ARIS

METHODENHANDBUCH

VERSION 10.0 - SERVICE RELEASE 27 AND HIGHER

OKTOBER 2024

This document applies to ARIS Version 10.0 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2010-2024 Software GmbH, Darmstadt, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software GmbH product names are either trademarks or registered trademarks of Software GmbH and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software GmbH and/or its subsidiaries is located at https://softwareag.com/licenses.

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software GmbH Products / Copyright and Trademark Notices of Software GmbH Products". These documents are part of the product documentation, located at https://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software GmbH.

Inhalt

Inhalt I

1 Einleitung 1

2 Architektur integrierter Informationssysteme (ARIS) 3
2.1 ARIS-Architektur-Konzept 3
2.2 Beschreibungssichten 3
2.3 Beschreibungsebenen 7

3 Modellierung innerhalb der Sichten und Ebenen des ARIS-Konzeptes 12
3.1 Funktionssicht 12
3.1.1 Fachkonzept 12
3.1.1.1 Funktionsbaum 13
3.1.1.2 Zieldiagramm 18
3.1.2 DV-Konzept - Anwendungssystemtypdiagramm 19
3.1.3 Implementierung - Anwendungssystemdiagramm 24
3.2 Datensicht 27
3.2.1 Fachkonzept 27
3.2.1.1 Das ERM-Grundmodell 27
3.2.1.2 Erweiterungen des ERM - eERM 32
3.2.1.3 Erweiterung um Konstruktionsoperatoren 32
3.2.1.4 Erweiterung der Kardinalitäten 37
3.2.1.5 Identifizierungs- und Existenzabhängigkeit 40
3.2.1.6 Modellierung von Fachbegriffen des Unternehmens-Fachbegriffsmodell 41
3.2.1.7 eERM-Attributzuordnungsdiagramm 43
3.2.1.8 IE-Datenmodell 44
3.2.1.9 Zusammenfassung der wichtigsten Begriffe und Darstellungsformen des eERM 45
3.2.1.10 Modellierung der Data Warehouse-Struktur 46
3.2.1.11 Projektmanagementdaten - Informationsträgerdiagramm 47
3.2.2 Implementierung - Tabellendiagramm 48
3.2.3 Rollenzuordnungsdiagramm (RAD - Role Allocation Diagram) 50
3.3 Organisationssicht 51
3.3.1 Fachkonzept 51
3.3.1.1 Organisatorische Gestaltung der Unternehmen 51
3.3.1.2 Organigramm 53
3.3.2 DV-Konzept - Netztopologie 58
3.3.3 Implementierung 60
3.3.3.1 Netzdiagramm 60
3.3.3.2 Modellierung des Materialflusses - Technische Ressourcen 62
3.4 Prozesssicht 65
3.4.1 Fachkonzept 65
3.4.1.1 Funktionen mit Organisation - EPK 65
3.4.1.2 Ereignissteuerung - ereignisgesteuerte Prozesskette (EPK) 66
3.4.1.3 Funktionszuordnungsdiagramm (I/O) 75
3.4.1.4 Ereignisdiagramm 77
3.4.1.5 Wertschöpfungskettendiagramm 78

3.4.1.6 Objektorientierte Modellierung 78
3.4.1.7 Prozessauswahlmatrix 79
3.4.1.8 Modellierung des Materialflusses 80
3.4.1.9 EPK (Materialfluss) 80
3.4.1.10 EPK (Spalten-/Zeilendarstellung) 81
3.4.1.11 Business Controls Diagram 82
3.4.1.12 E-Business scenario diagram 84
3.4.1.13 Gliederungsmodell 87
3.4.1.14 Rollendiagramm 88
3.4.1.15 Casual model 89
3.4.1.16 Maskendesign 91
3.4.1.17 Maskennavigation 93
3.4.1.18 Geschäftsfelder-Matrix 94
3.4.2 DV-Konzept 96
3.4.2.1 Zugriffsdiagramm 96
3.4.2.2 Funktionen mit Daten 96
3.4.2.3 Organisation mit Funktion 97
3.4.2.4 Programmablaufdiagramm 98
3.4.2.5 Programmablaufplan (PA) 99
3.4.2.6 Maskendiagramm 100
3.4.3 Implementierung - Zugriffsdiagramm (physikalisch) 102
3.4.3.1 Funktionen mit Daten 102
3.4.3.2 Organisation mit Daten 104
3.4.3.3 Organisation mit Funktionen 105
3.5 Leistungsmodellierung 108
3.5.1 Leistungsaustauschdiagramm 109
3.5.2 Leistungsbaum 110
3.5.3 Produktzuordnungsdiagramm 111
3.5.4 Produktbaum 114
3.5.5 Produktauswahlmatrix 115

4 Unified Modeling Language (UML) in ARIS 116
4.1 Einleitung 116
4.2 ARIS UML Designer - Unterstützter UML-Standard 116

5 Methoden für das Wissensmanagement 117
5.1 Einleitung 117
5.2 Objekttypen zur Modellierung der Wissensverarbeitung 118
5.2.1 Wissenskategorie 118
5.2.2 Dokumentiertes Wissen 121
5.3 Modelltypen zur Modellierung der Wissensverarbeitung 121
5.3.1 Wissensstrukturdiagramm 121
5.3.2 Wissenslandkarte 123
5.3.3 Darstellung der Wissensverarbeitung in Geschäftsprozessen 124

6 Balanced Scorecard-Methode 126
6.1 Einleitung 126
6.2 Das Konzept der Balanced Scorecard 126
6.2.1 Kernaussagen des BSC-Ansatzes 126
6.2.2 Der strategische Managementprozess und die Balanced Scorecard 127
6.2.2.1 Formulierung und Übersetzen der Vision und der Strategie 128

6.2.2.2 Die Standardperspektiven einer Balanced Scorecard 129
6.2.2.3 Ursache-Wirkungs-Kette 130
6.2.2.4 Definition der Früh- und Spätindikatoren 131
6.2.2.5 Kommunikation und Ableiten weiterer Scorecards 132
6.2.2.6 Planung und Vorgaben 132
6.2.2.7 Strategisches Lernen und Feedback 134
6.2.3 Vorteile und Nutzen der Balanced Scorecard 134
6.3 Entwicklung einer Balanced Scorecard mit ARIS BSC 135
6.3.1 Begriffe und Abkürzungen 135
6.3.2 Aufbau der Balanced Scorecards mit ARIS BSC 137
6.3.2.1 Perspektiven festlegen 137
6.3.2.2 Struktur des Balanced Scorecard-Systems festlegen 137
6.3.2.3 Ursache-Wirkungs-Beziehungen festlegen 138
6.3.2.4 Maßnahmen und Kennzahlen zur Zielkontrolle festlegen 141
6.3.2.5 Kennzahlen und deren Zusammenhänge beschreiben 144
6.3.3 Beziehungen zu anderen Modellen 145

7 eBusiness Scenario Diagram 146
7.1 Einleitung 146
7.2 Die Methode des eBusiness Scenario Diagrams 148
7.2.1 Die Idee 148
7.2.2 Das Modell und seine Objekte 148
7.2.3 Attributgruppe Übertragungsart 150
7.3 Auswertungen durch Reporte 150
7.3.1 Überprüfung der Datensicherheit 150
7.3.2 Systemunterstützung 151
7.3.3 Informationsfluss 151
7.3.4 Collaborative Business Maps 151
7.4 Verbindung zu anderen Methoden und Komponenten 151

8 IT City Planning 156
8.1 Enterprise Architecture und IT City Planning 156
8.2 An welche Unternehmen richtet sich IT City Planning? 156
8.3 IT City Planning mit ARIS 157
8.4 Service-Sicht 159
8.5 Servicetypen und ihre Daten 162
8.6 Detailbeschreibung der Servicetypen 162
8.7 Zeitlich-logische Abläufe zwischen IS-Elementen 163
8.8 IT-Sicht 164
8.9 IT-Elemente und ihre Daten 165
8.10 Detailbeschreibung der IT-Elemente 165
8.11 Organisatorische Aspekte 166
8.12 Zeitlich-logische Abläufe zwischen IT-Elementen 166
8.13 Zeitlich-logische Abläufe innerhalb der Architektur 167
8.14 Mögliche Auswertungen 168

9 Business Process Modeling 169
9.1 Prozessklassen und das Business process diagram 169
9.2 Realisierung der BPMN in ARIS 171
9.3 Die Elemente des Business process diagram 172
9.3.1 Pools und Lanes 172

9.3.2 Modellierungsrichtlinien für Pools und Lanes ... 174
9.3.3 Sequence flow ... 174
9.3.4 Modellierungsrichtlinien für Sequence flow-Kanten ... 174
9.3.5 Message flow ... 175
9.3.6 Modellierungsrichtlinien für Message flow-Kanten ... 175
9.3.7 Association ... 176
9.3.8 Ereignisse ... 177
9.3.9 Modellierungsrichtlinien für Ereignisse ... 178
9.3.10 Aktivitäten ... 179
9.3.11 Modellierungsrichtlinien für Aktivitäten ... 180
9.3.12 Gateway ... 181
9.3.13 Modellierungsrichtlinien für Gateways ... 182
9.3.14 Artifact ... 184
9.3.15 Quellenangabe zu den Abbildungen ... 186

10 Modeling BPMN 2.0 ... 187
10.1 Introduction ... 187
10.1.1 Initial situation and objective ... 187
10.1.2 Purpose of this chapter ... 187
10.2 BPMN core elements and their implementation in ARIS ... 188
10.2.1 Infrastructure ... 188
10.2.2 Foundation ... 189
10.2.3 Common Elements ... 191
10.2.3.1 Artifacts ... 191
10.2.3.2 Association ... 195
10.2.3.3 Group ... 195
10.2.3.4 Text annotation ... 196
10.2.3.5 Callable Elements ... 197
10.2.3.6 Event ... 197
10.2.3.7 Expression ... 198
10.2.3.8 Flow Element ... 198
10.2.3.9 Flow Elements Container ... 199
10.2.3.10 Gateways ... 200
10.2.3.11 Message ... 201
10.2.3.12 Message flow ... 202
10.2.3.13 Participant ... 204
10.2.3.14 Resource ... 207
10.2.3.15 Sequence flow ... 207
10.2.3.16 Elements not included in the current implementation ... 211
10.3 BPMN diagrams and ARIS model types: An overview ... 212
10.4 Process ... 213
10.4.1 Activities ... 215
10.4.1.1 Resource assignment ... 218
10.4.1.2 Performer ... 218
10.4.1.3 Activity type: Task ... 218
10.4.1.4 Human interactions ... 223
10.4.1.5 Activity type: Subprocess ... 223
10.4.1.6 Subprocess type: Subprocess ... 224
10.4.1.7 Subprocess type: Event subprocess ... 226
10.4.1.8 Subprocess type: Transaction ... 227
10.4.1.9 Subprocess type: Ad hoc subprocess ... 228
10.4.1.10 Subprocess type: Call Activity ... 230

10.4.1.11 Global task 232
10.4.1.12 Loop characteristics 232
10.4.1.13 Loop characteristics representations 232
10.4.1.14 Standard and multi-instance loop characteristics and complex behavior definition 233
10.4.2 Items and Data 237
10.4.2.1 Data object 237
10.4.2.2 Data store 239
10.4.3 Events 241
10.4.3.1 Catch events and throw events 243
10.4.3.2 Start event 245
10.4.3.3 Intermediate events 246
10.4.3.4 End event 247
10.4.3.5 Event definitions 248
10.4.4 Gateways 256
10.4.4.1 Exclusive gateway 257
10.4.4.2 Inclusive gateway 257
10.4.4.3 Parallel gateway 259
10.4.4.4 Complex gateway 259
10.4.4.5 Event-based gateways 260
10.4.5 Lanes 261
10.5 Collaboration 263
10.5.1 Pool and participant 264
10.5.2 Object types and connection types reused from a process 265
10.5.3 Message flow 265
10.6 Conversation 266
10.6.1 Conversation container 266
10.6.2 Conversation nodes 267
10.6.3 Participant 268
10.6.4 Artifacts 269
10.6.5 Conversation link 269
10.6.6 Message flow in a conversation 270
10.6.7 Model assignments 270
10.7 Enterprise BPMN collaboration diagram 271

11 Customer Experience Management (CXM) 272
11.1 Customer journey landscape 272
11.2 Customer journey map 274
11.3 Customer touchpoint-Zuordnungsdiagramm 277
11.4 Customer touchpoint map 278
11.5 Verknüpfung von CXM und BPM 279
11.5.1 Analysemöglichkeiten 280
11.5.1.1 Report 280
11.5.1.2 Abfragen 282
11.5.1.3 Gesamtübersicht einer oder mehrerer Customer journeys erzeugen 282
11.5.1.4 Zum Risiko gehörende Customer touchpoints 284
11.5.1.5 Zur Verantwortlichkeit gehörende Customer touchpoints 284
11.5.1.6 Zum Channel gehörende Customer touchpoints 285
11.5.1.7 Risiken und strategische Initiativen aller Customer touchpoints 286

11.5.1.8 Risiken und strategische Initiativen mangelhafter Customer touchpoints 287
11.5.1.9 Mit Customer journeys verbundene Prozesse 288

12 Anwendungsfälle 289
12.1 Allgemeine Unternehmensdokumentation 292
12.2 Datenbankmanagement/ Data Warehousing 293
12.3 PC-Hardware- und Netzwerkmanagement 294
12.4 Prozesskostenrechnung 295
12.5 Qualitätsmanagement 296
12.6 Reorganisationsmaßnahmen 297
12.7 SAP R/3-Einführung 299
12.8 Software-Entwicklung und -einführung 300
12.9 Wissensmanagement 302
12.10 Workflow-Management 303

13 Literaturverzeichnis 304
13.1 Allgemeine Literaturliste 304
13.2 Themenbezogene Literaturliste 306
13.2.1 Unified Modeling Language in ARIS 306
13.2.1.1 UML-Spezifikation 306
13.2.1.2 Anwendung von UML 306
13.2.1.3 UML und Geschäftsprozessmodellierung 306
13.2.2 Methoden für das Wissensmanagement 307
13.2.2.1 Wissensmanagement allgemein 307
13.2.2.2 ARIS-Einsatz für das Wissensmanagement 307
13.2.3 Balanced Scorecard-Methode 307
13.2.4 IT City Planning 307
13.2.5 Business Process Modeling 307

14 Rechtliche Informationen 308
14.1 Dokumentationsumfang 308
14.2 Support 308

1 Einleitung

Früher lag das größte Optimierungspotenzial im Systemaufbau und in der Systemintegration. Der Fokus hat sich in den letzten Jahren allerdings mehr und mehr zur Lösungsentwicklung für besondere Anforderungen der einzelnen Branchen verschoben. Dadurch, dass dezentralisierte Informationssysteme verfügbar wurden und dass diese in integrierte Informationssysteminfrastrukturen eingegliedert werden konnten, eröffnete sich neues Einsparpotential hinsichtlich der Organisationsgestaltung von Unternehmen.

Organisationsstrukturen wurden früher funktionell heruntergebrochen und zentral aufgebaut, da diese meistens auf zentralisierten Host-Umgebungen mit wenig Kapazitäten gründeten. Dadurch fehlte es den Unternehmen an Flexibilität. Zu Beginn erkannten oder beachteten nur wenige die neuen Chancen, die sich durch die steigende Dezentralisierung von Rechnerdienstleistungen und der parallelen Entwicklung neuer Konzepte für Informationssystemarchitekturen boten (z. B. Client/Server, Workflow-Management).

Heutzutage ist dieses Potential durch den ständig zunehmenden Wettbewerb zum wichtigsten Thema für jedes einzelne Unternehmen geworden. Flexible Strukturen, die sich anhaltend auf interne Geschäftsprozesse konzentrieren stellen mittlerweile den entscheidenden Wettbewerbsfaktor für die Unternehmen dar. Dennoch kann nur eine ganzheitliche Betrachtung aller Geschäftsprozesse ein Unternehmen in die Lage versetzen, durch eine optimierte Infrastruktur der Informationssysteme zusammenhängende Prozesse zu erkennen, zu bündeln und zu unterstützen. Im Vergleich zur Verwaltung von zentralisierten Unternehmensumgebungen ist die Verwaltung dieser neuen Strukturen deutlich komplexer. Das Angehen dieser Herausforderung erfordert eine eindeutige Zuordnung der Verantwortlichkeiten, ein Maximum an Transparenz der Strukturen, homogene Kommunikation auf allen Unternehmensebenen und ein optimiertes Projektmanagement auf der Grundlage von definierten Geschäftszielen.

Zur Erfüllung dieser komplexen Aufgaben erhalten Führungskräfte Unterstützung durch Methoden zur Unternehmensmodellierung. Eine entscheidende Voraussetzung für die Analyse von Geschäftsprozessen sind Unternehmensmodelle, da durch diese eine Ausrichtung der Projekte auf die allgemeinen Geschäftsziele erreicht werden kann sowie Infrastrukturen von Informationssystemen verwendet werden, die mit zusammengesetzten, verteilten und integrierten Systemen für die optimale Unterstützung dieser schlanken Organisationsstrukturen sorgen.

Daher rückt eine Modellierung der tatsächlichen Unternehmenssituation - und damit auch eine Untersuchung der ganzheitlichen Geschäftsprozesse - immer mehr in den Mittelpunkt der Diskussion. Die Vielfältigkeit und die steigende Anzahl an Modellierungsmethoden führte bisher eher zu Komplexität und Verwirrung. Folglich hat man sich bemüht, standardisierte Rahmenkonzepte (Architekturen) für die Entwicklung und für die Modellierungsmethoden zu definieren.

Eine dieser Architekturen stellt die **Architektur integrierter Informationssysteme** (**ARIS**®) dar, die von Scheer entwickelt wurde (siehe Scheer, Architektur integrierter Informationssysteme, 1992). Durch diese Architekturkonzepte wird eine Bewertung und Organisation der Methoden mit dem Hauptaugenmerk auf den zentralen Punkten möglich. Außerdem dient sie als Orientierungshilfe für komplexe Entwicklungsprojekte, da ihre Strukturelemente gleichzeitig ein Vorgehensmodell für die Entwicklung integrierter Informationssysteme darstellen.

Eine derartige Struktur führt unwillkürlich zu einer Vereinheitlichung bei der Verwendung der Methoden. Auf der Grundlage dieser Architektur werden bestehende und neue Modellierungsmethoden zusammengefasst, um eine ganzheitliche Methode für die Modellierung von Geschäftsprozessen zu bilden.

