstellt man mit dem UML-Profil „Anwendungsfälle“.

Man zieht die Elemente „Anwendungsfunktion“ und ggf. „Benutzeraktion“ in ein Activity

Diagramm und benennt sie passend. Den Verbinder „Ablauf“ nutzt man für die Übergänge zwischen den Elementen, außer bei Nachbarschnittstellen. Die Stereotypen im Ablauf-Verbinder blendet man aus.

Mit [Strg] + Mausklick kann man Ecken in den Verbindern hinzufügen.

Im Kontextmenü des Verbinders gibt man mit „General > Name“ in eckigen Klammern an, welche Bedingung nach einer Verzweigung gilt. Wenn die Verzweigung sich aus dem Ergebnis einer Anwendungsfunktion ergibt, kann man das Verzweigungssymbol weglassen, um die Lesbarkeit zu erhöhen. Im Kontextmenü der Start- und Ende-Elemente entfernt man unter „Properties …“ den Stereotyp „Start“ bzw. „Ende“ und setzt den Namen fachlich sinnvoll.

Anwendungsfall AWF_<Bezeichnung des Anwendungsfalls>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Anwendungsfälle der Anwendungskomponente zu beschreiben sind.

Batch BAT_<Bezeichnung des Batches>

Ein Batchprogramm realisiert eine eigenständige Verarbeitung ohne direkten Benutzereingriff während des Ablaufes. In diesem Abschnitt wird ein batchverarbeitendes Programm der Anwendung fachlich beschrieben (Konfiguration, Abhängigkeiten, Datenvolumen, etc.). Batches können Anwendungsfälle für die Durchführung ihrer Fachlichkeit aufrufen oder eigenständig arbeiten; die Definition und das Layout der Ein- und Ausgaben werden in den Vor- und Nachbedingungen des Batches beschrieben und bei entsprechender Komplexität als Druckstücke erfasst. Falls Abhängigkeiten zwischen Batchprogrammen bezüglich des Aufrufs bestehen, werden diese als Teil des Diagramms der Anwendungskomponente dargestellt.

Batch	
Kurzbeschreibung	*Ein oder zwei Sätze zum Zweck des Batches.*
Vorbedingungen/ auslösendes Ereignis	*Welche Kriterien müssen für den Start des Batchprogrammes erfüllt sein? Welche Kriterien müssen für die Verarbeitung eines Datensatzes erfüllt sein? Mit welchen Parametern kann das Batchprogramm gestartet werden?*
Nachbedingungen/ Ergebnisse	*Was muss nach Ablauf des Batchprogramms erfüllt sein? Welche Ausgänge kann der Ablauf des Batchprogramms haben?*
Erwartetes Datenvolumen	*Wie viele Datensätze werden maximal und durchschnittlich vom Batchprogramm verarbeitet?*
Wiederanlauffähigkeit	*Wie flexibel reagiert der Batch im Fehlerfall? Kann er erneut gestartet werden? Bearbeitet er dann nur die zuvor noch nicht bearbeiteten Daten (Restart), oder bearbeitet er dann alle Daten noch einmal (Rerun)?*
Standardablauf	*Wie erfolgt die Ablaufsteuerung? Welche Abhängigkeiten gibt es zu anderen Batches? In welcher Reihenfolge und mit welcher Priorität erfolgt der Ablauf?*
Alternative Abläufe	*Welche alternativen Abläufe zum Standardablauf sind möglich (z.B. technische Fehlerbehandlung)?*

Batch	
Verwendete Anwendungsfälle	*Welche Anwendungsfälle werden im Ablauf des Batchprogrammes aufgerufen?*

Falls Abhängigkeiten zwischen dem Batch und Anwendungsfällen oder anderen verknüpften Elemente der Spezifikation bestehen, werden diese in einem UML-Komponentendiagramm dargestellt.

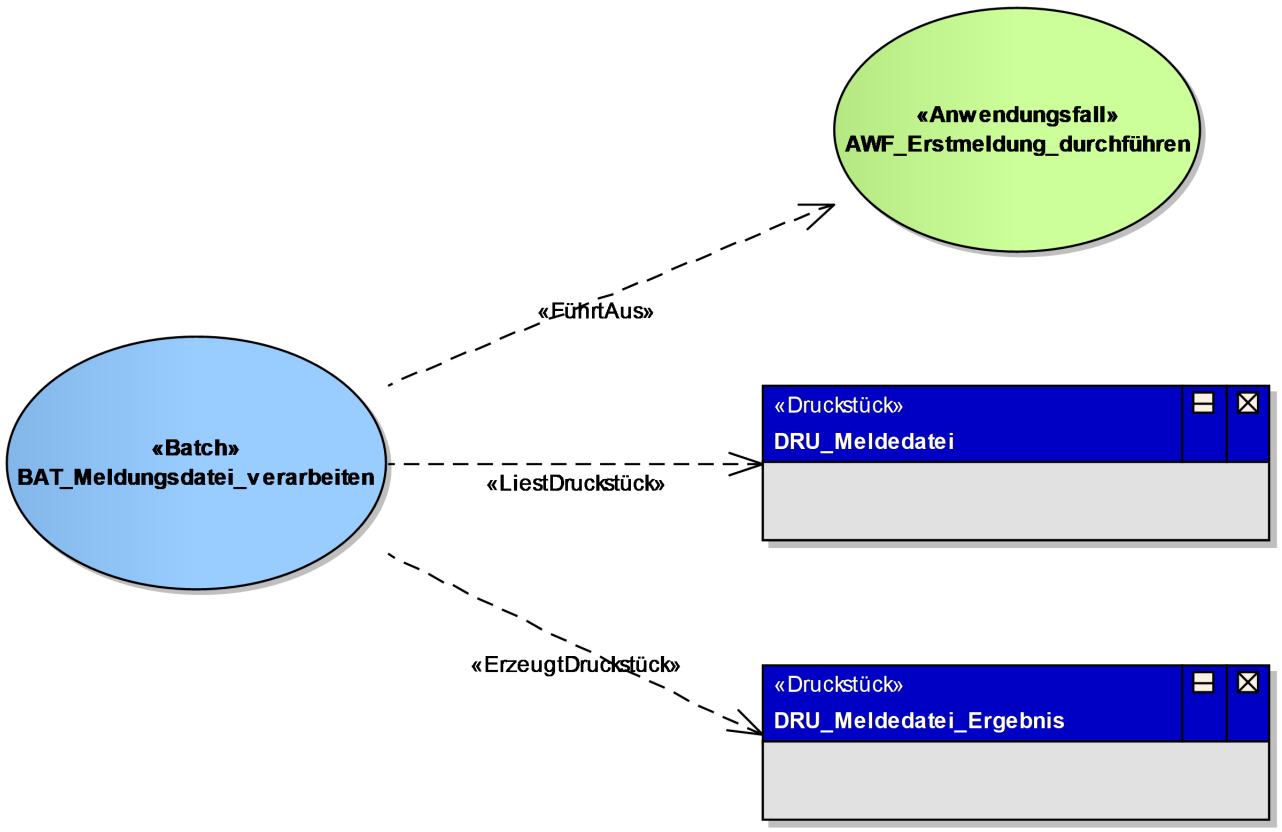


Figure 12. Batch: BAT_Meldungsdatei_verarbeiten

Erstellung des Diagramms im Enterprise Architect

Man zieht den Batch aus dem UML-Profil „Batches“ in ein Component Diagramm. Die Verbinder nutzt man analog zum Anwendungsfalldiagramm. Zusätzlich nutzt man den Verbinder mit Stereotyp „FührtAus“ aus dem UML-Profil, um den Batch mit dem Anwendungsfall zu verbinden.

BAT_<Bezeichnung des Batches>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Batches zu beschreiben sind.

Anwendungsfunktionen

Die Spezifikation der Anwendungsfunktion beschreibt Ausschnitte von Anwendungsfällen, die ohne Unterbrechung der Anwendung, gegebenenfalls unter Benutzung von Schnittstellen zum

Nachbarsystem, ausgeführt werden. Die Beschreibung der Anwendungsfunktionen ähnelt der Beschreibung eines Anwendungsfalls. Dabei wird wesentlich stärker auf den Aspekt: "Wie soll die Anwendung eine Verarbeitung durchführen", eingegangen. Die Anwendungsfunktionen sind der Komponente zugeordnet, deren Funktionalität sie umsetzen.

Anwendungsfunktionen dienen der Beschreibung komplexer Verarbeitungen, die von der Anwendung im Rahmen eines Anwendungsfalls durchgeführt werden. Die Anwendungsfälle verweisen auf die Anwendungsfunktionen.

Die Ausgliederung von Anwendungsfunktionen aus den Anwendungsfällen bringt die folgenden Vorteile:

- *Gliederung langer Anwendungsfall-Abläufe: Die Anwendungsfallspezifikation wird kompakter und leichter lesbar.*
- *Wiederverwendung von Funktionalität: Mehrfach verwendete Funktionalität wird in Anwendungsfunktionen nur einmal beschrieben.*

Anwendungsfunktionen können einander aufrufen. Hierarchische Beziehungen zwischen Anwendungsfunktionen werden nicht empfohlen. Sie erschweren die Verständlichkeit und nehmen das Design vorweg.

Namen von Anwendungsfunktionen beginnen mit dem Präfix „AFU_“ gefolgt von einem Substantiv, einem Unterstrich und einem Verb, z.B. „AFU_Treffer_bewerten“. Falls nötig kann noch ein Adjektiv vor das Substantiv gestellt werden. Der Titel ist ein im Kontext der Anwendung eindeutiger Bezeichner der Anwendungsfunktion. Er sollte so formuliert sein, dass er möglichst prägnant Hinweise auf Akteur und Ziel gibt.

Anwendungsfunktion AFU_<Bezeichnung der Anwendungsfunktion>

Anwendungsfunktionen werden gemäß der nachfolgenden Tabelle textuell beschrieben.

Anwendungsfunktion	
Kurzbeschreibung	*Ein erster Überblick darüber, was die Funktion tut.*
Vorbedingungen/ auslösendes Ereignis	*Vorbedingungen sind alle Randbedingungen, die für die Durchführung der Funktion erfüllt sein müssen und innerhalb der Funktion nicht mehr geprüft werden. Außerdem werden hier auch die Eingaben (Parameter etc.) angegeben.*
Nachbedingungen/ Ergebnisse	*Das Ergebnis ist die Außenwirkung der Ausführung der Funktion. Hierbei kann Bezug auf die Konsistenzbedingungen und den Ablauf genommen werden. Triviale Ergebnisse ("Ablauf ist abgelaufen", "Konsistenzbedingungen geprüft") können entfallen.*
Standardablauf	*Der Standardablauf beschreibt die einzelnen Teilschritte zur Durchführung der Funktion. Insbesondere können hier andere Anwendungsfunktionen aufgerufen werden. Hier erfolgt auch die Beschreibung komplexer Verarbeitungsschritte.* *Die einzelnen Schritte werden durchnummeriert.*

Anwendungsfunktion	
Alternative Abläufe	*Alternative Abläufe der Anwendungsfunktion (z.B. Fehlersituationen) können hier beschrieben werden.*

Anwendungsfunktion AFU_<Bezeichnung der Anwendungsfunktion>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Anwendungsfunktionen zu beschreiben sind.

Anwendungskomponente ANK_<Bezeichnung der Anwendungskomponente

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Anwendungskomponenten zu beschreiben sind.

Datenmodell

Das logische Datenmodell beschreibt die fachliche Sicht auf die Daten der Anwendung. Es enthält Entitätstypen und ihre Beziehungen zueinander. Jeder Entitätstyp enthält Attribute und ihre fachlichen Datentypen (siehe Abschnitt „Datentypen“). Den Inhalt des logischen Datenmodells bildet eine Beschreibung der Entitätstypen, Assoziationen und Attribute.

Die Entitätstypen, Assoziationen und eventuell auch die Attribute (mit ihren Datentypen) werden in einem Diagramm grafisch dargestellt. Das Diagramm dient dazu, einen ersten Überblick über die Entitätstypen zu bekommen und die Assoziationen „auf einen Blick“ zu erfassen. Die textuelle Beschreibung enthält die Beschreibungen der Entitätstypen und Attribute.

Für die Darstellung der Attribute spricht, dass so im Diagramm alle Informationen des Datenmodells bis auf die beschreibenden Texte enthalten sind. Gegen die Darstellung der Attribute spricht, dass diese dann im Diagramm redundant zur tabellarischen Darstellung gepflegt werden müssen. Hier kann es zu Inkonsistenzen kommen. Daher muss abgewogen werden, ob die Darstellung der Attribute wirklich notwendig ist.

Bei großen Datenmodellen werden Modellkomponenten definiert. Eine Modellkomponente ist eine Gruppierung, die fachlich zusammengehörige Entitätstypen inklusive ihrer Assoziationen zusammenfasst. Sie ist überschneidungsfrei, d.h. ein Entitätstyp gehört zu genau einer Modellkomponente. In einer Modellkomponente kann allerdings ein Entitätstyp einer anderen Modellkomponente referenziert werden.

Modellkomponenten werden in der textuellen Beschreibung durch entsprechende Benennung der Abschnitte gekennzeichnet.

Für die Modellierung gilt generell: Nicht die ganze Welt modellieren, sondern nur den für die zu realisierende Anwendung wichtigen fachlichen Ausschnitt. Die technische Umsetzung berücksichtigen wir hier nicht.

Ein Entitätstyp hat eine Reihe von Eigenschaften, die dabei helfen können zu entscheiden, ob eine

Information als Entitätstyp (oder als Datentyp bzw. Menge von Attributen) zu modellieren ist:

- *Er ist autonom, d.h. für sich allein lebensfähig (Gegenbeispiel: Alter einer Person).*
- *Die Entitäten sind (über einen Schlüssel) eindeutig identifizierbar.*
- *Er ist unverzichtbar, d.h. bestimmte Abläufe lassen sich nicht mehr darstellen, wenn man auf den Entitätstyp verzichtet.*
- *Man kann und will ihn durch Attribute näher beschreiben (Gegenbeispiel: Anzahl Mitarbeiter).*
- *Man kann und will ihn anlegen und löschen (siehe Eigenschaft autonom).*

Modellkomponente MKO_<Name der Modellkomponente>

Namen von Modellkomponenten beginnen mit dem Präfix „MKO_“ gefolgt von einem Substantiv, z.B. „MKO_Behörden“. Falls nötig kann noch ein Adjektiv vor das Substantiv gestellt werden.

Hier wird kurz textuell beschrieben, bezüglich welcher Geschäftslogik die Entitätstypen eine Modellkomponente bilden. Ein UML-Klassendiagramm enthält alle Entitäten mit Attributen und Assoziationen zwischen diesen. Fachliche Schlüsselattribute müssen im UML-Klassendiagramm hervorgehoben werden (in ETY_Person durch die Abtrennung "Schlüsselattribut").

Modellkomponenten fremder Entitätstypen, die in dem Übersichtsdiagramm der Modellkomponente referenziert werden, sind unterschiedlich zu kennzeichnen (ETY_Protokolleintrag weiß dargestellt). Ebenso sind transiente Entitätstypen im Übersichtsdiagramm kenntlich zu machen (z.B. durch kursiv geschriebenen Entitätsnamen).

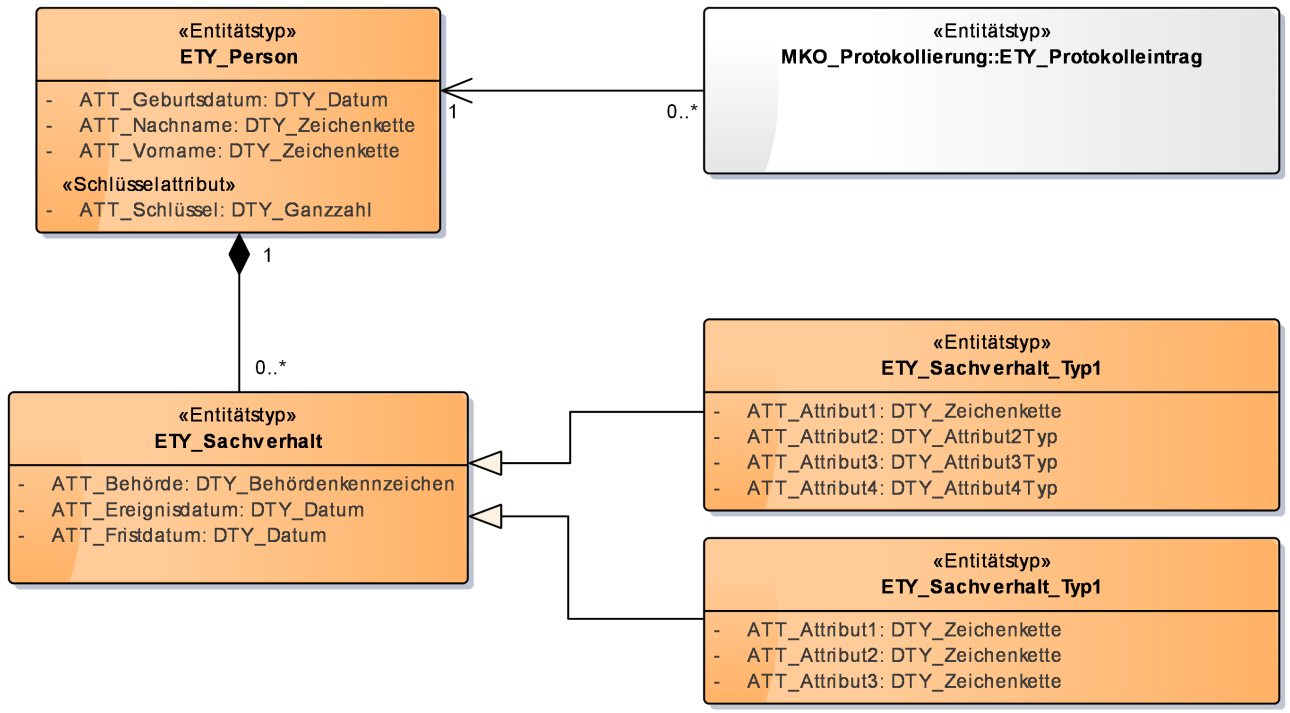


Figure 13. Modellkomponente MKO_xy

Wenn das Diagramm der Modellkomponente zu groß wird, kann man es auf mehrere Diagramme

aufteilen. Oft ergibt sich im Datenmodell eine baumartige Struktur. Hier kann man zentrale Entitäten als „Brücke“ zwischen den Diagrammen nutzen, die man im ersten Diagramm darstellt, als kämen sie aus einer anderen Modellkomponente. Im zweiten Diagramm stellt man sie dann vollständig dar, inklusive der ihnen untergeordneten Entitäten des Baumes. Ein Beispiel dafür ist nachfolgend abgebildet.

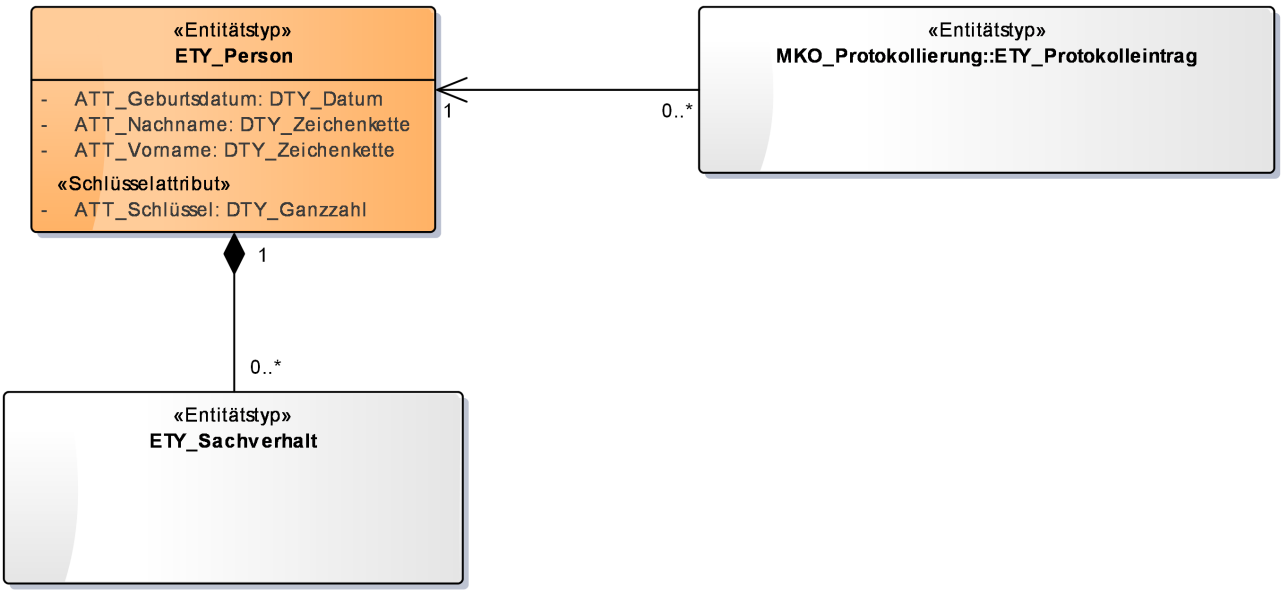


Figure 14. Modellkomponente MKO_xy (alternativ): Übersicht

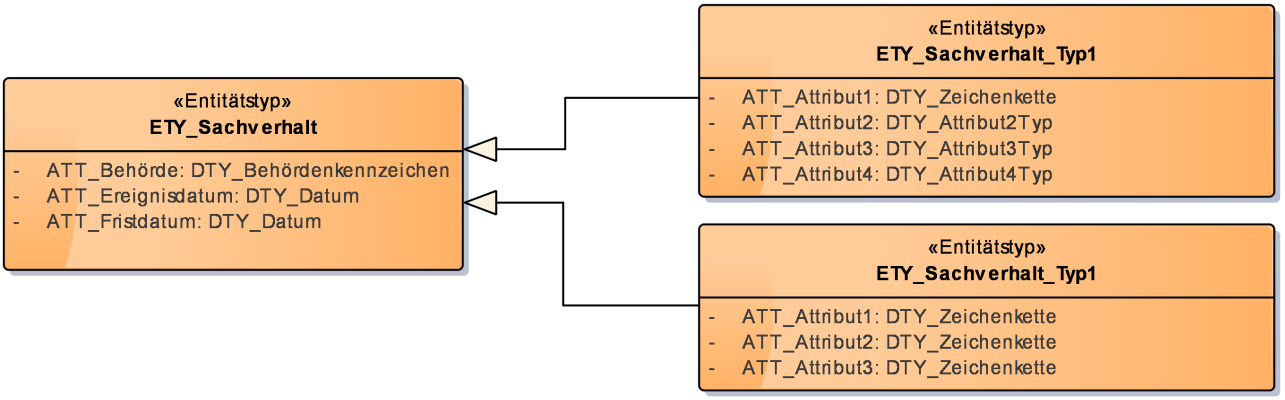


Figure 15. Modellkomponente MKO_xy: Details

Erstellung des Diagramms im Enterprise Architect

Man zieht die Entitätstypen aus dem UML-Profil „Datenmodell“ in ein Klassendiagramm, gibt ihnen Namen (mit Präfix „ETY_“) und falls gewünscht Attribute. Die Attribute haben ihrerseits jeweils Namen (mit Präfix „ATT_“) und Datentypen (mit Präfix „DTY_“).

Für die Verbinder nutzt man ebenfalls das UML-Profil. In der Source Role und Target Role kann man für die Enden der Assoziationen Multiplizitäten angeben. Wenn eine Assoziation ungerichtet ist, blendet man die Pfeilspitze übersetzen von „Navigability“ auf „unspecified“ in der Target Role aus.

Verknüpfte Entitäten aus anderen Modellkomponenten stellt man in diesem Diagramm weiß dar und blendet ihre Attribute aus. Es bietet sich hier an, die Namespaces ausnahmsweise darzustellen. So kann man sehen, aus welcher Modellkomponente die fremde Entität stammt.

Assoziationen zwischen Entitäten können, falls ihre Bedeutung nicht intuitiv klar ist, unterhalb der Diagramme textuell beschrieben werden: Bei den Entitäten selbst werden Assoziationen nicht beschrieben.

Entitätstyp ETY_<Bezeichnung des Entitätstypen>

Namen von Entitätstypen beginnen mit dem Präfix „ETY_“ gefolgt von einem Substantiv, z.B. „ETY_Person“. Falls nötig kann noch ein Adjektiv vor das Substantiv gestellt werden.