Die ARIS-Architektur integriert außerdem Produkte wie ARIS Architect in der Produktpalette von Software GmbH. Diese Produkte unterstützen Berater und Unternehmen bei der Erstellung, Analyse und Bewertung ihrer Geschäftsprozesse im Sinne einer Geschäftsprozess-Restrukturierung. Mit ARIS Designer haben Sie eine bequeme Möglichkeit, die relevanten Geschäftsprozesse in den operativen Fachabteilungen zu erfassen und zu modellieren.

In diesem Handbuch erhalten Sie eine erste Einführung in die entsprechenden Modellierungsmethoden. Außerdem werden Ihnen Ansätze und Methoden vorgestellt, die die volle Bandbreite der ARIS-Produkte inklusive ihrer Systemerweiterungen nutzen. In diesem Handbuch erhalten Benutzer außerdem eine optimale Unterstützung bei der Verwendung von Modellierungsmethoden, ohne dass Fragen oder Probleme im Hinblick auf die Verwendung der Werkzeuge offen bleiben.

2 Architektur integrierter Informationssysteme (ARIS)

2.1 ARIS-Architektur-Konzept

Die Konzeption der **AR**chitektur integrierter **I**nformations**S**ysteme (ARIS basiert auf einem Integrationskonzept, das aus einer ganzheitlichen Betrachtung von Unternehmensprozessen abgeleitet wird. Zur Herleitung der Architektur wird zunächst ein Modell für Unternehmensprozesse entwickelt, das alle wesentlichen Merkmale zur Beschreibung von Geschäftsprozessen beinhaltet. Die hohe Komplexität des dabei entstehenden Modells wird durch die Zerlegung in einzelne Sichten reduziert. Die Sichtenzerlegung ermöglicht es nun, die Beschreibung der Sichteninhalte durch spezielle, für diese Sicht geeignete Methoden durchzuführen, ohne die vielfältigen Beziehungen und Zusammenhänge zu den anderen Sichten berücksichtigen zu müssen. Danach werden die Beziehungen zwischen den Sichten aufgenommen und redundanzfrei zu einer Gesamtbetrachtung der Prozessketten zusammengeführt.

Der zweite Ansatz, der ebenfalls zur Reduktion der Komplexität beiträgt, ist die Betrachtung verschiedener Beschreibungen. Dabei werden einem Life-Cycle-Konzept folgend die unterschiedlichen Beschreibungsmethoden für Informationssysteme nach ihrer Nähe zur Informationstechnik unterschieden. Dadurch wird eine durchgängige Beschreibung von der betriebswirtschaftlichen Problemstellung bis hin zur technischen Umsetzung sichergestellt.

Das (ARIS)-Konzept bildet somit einen Rahmen, in dem integrierte Informationssysteme entwickelt, optimiert und ihre Umsetzung beschrieben werden können. Dabei führt vor allem die Betonung der fachlichen Beschreibungsebene dazu, dass das ARIS-Konzept als Orientierung bei der Erstellung, Analyse und Auswertung von betriebswirtschaftlichen Prozessketten dient. Eine genauere Beschreibung der Architektur integrierter Informationssysteme gibt Scheer (vgl. Scheer, Architektur integrierter Informationssysteme, 1992, und Scheer,ARIS Vom Geschäftsprozess zum Anwendungssystem, 1998).

2.2 Beschreibungssichten

Ausgangspunkt der Betrachtung ist ein Unternehmensprozess, wie er in folgender Abbildung dargestellt ist.

Der Prozess wird durch das Ereignis **Kundenauftrag ist eingetroffen** ausgelöst. Dieses Ereignis stößt die Funktion (Vorgang) **Kundenauftrag annehmen** an. Zur Durchführung dieses Vorgangs werden Zustandsbeschreibungen der relevanten Umwelt des Vorgangs benötigt. Hier sind insbesondere Daten über den Kunden und die Artikel relevant. Die Vorgangsbearbeitung kann zu einer Zustandsänderung der Umweltobjekte führen, z. B. können die Lagerbestandsdaten der Artikel mit den neuen Reservierungsdaten aktualisiert werden.

Die Vorgänge werden durch Sachbearbeiter durchgeführt, die wiederum Abteilungen zugeordnet werden können. Die Abteilung greift zur Durchführung ihrer Aufgaben auf bestimmte Ressourcen der Informationstechnik (Personal Computer, Drucker usw.) zu.

Mit Abschluss des Vorgangs **Kundenauftrag annehmen** tritt das Ereignis **Auftrag ist bestätigt** ein, das nun weitere Vorgänge zur Folge hat, z. B. **Produktionsplan erstellen** und **Auftrag verfolgen**. Das Objekt **Auftrag** befindet sich nun in einem neuen Zustand, denn aus dem Objekt **Eingetroffener Auftrag** ist ein **Bestätigter Auftrag** geworden. Durch das Ausführen der Funktion **Kundenauftrag annehmen** wurde eine Leistung erbracht, die bei der Bearbeitung der folgenden Vorgänge, zusammen mit personellen und sachlichen Ressourcen, als Input verwendet wird.

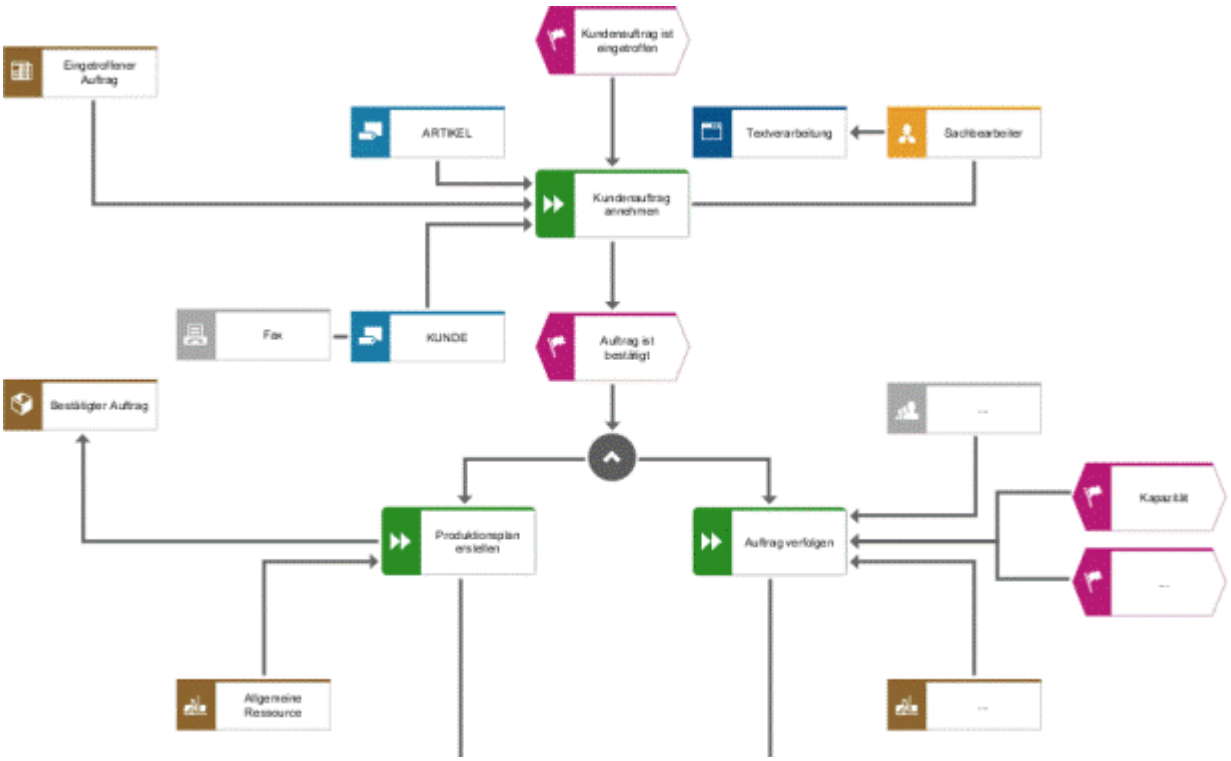


Abbildung 1: Modell eines Unternehmensprozesses

Die zur vollständigen Beschreibung eines Unternehmensprozesses benötigten Komponenten sind somit Vorgänge, Ereignisse, Leistungen (Zustände), Bearbeiter, Organisationseinheiten und Ressourcen der Informationstechnik. Müssten je betrachteten Vorgang alle Auswirkungen auf alle Elemente des Prozesses betrachtet werden, würde dies das Modell sehr komplizieren und auch zu Redundanzen in der Beschreibung führen.

Um diese Komplexität zu reduzieren, wird der Gesamtzusammenhang in einzelne Sichten zerlegt (vgl. nächste Abbildung), die eigene Modellierungs- und Entwurfsfelder darstellen (vgl. Scheer, Architektur integrierter Informationssysteme 1992, S. 13 ff.). Diese können zunächst weitgehend unabhängig voneinander bearbeitet werden. Die Zerlegung der Sichten erfolgt derart, dass die Beziehungen der Komponenten innerhalb einer Sicht sehr hoch sind, zwischen den Sichten jedoch nur eine relativ lose Kopplung besteht.

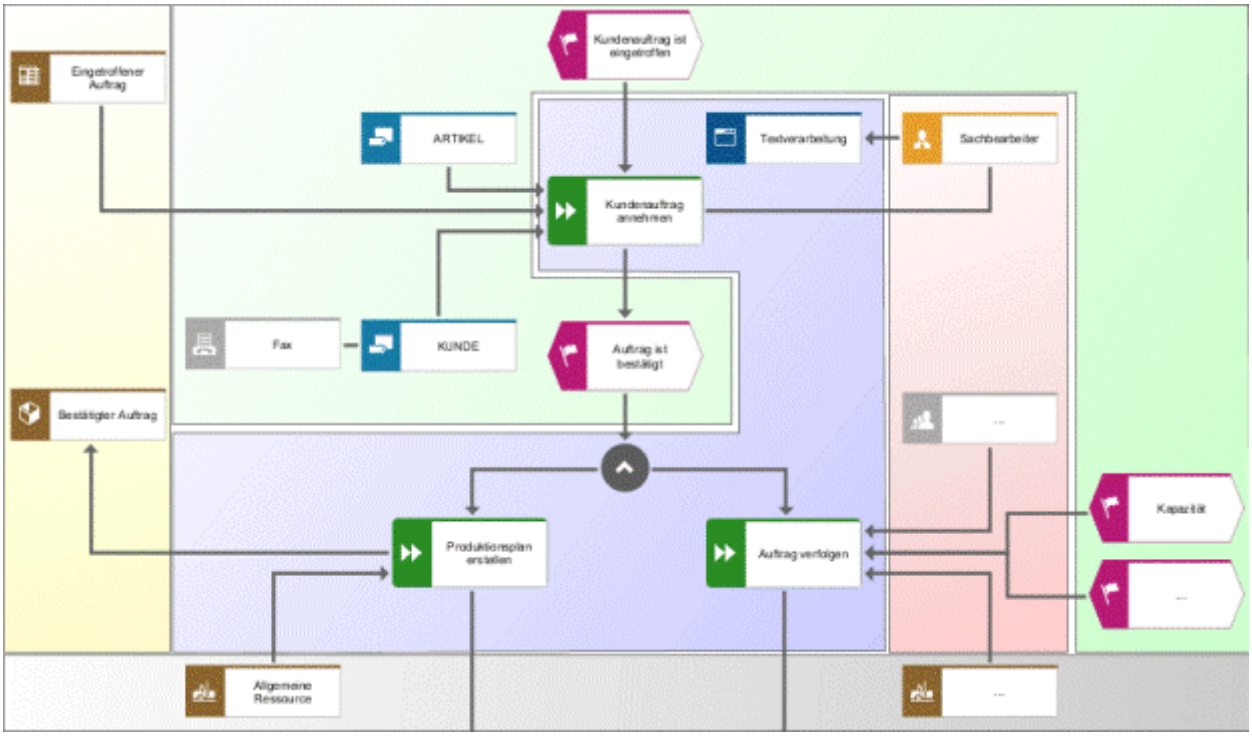


Abbildung 2: Sichten des Prozessmodells

Ereignisse wie **Kundenauftrag ist eingetroffen**, **Rechnung ist erstellt** definieren das Eintreten einer Zustandsänderung von Informationsobjekten (Daten). Sie werden in der Datensicht der ARIS-Architektur beschrieben.

Die Zustände im Umfeld der Objekte, z. B. im Umfeld des Auftrags, werden durch Leistungen repräsentiert. Eine Leistung kann entweder eine Sachleistung oder eine Dienstleistung sein. Dienstleistungen, die Informationen erzeugen und bereitstellen, sind Informationsdienstleistungen. Das Bereitstellen von finanziellen Mitteln gehört ebenfalls zu den Leistungen. Zusammenhänge zwischen Leistungen werden in der **Leistungssicht** der ARIS-Architektur beschrieben.

Die auszuführenden Funktionen (Vorgänge) sowie ihre Zusammenhänge untereinander bilden die zweite Sicht, die **Funktionssicht**. Sie beinhaltet die Beschreibung der Funktionen, die Aufzählung der einzelnen Teilfunktionen die zum Gesamtzusammenhang gehören, sowie die zwischen den Funktionen geltenden Anordnungsbeziehungen.

In der **Organisationssicht** werden die Bearbeiter und Organisationseinheiten sowie deren Beziehungen und Strukturen zusammengefasst.

Die Ressourcen der Informationstechnik bilden ein weiteres Betrachtungsfeld, die **Ressourcensicht**. Diese ist aber für die fachliche Betrachtung eines Geschäftsprozesses nur insoweit von Bedeutung, als sie Rahmenbedingungen für die Beschreibung der anderen, stärker betriebswirtschaftlich ausgerichteten Komponenten bildet. Aus diesem Grunde werden die Komponentenbeschreibungen der anderen Sichten (Daten, Funktionen, Organisation) in Abhängigkeit von der Nähe zu den Ressourcen der Informationstechnik beschrieben. Die Ressourcen werden somit auf den Beschreibungen DV-Konzept und Implementierung der anderen Sichten behandelt (vgl. Kapitel **Beschreibungsebenen** (Seite 7)). Das durch die Ebenenbetrachtung definierte Life-Cycle-Modell ersetzt somit die Ressourcensicht als eigenes Beschreibungsfeld.

Mit der Zerlegung des Prozesses in einzelne Sichten wird zwar das Ziel der Komplexitätsreduzierung erreicht, allerdings gehen die Zusammenhänge der Prozesselemente zwischen den Sichten verloren. Aus diesem Grunde wird eine weitere Sicht, die **Steuerungssicht** aufgenommen, in der die Verbindungen zwischen den Sichten beschrieben werden. Die Aufnahme dieser Beziehungen in einer eigenen Sicht ermöglicht es, alle Beziehungen systematisch und redundanzfrei zu erfassen.

Die Steuerungssicht ist eine wesentliche Komponente von ARIS. Hierin liegt auch der hauptsächliche Unterschied des ARIS-Konzeptes zu anderen Architekturvorschlägen (zum Vergleich mit anderen Architekturvorschlägen s. Scheer, Architektur integrierter Informationssysteme, S. 24 ff.).

Es ergeben sich somit insgesamt die fünf ARIS-Sichten, denen auch in den weiteren Methodenbeschreibungen gefolgt wird.

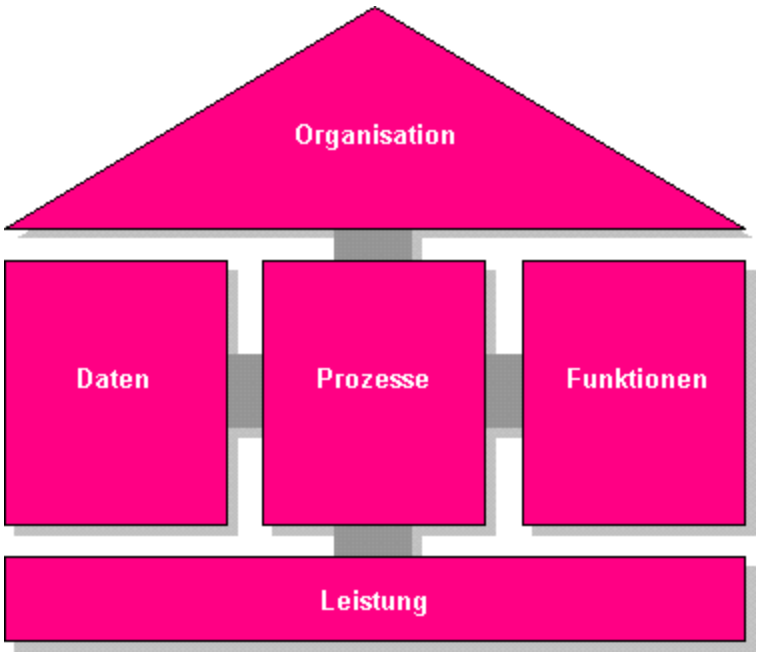


Abbildung 3: Zerlegungssichten des Prozessmodells

2.3 Beschreibungsebenen

Wie bereits erläutert wurde, wird die Ressourcensicht in ARIS durch ein Life-Cycle-Konzept der Beschreibungen eines Informationssystems aufgelöst.

Life-Cycle-Modelle in Form von Stufen- oder Phasenkonzepten beschreiben den Lebenslauf eines Informationssystems. Das Life-Cycle-Modell von ARIS hat jedoch nicht die Bedeutung eines Vorgehensmodells zur Entwicklung eines Informationssystems. Es definiert vielmehr unterschiedliche Beschreibungen, die sich in ihrer Nähe zur Informationstechnik unterscheiden.

Dabei unterscheidet ARIS die in der folgenden Abbildung gezeigte dreistufige Unterteilung (vgl. Scheer, Architektur integrierter Informationssysteme 1992, S. 16 f.).

Ausgangspunkt der Betrachtung ist die **betriebswirtschaftliche Problemstellung**. Die Beschreibung umfasst hier grobe Tatbestände, die sehr nahe an den fachlichen Zielsetzungen und der fachlichen Sprachwelt orientiert sind. Dabei werden auch die Möglichkeiten der Informationstechnik zur Unterstützung betriebswirtschaftlicher Prozesse und Entscheidungen mit einbezogen. Zur Darstellung werden deshalb auch nur halbformale Beschreibungsmethoden eingesetzt. Sie können aufgrund ihrer fehlenden Detailliertheit und des in der Fachwelt spezifischen Vokabulars nicht als Ausgangspunkt einer formalisierten Umsetzung in die Implementierung dienen.