Entitätstyp	
Kurzbeschreibung	*Die Beschreibung des Entitätstyps ermöglicht dem Leser, den Namen der Entität und ihre Bedeutung zu verstehen.* *Für die Begriffserklärung können Synonyme oder Oberbegriffe benutzt werden (Beispiel: „Ein Mitarbeiter ist eine natürliche Person …“).* *Anschließend wird der Begriff inhaltlich und zeitlich abgegrenzt, d.h. wir erläutern, unter welchen Bedingungen eine Ausprägung zu diesem Entitätstyp gehört.* *Die Beschreibung kann Beispiele, weitere Erläuterungen und Anmerkungen enthalten. Sie sollte auf eine standardisierte Weise erfolgen, so dass sich die Beschreibungen über das Datenmodell hinweg in der Form ähneln.*

Attribut	Datentyp	Bedeutung
Name des Attributs beginnend mit dem Präfix „ATT_“.	*Verweis auf den fachlichen Datentyp (siehe Abschnitt „Datentypen“) oder auf einen einfachen Datentyp („Zeichenkette“, „Ganzzahl“, „Fließkommazahl“ etc.).*	*Die Beschreibung des Attributes sollte einen inhaltlichen Mehrwert bringen (also Beschreibungen wie „Datum ist das Datum der Buchung“ vermeiden).* *Es hat sich bewährt, die Beschreibung soweit möglich mit „Enthält“ zu beginnen.* *Folgende Fragestellungen können bei der Beschreibung helfen:* • *Ist diese Information immer oder nur unter bestimmten Bedingungen vorhanden?* • *Wann und wo entsteht diese Information?* • *Wie entsteht diese Information im Unternehmen? (Die Information kann festgestellt, festgelegt und abgeleitet sein.)* • *Für welchen Zeitraum bzw. bis zu welchem Zeitpunkt ist diese Information gültig?*
Weitere Attribute in nachfolgenden Zeilen	*Weitere Datentypen*	*Weitere Beschreibungen*

Entitätstyp ETY_<Bezeichnung des Entitätstypen>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Entitätstypen der Modellkomponente zu beschreiben sind.

Modellkomponente MKO_<Name der Modellkomponente>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Modellkomponenten mit ihren Entitätstypen zu beschreiben sind.

Datentypen

Fachliche Datentypen gruppieren Typen und Wertebereichsangaben von Attributen. Die Datentypen werden in einem Datentypverzeichnis verwaltet. Beispiele: ISBN, Fahrgestellnummer, Aufzählungstypen.

Im Fall von trivialer Fachlichkeit (z.B. Beschreibungstexte, einfache Nummern) verzichten wir auf fachliche Datentypen und verwenden direkt die technischen Basistypen Zeichenkette, Ganzzahl, Kommazahl etc. Eigenschaften des Attributes und der Datentyp sollten voneinander getrennt werden.

Typischerweise verwenden verschiedene Anwendungen ähnliche Datentypen. Innerhalb einer Anwendungslandschaft müssen gleich benannte Datentypen auch den gleichen Inhalt haben. Ähnliche, aber inhaltlich unterschiedliche Datentypen sollten auch über die Anwendungen der Anwendungslandschaft explizit unterschiedlich benannt werden, um hier Verwirrung zu vermeiden.

Falls Datentypen für Schlüsselwerte verwendet werden, welche im Schlüsselverzeichnis abgelegt sind, so ist in der Beschreibung des Datentyps die Schlüsselkategorie des SVZ zu nennen. Falls die Werte nicht im Schlüsselverzeichnis abgelegt sind, so ist ein Kapitel im Anhang der Spezifikation zu referenzieren, in dem alle fachlichen Ausprägungen des Schlüssels genannt werden.

Datentyp	**Basistyp**	**Bedeutung**	**Wertebereich**
Name des Datentyps beginnend mit dem Präfix „DTY_“.	*Technischer Basistyp wie „String“, „Integer“, „Float“, „Alphanum“ oder ähnliche.*	*Fachliche Bedeutung des Datentyps. Hier sollen auch Plausibilisierungen und Prüfungen beschrieben werden.*	*Mögliche Ausprägungen des Datentyps.*

Dialoge

In dem folgenden Abschnitt wird beschrieben, wie der Benutzer mit der Anwendung kommuniziert, d.h. welche Funktionen er wie in welchen Dialogen aufruft und welche Daten er pflegen kann. Die Beschreibung umfasst das statische Aussehen und das dynamische Verhalten der Dialoge.

Der Begriff „Dialog“ bezeichnet die gesamte Benutzerschnittstelle eines Anwendungsfalls (Masken, Zustände, Übergänge). Die Spezifikation eines Dialogs umfasst die Beschreibung seiner Masken und Maskenzustände sowie der Übergänge zwischen den Masken. Ein Dialog ist in sich abgeschlossen und gegenüber anderen Dialogen abgegrenzt.

Eine „Maske“ entspricht einem Bildschirmbereich zur Bearbeitung eines Arbeitsschritts eines Anwendungsfalls, z.B. ein Fenster. Eine Maske entspricht in der grafischen Benutzungsschnittstelle meist genau einem Fenster. Die Übergänge zwischen den Masken bzw. Zuständen eines Dialoges werden textuell bzw. grafisch beschrieben.

Wird die grafische Benutzeroberfläche innerhalb der Anwendungsfälle spezifiziert, hat das die folgenden Nachteile:

- *Die Beschreibung der Anwendungsfälle wird sehr umfangreich und sehr detailliert und lenkt von*

der Beschreibung des fachlichen Ablaufes ab.

- *Die Anwendungsfälle werden eventuell unnötigerweise zu detailliert beschrieben, da bestimmte Masken in mehreren Fällen benutzt werden und damit als separater Anwendungsfall ausgelagert werden können.*

Die Maskenbeschreibungen werden nach Anwendungskomponenten oder anderen fachlichen Kriterien gruppiert.

Dialog DIA_<Bezeichnung des Dialogs>

Der Dialog wird kurz textuell beschrieben. Es folgt eine Übersicht der zum Dialog zugehörigen Masken und deren Navigationsmöglichkeiten untereinander. Die Abhängigkeiten der Masken zu den Entitäten des Datenmodells bezüglich Eingabe und Ausgabe werden grafisch in einem UML-Komponentendiagramm dargestellt.

Namen von Dialogen beginnen mit „DIA_“ gefolgt von einem Substantiv und einem Verb, z.B. „DIA_Person_hinzufügen“. Falls nötig kann noch ein Adjektiv vor das Substantiv gestellt werden.

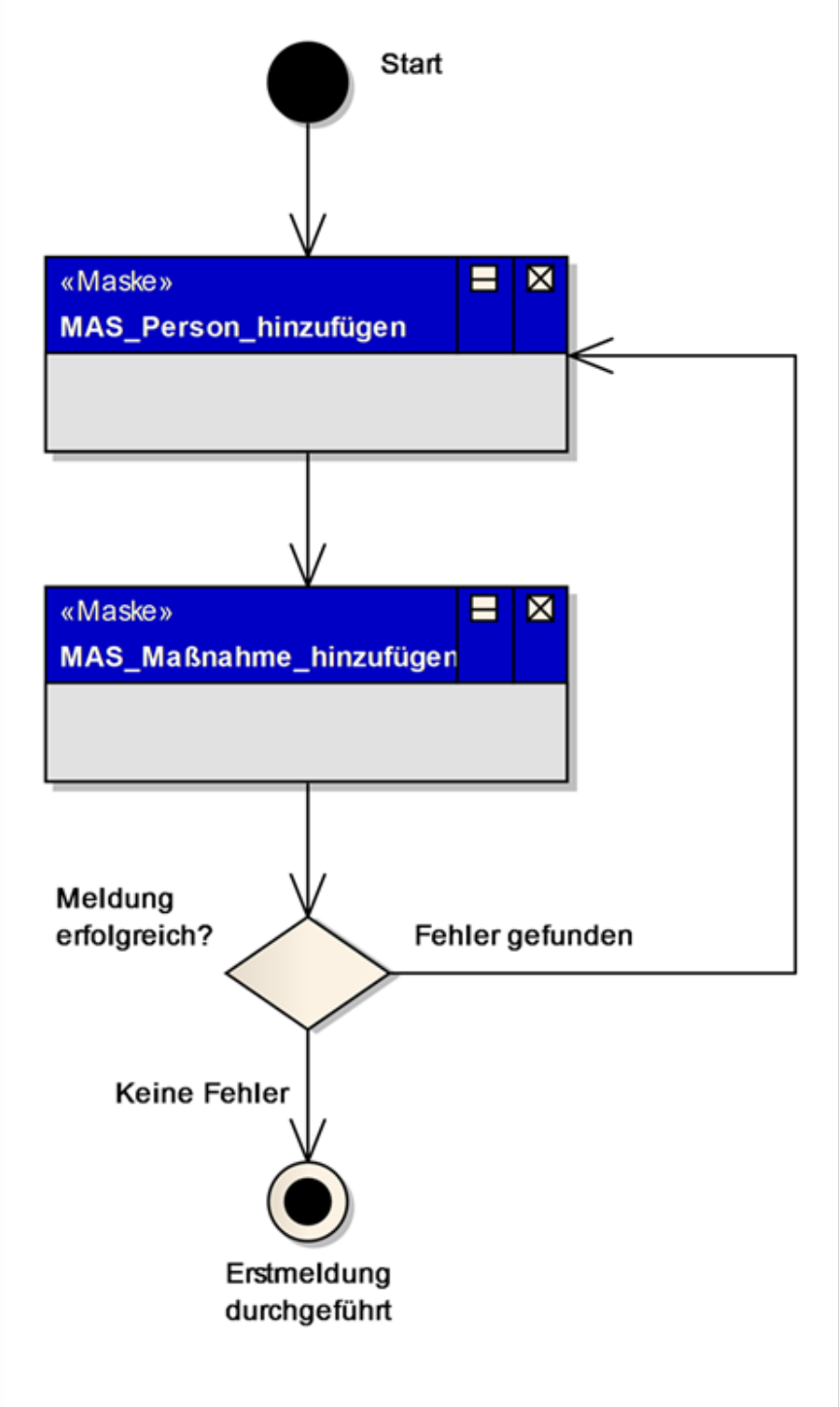


Figure 16. Dialog: DIA_Person_hinzufuegen

Erstellung des Diagramms im Enterprise Architect

Man erstellt ein Composite Structure Diagramm. Typischerweise beschreibt man die im Dialog verwendeten Masken zuerst (siehe Kapitel 6.3) und zieht die dort erzeugten Masken in das Diagramm. Dann nutzt man die Verbinder aus demselben UML-Profil „Dialoge“, um den Ablauf darzustellen. Man nutzt Verzweigung, Start- und Ende-Elemente aus dem UML-Profil „Anwendungsfälle“. Die Modellierung des Ablaufs führt man analog der Modellierung des Anwendungsfall-Ablaufs durch.

Dialog DIA_<Bezeichnung des Dialogs>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Dialoge zu beschreiben sind.

Masken

Die Masken werden in Kapiteln analog zu den Anwendungskomponenten aufgeteilt. Ein Übersichtsdiagramm zeigt die Masken je Anwendungskomponente.

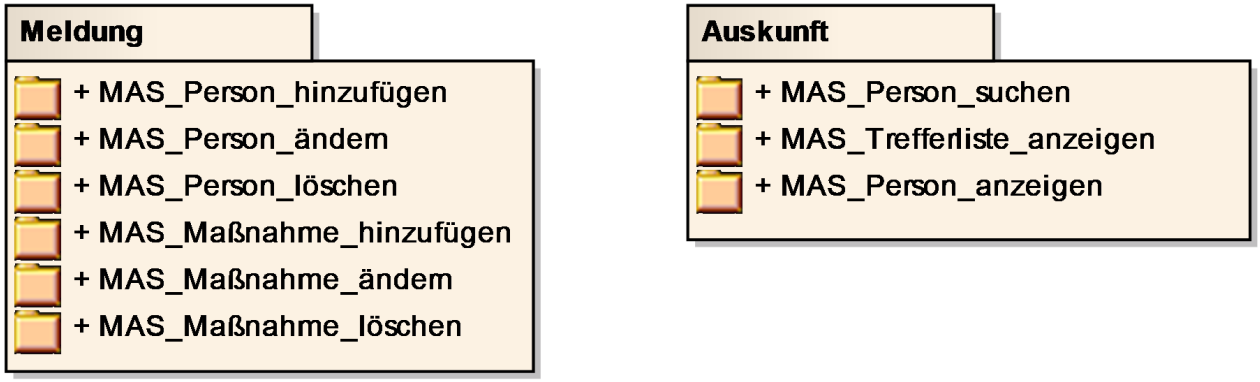


Figure 17. Übersicht Masken

> *Erstellung des Diagramms im Enterprise Architect*
>
> *Man zieht die Ordner, die die Masken enthalten, in ein Composite Structure Diagramm.*

<Bezeichnung der Anwendungskomponente>

Maske MAS_<Bezeichnung der Maske>

Der Zweck der Maske wird in ein bis zwei Sätzen erläutert. Namen von Masken beginnen mit „MAS_“ gefolgt von einem Substantiv und einem Verb, z.B. „MAS_Suchkriterien_eingeben“.

Die Entitätstypen des Datenmodells, die durch die Maske angezeigt oder eingegeben werden, können optional in einer Übersicht dargestellt werden. Alternativ können sie auch in der Kurzbeschreibung der Maske aufgezählt werden.

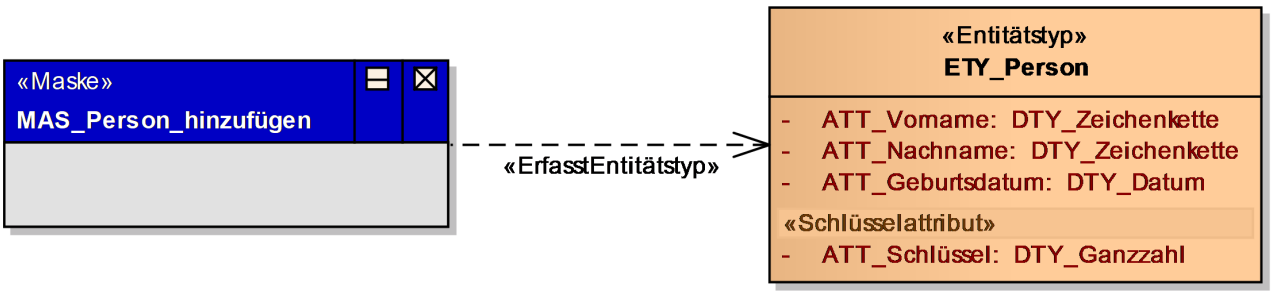


Figure 18. UML: MAS_Person_hinzufuegen

Erstellung des Diagramms im Enterprise Architect

Man zieht die Maske als zentrales Element des Diagramms aus dem UML-Profil „Dialoge“ in die Mitte eines „Composite Structure“-Diagramms und ordnet Entitäten aus dem Datenmodell oder Schnittstellenentitäten darum an. Die Verbinder stammen ebenfalls aus dem UML-Profil „Dialoge“.

Die Maske kann optional als Prototyp aufgezeichnet werden. Hierzu stehen verschiedene Elemente im UML-Profil „Dialoge“ zur Verfügung. Dies kann dazu dienen, vor Erstellung der Masken bereits das Bedienkonzept grob zu beschreiben und Maskenelemente eindeutig zu benennen. Alternativ kann ein Masken-Prototyp (Englisch „Mockup“) mit anderen Mitteln erstellt und hier eingefügt werden, oder es kann nach Inbetriebnahme ein Masken-Screenshot des produktiven Systems eingefügt werden.

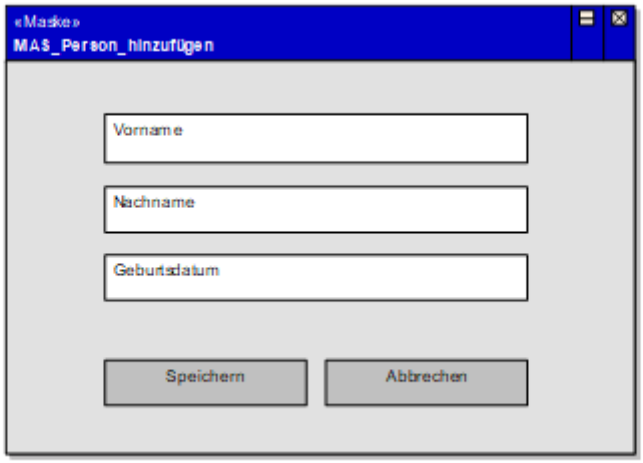


Figure 19. Maske: MAS_Person_hinzufuegen

Erstellung des Diagramms im Enterprise Architect

Man zieht die Maske in ein Component Diagramm und vergrößert sie. Dann zieht man aus dem UML-Profil „Dialoge“ einzelne Maskenelemente in die Maske hinein, benennt sie und ordnet sie an.

Maske	
Typ	*<Maskentyp>*
Referenzen	*Optionale Referenzen auf bestehende technische Maskennamen.*
Kurzbeschreibung	*Hier werden der Zweck sowie die Anzeige- und Eingabe-Möglichkeiten der Maske beschrieben. Falls kein Diagramm zu den genutzten Entitätstypen erstellt wird, werden diese hier zusätzlich aufgezählt.*

Maske MAS_<Bezeichnung der Maske>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Masken der Anwendungskomponente zu beschreiben sind.

<Bezeichnung der Anwendungskomponente>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier Masken der weiteren Anwendungskomponenten zu beschreiben sind.

Druckausgaben

Der Abschnitt Druckausgaben beschreibt das Layout und den Inhalt von Druckstücken. Hierzu gehören zum Beispiel Drucklisten, Serienbriefe, sowie die Ein- und Ausgabeformate für Batches. Für komplexe Auswertungen werden mögliche Selektions- und Verdichtungskriterien beschrieben.

Da Druckstücke für die Beschreibung sehr unterschiedlicher Formate verwendet werden, wird die Spezifikation den Notwendigkeiten des beschriebenen Formats angepasst. Dabei können Layouts, schematische Darstellungen oder fertige Vorlagen im Anhang oder als externe Dokumente mit Verweis beigefügt werden. Hier können allgemeine Informationen zu den Druckstücken beschrieben werden.

Danach folgt die Liste der Druckstücke, gruppiert nach Anwendungskomponenten und/oder ein UML-Komponentendiagramm, welches einen Überblick über die Anwendungskomponenten und deren Druckstücke gibt.

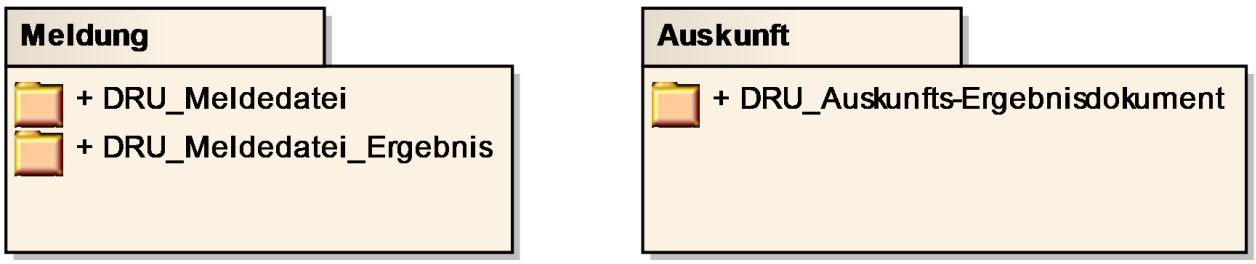


Figure 20. Übersicht der Druckausgaben

<Bezeichnung der Anwendungskomponente>

Druckstück DRU_<Bezeichnung der Druckausgabe>

Das Druckstück wird tabellarisch beschrieben. Zusätzliche Darstellungen können direkt im Dokument unter der Tabelle eingebunden oder im Anhang als Verweis auf externe Dokumente aufgenommen werden. In diesen zusätzlichen Darstellungen können Textbausteine mit Präfix „TBS_“ und Druckvariablen mit Präfix „DRV_“ markiert werden, um Teilen des Druckstücks eindeutige Namen zu geben. Für die Markierung können zum Beispiel Microsoft Word-Kommentare verwendet werden. Textbausteine sind typischerweise feste Textteile, Druckvariablen sind typischerweise variabel belegte Textteile.

Table 2. Druckausgabe <DRU_abc>

Druckausgabe	
Kurzbeschreibung	*Welche fachlichen Dienste werden durch die Schnittstelle bereitgestellt? Welchen fachlichen Hintergrund haben die ausgetauschten Daten?*
Empfänger	*An wen sind die Inhalte der Druckausgabe gerichtet?*
Medium	*Wie liegt das Druckstück vor? Z.B. „elektronisch“, „Papier“*

Sehr geehrte Damen und Herren,

sie erhalten eine automatisch generierte Mitteilung aus der Beispiel-Anwendung. Die entsprechenden Informationen entnehmen Sie bitte der Anlage.

Mit freundlichen Grüßen

Ihr Fachbereich

Behörde der Beispiel-Anwendung

Fachbereich

Besucheradresse: Beispielstraße 23, 12345 Teststadt

Postadresse: Beispielstraße 42, 12345 Teststadt

Servicezeiten: montags bis freitags 7:00 Uhr bis 18:30 Uhr

Telefon: +49 (0) 12345 - 678

+49 (0) 12345 -876

Telefax: +49 (0) 12345 - 123

+49 (0) 12345 - 321

E-Mail: fachbereich@behoerde.de

Internet: https://www.beispielbehoerde.de

Druckstück DRU_<Bezeichnung der Druckausgabe>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere Druckausgaben eines Druckausgabetyps zu beschreiben sind.

<Bezeichnung der Anwendungskomponente>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier Druckstücke weiterer Anwendungskomponenten zu beschreiben sind.

Nachbarsystemschnittstellen

Das Kapitel Nachbarsystemschnittstellen beschreibt die Verbindung zwischen der Anwendung und den Nachbarsystemen, mit denen die Anwendung kommunizieren soll.

Alle zu berücksichtigenden Nachbarsysteme werden im Abschnitt "Fachlicher Architekturüberblick" vorgestellt. Hier werden die einzelnen Schnittstellen beschrieben, wobei zwischen von der Anwendung benutzten und durch die Anwendung angebotenen Schnittstellen unterschieden wird.

Namen von Nachbarsystem-Schnittstellen beginnen mit „NST_“ gefolgt von einem Substantiv und einem Verb, z.B. „NST_Auskunft_durchführen“. Falls nötig, kann noch ein Adjektiv gefolgt von einem Unterstrich vor das Substantiv gestellt werden.

Angebotene Schnittstellen

Ein UML-Komponentendiagramm gibt einen Überblick über die von der Anwendung angebotenen Schnittstellen.

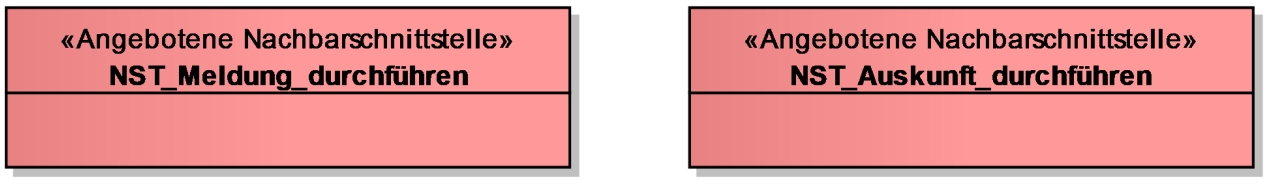


Figure 21. Angebotene Schnittstellen

Angebotene Schnittstelle NST_<Bezeichnung der Nachbarschnittstelle>

Die Nachbarsystemschnittstelle wird in einem oder mehreren Diagrammen grafisch dargestellt. Dabei werden die Parameter und Rückgabewerte als Nachbarschnittstellenentitäten dargestellt. Diese beschreiben analog zu den Entitäten des Datenmodells den Aufbau der Schnittstelle. Die Vor- und Nachteile der Darstellung der Schnittstellenattribute (NSA_ ...) gelten analog. Auch der Aufbau der Schnittstellenentitäten kann dem Aufbau des Datenmodells ähneln. Meist ergeben sich aber doch Abweichungen, zum Beispiel weil Attribute erst vom aufgerufenen Anwendungsfall berechnet und gespeichert werden.

Nachbarschnittstellenentitäten können über mehrere Schnittstellen hinweg wiederverwendet werden.

Wenn das Diagramm der Schnittstelle zu umfangreich wird, kann es auf mehrere Diagramme aufgeteilt werden, analog zur Aufteilung einer Modellkomponente des Datenmodells.