Deshalb wird in einem **Fachkonzept** das zu unterstützende betriebswirtschaftliche Anwendungskonzept in einer soweit formalisierten Beschreibungssprache dargestellt, sodass es Ausgangspunkt einer konsistenten Umsetzung in die Informationstechnik sein kann. Dieser Vorgang wird auch als (semantische) Modellierung bezeichnet. Das Fachkonzept ist sehr eng mit der betriebswirtschaftlichen Problemstellung verbunden, wie es die Breite des Pfeils in der nächsten Abbildung ausdrückt.

Wird die Begriffswelt des Fachkonzeptes in die Kategorien der DV-Umsetzung übertragen, erreicht man die Ebene des **DV-Konzeptes**. Hier werden anstelle von fachlichen Funktionen die sie ausführenden Module oder Transaktionen definiert. Diese Ebene kann auch als Anpassung der Fachbeschreibung an generelle Beschreibungskonstrukte der Informationstechnik bezeichnet werden. Fachkonzept und DV-Konzept sind dabei nur lose gekoppelt. Ein DV-Konzept kann geändert werden, ohne dass dies Auswirkungen auf das Fachkonzept hat. Dies bedeutet jedoch nicht, dass Fachkonzept und DV-Konzept isoliert voneinander entwickelt werden können. Vielmehr soll nach Abschluss der fachkonzeptionellen Beschreibungen der betriebswirtschaftliche Inhalt so festgelegt sein, dass rein DV-bezogene Argumente wie das Leistungsverhalten eines Informationssystems keinen Einfluss auf die Fachinhalte nehmen.

Auf der Ebene der **Implementierung** wird das DV-Konzept auf konkrete hardware- und softwaretechnische Komponenten übertragen. Hiermit wird die Verbindung zur Informationstechnik hergestellt.

Die Beschreibungen sind durch unterschiedliche Änderungszyklen gekennzeichnet. Die Änderungsfrequenz ist auf der Ebene der Fachkonzepte am geringsten, auf der Ebene der Implementierung am höchsten.

Die Ebene der Implementierung ist eng an die Entwicklung der Informationstechnik gekoppelt und erfährt deshalb durch deren schnelle Innovationszyklen ebenfalls ständige Änderungen.

Die Ebene des Fachkonzeptes ist von besonderer Bedeutung, da sie einmal Träger des langfristig geltenden betriebswirtschaftlichen Anwendungskonzeptes ist und gleichzeitig den Ausgangspunkt für die weiteren Generierungsschritte zur Umsetzung in die technische Implementierung darstellt. Fachkonzepte besitzen die größte Lebensdauer und dokumentieren durch ihre enge Nähe zur betriebswirtschaftlichen Problemstellung auch den fachlichen Nutzen des Informationssystems. Aus diesem Grunde steht die Sicht der Entwicklung von Fachkonzepten oder semantischen Modellen im Vordergrund. Die semantischen Modelle bilden die Verbindung zwischen dem Anwender und der ersten Umsetzung seiner Problemstellung in eine DV-nahe Sprache.

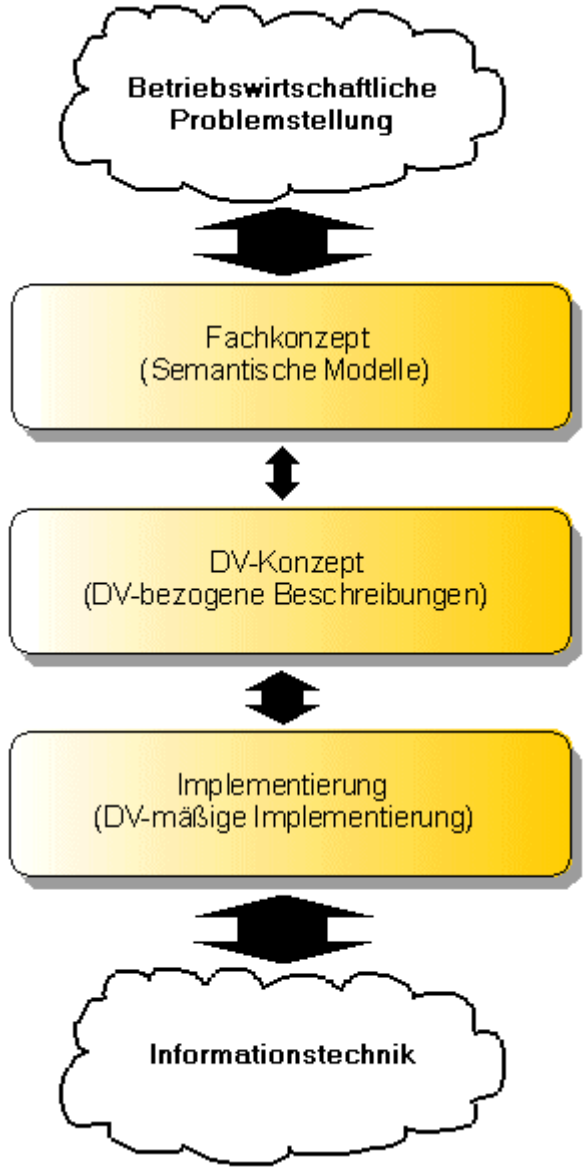


Abbildung 4: Beschreibungen eines Informationssystems

Mit der Sichtenbildung und den Beschreibungen einschließlich der betriebswirtschaftlichen Ausgangslösung ist das ARIS-Konzept entwickelt. Jede der Beschreibungssichten wird, wie in der folgenden Abbildung gezeigt, in den drei Ebenen **Fachkonzept**, **DV-Konzept** und **Implementierung** beschrieben.

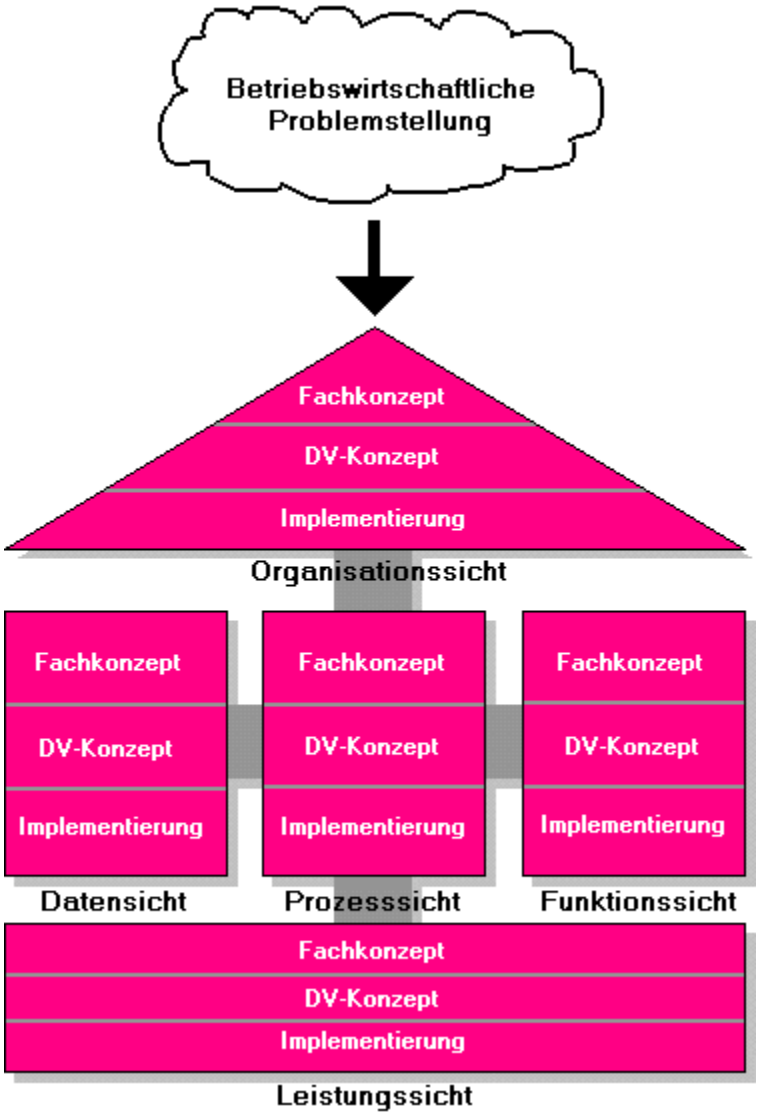


Abbildung 5: ARIS-Konzept

Mit dem entwickelten ARIS-Konzept liegen nun die Beschreibungsfelder, definiert durch die Beschreibungssichten und -ebenen der Architektur fest. Es ergeben sich somit einschließlich der als Ausgangspunkt der Betrachtung dienenden betriebswirtschaftlichen Problemstellung dreizehn Komponenten. Für jedes Betrachtungsfeld gilt es nun, die geeigneten Beschreibungsmethoden auszuwählen und zu erläutern.

Als Auswahlkriterien der Methoden gelten hierbei (vgl. Scheer, Wirtschaftsinformatik 1994, S. 18):

- die Einfachheit und Verständlichkeit der Darstellungsmittel,
- die Eignung für die speziell auszudrückenden Inhalte,
- die Möglichkeit, für alle darzustellenden Anwendungen einheitliche Methoden einsetzen zu können,
- der vorhandene oder zu erwartende Bekanntheitsgrad der Methoden sowie
- die weitgehende Unabhängigkeit der Methoden von technischen Entwicklungen der Informationstechnik.

Die einzelnen Methoden der Beschreibungsfelder werden nachfolgend beschrieben.

3 Modellierung innerhalb der Sichten und Ebenen des ARIS-Konzeptes

3.1 Funktionssicht

3.1.1 Fachkonzept

Modellierungsmethoden betrachten Funktionen oft im Zusammenhang mit Objekten der anderen Beschreibungssichten von ARIS. So wird z. B. die Verbindung von Daten und Funktionen dargestellt, um den Transformationsprozess einer Funktion über die Input-/Output-Daten der Funktion zu spezifizieren.

Bei der ARIS-Architektur wird jedoch eine strikte Trennung der Betrachtungsfelder verfolgt (vgl. Scheer, Architektur integrierter Informationssysteme 1992, S. 62). Innerhalb der Funktionssicht werden somit nur solche Darstellungsmittel behandelt, die Verbindungen zwischen Funktionen aufzeigen. So werden z. B. Verbindungen zwischen Funktionen und Daten in der Prozesssicht von ARIS dargestellt.

Eine Funktion ist eine fachliche Aufgabe oder Tätigkeit an einem Objekt zur Unterstützung eines oder mehrerer Unternehmensziele (vgl. Scheer, Architektur integrierter Informationssysteme 1992, S. 63).

Funktionen werden durch Rechtecke mit abgerundeten Ecken dargestellt:

Abbildung 6: Darstellung der Funktion Kundenanfrage prüfen

Bildungskriterium für eine solche Funktion ist in der Regel ein Informationsobjekt wie eine Kundenanfrage oder ein Fertigungsauftrag. Dies sollte auch bei der Bezeichnung einer Funktion zum Ausdruck kommen. In obiger Abbildung wird dies verdeutlicht. **Kundenanfrage** definiert das Objekt, **prüfen** definiert die Verrichtung, die an diesem Objekt durchgeführt wird. Auf hoher Ebene wird jedoch meist nur ein Substantiv als Funktionsbezeichnung verwendet (Beschaffungslogistik, Produktion, Vertrieb).

3.1.1.1 Funktionsbaum

Funktionen können auf unterschiedlichen Verdichtungsstufen beschrieben werden. Oberste Verdichtungsstufe sind Funktionsbündelungen in Form von Geschäftsprozessen oder Vorgangsketten. Beispiel hierfür ist die Bearbeitung eines Kundenauftrags von der Bearbeitung der Kundenanfrage bis zum Versand. Ein solcher Geschäftsprozess stellt somit eine komplexe Funktion dar, die zur Reduzierung der Komplexität in Teilfunktionen zerlegt werden kann. Der Begriff Funktion kann somit auf allen Hierarchieebenen verwendet werden. Oft werden jedoch auch Begriffe wie Vorgang, Prozess, Teilfunktion oder Elementarfunktion zur Verdeutlichung der Hierarchieebene verwendet.

Die Zerlegung der Funktionen kann über mehrere Hierarchieebenen erfolgen. In semantischen Funktionsbäumen stellen hierbei Elementarfunktionen die unterste Stufe dar.

Elementarfunktionen sind Funktionen, die betriebswirtschaftlich nicht mehr sinnvoll zerlegbar sind.

Zur Darstellung dieser Untergliederung werden Funktionsbäume oder Hierarchiemodelle verwendet.

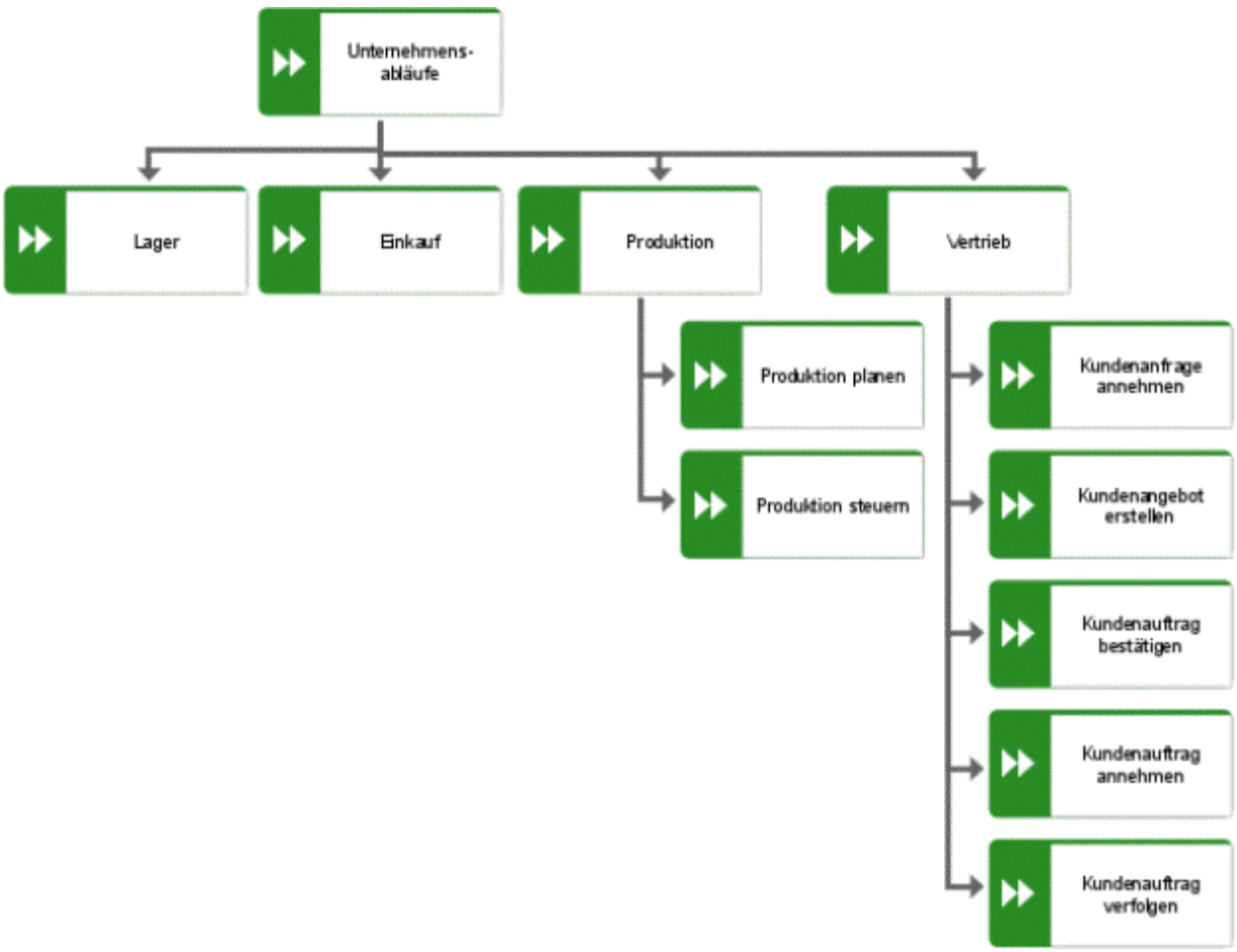


Abbildung 7: Funktionsbaum (Ausschnitt)

Die Gruppierung der Funktionen in einem Funktionsbaum kann nach verschiedenen Kriterien erfolgen (vgl. Brombacher/Bungert, Praxis der Unternehmensmodellierung 1992). Häufig verwendete Kriterien sind hierbei die Bearbeitung des gleichen Objekts (objektorientiert), die Zerlegung nach der Prozesszugehörigkeit (prozessorientiert) oder die Zusammenfassung von Funktionen nach gleichen Verrichtungen (verrichtungsorientiert).

Die nächste Abbildung zeigt das Beispiel einer objektorientierten Zerlegung. Die übergeordnete Funktion **Fertigungsauftrag bearbeiten** wird in die Funktionen **Fertigungsauftrag erstellen**, **Fertigungsauftrag rückmelden**, **Fertigungsauftrag ändern**, **Fertigungsauftrag stornieren**, **Fertigungsauftrag freigeben** und **Fertigungsauftrag überwachen** untergliedert. Die Funktionen beschreiben unterschiedliche Verrichtungen (erstellen, ändern, stornieren, ...), jedoch immer am gleichen Objekt **Fertigungsauftrag**.

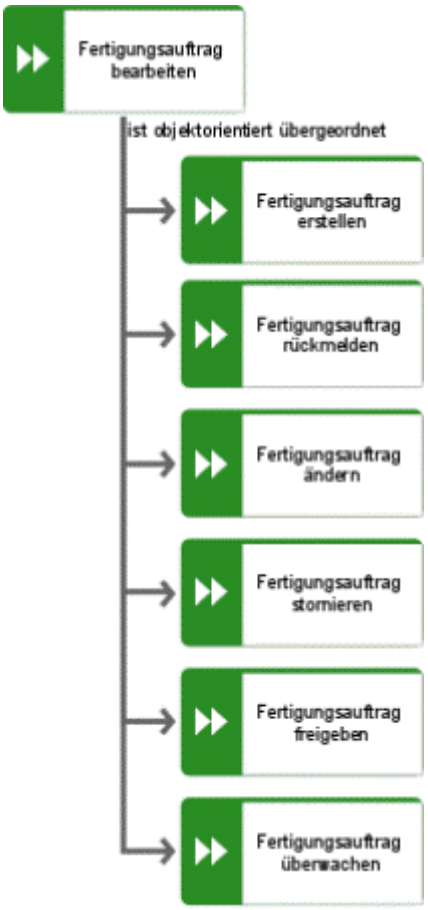


Abbildung 8: Funktionsbaum objektorientiert

Stellen die Funktionsbäume Ergebnisse der Modellierung von Geschäftsprozessen dar, bietet sich die Darstellung prozessorientierter Funktionsbäume an. Die folgende Abbildung zeigt das Beispiel einer prozessorientierten Funktionszerlegung.