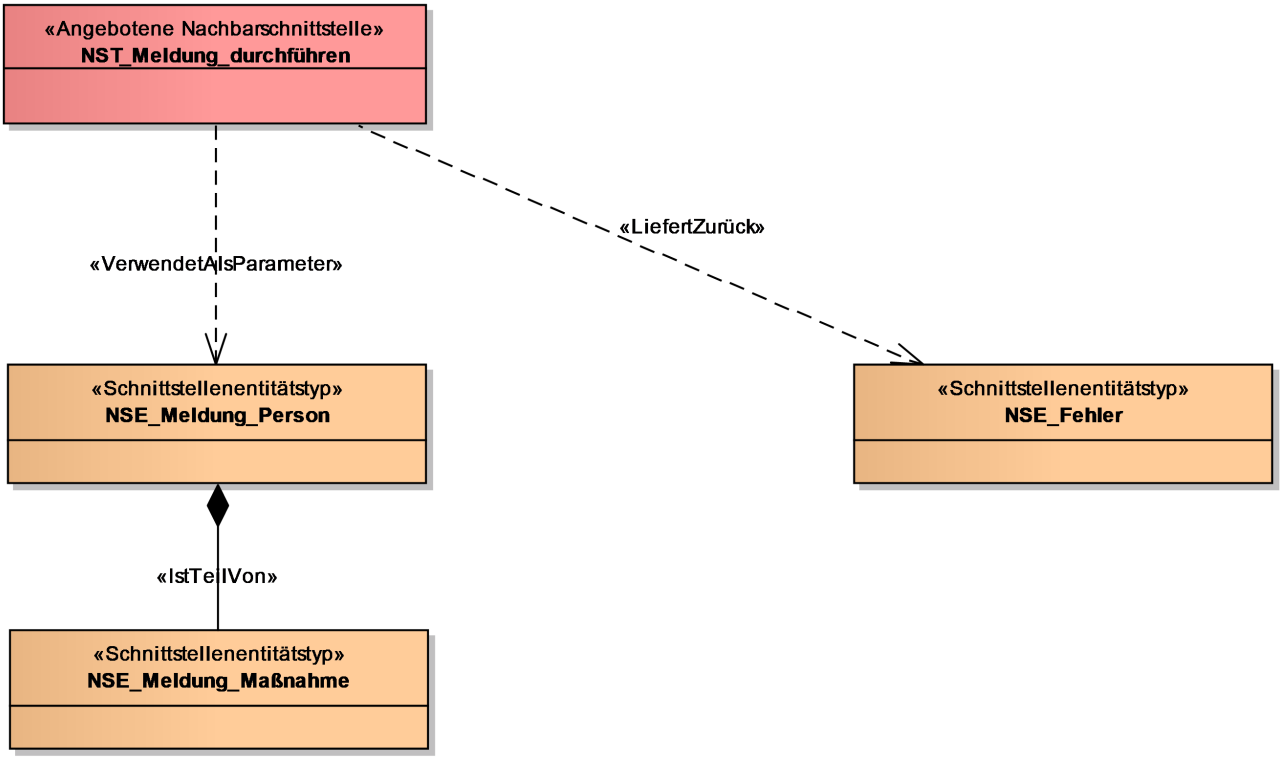


Figure 22. NST_Meldung_durchführen

Erstellung des Diagramms im Enterprise Architect

Man zieht die Nachbarsystemschnittstelle aus dem UML-Profil „Nachbarschnittstellen“ in ein Composite Structure Diagramm und benennt sie richtig. Die Schnittstellenentitätstypen werden erst wie im Kapitel Schnittstellenentitätstyp NSE_<Name des Schnittstellenentitätstyps> beschrieben erstellt und dann in diesem Diagramm verwendet. Zwischen den Schnittstellenentitätstypen werden Verknüpfungen definiert. Typischerweise ergeben sich eine oder mehrere baumartige Strukturen, analog zum Datenmodell. Deren „Wurzeln“ werden mit der Schnittstelle verknüpft, entweder als Parameter oder als Rückgabe. Beide Verknüpfungen stammen aus dem UML-Profil „Nachbarschnittstellen“.

Table 3. Nachbarschnittstelle <NST_abc>

Nachbarschnittstelle	
Kurzbeschreibung	*Welche fachlichen Dienste werden durch die Schnittstelle bereitgestellt? Welchen fachlichen Hintergrund haben die ausgetauschten Daten?*
Nutzende Nachbarssysteme	*Liste der Anwendungen, welche auf die Schnittstelle zugreifen.*
Komplexität	*Einschätzung der fachlichen Komplexität der Schnittstelle: „hoch“, „mittel“, „gering“. Die hierfür notwendigen Regeln müssen vorab systemabhängig festgelegt und beschrieben werden.*

Nachbarschnittstelle	
Synchron/ Asynchron	*Bei synchronen Schnittstellen blockiert der aufrufende Bearbeitungsablauf so lange, bis die aufgerufene Anwendung eine Antwort zurückgesendet hat (enge Kopplung). Bei asynchronen Schnittstellen fährt der aufrufende Bearbeitungsablauf sofort mit weiteren, lokalen Schritten fort (lose Kopplung). Falls eine Reaktion von der aufgerufenen Anwendung erwartet wird, so muss sie über einen asynchronen Mechanismus wie z.B. ein Aufgabensystem in den Bearbeitungsprozess wieder eingegliedert werden.*
Online/ Offline	*Bei Online-Schnittstellen führt der Datenaustausch sofort zu einer Verarbeitung in der angesprochenen Anwendung. Bei Offline-Schnittstellen wird eine Datenstruktur (Nachricht, Datei, ...) für die angesprochene Anwendung hinterlegt, die zu einem späteren Zeitpunkt, z.B. im Rahmen eines nächtlichen Batch-Laufes verarbeitet wird. Offline-Schnittstellen werden alternativ als Batches mit Druckstücken spezifiziert.*
Verwendete Entitätstypen (Input)	*Liste der Schnittstellenentitätstypen, die Parameter beim Aufruf der Schnittstelle durch ein Nachbarsystem sind. Durch das Schnittstellendiagramm ist diese Information bereits dargestellt, daher ist die Auflistung optional.*
Verwendete Entitätstypen (Output)	*Liste der Schnittstellenentitätstypen, die von der Schnittstelle zurückgeliefert werden. Durch das Schnittstellendiagramm ist diese Information bereits dargestellt, daher ist die Auflistung optional.*
Nutzende Anwendungsfälle	*Liste der Anwendungsfälle, die bei Nutzung der Schnittstelle aufgerufen werden.*
Nichtfunktionale Anforderungen	*Liste der nichtfunktionalen Anforderungen, welche die Schnittstelle erfüllen muss.*

Eingabeparameter

Die Eingabeparameter der Nachbarsystemschnittstelle werden beschrieben. Wenn es viele Überschneidungen zwischen den Eingabeparametern der verschiedenen angebotenen Nachbarsystemschnittstellen gibt, kann dieser Abschnitt auch einmalig nach den Nachbarsystemschnittstellen beschrieben werden.

Schnittstellenentitätstyp NSE_<Name des Schnittstellenentitätstyps>

Namen von Schnittstellenentitätstypen beginnen mit dem Präfix „NSE_“ gefolgt von einem Substantiv, z.B. „NSE_Person“. Falls nötig, kann noch ein Adjektiv vor das Substantiv gestellt werden. Es hat sich als sinnvoll erwiesen, zusätzlich eine Kurzfassung des Namens der Schnittstelle in den Namen der Entität aufzunehmen, z.B. „NSE_Meldung_Person“. Wenn die Schnittstellenentität einem Entitätstyp im Datenmodell entspricht, sollte sie auch analog heißen. Die hier beschriebene Füllung der Kurzbeschreibung der Schnittstellenentität und die Bedeutung des Attributs entsprechen unter Umständen den analogen Entitäten aus dem Datenmodell. In diesem Fall kann man in den Textfeldern dorthin verweisen, um Dopplungen zu vermeiden.

Nachbarschnittstellen-Entitätstyp	
Kurzbeschreibung	*Die Beschreibung des Schnittstellenentitätstyps ermöglicht dem Leser, den Namen der Entität zu verstehen. Für die Begriffserklärung können Synonyme oder Oberbegriffe benutzt werden (Beispiel: „Ein Mitarbeiter ist eine natürliche Person“). Anschließend wird der Begriff inhaltlich und zeitlich abgegrenzt, d.h. wir erläutern, unter welchen Bedingungen eine Ausprägung zu diesem Entitätstyp gehört. Die Beschreibung kann Beispiele, weitere Erläuterungen und Anmerkungen enthalten. Sie sollte auf eine standardisierte Weise erfolgen, sodass sich die Beschreibungen über das Datenmodell hinweg in der Form ähneln.*

Attribut	Datentyp	Beschreibung
Name des Nachbarschnittstellenattributs beginnend mit dem Präfix „NSA“.	Verweis auf den fachlichen Datentyp (siehe Abschnitt „Datentypen“ des Kapitels zum Datenmodell) oder auf einen einfachen Datentyp („Zeichenkette“, „Ganzzahl“, „Fließkommazahl“ etc.).	*Die Beschreibung des Attributes sollte einen inhaltlichen Mehrwert bringen (also Beschreibungen wie „Datum ist das Datum der Buchung“ vermeiden).* *Folgende Fragestellungen können bei der Beschreibung helfen:* *- Ist diese Information immer oder nur unter bestimmten Bedingungen vorhanden?* *- Wann und wo entsteht diese Information?* *- Wie entsteht diese Information im Unternehmen? (Die Information kann festgestellt, festgelegt und abgeleitet sein.)* *- Für welchen Zeitraum bzw. bis zu welchem Zeitpunkt ist diese Information gültig?*
Weitere Nachbarschnittstellenattribute in den nachfolgenden Zeilen	*weitere Datentypen*	*Weitere Beschreibungen*

NST_<Bezeichnung der Nachbarschnittstelle>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere angebotene Schnittstellen mit ihren Ein- und Ausgabeparametern zu beschreiben sind.

Benutzte Schnittstellen

Ein UML-Komponentendiagramm gibt einen Überblick über die von der Anwendung benutzten Schnittstellen, geordnet nach den anbietenden Anwendungen. Benutzte Schnittstellen werden normalerweise im anbietenden Nachbarsystem modelliert. Falls keine derartige Dokumentation vorliegt, können sie hier analog der angebotenen Schnittstellen modelliert werden.

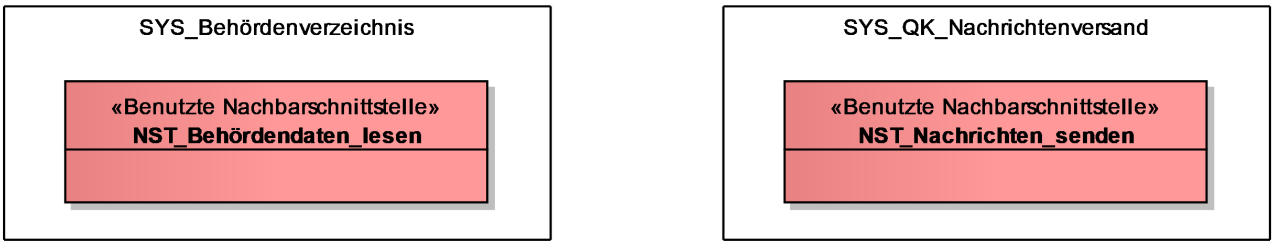


Figure 23. Benutzte Schnittstellen

Erstellung des Diagramms im Enterprise Architect

Hier hat man zwei Alternativen: Entweder die Nachbarsysteme sind bereits im EA modelliert, dann nimmt man die angebotenen Schnittstellen aus diesen Anwendungen und zieht sie in ein Composite Structure Diagramm in der eigenen Anwendung. Wenn nötig, kann man dazu die Schnittstellen oder die gesamten Anwendungen ins eigene EA-Modell importieren. Oder die Nachbarsysteme werden nicht modelliert (z.B. bei Anwendungen außerhalb der Anwendungslandschaft), dann spezifiziert man benutzte Schnittstellen in der eigenen Anwendung.

Zur Abgrenzung verschiedener Nachbarsysteme zieht man je eine Boundary um die Schnittstellen eines Nachbarsystems und gibt ihr über ihre Properties den Namen des Nachbarsystems.