Die Funktionen **Kundenauftrag annehmen**, **Kundenauftrag prüfen**, **Kundendaten anlegen**, **Kundenbonität prüfen**, **Produktverfügbarkeit prüfen** und **Kundenauftrag bestätigen** sind Funktionen des Geschäftsprozesses **Kundenauftrag bearbeiten**. Im Gegensatz zu einer objektorientierten Zerlegung werden hier Verrichtungen an verschiedenen Objekten durchgeführt (Kundenauftrag, Produktverfügbarkeit).

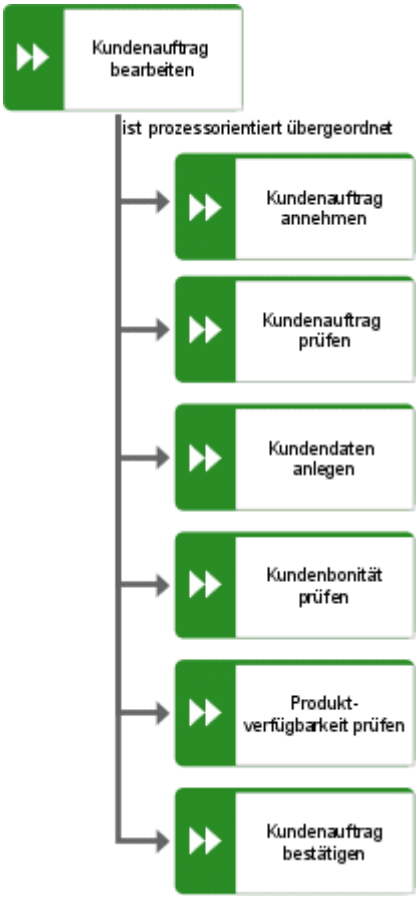


Abbildung 9: Funktionsbaum prozessorientiert

Verrichtungsorientierte Gruppierung bedeutet, dass alle Funktionen, die die gleiche Verrichtung (prüfen, anlegen, löschen) an unterschiedlichen Informationsobjekten durchführen, zusammengefasst werden. Ein Beispiel für die Verrichtung **Ändern** ist in der nächsten Abbildung dargestellt. Die gezeigten Funktionen können in unterschiedlichen Prozessen auftreten und bearbeiten auch unterschiedliche Objekte. Die Art der Verrichtung an den unterschiedlichen Objekten ist jedoch immer die gleiche.

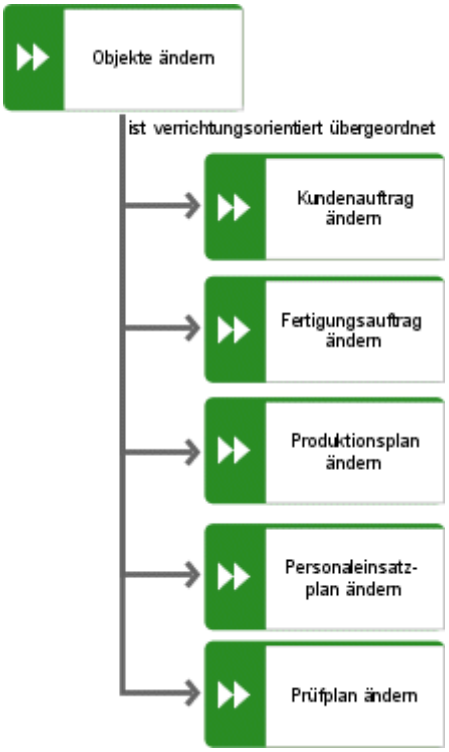


Abbildung 10: Funktionsbaum verrichtungsorientiert

Die Darstellung von Funktionen in einem Funktionsbaum dient zwar zur Reduzierung der Komplexität, ist jedoch statisch. Neben dieser statischen Darstellung kann aber auch die Ablauffolge im Sinne der zeitlichen Aufeinanderfolge von Funktionen von Interesse sein. Zur Darstellung zeitlich-logischer Ablauffolgen dienen so genannte ereignisgesteuerte Prozessketten (EPK). Diese beinhalten jedoch neben den Funktionen auch Ereignisse als Bindeglieder zwischen den Funktionen. Ereignisse sind der Datensicht von ARIS zuzuordnen. Der strikten Sichtentrennung von ARIS folgend, werden ereignisgesteuerte Prozessketten in der Steuerungssicht von ARIS beschrieben (vgl. Kapitel **Fachkonzept** (Seite 12)).

Bei der fachkonzeptionellen Beschreibung von Funktionen sind neben der Eigenschaft der Funktionszergliederung in Teilfunktionen auch weitere Eigenschaften einer Funktion von Interesse; insbesondere solche Eigenschaften, die auf die Gestaltung der Geschäftsabläufe Einfluss nehmen.

So sollte für jede Funktion bereits grob definiert werden, ob ein Benutzereingriff notwendig ist oder die Funktion weitgehend automatisiert ablaufen kann. Gleichartige Funktionen, die ohne Benutzereingriff durchführbar sind, können somit gebündelt und geschlossen verarbeitet werden (Batch-Lauf).

Weitere Anhaltspunkte für die Neugestaltung der betrieblichen Abläufe bieten Informationen über das Mengengerüst einer Funktion (Bsp.: Anzahl der Anfragen, die an einem Tag bearbeitet werden) und die Gesamtdauer der Funktionsausführung. Die Gesamtdauer kann nochmals in Teilzeiten (Einarbeitungszeit, Bearbeitungszeit, Liegezeit) untergliedert werden. In ARIS können diese Informationen in den Attributen des Objekttyps **Funktion** abgelegt werden. Eine Auflistung der zur Verfügung stehenden Attributtypen finden Sie in der **ARIS-Methode Reference**-Hilfe.

3.1.1.2 Zieldiagramm

Vor dem Beginn jeder Geschäftsprozessmodellierung, -analyse und -optimierung (Business Process Reengineering) sollte die Frage stehen, welche Ziele das Unternehmen mit der Geschäftsprozessmodellierung verfolgt.

Im Zieldiagramm können u. a. (Unternehmens-)Ziele definiert und Zielhierarchien aufgebaut werden.

Mit einem Ziel werden die zukünftigen Unternehmenszielsetzungen definiert, die durch Unterstützung der Erfolgsfaktoren und Realisierung neuer Geschäftsprozesse erreicht werden sollen.

Die möglichen Erfolgsfaktoren zur Zielerreichung können festgelegt, hierarchisiert und den Zielen, deren Erreichung sie unterstützen, zugeordnet werden.

Erfolgsfaktoren spezifizieren die zur Erreichung des jeweiligen Unternehmenszieles zu beachtenden Aspekte. Sie werden im Zieldiagramm den Unternehmenszielen zugeordnet.

Die Verbindung dieses Modelltyps zu den anderen Modelltypen des Fachkonzepts erfolgt über den Objekttyp **Funktion**. Für jedes Ziel kann dargestellt werden, welche(r) Funktion (Geschäftsprozess) des Unternehmens die Erreichung des Ziels unterstützt. In der Phase der Geschäftsprozessmodellierung und -optimierung sollten bei der Festlegung des Vorgehensmodells die hier festgelegte Zielpriorisierung und die zugeordneten Funktionen berücksichtigt werden.

Ein Beispiel für ein Zieldiagramm zeigt diese Abbildung.

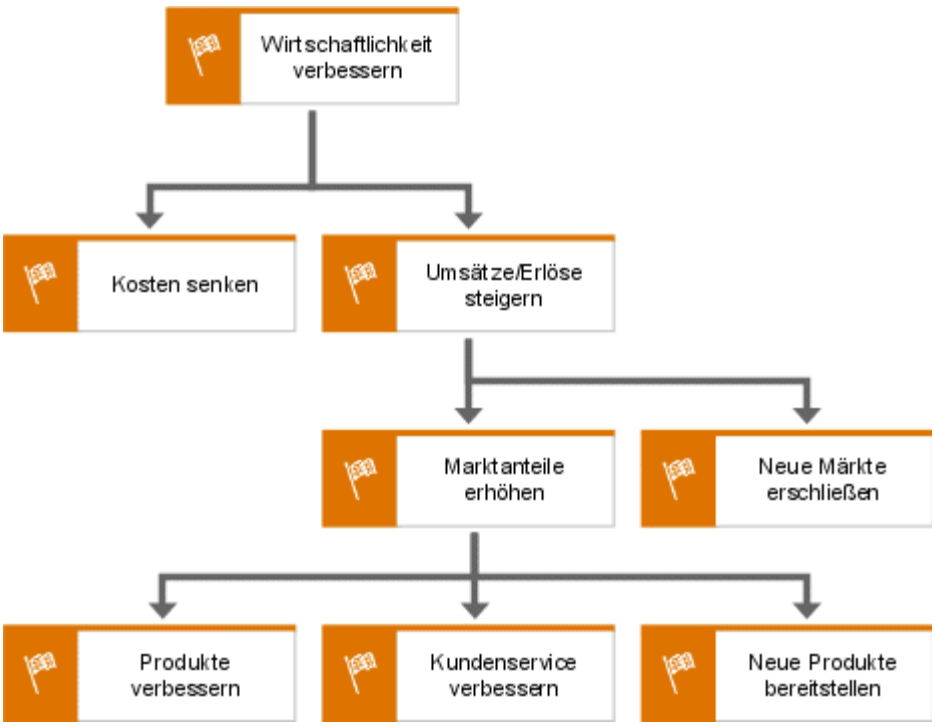


Abbildung 11: Zieldiagramm

3.1.2 DV-Konzept - Anwendungssystemtypdiagramm

Das DV-Konzept der Funktionssicht beinhaltet die Festlegung der Anwendungssystem- und Modultypen, den modularen Aufbau des Anwendungssystemtyps, den Entwurf einzelner Transaktionsschritte und die Definition der Ein- und Ausgabepräsentationen in Form von Listen- und Maskenentwürfen.

Zentrale Fragestellungen, die mit der Festlegung des DV-Konzepts der Funktionssicht beantwortet werden, sind:

- Wie können die im Fachkonzept definierten Funktionen durch Einsatz von Anwendungssystemtypen, Modultypen oder DV-Funktionen unterstützt werden?
- Wie sind Anwendungssystemtypen oder Modultypen modular aufgebaut?
- Welche Listen und Masken werden zur Durchführung einer Funktion benötigt?
- Welche Listen können mit einem Anwendungssystemtyp oder einem Modultyp erstellt werden und welche Masken verwenden Anwendungssystemtypen und Modultypen?
- Auf welcher technologischen Basis (Betriebssysteme, Benutzeroberflächen oder Datenbankmanagementsysteme) beruht ein Anwendungssystemtyp?
- Welche unternehmerischen Ziele werden beim Einsatz eines bestimmten Anwendungssystemtyps verfolgt?

Der zentrale Objekttyp des DV-Konzeptes der Funktionssicht ist somit der **Anwendungssystemtyp**.

Im Gegensatz zu einem konkreten Anwendungssystem, das erst auf der Implementierungsebene der Funktionssicht betrachtet wird und ein einzelnes z. B. durch eine Lizenznummer im Unternehmen identifizierbares Anwendungssystem darstellt, entsteht ein Anwendungssystemtyp durch Typisierung aller Anwendungssysteme, die sich auf exakt der gleichen technologischen Basis befinden.

Ein Anwendungssystemtyp stellt die Typisierung einzelner Anwendungssysteme dar, die sich auf exakt der gleichen technologischen Basis befinden.

Beispiel: ARIS Architect stellt einen Anwendungssystemtyp dar. Von diesem Anwendungssystemtyp kann man mehrere Lizenzen und damit mehrere einzelne Anwendungssysteme erwerben.

Anwendungssystemtypen werden grafisch wie folgt dargestellt:

Abbildung 12: Grafische Darstellung eines Anwendungssystemtyps

Anwendungssystemtypen sind meist modular aufgebaut. Das Anwendungssystemtypdiagramm ermöglicht die Darstellung dieses modularen Aufbaus. Die einzelnen Teile eines Anwendungssystemtyps sind Modultypen. Diese Abbildung zeigt hierzu ein Beispiel:

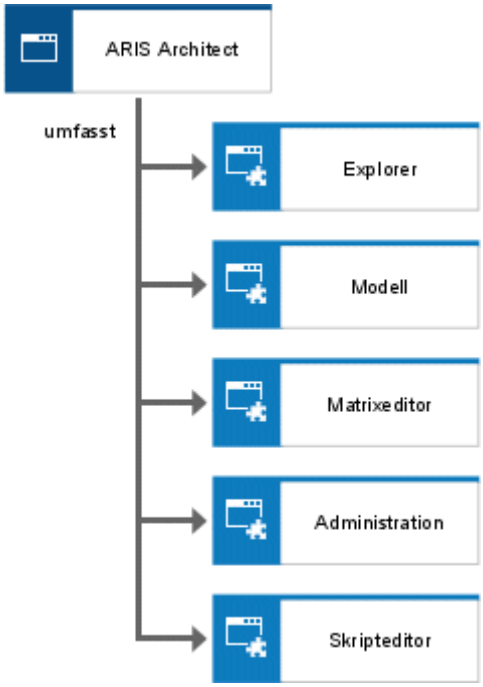


Abbildung 13: Modularer Aufbau eines Anwendungssystemtyps

ARIS Architect setzt sich in obigem Beispiel aus den Modultypen **Explorer**, **Modell**, **Matrixeditor**, **Administration** und **Skripteditor** zusammen. Wie schon bei den Anwendungssystemtypen handelt es sich auch bei den Modultypen um die Typisierung einzelner Module, die sich auf exakt der gleichen technologischen Basis befinden. Modultypen sind Bestandteile von Anwendungssystemtypen. Sie stellen für sich alleine ablauffähige Komponenten dar.

Ein Modultyp ist ein für sich alleine ablauffähiger Bestandteil eines Anwendungssystemtyps. Modultypen stellen die Typisierung einzelner Module dar, die sich auf exakt der gleichen technologischen Basis befinden.

Anwendungssystemtypen und Modultypen sind beliebig hierarchisierbar. Modultypen können auf der untersten Stufe in DV-Funktionstypen unterteilt werden.

DV-Funktionstypen sind im Sinne einer Transaktion die kleinsten Einheiten eines Modultyps. Sie werden durch einzelne Programmteile realisiert und müssen zur Bearbeitung eines einzelnen Arbeitsschrittes immer vollständig durchgeführt werden.

Abbildung 14: Grafische Darstellung des DV-Funktionstyps

Das Anwendungssystemtypdiagramm erlaubt auch die Festlegung der Funktionen des Fachkonzeptes, die durch die definierten Anwendungssystemtypen und Modultypen unterstützt werden. Diese Zuordnung bildet das Bindeglied zwischen Fachkonzept und DV-Konzept der Funktionssicht. Diese Abbildung zeigt hierzu ein Beispiel:

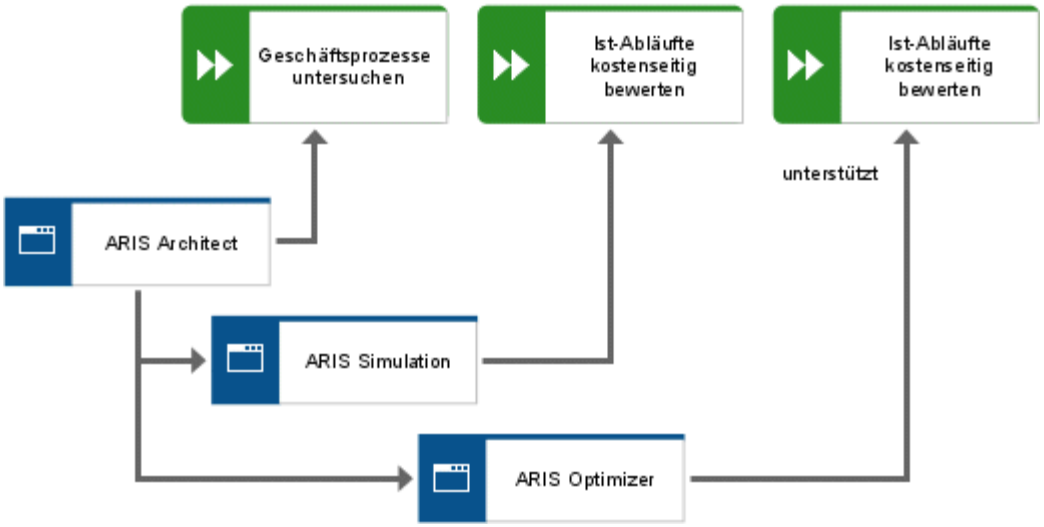


Abbildung 15: Zuordnung der Funktionen zu Anwendungssystemtypen

Zur genaueren Bestimmung der technologischen Basis von Anwendungssystemtypen und Modultypen können diesen auch die möglichen Typen von Benutzeroberflächen, Datenbankmanagementsystemen und Betriebssystemen, unter denen sie laufen, sowie die Programmiersprachen, mit denen sie realisiert sind, zugeordnet werden. Da es sich um Typen und nicht um konkrete Exemplare handelt, ist eine Mehrfachbeziehung durchaus möglich. So kann einem Anwendungssystemtyp die Benutzeroberfläche **Windows 7** und **Windows 8** zugeordnet sein, was bedeutet, dass die Version des Anwendungssystemtyps unter beiden Benutzeroberflächen lauffähig ist. Erst bei der Zuordnung der Benutzeroberfläche zu einem konkreten Exemplar (also einem Anwendungssystem) in der Implementierungsebene der Funktionssicht ist eine eindeutige Beziehung notwendig. Diese beschreibt dann genau die Konfigurierung der einzelnen Lizenz des Anwendungssystemtyps, die vom Unternehmen erworben wurde.

Ein Beispiel für die möglichen Zuordnungen im Anwendungssystemtypdiagramm zeigt diese Abbildung.

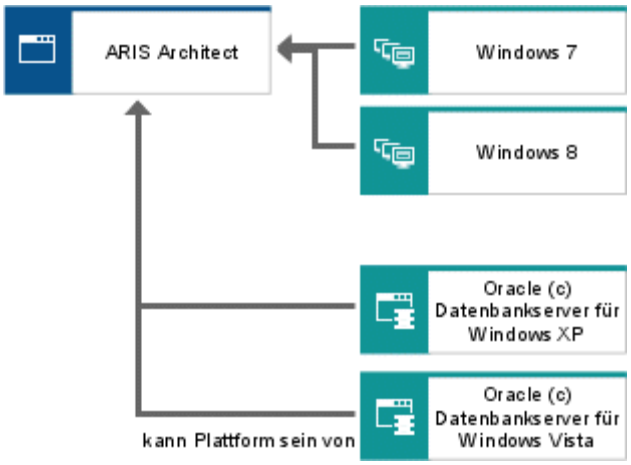


Abbildung 16: Konfigurierung des Anwendungssystemtyps

Die Bearbeitung einer fachlichen Funktion mit Unterstützung eines Anwendungssystems führt zur Benutzung unterschiedlicher Bildschirmmasken und zur Erzeugung oder Verwendung unterschiedlicher Listen, die vom entsprechenden Anwendungssystem angeboten werden. Zur Darstellung dieses Sachverhalts stehen die Objekte **Liste** und **Maske** zur Verfügung, die entweder der fachlichen Funktion oder den Anwendungssystemtypen und Modultypen zugeordnet werden können.

Sollen zunächst allgemeine Ablauffolgen definiert werden, ohne Bezug auf konkrete Anwendungssystemtypen, so können zur Spezifikation der benötigten Masken und Listen auch die Objekte **Listenentwurf** und **Maskenentwurf** verwendet werden. Beide Objekttypen spezifizieren zunächst allgemein, welche Art von Liste oder Maske verwendet werden soll (Beispiel **Eingabe Kundendaten**) ohne einen konkreten Bezug zu Listen oder Masken eines Anwendungssystemtyps herzustellen. Danach können diese Listen- und Maskenentwürfe mit konkreten Listen und Masken verbunden werden. Die Zuordnung definiert dabei die zur Verfügung stehenden Realisierungsmöglichkeiten. Ein Beispiel zeigt die folgende Abbildung.

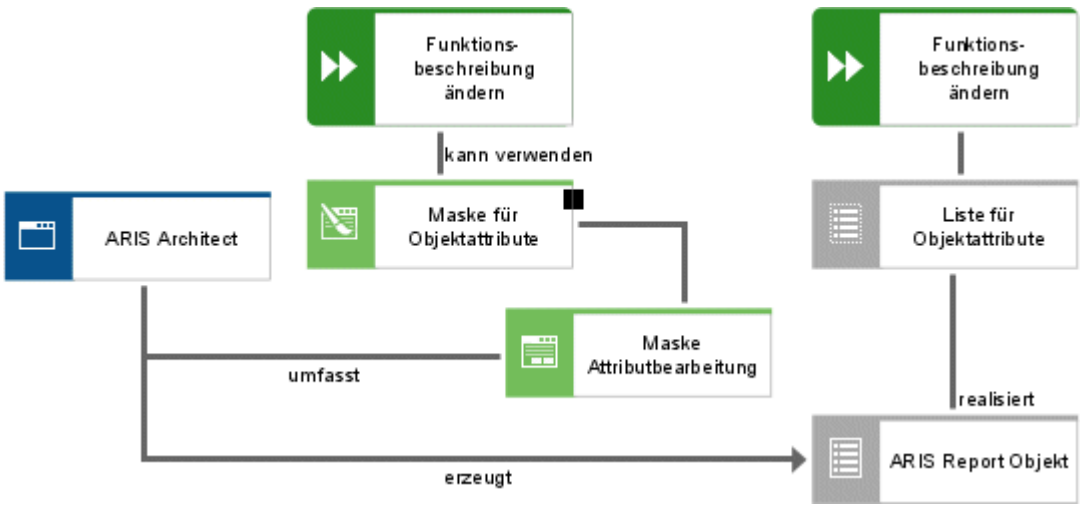


Abbildung 17: Masken- und Listenzuordnungen

Eine Zusammenfassung der im Anwendungssystemtypdiagramm vorhandenen Objekttypen und Beziehungen finden Sie in der **ARIS-Methode Reference**-Hilfe.

3.1.3 Implementierung - Anwendungssystemdiagramm

Im Anwendungssystemtypdiagramm können den im DV-Konzept beschriebenen Anwendungssystem- und Modultypen die konkreten Anwendungssysteme und Module zugeordnet werden. Es handelt sich hierbei um die im Unternehmen vorhandenen Exemplare eines Anwendungssystemtyps, die z. B. durch die Lizenznummern eindeutig identifizierbar sind.

Ein Anwendungssystem (Modul) ist ein einzelnes Exemplar eines Anwendungssystemtyps (Modultyps) das z. B. durch die Lizenznummer genau identifiziert werden kann.

Grafisch werden Anwendungssysteme und Module wie folgt dargestellt.

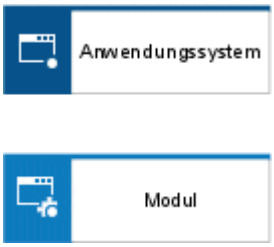


Abbildung 18: Grafische Darstellung des Anwendungssystems und des Moduls

Da von einem Anwendungssystemtyp (Modultyp) mehrere Lizenzen in einem Unternehmen vorhanden sein können, können somit auch im Anwendungssystemdiagramm mehrere Anwendungssysteme (Module) einem Anwendungssystemtyp (Modultyp) zugeordnet werden.

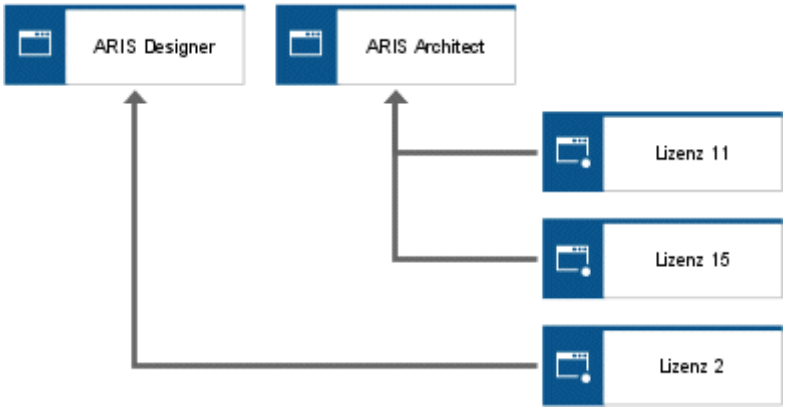


Abbildung 19: Zuordnung der Anwendungssysteme zu ihren Anwendungssystemtypen

Im Anwendungssystemdiagramm wird der tatsächliche modulare Aufbau eines Anwendungssystems dargestellt. Wurden im DV-Konzept alle möglichen modularen Bestandteile eines Anwendungssystemtyps aufgezeigt, liegt nun eine einzelne Lizenz des Anwendungssystems vor, sodass die modularen Bestandteile jeder einzelnen Lizenz eindeutig definierbar sind. So können im Unternehmen von einem Anwendungssystemtyp mehrere Anwendungssysteme mit völlig unterschiedlichem modularem Aufbau vorhanden sein.

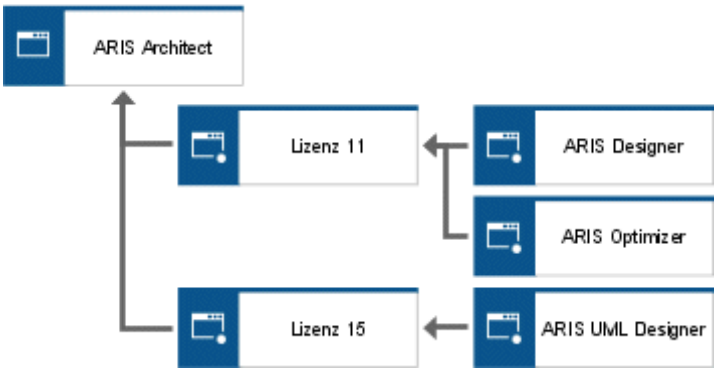


Abbildung 20: Unterschiedlicher modularer Aufbau zweier Anwendungssysteme des gleichen Typs

Neben der Darstellung der tatsächlich vorhandenen Anwendungssysteme und Module erlaubt die Implementierungsebene auch die Definition der programmtechnischen (physikalischen) Umsetzung der Anwendungssysteme in Form einzelner Programmdateien.

Hierzu kann im Anwendungssystemtypdiagramm dargestellt werden, welche Programmteiltypen zur Realisierung eines Anwendungssystemtyps oder Modultyps benötigt werden.

Ein Programmteil ist jede durch den Kauf einer Lizenz erworbene Programmdatei auf einem Datenträger (z. B. eine EXE-Datei oder COM-Datei). Ein Programmteiltyp entsteht durch Typisierung von Programmteilen, die sich auf exakt der gleichen technologischen Basis befinden.

Die Zuordnung von Programmteiltypen zu einem Anwendungssystemtyp und von einzelnen Programmteilen zu Programmteiltypen zeigt die folgende Abbildung.

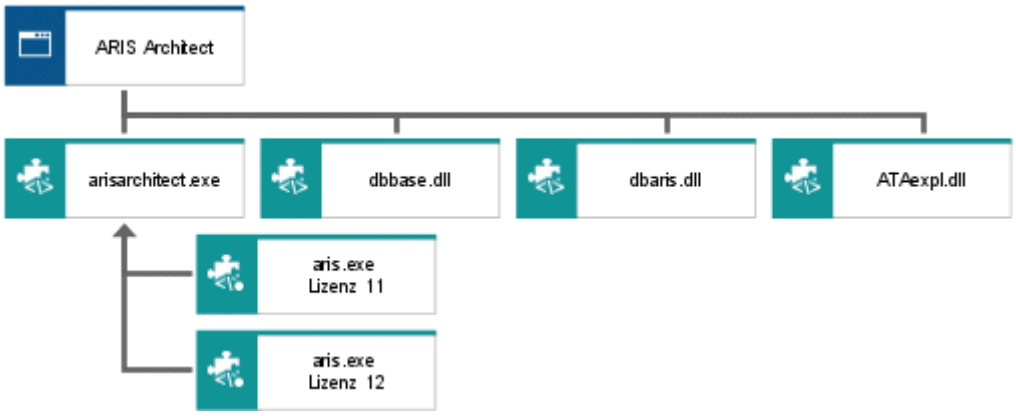


Abbildung 21: Zuordnung von Anwendungssystemtypen, Programmteiltypen und Programmteilen

Der Anwendungssystemtyp **ARIS Architect**besteht u. a. aus den Programmteiltypen **arisarchitect.exe**, **dbbase.dll**, **abaris.dll** und **ATAexpl.dll**. Von jedem Programmteiltyp können im Unternehmen mehrere Exemplare (Programmteile) durch Erwerb mehrerer Lizenzen oder durch das Erstellen von Sicherungskopien vorhanden sein.

Programmteiltypen und Programmteile sind beliebig hierarchisierbar. Zur näheren technologischen Spezifikation des Programms können auch die Zugriffe von Programmteiltypen auf Programmbibliotheken im Anwendungssystemtypdiagramm dargestellt werden.

Eine Zusammenfassung der im Anwendungssystemtypdiagramm vorhandenen Objekttypen und Beziehungen finden Sie in der **ARIS-Methode Reference**-Hilfe.

3.2 Datensicht

3.2.1 Fachkonzept

Das Fachkonzept der Datensicht beinhaltet die Beschreibung des semantischen Datenmodells des betrachteten Untersuchungsfeldes. Dem Zerlegungsprinzip von ARIS folgend werden hierbei sowohl die Objekte zur Spezifizierung der Start- und Endereignisse als auch die Zustandsbeschreibungen des relevanten Umfeldes einer Vorgangskette beschrieben.

Gegenüber der Funktionsmodellierung ist die Datenmodellierung methodisch besonders anspruchsvoll. In der Funktionssicht wird ausschließlich ein Objekt, die Funktion, betrachtet. Außerdem werden als Beziehungen zwischen Funktionen lediglich die Über- und Unterordnungen dargestellt.

Die am weitesten verbreitete Entwurfsmethode für semantische Datenmodelle ist das Entity-Relationship-Modell (ERM) von Chen (vgl. Chen, Entity-Relationship Model 1976). Bei dieser Modellierungsmethode werden vielfältige Begriffe wie Entitytyp, Beziehungstyp, Attribut usw. verwendet. Zwischen diesen Objekten bestehen zahlreiche und - im Vergleich zur Funktionsmodellierung - wesentlich schwieriger zu klassifizierende Beziehungen.

Im Folgenden wird die Modellierung mit Entity-Relationship-Modellen (ERM) vorgestellt. Dabei werden zunächst die Objekte und Beziehungen des Grundmodells nach Chen erläutert. Dieses Grundmodell wird dann im darauf folgenden Kapitel um einige Regeln erweitert.

3.2.1.1 Das ERM-Grundmodell

Das Grundmodell unterscheidet zwischen Entities, Attributen und Beziehungen. Hierbei kann grundsätzlich die Typebene von der Ausprägungsebene unterschieden werden.

Entities sind reale oder abstrakte Dinge, die für den betrachteten Ausschnitt der Aufgaben einer Unternehmung von Interesse sind.

Beispielsweise kann ein Geschäftsprozess ein betrachtetes Objekt sein. Die interessierenden Datenobjekte sind dann entsprechend dem Zerlegungsmodell von ARIS die Objekte des Umfeldes und die Objekte, die Ereignisse spezifizieren. Beispiele für Entities eines Prozesses **Kundenauftragsbearbeitung** sind:

- Kunde 1235,
- Artikel 471,
- Auftrag 11.

Entities werden durch bestimmte Attribute (Eigenschaften) näher beschrieben. So kann z. B. ein Kunde über seinen Namen, Vornamen und die Anschrift näher spezifiziert werden.

Werden gleichartige Entities zu Mengen zusammengefasst, so werden diese als Entitytypen bezeichnet, deren einzelne Ausprägungen die Entities sind.

Die Gleichartigkeit von Entities drückt sich dadurch aus, dass sie durch gleiche Attribute beschrieben werden können. So wird der Kunde **Maier** und der Kunde **Müller** zu dem Entitytyp **Kunden** zusammengefasst, der Artikel **4710** und der Artikel **4712** zu dem Entitytyp **Artikel**. Entitytypen werden im ERM-Modell durch Rechtecke dargestellt (vgl. nächste Abbildung). Im Folgenden werden Entitytypen durch großgeschriebenen Text gekennzeichnet.

Abbildung 22: Beispiele für Entitytypen

Attribute sind Eigenschaften, durch die Entitytypen beschrieben werden.

Attributausprägungen sind konkrete Werte von Attributen einzelner Entities. Z. B. kann der Kunde **1235** durch die Attributausprägungen **Maier**, **Heinz**, **München** usw. beschrieben werden. Die dazugehörigen Attribute lauten **Name**, **Vorname** und **Wohnort**.

Attribute werden meist durch ein Oval oder einen Kreis dargestellt. Im Folgenden wird als Darstellungsmittel das Oval verwendet. Die folgende Abbildung zeigt beispielhaft Attribute für den Entitytyp KUNDE.

Abbildung 23: Beispiele für Attribute des Entitytyps Kunden

Eine Unterscheidung zwischen Entitytypen und Attributen ist oft sehr schwierig und kann nur aus dem Kontext der Modellierung heraus getroffen werden. So können z. B. die Anschriften der Kunden auch als Entities aufgefasst werden und nicht als Attribute zu KUNDEN. In diesem

Fall würde ein eigener Entitytyp ANSCHRIFT mit einer Beziehung zu KUNDEN modelliert werden. Ein hilfreiches Unterscheidungsmerkmal bei der Festlegung, ob es sich um einen Entitytyp oder ein Attribut handelt, ist hierbei, dass Entities Attribute besitzen, Attribute jedoch keine eigenen Attribute besitzen. Wird somit ein Attribut in einem ERM angelegt, das man später durch weitere Attribute beschreiben möchte, so wird es zum Entitytyp. Eine weitere hilfreiche Unterstützung bietet die Frage, ob man zu einem Objekt Beziehungen zu anderen Entitytypen herstellen möchte. Ist dies der Fall, so ist das betrachtete Objekt ebenfalls ein Entitytyp.

Eine Beziehung ist eine logische Verknüpfung zwischen Entities.

Beziehungen können damit nur existieren, wenn auch Entities existieren.

Werden gleichartige Beziehungen zu Mengen zusammengefasst, werden diese als Beziehungstypen bezeichnet.

Ein Beziehungstyp zwischen LIEFERANT und TEIL könnte z. B. LIEFERT heißen. Beziehungstypen werden im folgenden Text ebenfalls durch Großschreibung gekennzeichnet. Im ERM werden Beziehungstypen durch eine Raute dargestellt und über Kanten mit den Entitytypen verbunden (vgl. folgende Abbildung).

Abbildung 24: Beispiel für einen Beziehungstyp

Bei der Bezeichnung der Beziehungstypen kann oft nur eine Leserichtung zu sinnvollen Verbindungen führen. So soll im obigen Beispiel die Beziehung **Lieferant liefert Teil** ausgedrückt werden. In anderer Leserichtung wäre das Ergebnis **Teil liefert Lieferant**, was eindeutig nicht sinnvoll ist. Ist die Leserichtung nicht eindeutig bestimmbar, muss diese Schwierigkeit durch eine geschickte Wahl übergeordneter Begriffe umgangen werden.

Es werden mehrere Arten von Beziehungstypen unterschieden. Unterscheidungsmerkmale sind hierbei zum einen die Anzahl der durch sie verbundenen Entitytypen und zum anderen der Komplexitätsgrad einer Beziehung.

Bei der Anzahl der durch einen Beziehungstyp verbundenen Entitytypen können 1-, 2- oder n-stellige Beziehungen unterschieden werden.

Der Komplexitätsgrad oder die Kardinalität gibt an, wie viele Entities eines Entitytyps mit einem Entity des anderen Entitytyps in Beziehung stehen können.

Die dabei zu unterscheidenden Beziehungen werden in folgender Abbildung gezeigt (vgl. Scheer, Wirtschaftsinformatik 1994, S. 34).