Benutzte Nachbarschnittstelle NST_<Bezeichnung der Nachbarschnittstelle>

Falls benutzte Nachbarschnittstellen ausmodelliert werden, erfolgt dies hier analog der angebotenen Nachbarsystemschnittstellen.

Benutzte Nachbarschnittstelle NNST_<Bezeichnung der Nachbarschnittstelle>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere benutzte Nachbarschnittstellen zu beschreiben sind.

Nichtfunktionale Anforderungen

Die Spezifikation der nichtfunktionalen Anforderungen umfasst alle Anforderungen, die nicht der

Fachlichkeit und dem funktionalen Verhalten der Anwendung zugerechnet werden können und die nicht in anderen Abschnitten der Spezifikation behandelt werden. Nichtfunktionale Anforderungen, die in anderen Spezifikationsabschnitten behandelt werden, sind vor allem die Benutzbarkeitsanforderungen der Dialogspezifikation und die Projektanforderungen.

Nichtfunktionale Anforderungen verdienen besondere Berücksichtigung, da sie gerne vernachlässigt werden, mitunter schwer umzusetzen sind und gleichzeitig eine hohe Bedeutung für den Projekterfolg haben können (z.B. Laufzeitverhalten der Anwendung).

Erfahrungsgemäß können hier beschriebene Anforderungen projektweite Auswirkungen entfalten oder zu erheblichen Mehraufwänden führen. Deshalb sind solche Anforderungen sehr gewissenhaft aufzustellen und auf das fachlich notwendige Maß zu beschränken.

Beispiel: Es ist natürlich immer wünschenswert, bei Online-Funktionen eine Antwortzeit von maximal 3 Sekunden zu fordern. Eine solche Forderung kann aber je nach Komplexität der Anwendungsarchitektur dazu führen, dass Hard- und Softwarekosten entstehen, die in keinem Verhältnis zum Nutzen stehen.

Performanz und Antwortzeitverhalten

Die Anforderungen an das Antwortzeitverhalten im Dialog und die Verarbeitungsdauer im Batch sind wichtige Parameter für die technische Architektur der Anwendung. Oft erhält man hier Angaben wie "Antwortzeit im Dialog max. 10 Sekunden". Idealerweise sollten aber Performanz-Anforderungen je Anwendungsfall spezifiziert werden. Hier ist dann der Ort, an dem diese Informationen dann noch einmal tabellarisch zusammengefasst dargestellt werden.

NFA_<Name der Anforderung>

Table 4. Anforderungen an <SYSTEM>

Kurzbeschreibung	*Ein kurzer Satz, was gefordert wird.*
Prüfkriterien	*Wie wissen Sie, ob die Anforderung erfüllt ist?*
Quellen	*Wer hat die Anforderung gestellt?*
Priorisierung	*Vorzugsweise Bewertung mittels Unzufriedenheits-/ Zufriedenheits-Maß*

Optional können zusätzlich folgende Felder aufgenommen werden:

- *Begründung (Warum ist die Anforderung wichtig?)*
- *Abhängigkeiten (Verweise auf andere Anforderungen, die das gleiche Thema betreffen)*
- *Zusätzliche Materialien (Verweise auf Definitionen, Modelle und Dokumente, die die Anforderung verdeutlichen)*
- *Historie (Änderungshistorie der Anforderung)*
- *Verweise auf die Anwendungsfälle, auf welche sich die Anforderung bezieht.*

NFA_Performanz_und_Antwortzeitverhalten (optional)

Table 5. Performanz und Antwortzeitverhalten

Kurzbeschreibung	*Die Verarbeitung einer <Meldung / Auskunft / …> muss im <System> in <Prozent> der Fälle innerhalb von <Sekunden> durchgeführt werden.* *In weiteren <Prozent> der Fälle muss sie innerhalb von <Sekunden> durchgeführt werden.* *In <Prozent> der Fälle sind längere Antwortzeiten erlaubt.* *Die Messung der Antwortzeit erfolgt an der <Service-Schnittstelle des Systems / …>.* *Die Antwortzeiten von Nachbarsystemen sind <nicht> in den geforderten Antwortzeiten enthalten.*
Prüfkriterien	*<Test in Staging-Umgebung mit Echtdaten / Verwendung von JMeter / …>* *<Nachbarsysteme werden durch Stubs ersetzt / …>*
Quellen	*<Quelle der Anforderung>*
Priorisierung	*<hoch / mittel / niedrig>*

NFA_<Name der Anforderung>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere nichtfunktionale Anforderungen zu beschreiben sind.

Lastverhalten und Mengengerüst

Mengengerüste können sich auf Aspekte wie Benutzerzahlen, Benutzungshäufigkeit, zeitliche und räumliche Verteilung der Benutzung, Menge fachlicher Objekte und Datenvolumen beziehen. Die Mengengerüste sind eine wichtige Bezugsgröße für nichtfunktionale Anforderungen. Zum Beispiel sind Performanzanforderungen nur dann aussagekräftig, wenn sie sich als Voraussetzung und Annahme auf ein konkretes Mengengerüst beziehen. Das Mengengerüst selbst ist auch eine wichtige Anforderung, die für die Konstruktion große Bedeutung besitzt.

Die maximale Last, welche die Anwendung verarbeiten können muss, wird spezifiziert. Idealerweise kann das je Anwendungsfall angegeben werden. Außerdem sind Angaben zu der maximalen gleichzeitig zu erwartenden Benutzeranzahl (parallele Sessions) zu machen.

NFA_Lastverhalten_und_Mengengerüst (optional)

Table 6. Lastverhalten und Mengengerüst

Kurzbeschreibung	*<Das System> muss in der Lage sein, in Stoßzeiten <Zahl> <Auskünfte/Meldungen/ … pro Zeiteinheit> zu verarbeiten.*
Prüfkriterien	*<Test in Staging-Umgebung mit Echtdaten / Verwendung von JMeter / …> < Nachbarsysteme werden durch Stubs ersetzt / …>*
Quellen	*<Quelle der Anforderung>*
Priorisierung	*<hoch / mittel / niedrig>*

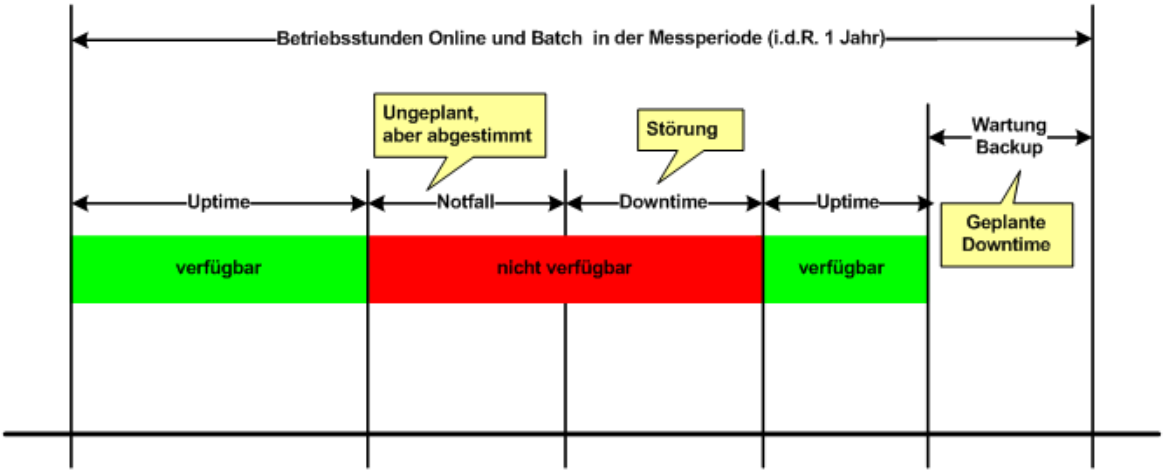
NFA_<Name der Anforderung>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere nichtfunktionale Anforderungen zu beschreiben sind.

Verfügbarkeit

In diesem Kapitel werden besondere Anforderungen an die Verfügbarkeit spezifiziert. Die Verfügbarkeit wird über einen Prozentwert gemessen, der unter Berücksichtigung von definierten Betriebs- und Wartungsfenstern angibt, wie ausfallsicher eine Anwendung sein soll und tatsächlich ist. Spezifiziert wird die geplante Betriebszeit, unterschieden nach Online und Batchbetrieb. Zum Beispiel wird hier angegeben, dass der Dialog jeweils Montag bis Freitag von 07:00 bis 19:00 Uhr zur Verfügung stehen muss. Alternativ wird auch einfach nur ein Prozentwert angegeben, zum Beispiel 99 %.

Wichtig ist, dass klar spezifiziert wird, wie die Verfügbarkeit der Anwendung exakt definiert ist. Abbildung 24 *gibt einen Überblick, was zu berücksichtigen ist.*



$$\text{Verfügbarkeit}_{\%} = \frac{\text{Betriebszeitraum}_{\text{Std}} - \Sigma\,\text{Wartung}_{\text{Std}} - \Sigma\,\text{Notfall}_{\text{Std}} - \Sigma\,\text{Störung}_{\text{Std}}}{\text{Betriebszeitraum}_{\text{Std}} - \Sigma\,\text{Wartung}_{\text{Std}}} \times 100$$

Figure 24. Verfügbarkeit

NFA_Verfügbarkeit (optional)

Table 7. Verfügbarkeit

Kurzbeschreibung	*<Das System> muss <durchgängig (7x24) / zu den normalen Büro-Zeiten (Mo-Fr, 06:30 Uhr bis 21 Uhr) / ...> zu <Prozent> der Zeit verfügbar sein. Ausfälle von Nachbarsystemen werden <nicht> als Ausfall des <Systems> gewertet.*
Prüfkriterien	*<Das System> wird zunächst in der Staging-Umgebung einem Stabilitätstest unterzogen. Im laufenden Betrieb wird die Erreichung der Anforderung durch Statistiken nachgewiesen.*
Quellen	*<Quelle der Anforderung>*

Kurzbeschreibung	*<Das System> muss <durchgängig (7x24) / zu den normalen Büro-Zeiten (Mo-Fr, 06:30 Uhr bis 21 Uhr) / …> zu <Prozent> der Zeit verfügbar sein. Ausfälle von Nachbarsystemen werden <nicht> als Ausfall des <Systems> gewertet.*
Priorisierung	*<hoch / mittel / niedrig>*

NFA_<Name der Anforderung>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere nichtfunktionale Anforderungen zu beschreiben sind.

Systemsicherheit

In diesem Kapitel werden besondere Anforderungen an den Systemzugang spezifiziert. Hier ist zu spezifizieren, welche Besonderheiten für den Administrator-, Entwickler- und Anwenderzugang sowie für die Konfiguration des Systems und des unterliegenden Betriebssystems zu berücksichtigen sind.

NFA_<Name der Anforderung>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere nichtfunktionale Anforderungen zu beschreiben sind.

Vertraulichkeit

In diesem Kapitel werden besondere Anforderungen an den Schutz der Vertraulichkeit von Daten spezifiziert. Diese dienen der Verhinderung der Preisgabe von Informationen an Unbefugte, beispielsweise durch unverschlüsselte Übertragung.

NFA_<Name der Anforderung>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere nichtfunktionale Anforderungen zu beschreiben sind.

Datensicherheit

In diesem werden besondere Anforderungen an den Schutz von Daten vor Verlust und unberechtigter Veränderung spezifiziert. Die zu spezifizierende Anwendung wird sich nach seiner Fertigstellung in eine bestehende Systemlandschaft integrieren. I.d.R. sind für die Systemlandschaft bereits Regeln zur Datensicherheit definiert, die dokumentiert und zu berücksichtigen sind. Gelten für das neue System keine besonderen Regeln, reicht hier der Verweis auf ein übergeordnetes Dokument.

NFA_<Name der Anforderung>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere nichtfunktionale Anforderungen zu beschreiben sind.

Nachvollziehbarkeit

In diesem Kapitel werden besondere Anforderungen an die Nachvollziehbarkeit spezifiziert. Nachvollziehbar gemacht werden sollen die durchgeführten Aktionen und die Bedingungen, unter denen diese ausgeführt wurden. Dies kann sich beispielsweise auf die An- und Abmeldungen eines Benutzers beziehen und auf die Aktionen, die dieser durchgeführt hat, oder auf die Durchführung eines Batches. Zur Herstellung von Nachvollziehbarkeit können Protokolleinträge verwendet werden. Die Speicherdauer von Protokolleinträgen sollte hierbei immer spezifiziert werden.