Es ergeben sich somit vier verschiedene Beziehungsarten (Kardinalitäten):

- 1:1 - Beziehung,
- 1:n - Beziehung,
- n:1 - Beziehung und
- n:m - Beziehung.

Bei einer 1:1-Beziehung wird jeder Entity der ersten Menge genau einem Entity der zweiten Menge zugeordnet.

Eine 1:n-Beziehung drückt aus, dass jeder Entity der ersten Menge genau einem Entity der zweiten Menge zugeordnet wird, jeder Entity der zweiten Menge jedoch Beziehungen zu verschiedenen Entities der ersten Menge eingehen kann.

Die n:1-Beziehung drückt den gleichen Sachverhalt in umgekehrter Reihenfolge aus.

Werden jedem Entity der ersten Menge mehrere Entities der zweiten Menge zugeordnet und umgekehrt, so liegt eine n:m-Beziehung vor.

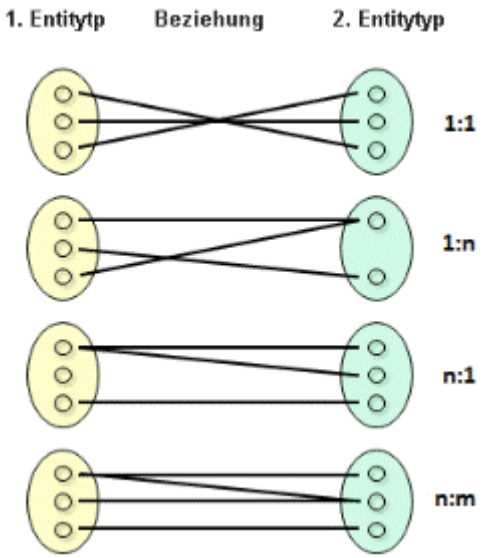


Abbildung 25: Kardinalitäten der Beziehungen zwischen zwei Entitytypen

Die Kardinalitäten des Beziehungstyps (Attributtyp **Komplexitätsgrad**) werden an die Kanten des Entity-Relationship-Modells geschrieben.

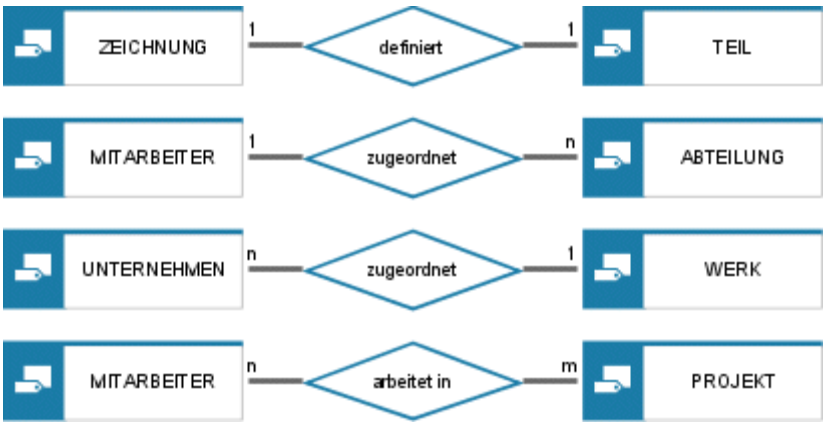


Abbildung 26: Darstellung der Kardinalitäten im ERM

Die Kardinalität gibt an, an wie vielen Beziehungen des Beziehungstyps ein Entity des Entitytyps maximal beteiligt sein kann. Im Beispiel der n:1-Beziehung der obigen bedeutet dies, dass ein Unternehmen des Entitytyps UNTERNEHMEN in mehreren Beziehungen ZUGEORDNET vorkommen kann, da ein Unternehmen aus mehreren Werken besteht; ein

konkretes Werk kann jedoch maximal an einer Beziehung ZUGEORDNET beteiligt sein, muss also einem Unternehmen eindeutig zugeordnet sein.

In der Originalarbeit von Chen werden die Kardinalitätsangaben in anderer Weise interpretiert. Die hier dargestellte Notation ermöglicht aber, insbesondere bei der Darstellung von Beziehungen zwischen mehreren Entitytypen, eindeutigere Formulierungen. Auf die Originalfassung von Chen wird an dieser Stelle daher nicht eingegangen, da dies nur zu unnötigen Verwirrungen führen würde.

Da auch Beziehungen zwischen Entities eines Entitytyps möglich sind, ist auch die Darstellung zweier paralleler Kanten zwischen einem Entitytyp und einem Beziehungstyp zulässig. Den Kanten können dann zur Unterscheidung Rollennamen vergeben werden. Ein Beispiel zur Darstellung rekursiver Beziehungen zeigt die folgende Abbildung. Ein übergeordnetes Teil besteht aus mehreren untergeordneten Teilen, ein untergeordnetes Teil kann aber auch als Komponente in mehreren übergeordneten Teilen verwendet werden.

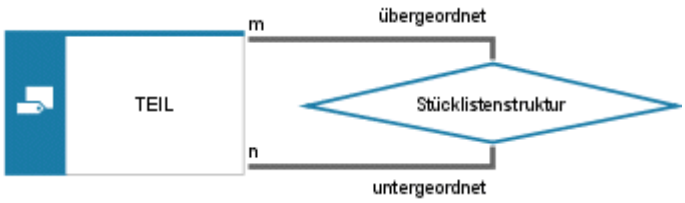


Abbildung 27: ERM für eine Stückliste

Sowohl Entitytypen als auch Beziehungstypen können durch Attribute beschrieben werden (vgl. folgende Abbildung).

Die Wertebereiche von Attributen werden als Domänen bezeichnet.

Die Zuordnungen von Elementen der Domäne zu Elementen der Entity- und Beziehungstypen stellen auch Beziehungen dar und können durch eine Kante dargestellt werden, die mit dem Namen gekennzeichnet ist.

Zwischen einem Entitytyp und mindestens einer Domäne muss eine 1:1-Beziehung bestehen. Die Werte dieser Domäne identifizieren die einzelnen Entities eindeutig und werden deshalb auch Schlüsselattribut des Entitytyps genannt.

Im Beispiel der nächsten Abbildung (vgl. Scheer, Wirtschaftsinformatik 1994, S. 33) werden die Entities von KUNDE über das Schlüsselattribut **Kundennummer** eindeutig identifiziert.

Beziehungen werden durch die Verschmelzung der Schlüsselattribute der verbundenen Entities identifiziert. Schlüsselattribute des Beziehungstyps WOHNEN sind somit die Kundennummer und die Adressennummer.

Werte aus Domänen mit einer 1:n - Beziehung zu Entity- oder Beziehungstypen definieren die beschreibenden Attribute der jeweiligen Datenobjekte.

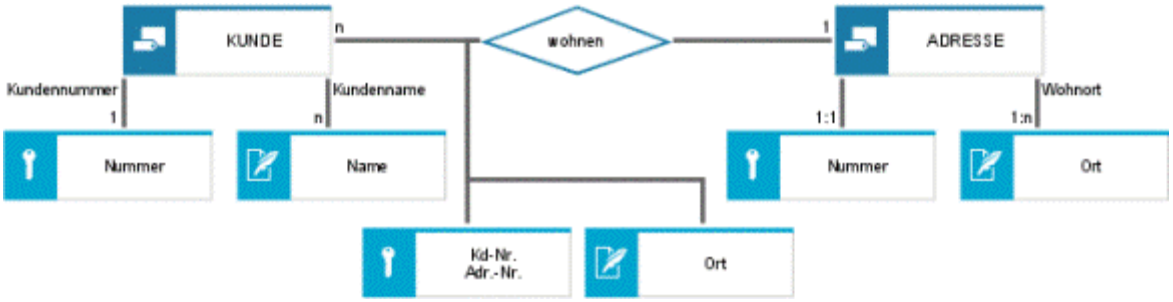


Abbildung 28: Zuordnung von Attributen im ERM

3.2.1.2 Erweiterungen des ERM - eERM

Das Grundmodell von Chen hat in den letzten Jahren zahlreiche Erweiterungen erfahren. Im Folgenden werden jedoch nur solche Erweiterungen angeführt, die für die Modellierung der Datensicht im ARIS-Konzept von Bedeutung sind.

3.2.1.3 Erweiterung um Konstruktionsoperatoren

Konstruktionsoperatoren bieten eine formale Hilfestellung bei der Erstellung eines Datenmodells. Sie sichern ab, dass Datenstrukturen nicht „willkürlich“ entstehen und geben dem Betrachter einer gegebenen Datenstruktur Hinweise über den Entwurfsprozess. Von eingeführten Begriffen ausgehend werden neue Begriffe durch Anwendung der Konstruktionsoperatoren erzeugt. Die Erstellung stellt dabei einen intellektuellen Prozess dar, der auf der Ebene des betriebswirtschaftlichen Fachwissens abläuft. Die Untersuchung der betriebswirtschaftlichen Tatbestände unter dem Gesichtspunkt der Datenstrukturen führt dabei entweder zur Strukturierung bekannter Tatbestände aus neuer Sicht oder auch zur Gewinnung neuer, bisher nicht berücksichtigter Zusammenhänge.

In der Vielzahl der unterschiedlichen Ansätze zur Erweiterung der ERM-Modellierung haben sich vier grundsätzliche Konstruktionsoperatoren durchgesetzt (vgl. Scheer, Wirtschaftsinformatik 1994, S. 35 ff):

- Klassifizierung,
- Generalisierung,
- Aggregation und
- Gruppierung.

KLASSIFIZIERUNG

Bei der Klassifizierung werden gleichartige Objekte (Entities) erkannt und einem Begriff (Entitytyp) zugeordnet. Ein Objekt ist mit einem anderen Objekt gleich, wenn es durch die gleichen Merkmale (Attribute) beschrieben wird.

Die Klassifizierung führt somit zur bereits beschriebenen Ermittlung von Entitytypen.

Abbildung 29: Klassifizierung von Kunden

GENERALISIERUNG/SPEZIALISIERUNG

Bei der Generalisierung werden ähnliche Objekttypen zu einem übergreifenden Objekttyp zusammengefasst.

So werden in der nächsten Abbildung der Entitytyp **Kunde** und der Entitytyp **Lieferant** zu dem Gattungsbegriff **Geschäftspartner** generalisiert. Die Eigenschaften (beschrieben durch die Attribute), die beiden Ausgangsobjekten gemeinsam sind, werden auf den generalisierten Objekttyp übertragen. Damit müssen nur noch die Attribute beschrieben werden, in denen sich die Ausgangsobjekttypen unterscheiden. Die Bildung des neuen Entitytyps **Geschäftspartner** wird grafisch durch das Dreieck ausgedrückt, das auch als is-a-Beziehung bezeichnet wird.

Wird ein Gattungsbegriff in Teilbegriffe zerlegt (**Geschäftspartner** wird in **Kunde** und **Lieferant** aufgespaltet), so liegt eine Spezialisierung vor.

Es handelt sich hierbei um die umgekehrte Form der Generalisierung. Die Eigenschaften des generalisierten Objektes werden auf die spezialisierten Objekte vererbt. Neben diesen vererbten Attributen können die spezialisierten Objekttypen auch eigene Attribute besitzen. Die Darstellung der Spezialisierung unterscheidet sich grafisch nicht von der Darstellung der Generalisierung.

Dies drückt sich in der Grafik auch dadurch aus, dass die Verbindungslinien nicht gerichtet dargestellt werden.

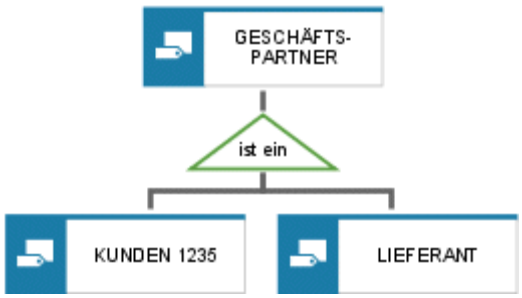


Abbildung 30: Generalisierung/Spezialisierung

Während die Spezialisierung hauptsächlich eine Top-down-Vorgehensweise bei der Erstellung eines Datenmodells unterstützt, wird die Generalisierung zur Unterstützung einer Bottom-up-Vorgehensweise eingesetzt.

Im Rahmen der Spezialisierung kann bei der Bildung von Teilmengen die Vollständigkeit und Disjunktheit (Alternative) der entstehenden Teilmengen spezifiziert werden.

Nicht-disjunkte Teilmengen liegen vor, wenn die Ausprägung eines Objektes sowohl in der einen Teilmenge als auch in der anderen Teilmenge vertreten sein kann. So kann im obigen Beispiel ein Kunde auch gleichzeitig Lieferant sein. Kann eine Ausprägung immer nur genau einer Teilmenge zugeordnet werden, so liegen disjunkte Mengen vor.

Eine vollständige Spezialisierung liegt dann vor, wenn für einen generalisierten Objekttyp alle möglichen spezialisierten Objekttypen eines Spezialisierungsgesichtspunktes aufgeführt werden. So kann z. B. der Entitytyp **Mensch** in die Entitytypen **Frau** und **Mann** spezialisiert werden (vgl. folgende Abbildung). Die Spezialisierung unter dem Gesichtspunkt Geschlecht wäre damit vollständig.

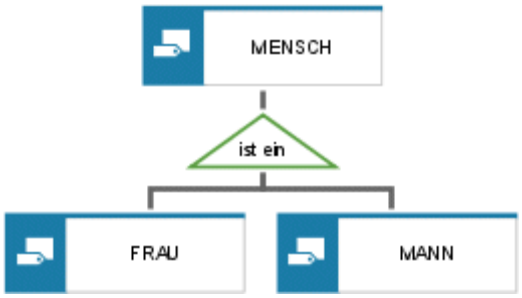


Abbildung 31: Vollständig beschriebene Spezialisierung

Die Kombinationen dieser Kriterien ergeben somit die folgenden vier Ausprägungen zur näheren Spezifizierung einer Generalisierung/ Spezialisierung:

- disjunkt/vollständig,
- disjunkt/nicht vollständig,
- nicht disjunkt/vollständig,
- nicht disjunkt/nicht vollständig.

AGGREGATION

Die Aggregation beschreibt die Bildung neuer Objekttypen durch die Zusammenfassung vorhandener Objekttypen. Der neue Objekttyp kann hierbei Träger neuer Eigenschaften sein.

Die Aggregation wird im ERM durch die Bildung von Beziehungstypen ausgedrückt (vgl. nächste Abbildung). Das neue Objekt **Auftragsarbeitsplan** entsteht durch Aggregation der Entitytypen **Fertigungsauftrag** und **Arbeitsplan**.

Abbildung 32: Beispiel für eine Aggregation

Der Aggregationsoperator kann aber auch auf Beziehungen angewendet werden. Ein dabei vorhandener Beziehungstyp wird dann als Entitytyp betrachtet und kann somit selbst wieder Ausgangspunkt für die Bildung neuer Beziehungen sein. Ein Beispiel hierfür wird in folgender Abbildung gezeigt.

Aus den Entitytypen **Fertigungsauftrag** und **Arbeitsplan** wurde durch eine erste Aggregation der Beziehungstyp **Auftragsarbeitsplan** gebildet. Die Schlüsselattribute **Fertigungsauftragsnummer** (FANR) und **Arbeitsplannummer** (APLNR) bilden den komplexen Schlüssel des Auftragsarbeitsplans. Dem Auftragsarbeitsplan können nun mehrere Arbeitsgänge zugeordnet werden. Aus diesem Grunde wird zwischen dem Beziehungstyp **Auftragsarbeitsplan** und dem Entitytyp **Arbeitsgang** die Beziehung **Auftragsarbeitsgang** gebildet. Da Beziehungen nur zwischen Entitytypen erzeugt werden können, muss der Beziehungstyp **Auftragsarbeitsplan** uminterpretiert werden. Wie in der nächsten Abbildung dargestellt ist, wird hierzu die Raute umrandet. Der entstehende uminterpretierte Beziehungstyp wird im Folgenden dann wie ein „normaler“ Entitytyp behandelt. Um die Entstehung des Beziehungstyps weiterhin grafisch darzustellen, werden die Kanten der an der Entstehung des Beziehungstyps beteiligten Entitytypen bis an die Raute herangezogen. Die von dem uminterpretierten Beziehungstyp ausgehenden Kanten zur Bildung neuer Beziehungen werden lediglich bis an die Ränder des Rechtecks geführt und berühren die Raute im Innern des Symbols nicht.

Obwohl es grundsätzlich möglich ist, die komplexen Schlüssel durch einfache Schlüssel zu ersetzen, hilft die Mitführung der komplexen Schlüssel, den Entstehungsprozess eines Datenmodells nachzuvollziehen.

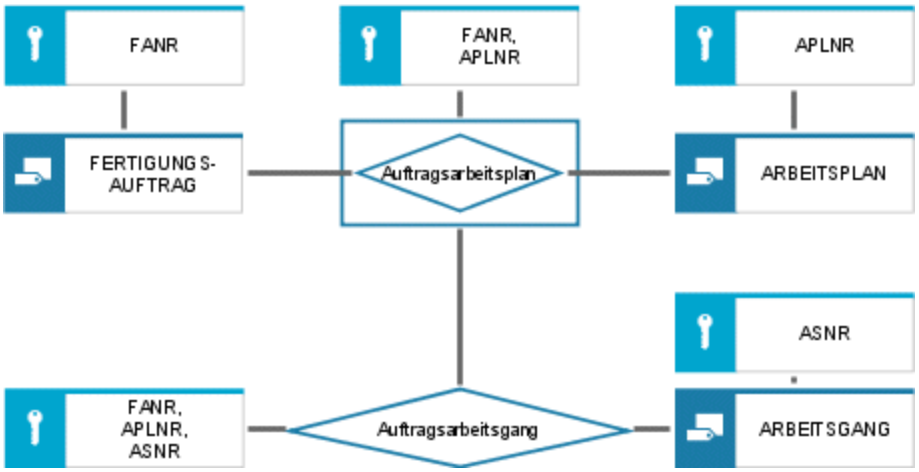


Abbildung 33: Aggregation mit uminterpretiertem Beziehungstypen

Ein ERM zerlegt einen komplexen Zusammenhang in eine übersichtliche Struktur. Da dabei jedoch der Bezug zum Gesamtkomplex nicht immer unmittelbar sichtbar bleibt, werden komplexe Objekte in Form von Clustern/Datenmodellen eingeführt.