NFA_<Name der Anforderung>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere nichtfunktionale Anforderungen zu beschreiben sind.

Verbindlichkeit

In diesem Kapitel werden besondere Anforderungen an die Verbindlichkeit spezifiziert. Verbindlichkeit wird hier als Kombination von Authentizität und Nichtabstreitbarkeit definiert. Authentizität beschreibt die Sicherstellung der Identität eines Kommunikationspartners. Dies kann sich sowohl auf den Versender als auch auf den Empfänger von Nachrichten bzw. den Anbieter und den Nutzer von Diensten beziehen. Nichtabstreitbarkeit beschreibt die Eigenschaft, dass der Empfang von Nachrichten bzw. die Verwendung von Diensten nicht in Abrede gestellt werden kann.

NFA_<Name der Anforderung>

Dieser Abschnitt ist ein Platzhalter, um zu verdeutlichen, dass ab hier weitere nichtfunktionale Anforderungen zu beschreiben sind.

Querschnittskonzepte

Die Spezifikation der Querschnittskonzepte enthält fachliche und technische Anforderungen, die für die gesamte zu erstellende Anwendung gelten und in keinen der anderen Spezifikationsteile sinnvoll eingeordnet werden können oder mehrere, andere Spezifikationsteile zugleich betreffen würden.

Die folgenden Querschnittskonzepte finden sich in fast jedem größeren Projekt:

- *Protokollierung*
- *Berechtigungen, Zugriffsschutz und Sicherheit*
- *Historisierung*
- *Mandantenfähigkeit*
- *Mehrsprachigkeit*
- *Konfiguration*
- *Monitoring (System-Logs, Fehlermeldungen)*

Im Rahmen von Großprojekten können diese Querschnittskonzepte systemübergreifend erstellt,

gepflegt und für jedes System in der Anwendungslandschaft des Großprojekts wiederverwendet werden. Hierdurch ergibt sich sowohl Kostenersparnis als auch eine einheitlichere Umsetzung der Anwendungslandschaft.

Betriebliche Anforderungen

Betriebsanforderungen sind die Anforderungen, die der Betrieb an die zu entwickelnde Anwendung stellt. Sie dienen dazu, die Belange des späteren Betriebes schon frühzeitig bei der Systementwicklung zu berücksichtigen und spätere Nachbesserungen zu vermeiden.

Der Betrieb spielt eine gewichtige Rolle und trägt wesentlich zum Projekterfolg bei, sobald die Anwendung implementiert ist und schrittweise in den Produktivbetrieb überführt wird. Allerdings stehen zu Projektbeginn funktionale Aspekte der zu entwickelnden Anwendung im Vordergrund und die Belange des Betriebes werden oft übersehen. Dadurch entstehen später dann häufig Reibungen und aufwändiger Nachbesserungsbedarf.

Betriebsanforderungen unterscheiden sich nicht von anderen Anforderungen an die Anwendung. Die betrieblichen Anforderungen betreffen jedoch jede Anwendung der Anwendungslandschaft, und sind oft über verschiedene Anwendungen hinweg sehr ähnlich. Somit besteht bei den Betriebsanforderungen ein hohes Wiederverwendungspotenzial über Projektgrenzen hinweg.

Die Betriebsanforderungen umfassen in der Regel Anforderungen aus den folgenden Themengebieten:

- *Technische Plattform (relevant insbesondere für TI-Architektur)*
- *Schnittstellen zur Anwendung für den Betrieb*
- *Migration*
- *Überwachbarkeit, Konfigurierbarkeit, Wiederherstellbarkeit, Deployment*
- *Systemeinführung*
- *Betriebsdokumentation*
- *Art und Umfang von Datensicherungen, Sicherungszyklen, Aufbewahrungsfristen, etc.*

Die Betriebsanforderungen stehen oft in engem Zusammenhang mit anderen technischen, projektbezogenen oder querschnittlichen Anforderungen. Falls für diese anderen Anforderungen separate Spezifikationskapitel existieren, sollten sie eher an diesen anderen Stellen zusammenhängend beschrieben werden. Hier finden sich dann kurz erläuterte Querverweise auf jene Kapitel.

Offene Punkte

Die zur Abnahme des Dokuments noch zu klärenden offenen Punkte in Listenform.

Table 8. Beschreibungstext für diese Tabelle

Nummer	***Beschreibung***
1	*Ein offener Punkt*

Anhang

Glossar

Das Glossar definiert die wichtigen fachlichen und technischen Begriffe und Abkürzungen des Projekts. Zudem werden häufig verwendete fremdsprachige Begriffe übersetzt und erläutert. Das Glossar klärt somit Begriffsbedeutungen und beugt Missverständnissen vor. Es erleichtert das gemeinsame Verständnis von Fachlichkeit und Technik, sowie die Einarbeitung der Projektbeteiligten. Schließlich etabliert das Glossar eine gemeinsame Sprache aller Projektbeteiligten, sowohl im Entwicklungsprojekt als auch auf Kundenseite.

Beispielhaft ein Glossar zu den in diesem Dokument verwendeten Begriffen:

Table 9. Glossar

Begriff	Kategorie	Erläuterung	Synonym(e) oder Übersetzung(en)
Ablaufverfolgung	*Nichtfunktionale Anforderungen*	*Mit Ablaufverfolgung bezeichnet man in der Programmierung eine Funktion zur Analyse von Fehlersuche von Programmen. Dabei wird z.B. bei jedem Einsprung in eine Funktion, sowie bei jedem Verlassen eine Meldung ausgegeben, sodass der Programmierer mitverfolgen kann, wann und von wo welche Funktion aufgerufen wird. Die Meldungen können auch die Argumente an die Funktion enthalten. Zusammen mit weiteren Diagnose-Ausgaben lässt sich so der Programmablauf eines fehlerhaften Programmes häufig sehr schnell bis zu der fehlerverursachenden Funktion zurückverfolgen.*	*Tracing*
Akteur	*Abläufe*	*Ein Akteur ist ein außerhalb der Anwendung agierender Beteiligter, der den in einem Anwendungsfall beschriebenen Ablauf anstößt.*	
Aktivität	*Abläufe*	*Eine Aktivität ist eine Tätigkeit, die einen elementaren, logischen Schritt innerhalb eines Geschäftsvorfalls bildet. Sie wird von einem Akteur mit einem bestimmten Ziel ausgeführt. Eine Aktivität kann sowohl manuell als auch teilweise oder vollständig automatisiert (computer-unterstützt) ablaufen.*	

Begriff	Kategorie	Erläuterung	Synonym(e) oder Übersetzung(en)
Anwendungsfall	*Abläufe*	*Ein Anwendungsfall ist eine Abfolge zielgerichteter Interaktionen zwischen Akteuren und einer Anwendung.*	*Use Case*
Anwendungsfunktion	*Abläufe*	*Eine Anwendungsfunktion ist ein Ausschnitt eines Anwendungsfalls, der ohne Unterbrechung der Anwendung ausgeführt wird.*	
Anwendungskomponente	*Architektur*	*Eine Anwendungskomponente beschreibt eine Menge funktional zusammenhängender Anwendungsfälle.*	
Anwendungslandschaft	*Architektur*	*Die Anwendungslandschaft einer Organisation ist bestimmt durch die Menge der von der Organisationseinheit betriebenen Anwendungen.*	
Anwendung	*Architektur*	*glossary::terms-definitions.adoc*	
Assoziation	*Daten*	*Eine Assoziation beschreibt eine Beziehung zwischen zwei oder mehr Entitätstypen.*	*Beziehung*
Attribut	*Daten*	*Als „Attribut“ versteht man eine dem Entitätstyp zugeordnete Information.*	
Batch	*Benutzungsschnittstelle*	*Ein Batchprogramm realisiert eine eigenständige Verarbeitung ohne direkten Benutzereingriff während des Ablaufes.*	*Batchprogramm, Batchverarbeitung*
Business Process Diagram (BPD)	*Methodik*	*Diagramme in der BPMN heißen Business Process Diagram (BPD).*	
Business Process Modelling Notation (BPMN)	*Methodik*	*Die Business Process Modeling Notation (BPMN) ist eine grafische Spezifikationssprache in der Wirtschaftsinformatik. Sie stellt Symbole zur Verfügung, mit denen Fach- und Informatikspezialisten Geschäftsprozesse modellieren können.*	
Datenhaltung	*Architektur*	*Eine Datenhaltung ist eine physikalisch abgrenzbare Menge nicht-flüchtiger Daten.*	*Datensenke*

Begriff	Kategorie	Erläuterung	Synonym(e) oder Übersetzung(en)
Dialog	*Benutzungsschnittstelle*	*Der Begriff „Dialog“ bezeichnet die gesamte Benutzungsschnittstelle eines Anwendungsfalls (Masken, Zustände, Übergänge). Die Spezifikation eines Dialogs umfasst die Beschreibung seiner Masken und Maskenzustände sowie der Übergänge zwischen den Masken. Ein Dialog ist in sich abgeschlossen und gegenüber anderen Dialogen abgegrenzt.*	
Dialog-Gestaltungsvorgabe	*Benutzungsschnittstelle*	*Die Dialog-Gestaltungsvorgabe spezifiziert grundlegende Eigenschaften der Benutzerschnittstelle. Sie umfasst Grundprinzipien des Dialogaufbaus, des Layouts und der Benutzerinteraktion. Oft setzt die Dialog-Gestaltungsvorgabe unternehmensweit gültige Vorgaben für die zu entwickelnde Anwendung um. Sie vereinheitlicht die Benutzerschnittstelle und trägt wesentlich zur effizienten Dialogspezifikation bei.*	*Bedienkonzept*
Entität	*Daten*	*Als Entität wird ein eindeutig zu bestimmendes Objekt bezeichnet, dem Informationen zugeordnet werden. Die Objekte können materiell oder immateriell sein.*	*Informationsobjekt, Objekt*
Entitätstyp	*Daten*	*Jede Entität (das einzelne Objekt) wird einem Entitätstyp zugeordnet. Entitäten sind konkrete Ausprägungen eines Entitätstyps.*	*Klasse*
Fachlicher Datentyp	*Daten*	*Fachliche Datentypen werden verwendet, um Typ und Wertebereichsangaben von Attributen gruppieren zu können.*	

Begriff	Kategorie	Erläuterung	Synonym(e) oder Übersetzung(en)
Geschäftsprozess	*Abläufe*	*Ein Geschäftsprozess ist eine funktions- und stellenübergreifende Folge von Arbeitsschritten zur Erreichung eines geplanten Arbeitsergebnisses in einer Organisation (Unternehmen, Behörde, etc.). Er dient direkt oder indirekt zur Erzeugung einer Leistung für einen Kunden oder den Markt. Ein Geschäftsprozess kann sich aus Aufgaben im Sinn von elementaren Tätigkeiten (Aktivitäten) zusammensetzen.*	
Geschäftsvorfall	*Abläufe*	*Ein Geschäftsvorfall ist die Bündelung elementarer Tätigkeiten (Aktivitäten) innerhalb eines Geschäftsprozesses, die durch ein Ereignis ausgelöst werden.*	
Kernprozess	*Abläufe*	*Kernprozesse sind die wertschöpfenden Prozesse. Im Dienstleistungsbereich beschäftigen sich die Kernprozesse mit denjenigen Leistungen, die direkt von einem externen Kunden bezahlt werden.*	
Logisches Datenmodell	*Daten*	*Das logische Datenmodell einer Anwendung beschreibt die Struktur der permanent gespeicherten Daten aus fachlicher Sicht.*	
Logging	*Nichtfunktionale Anforderung*	*Unter Logging sind systemnahe und sicherheitsrelevante Meldungen zu verstehen für das Erkennen, Behandeln und Beheben von Fehlern, Analyse und Nachvollziehen von Systemereignissen und des Systemzustands und weitere systemspezifische Auswertungen.* *Es handelt sich nicht um Protokollierung!*	
Maske	*Benutzungsschnittstelle*	*Eine „Maske“ entspricht einem Bildschirmbereich zur Bearbeitung eines Arbeitsschritts eines Anwendungsfalls, z.B. ein Fenster. Eine Maske entspricht bei einer GUI meist genau einem Fenster.*	*Screen*

Begriff	Kategorie	Erläuterung	Synonym(e) oder Übersetzung(en)
Maskentyp	*Benutzungsschnittstelle*	*Maskentypen fassen Masken mit gleichartigem Verhalten zusammen (z.B. Eingabe).*	
Modellkomponente	*Daten*	*Eine Modellkomponente ist eine Gruppierung, die fachlich zusammengehörige Entitätstypen inklusive ihrer Assoziationen zusammenfasst. Sie ist überschneidungsfrei, d.h. ein Entitätstyp gehört zu genau einer Modellkomponente. In einer Modellkomponente kann allerdings ein Entitätstyp einer anderen Modellkomponente referenziert werden.*	
Nachbarsystem	*Architektur*	*glossary::terms-definitions.adoc*	
Nachbarsystemschnittstelle	*Architektur*	*Über eine Nachbarsystemschnittstelle werden zwischen zwei Anwendungen Daten ausgetauscht oder externe Dienste benutzt.*	*Schnittstellen*
Nichtfunktionale Anforderung	*Architektur*	*Während die funktionalen Anforderungen die geforderten Fähigkeiten der Anwendung beschreiben, stellen die nichtfunktionalen Anforderungen die zu erfüllenden Rahmenbedingungen für die Anwendung dar (z.B. Performanz, Verfügbarkeit).*	*Non-functional requirement*
Organisationseinheit	*Abläufe*	*Einheiten des Unternehmens, die eine Aktivität ausführen bzw. Personen, die in einer bestimmten Rolle am Prozess beteiligt sind.*	
Projekt	*Vorgehen*	*Ein Projekt ist ein Vorhaben, bei dem innerhalb einer definierten Zeitspanne ein definiertes Ziel erreicht werden soll, und das sich dadurch auszeichnet, dass es im Wesentlichen ein einmaliges Vorhaben ist.*	*IT-Projekt*

Begriff	Kategorie	Erläuterung	Synonym(e) oder Übersetzung(en)
Protokollierung	*Nichtfunktionale Anforderungen*	*Die Protokollierung erfasst Informationen zu fachlichen Abläufen, um diese zu einem späteren Zeitpunkt nachvollziehbar zu machen.* *Es handelt sich nicht um Logging!*	
Stützprozess	*Abläufe*	*Stützprozesse sind die unterstützenden Prozesse, die notwendig sind, um die Kernprozesse am Laufen zu halten. Externe Nutzer nehmen sie nicht wahr.*	
UML-Aktivitätsdiagramm	*Methodik*	*Das Aktivitätsdiagramm ist ein Verhaltensdiagramm. Es zeigt eine bestimmte Sicht auf die dynamischen Aspekte der modellierten Anwendung.*	
UML-Klassendiagramm	*Methodik*	*Ein Klassendiagramm ist in der Informatik eine grafische Darstellung von Entitätstypen sowie der Assoziationen zwischen diesen Entitätstypen.*	
UML-Komponentendiagramm	*Methodik*	*Das Komponentendiagramm ist ein Strukturdiagramm. Es zeigt eine bestimmte Sicht auf die Struktur der modellierten Anwendung.*	
Unified Modelling Language (UML)	*Methodik*	*Die Unified Modeling Language (UML) ist eine von der Object Management Group (OMG) entwickelte und standardisierte Sprache für die Modellierung von Software und anderen Anwendungen.*	

Fachliche Grundlagen

Die fachlichen Grundlagen dokumentieren Fakten und Hintergrundinformationen zum Anwendungsbereich. Es handelt sich dabei um solche Informationen, die für das Verständnis der Spezifikation und für die Entwicklung der Anwendung hilfreich sind, sich aber nicht in den Anforderungen direkt widerspiegeln.

Fachliche Grundlagen werden in diesem Abschnitt beschrieben und können aus allen anderen Abschnitten der Systemspezifikation referenziert werden.

Rollen und Berechtigungen

Das Kapitel Rollen und Berechtigungen dokumentiert, welche Rollen es in einem System gibt. Es liefert eine kurze Beschreibung, welche Bedeutung jede Rolle hat. Darüber hinaus muss aufgeführt werden, welche Rechte die jeweiligen Rollen in dem System haben (d.h. z. B. welchen Anwendungsfall ein Anwender mit einer bestimmten Rolle nutzen darf oder welche Maskenelemente diesem zur Verfügung stehen). Rollen können dabei Anwender oder Systeme repräsentieren. Außerdem dokumentiert das Kapitel, welche Rollen die jeweiligen Akteure eines Systems einnehmen können.

Für das Bestimmen und Benennen der Rollen der gibt es folgende Richtlinien.

Rollenschnitt

Die Rollen des Benutzerverzeichnisses sind in fachliche und technische Rollen aufgeteilt. Fachliche Rollen können im Gegensatz zu technischen Rollen über die Oberfläche des Benutzerverzeichnisses administriert werden. Technische Rollen können dafür als Unterrollen von anderen technischen oder fachlichen Rollen dienen.

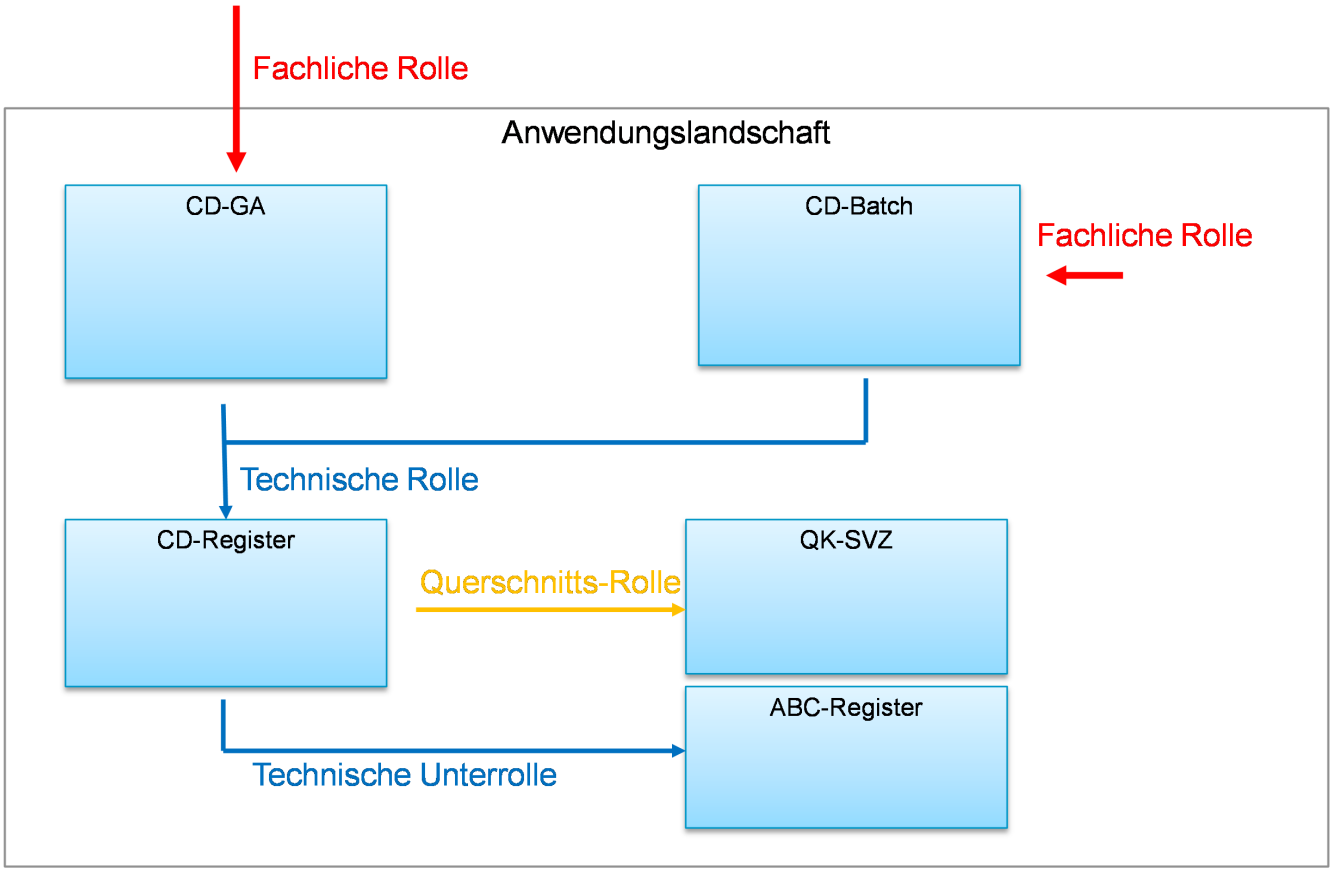


Figure 25. Rollenschnitt am Beispiel einer CD-Verwaltung

- *Fachliche Rollen werden für Anwendungen oder deren Bestandteile vergeben, sofern sie eine Autorisierung der Anfrage vornehmen (grafische Oberflächen und Service-Gateways) bzw. sich selbst autorisieren müssen (Batches und Systemtasks). Die einzelnen angebotenen Services werden über Rechte abgesichert.*
- *Technische Rollen sichern die Kommunikationswege innerhalb der Anwendungslandschaft ab. Sie werden für die Schnittstellen von Anwendungen verwendet, welche nur von anderen Anwendungen aufgerufen werden. Die einzelnen Schnittstellen werden durch Rechte abgesichert.*

- *Die Querschnittsrolle „QK_Nutzer“ ist eine einheitliche Rolle zur Absicherung von unkritischen Querschnittssysteme, deren Services von fast jeden Aufruf benötigt werden. Sie ist als Unterrolle von Zugang_Portal und Zugang_SGW definiert.*

Namenskonventionen für die Benennung

Die Benennung von Rollen muss fachlich getrieben sein. Rollen werden für eine fachliche Operation bzw. den Akteur angelegt. Grundsätzlich werden die Rollen in CamelCase-Schreibweise geschrieben, sofern der Name der Rolle nicht zu lang wird. In diesem Fall sollte auf eine Abkürzung des Rollennamens zurückgegriffen werden und ein sprechendes Label für die Administration der Rollen vergeben werden.

Das Schema zur Benennung einer fachlichen Rolle für GUI oder Service-Gateway ist:

<Fachlicher Systemname>_<Funktion>

Für <Fachlicher Systemname> wird der abgekürzte Name des Systems bzw. der Domäne aus der Spezifikation ohne Zusätze wie GA, Register eingesetzt, z. B. „CD_Auskunft“.

Das Schema zur Benennung einer fachlichen Rolle für einen Batch oder Task ist:

<Fachlicher Systemname>_SYSTEM_<Suffix>

Auch hier wird <Fachlicher Systemname> mit dem abgekürzten Namen des Systems aus der Spezifikation ohne Zusätze ersetzt, z. B. „CD_SYSTEM“ oder komplexer „CD_SYSTEM_ABLAGE“.

Das Schema zur Benennung einer technischen Rolle für interne Services ist:

<Technischer Systemname>_<Servicename>

<Technischer Systemname> wird mit dem abgekürzten Namen des Systems aus der Spezifikation ersetzt, inklusive der Bezeichnung um welche Art von System es sich handelt (Register, GA, usw.), z. B. „CD-GA_AntragEmpfangen“ oder „CD-REG_Nutzung“.

IMPORTANT	Prinzipiell sollten so wenig Rollen wie möglich und so viele wie nötig vergeben werden.

Eine beispielhafte Liste mit Rollen, deren Beschreibung und zugeordneten Rechten sieht wie folgt aus:

Rolle	***Beschreibung***	***Rechte***
Rolle_Nutzung	*Für den Zugriff auf das System ohne Administrationsfunktionalität.*	*Recht_1, Recht_2*
Rolle_Reporting	*Für die Verwaltung von Reports.*	*Recht_3*
Rolle_System	*Interne Rolle für den Systembenutzer.*	*Recht_4*
Rolle_Batch	*Interne Rolle für den Batchbenutzer.*	*Recht_5*

Eine beispielhafte Liste von Akteuren eines Systems und deren Rolle(n) sieht wie folgt aus:

Akteur	*Rolle*
AKR_Nutzer	*Rolle_Nutzung*
AKR_Reporter	*Rolle_Reporting*
AKR_Interner_Nutzer	*Rolle_System, Rolle_Batch*

Weiterführende Dokumente

Liste mit Referenzen auf weiterführende Dokumentation.

Kürzel	Beschreibung	Ablage
[Kürzel]	Eine Beschreibung des Dokuments	Verweis auf den Ablageort des Dokuments, z.B. eine URL.

[1] Siehe 40_Methodik\10_Systemspezifikation\IsyFact-Vorlage-Anforderungsliste.xlsx