Ein Cluster/Datenmodell beschreibt eine logische Sicht auf eine Ansammlung von Entity- und Beziehungstypen eines Datenmodells, die zur Beschreibung eines komplexen Objektes benötigt wird.

Neben Entity- und Beziehungstypen kann auch Cluster/Datenmodell selbst wieder Bestandteil eines Cluster/Datenmodells sein. Cluster/Datenmodell ist somit - im Gegensatz zu Entity- und Beziehungstypen - beliebig hierarchisierbar und unterstützt somit hauptsächlich ein Top-Down-Vorgehen bei der Datenmodell-Erstellung. Aber auch bei der Zusammenführung und Konsolidierung von Teilmodellen im Zuge einer Bottom-up-Vorgehensweise kann die Bildung von Clustern/Datenmodellen eine hilfreiche Unterstützung sein.

Die nächste Abbildung zeigt die grafische Darstellung eines Cluster/Datenmodells.

Abbildung 34: Datencluster (grafisches Symbol)

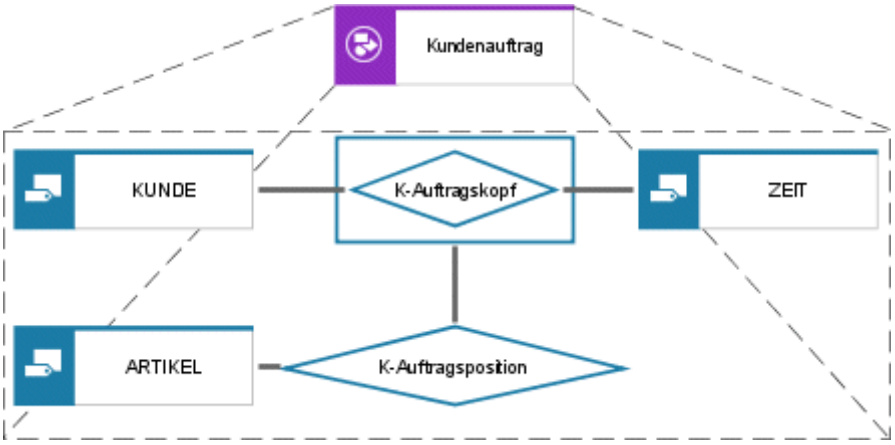


Abbildung 35: Datenclustersicht auf mehrere Objekte

Das Cluster/Datenmodell ist eine logische Sicht auf mehrere Entity- und Beziehungstypen. Zur Beschreibung des komplexen Objektes **Kundenauftrag** werden die Entity- und Beziehungstypen **Kunde**, **Zeit**, **Auftragskopf**, **Artikel** und **Auftragsposition** benötigt.

GRUPPIERUNG

Bei der Gruppierung werden aus den Elementen einer Entitymenge Gruppen gebildet.

So werden zum Beispiel in der folgenden Abbildung die Betriebsmittel zu Betriebsmittelgruppe zusammengefasst. Die Betriebsmittelgruppe ist ein eigenständiges Objekt, das durch weitere Attribute (Name der Betriebsmittelgruppe, Anzahl der Betriebsmittel), die nicht zu einzelnen Betriebsmitteln gehören, näher beschrieben werden kann. Weitere Beispiele sind die Gruppierung von Arbeitsplätzen zu Abteilungen oder die Zusammenfassung von Auftragszeilen zu Aufträgen.

Abbildung 36: Gruppierung

3.2.1.4 Erweiterung der Kardinalitäten

Bisher wurde bei der Angabe der Kardinalitäten immer nur die Obergrenze der möglichen Beziehungsausprägungen angezeigt. So wird mit den Kardinalitäten in der folgenden Abbildung ausgedrückt, das einem Projekt maximal mehrere (m) Mitarbeiter zugeordnet werden können, ein Mitarbeiter maximal an mehreren (n) Projekten beteiligt sein kann.

Abbildung 37: Ober-/Untergrenze (1)

Neben dieser Angabe der Obergrenze kann aber auch die Untergrenze zur Spezifikation der minimalen Anzahl der Beziehungsausprägungen von Interesse sein. Hierzu können z. B. die Kardinalitäten als Buchstabenpaar (a,b) dargestellt werden (vgl. Scheer, Wirtschaftsinformatik). In der nächsten Abbildung wird durch das Buchstabenpaar (a1, b1) ausgedrückt, dass jedes Projekt an **mindestens** a1 und **höchstens** b1 Beziehungsausprägungen vom Typ **arbeitet in** teilnehmen kann, was bedeutet, dass jedem Projekt **mindestens** a1 und **höchstens** b1 Mitarbeiter zugeordnet werden können. Das Buchstabenpaar (a2, b2) drückt aus, dass ein Mitarbeiter **mindestens** an a2 und **höchstens** an b2 Projekten beteiligt sein kann.

Abbildung 38: Ober-/Untergrenze (2)

Jede Beziehung wird somit durch zwei Komplexitätsgrade (Minimum, Maximum) bestimmt. Als Werte für die Untergrenze werden hierbei oft nur 0 und 1 verwendet, die Obergrenze hat als Wertebereich 1 <= max <= * (* bedeutet hierbei „beliebig viele“).

Eine Untergrenze min = 0 bedeutet, dass ein Entity an einer Beziehung beteiligt sein kann, aber nicht muss. Eine Untergrenze min = 1 bedeutet, dass ein Entity mindestens an einer Beziehung teilnehmen muss.

In der nächsten Abbildung wird durch die Untergrenzen ausgedrückt, dass ein Mitarbeiter an einer Beziehung teilnehmen kann, aber nicht muss (min = 0), ein Projekt aber mindestens an einer Beziehung teilnehmen muss (min = 1). Damit wird ausgedrückt, dass Mitarbeiter vorhanden sein können, die keinem Projekt zugeordnet sind. Umgekehrt muss jedoch jedes Projekt mindestens einem Mitarbeiter zugeordnet sein.

Abbildung 39: Ober- Untergrenze (3)

Werden als Minimalwerte nur 0 oder 1 und als Maximalwerte 1 oder * zugelassen, können somit vier Fälle einer (min,max) - Notation unterschieden werden: (1,1), (1,m), (0,1) und (0,m). Ist diesem Fall kann auch folgende abgekürzte Schreibweise verwendet werden (vgl. Schlageter/Stucky, Datenbanksysteme 1983, S. 51):

- 1 (entspricht (1,1))
- c (entspricht (0,1))
- m (entspricht (1,m))
- cm (entspricht (0,m)).

Die folgende Abbildung zeigt das vorherige Abbildungsbeispiel mit dieser Notation.

Abbildung 40: Ober- Untergrenze (4)

3.2.1.5 Identifizierungs- und Existenzabhängigkeit

Die in Kapitel **Erweiterungen des ERM - eERM** (Seite 32) diskutierte Erweiterung der Kardinalitäten um die Angabe von Unter- und Obergrenzen ermöglicht es nun, bestimmte Abhängigkeiten zwischen Datenobjekten zu definieren.

Beziehungstypen und uminterpretierte Beziehungstypen existieren definitionsgemäß nicht aus sich heraus, sondern aus der Existenz der Entitytypen, die sie verbinden. Sie sind somit identifikations- und existenzabhängig von anderen Entitytypen.

Daneben gibt es jedoch auch Entitytypen, die, obwohl sie ein eigenes Schlüsselattribut besitzen, von der Existenz anderer Entitytypen abhängen. Diese Abhängigkeiten sind z. B. durch die Operation der Gruppierung begründet. So ist in der nächsten Abbildung eine Abteilung nur dann sinnvoll, wenn ihr mindestens ein Arbeitsplatz zugeordnet ist, umgekehrt ist die Definition eines Arbeitsplatzes nur sinnvoll, wenn er einer Abteilung zugeordnet ist. Wie in der folgenden Abbildung dargestellt, werden diese existentiellen Abhängigkeiten durch die Angabe der Komplexitätsgrade ausgedrückt. Bei einer (min, max) - Notation durch (1,1) und (1,*). Die Definition existentieller Abhängigkeiten im Datenmodell führt in einer späteren Umsetzung zu Bedingungen der referentiellen Integrität von Daten. Einfach erklärt, stellt die Beachtung dieser Bedingungen sicher, dass sich die Inhalte einer Datenbank nach bestimmten Transaktionen immer noch in einem konsistenten Zustand befinden. Für das Beispiel bedeutet dies, dass eine Abteilung nur dann gelöscht werden kann, wenn auch alle Arbeitsplätze, die dieser Abteilung zugeordnet sind, gelöscht werden.

Abbildung 41: Existentielle Abhängigkeit

3.2.1.6 Modellierung von Fachbegriffen des Unternehmens-Fachbegriffsmodell

Eine häufig auftretende Schwierigkeit der Modellierung - insbesondere der Datenmodellierung - ist die Begriffsvielfalt, die sich für Informationsobjekte in größeren Unternehmen ergibt. So wird unter dem Begriff **Auftrag** in der Einkaufsabteilung etwas Anderes verstanden als in der Produktion. Die Akzeptanz der erhobenen Informationen kann jedoch durch die Verwendung der Unternehmens- oder Abteilungsbegrifflichkeit erheblich erhöht werden. Aus diesem Grunde bietet die ARIS-Methode sog. Fachbegriffsmodelle an, die sowohl die Verwaltung der unterschiedlichen Begriffe im Sinne einer Synonym-Verwaltung für Datenobjekte als auch die Pflege der Beziehungen zwischen den Objekten der Datenmodelle (Entitytyp, Beziehungstyp,...) und den erhobenen Fachbegriffen des Unternehmens erlaubt.

Zur Darstellung dieser Beziehungen wird der Objekttyp des Fachbegriffs eingeführt. Jedem Informationsobjekt des Datenmodells können nun mehrere Fachbegriffe zugeordnet werden. Ein Beispiel zeigt die folgende Abbildung.

Abbildung 42: Fachbegriffe (1)

Fachbegriffe können untereinander in Beziehung stehen und hierarchisiert werden. Die folgende Abbildung verdeutlicht die Verwendung der verschiedenen Kantentypen zwischen Fachbegriffen.

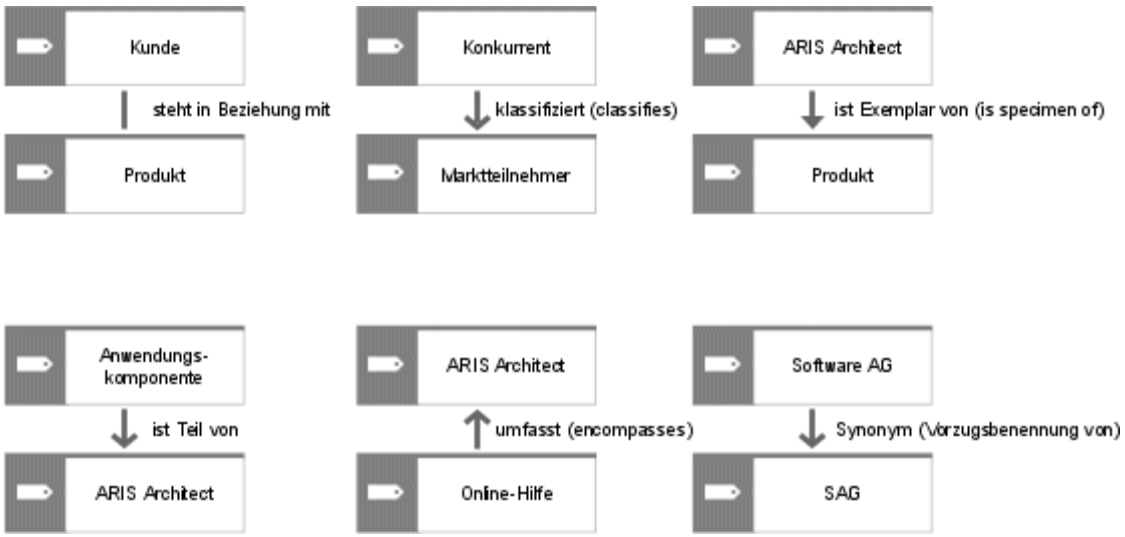


Abbildung 43: Fachbegriffe (2)

Die im Fachbegriffsmodell definierten Fachbegriffe können nun auch in anderen Modelltypen, in denen Informationsobjekte verwendet werden, herangezogen werden, so z. B. in Prozessketten zur Darstellung der Input-/Output-Daten einer Funktion.

3.2.1.7 eERM-Attributzuordnungsdiagramm

Datenmodelle in Form von eERM-Darstellungen, in denen nur die Entitytypen und Beziehungstypen dargestellt werden, weisen meist schon recht komplexe Strukturen auf. Eine Aufnahme der ERM-Attribute in diese Modelle führt dazu, dass die Modelle nicht mehr lesbar sind.

eERM-Attributzuordnungsdiagramme bieten die Möglichkeit, für jeden Entity- und Beziehungstyp die ERM-Attributzuordnungen in einem eigenen Modell zu hinterlegen. Der Objekttyp des eERM (Entitytyp oder Beziehungstyp) kann per Ausprägungskopie in dieses Modell aufgenommen werden und die Beziehungen zu ERM-Attributen modelliert werden. Hierbei kann unterschieden werden, ob es sich bei dem verbundenen ERM-Attribut um ein Schlüsselattribut, einen Fremdschlüssel oder ein beschreibendes Attribut handelt. Ein Beispiel zeigt die folgende Abbildung.

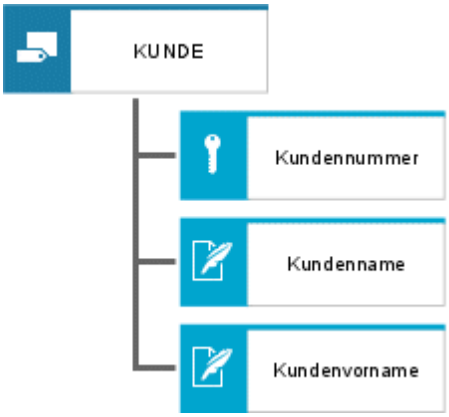


Abbildung 44: Zuordnung von ERM-Attributen zu einem Entitytyp

Neben der Darstellung und Zuordnung einzelner ERM-Attribute können in diesem Modelltyp auch Attributtypgruppen und ihre Zuordnungen dargestellt werden.

Eine Attributtypgruppe stellt die Gruppierung von ERM-Attributen eines Entitytyps dar, die eine enge semantische Zusammengehörigkeit aufweisen. So können z. B. die ERM-Attribute eines Entitytyps, die in ihrer Gesamtheit einen Sekundärschlüssel bilden, zu einer Attributtypgruppe zusammengefasst werden.

Attributtypgruppen werden wie folgt dargestellt:

Abbildung 45: Darstellung einer Attributtypgruppe

Eine Zusammenfassung der möglichen Beziehungen des ERM-Attributzuordnungsdiagramms in der **ARIS-Methode Reference**-Hilfe.

3.2.1.8 IE-Datenmodell

Das IE-Datenmodell entspricht der Notation der Datenmodellierung des CASE-Tools Information Engineering Facility (IE) der Texas Instruments Inc.

Es handelt sich hierbei ebenfalls um eine Notation, die keine eigenen Objekttypen für Beziehungen zwischen Entitytypen kennt.

Ein Beispiel für ein Datenmodell in IE-Notation zeigt diese Abbildung.

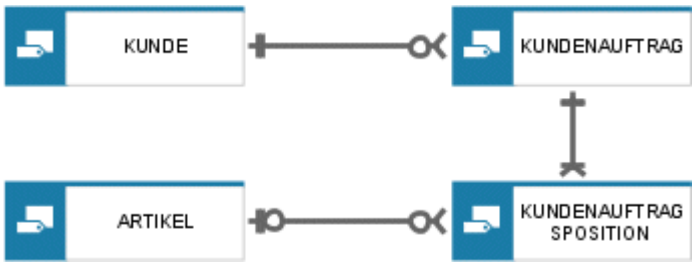


Abbildung 46: Datenmodell in IE-Notation

3.2.1.9 Zusammenfassung der wichtigsten Begriffe und Darstellungsformen des eERM

Die Begriffe und Darstellungsformen der Strukturelemente und Konstruktionsoperatoren des erweiterten Entity-Relationship-Modells (eERM) sind zusammenfassend in der folgenden Abbildung dargestellt (vgl. Scheer, Wirtschaftsinformatik 1994, S. 45).

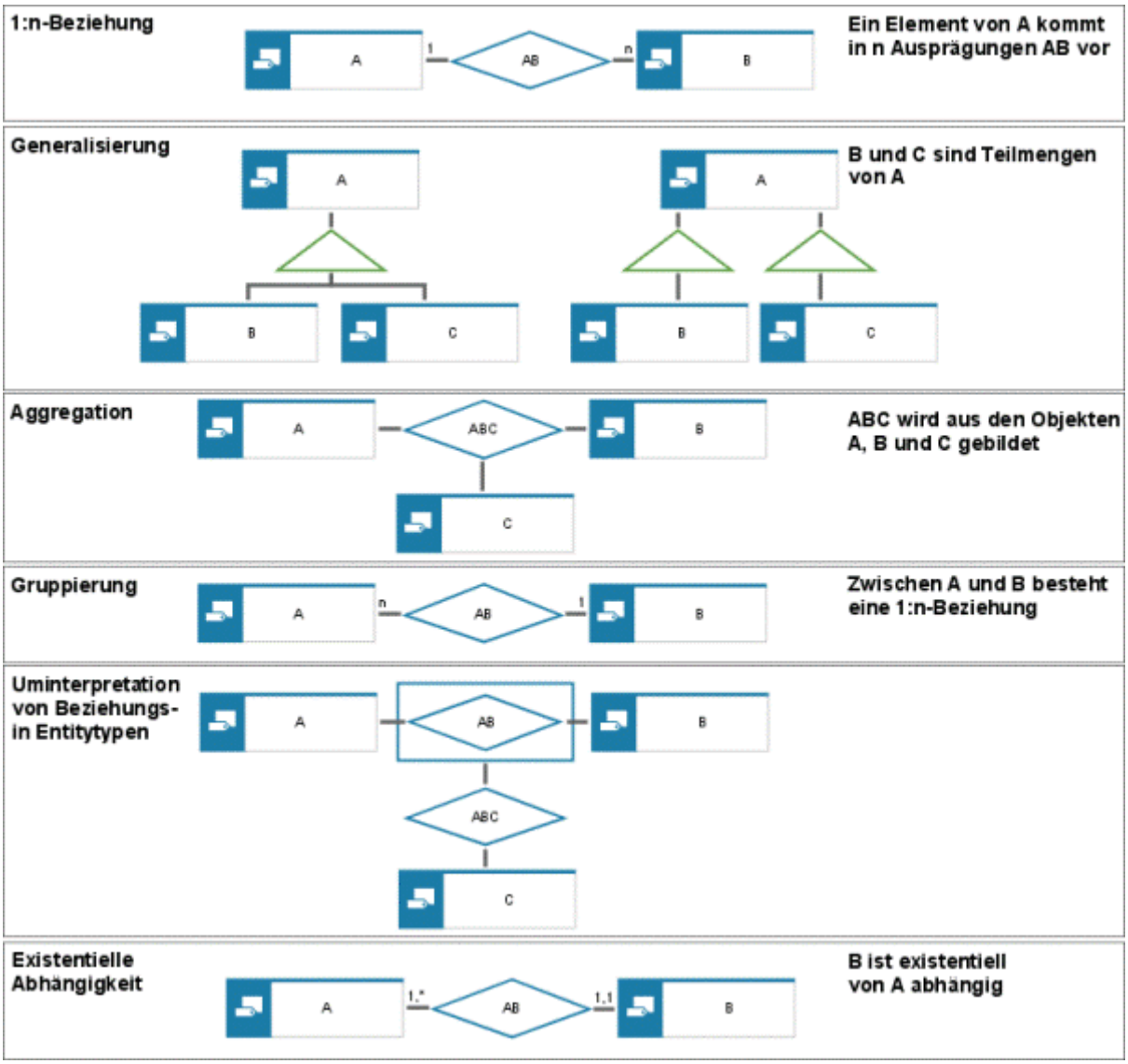


Abbildung 47: Begriffe und Darstellungsformen des eERM

3.2.1.10 Modellierung der Data Warehouse-Struktur

Das Data Warehouse-Strukturdiagramm dient der Beschreibung des Aufbaus eines Data Warehouse. Primär stellt es die statische Beschreibung dar, d. h. die Relationen der Daten untereinander und deren Standorte. In der ARIS-Architektur wird diese Art der Beschreibung in der Datensicht realisiert. Als zentraler Aspekt werden die Relationen der Informationen betrachtet sowie ihre Anordnung zueinander. Die Daten-Dimensionen werden durch den Info Cube beschrieben. Das Zusammenspiel der Dimensionen wird durch das Sternen-Schema (Star scheme) präsentiert (siehe folgende Abbildung). Hierbei kann eine Dimension als Schlüssel andere miteinander verbinden. Die Objekte der einzelnen Dimensionen können bestimmte Werte annehmen, die in den Faktentabellen katalogisiert und durch Kennzahlen genau bestimmt werden. Die Abhängigkeiten werden durch Dimensionstabellen mit ihren Schlüsselattributen und ihren Merkmalen beschrieben. Die Hierarchiebeziehungen der Merkmale untereinander sind durch Baumstrukturen beschrieben. Schließlich kann die Zuordnung von Dimensionen zu Stammdatentabellen mit Hilfe des Strukturdiagramms erfolgen.

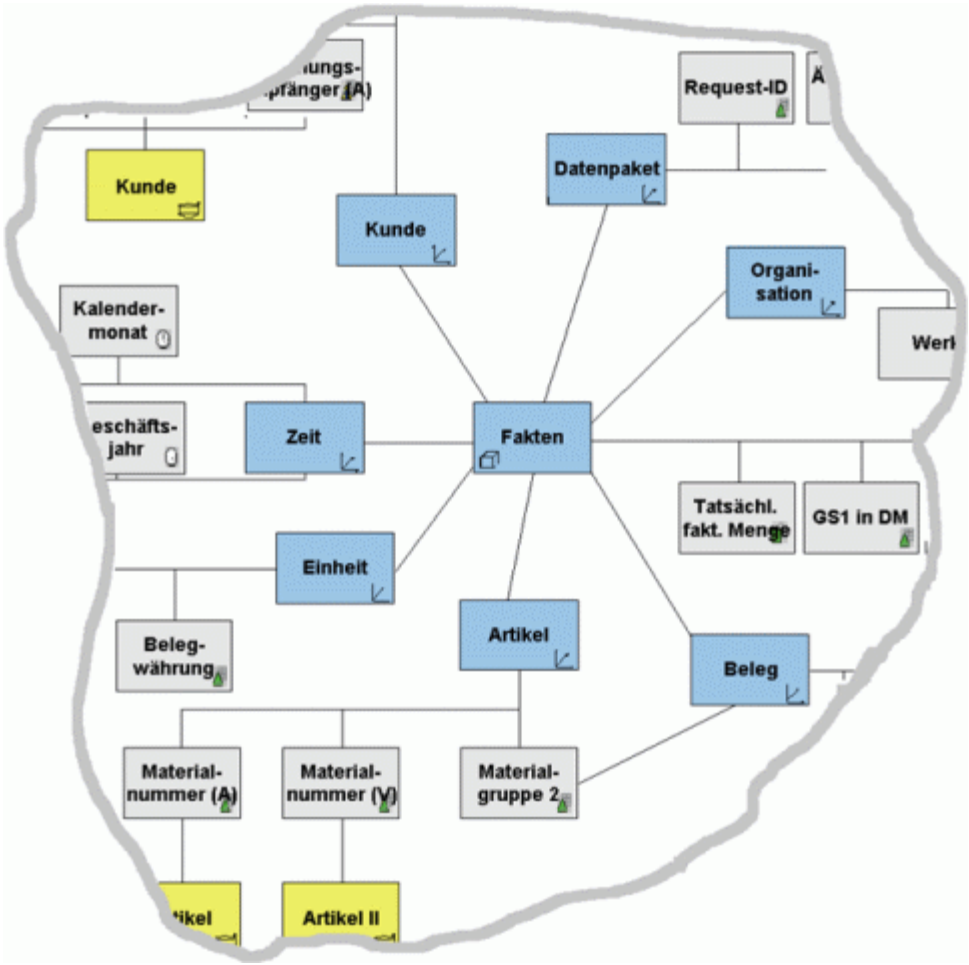


Abbildung 48: Data Warehouse im Sternen-Schema

3.2.1.11 Projektmanagementdaten - Informationsträgerdiagramm

Das Informationsträgerdiagramm ist ein optionaler Bestandteil des Projektmanagements mit ARIS. Es ist dem Fachkonzept der Datensicht zugeordnet und dient der Dokumentation ein- und ausgehender Daten in Form von Dokumenten, Protokollen und ARIS-Modellen.

Die tatsächlich benötigten Dokumente, z. B. eine Textverarbeitungsdatei, sind somit explizit darstellbar und durch die Attribute **Verknüpfung 1** bis **Verknüpfung 4** von ARIS aufrufbar.

Abbildung 49: Informationsträgerdiagramm

3.2.2 Implementierung - Tabellendiagramm

Das Tabellendiagramm erlaubt die Beschreibung der Tabellen und Felder eines Datenbanksystems. Die grafische Darstellung von Tabellen und Feldern wird in folgender Abbildung gezeigt.

Abbildung 50: Grafische Darstellung von Tabelle und Feld

Für jede Tabelle können die einzelnen Felder, die dieser Tabelle zugeordnet sind, dargestellt werden. Dem einzelnen Feld kann zur näheren Spezifikation ein Sortier-Index und die Domäne zugeordnet werden. Ein Beispiel zeigt die folgende Abbildung.

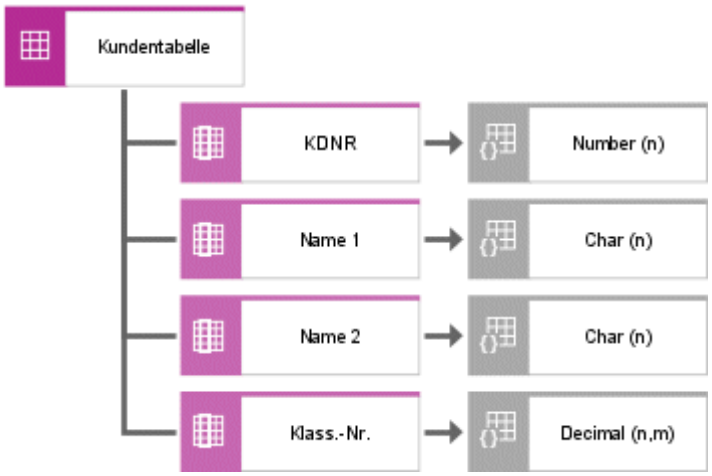


Abbildung 51: Feldzuordnungen

Es können mehrwertige Beziehungen zwischen Tabellen und Relationen oder Entitytypen vorkommen. Im Tabellendiagramm sind diese Beziehungen durch Auswahl der entsprechenden Kanten darstellbar.

Da die Umsetzung oder Dokumentation der im Unternehmen verwendeten Datenbanktabellen und -Felder nicht unbedingt über die Definition eines Relationenschemas erfolgen muss, sind sowohl die Realisierungsbeziehungen zwischen Relationen (bzw. Attributen) und Tabellen (bzw. Feldern) als auch zwischen Entitytypen (bzw. ERM-Attributen) und Tabellen (bzw. Feldern) darstellbar.

Es kann entweder gezeigt werden, welche Relationen und Attribute durch die dargestellten Tabellen und Felder realisiert werden oder - ohne den Weg über die Relationendefinitionen zu gehen - welche Entitytypen, Beziehungstypen und ERM-Attribute durch die dargestellten

Tabellen und Felder abgebildet werden. Beispiele für beide Darstellungsformen werden in der nächsten Abbildung gezeigt.

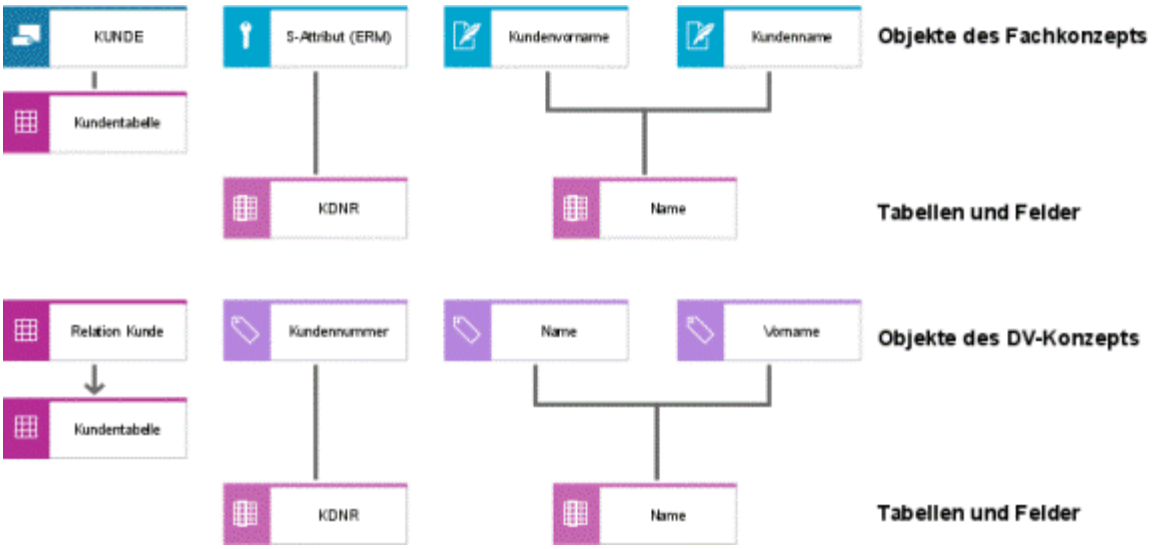


Abbildung 52: Zuordnung der Objekte des Fachkonzeptes und des DV-Konzeptes

Um den genauen Standort bestimmter Tabellen und Felder im Unternehmen definieren zu können, muss es auch möglich sein, jedes einzelne Exemplar einer Tabelle definieren zu können. Gleiches gilt dann, wenn die Zugriffsrechte von Organisationseinheiten auf Tabellen und Felder festgelegt werden sollen. Der bisher vorgestellte Objekttyp Tabelle definiert auf der **Typ-Ebene** den logischen Aufbau einer physikalischen Tabelle und ihrer Felder. Von jeder so definierten Tabelle können im Unternehmen jedoch mehrere Exemplare, verteilt an verschiedenen Standorten, gespeichert auf verschiedenen Medien existieren. Um dies darstellen zu können, werden die Objekttypen Tabelle (Exemplar) und Feld (Exemplar) eingeführt.

Mit Hilfe dieser Objekte kann nun genau definiert werden, wie viele Exemplare einer Tabelle oder eines Feldes existieren. Die folgende Abbildung verdeutlicht diesen Zusammenhang.

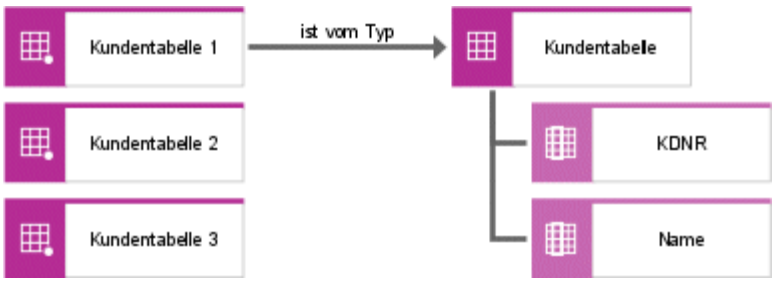


Abbildung 53: Tabellenexemplare

Eine Zusammenstellung der Objekte und Beziehungen des Tabellendiagramms finden Sie in der **ARIS-Methode Reference**-Hilfe.

3.2.3 Rollenzuordnungsdiagramm (RAD - Role Allocation Diagram)

Das SAP®-Referenzmodell wird mit Hilfe von EPKs dargestellt. Sie bilden die betrieblichen Abläufe auf unterschiedlichen Detaillierungsstufen ab. In den EPKs mit dem höchsten Detaillierungsgrad, die der SAP®-Terminologie als **Prozesse** bezeichnet werden, werden die Handlungsabläufe beim Abarbeiten von ausführbaren Einheiten im SAP®-System modelliert. Diesen Prozessen können sowohl Rollen als auch ausführbare Einheiten zugeordnet werden.

In ARIS geschieht dies im Funktionszuordnungsdiagramm, wobei die EPK, die den modellierten Prozess enthält, der entsprechenden Funktionsdefinition hinterlegt sein muss. Im Funktionszuordnungsdiagramm ist also ersichtlich, welche Rollen zur Durchführung der ausführbaren Einheiten notwendig sind. Da es jedoch keine direkten Beziehungen zwischen Rollen und ausführbaren Einheiten gibt, kann bei mehreren auftretenden Rollen nicht entschieden werden, für welche ausführbare Einheit diese zuständig sind. Das Zuordnen von Rollen zu ausführbaren Einheiten erfolgt deshalb im Rollenzuordnungsdiagramm (RAD). Pro Spalte wird eine Rolle dargestellt. In den Spalten werden die ausführbaren Einheiten platziert, wodurch implizite Beziehungen erzeugt werden.

Die Informationen können während der R/3-Einführung verwendet werden, um die für den Betrieb des SAP®-Systems notwendigen Benutzerprofile und Berechtigungskonzepte zu erstellen.

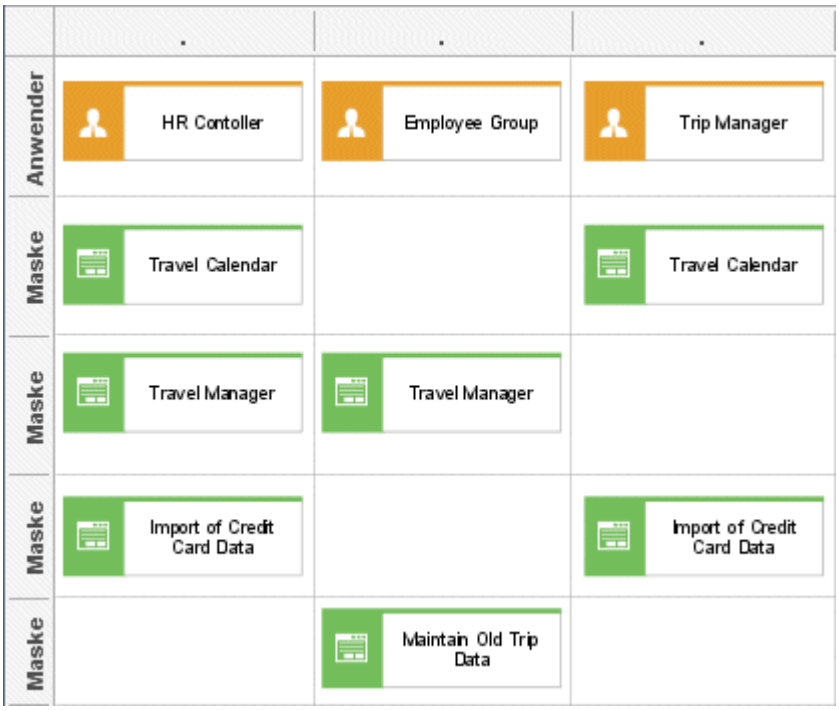


Abbildung 54: Rollenzuordnungsdiagramm (RAD - Role Allocation Diagram)

3.3 Organisationssicht

3.3.1 Fachkonzept

3.3.1.1 Organisatorische Gestaltung der Unternehmen

Unternehmen stellen komplexe soziale Gebilde dar, die in überschaubare Einheiten unterteilt werden. Zur Komplexitätsbewältigung werden Ordnungsmuster definiert und Regeln festlegt. Das Ergebnis dieses Ordnungsprozesses wird dann als Organisation bezeichnet. Welche Rolle organisatorische Betrachtungen im Zusammenhang mit der Entwicklung von Informationssystemen spielen, wurde bisher nur selten untersucht. Neuere Unternehmenskonzepte wie Lean Produktion, Lean Management oder CIM haben jedoch einen engen Bezug zur organisatorischen Gestaltung des jeweiligen Betrachtungsumfeldes. Das ARIS-Konzept besitzt aus diesem Grunde eine eigenständige Beschreibungssicht für die Organisation.

Bei der organisatorischen Strukturierung des Unternehmens können die Aufbauorganisation und die Ablauforganisation unterschieden werden.

Die Aufbauorganisation umfasst die Regeln zur statischen Strukturierung des Unternehmens. Die Ablauforganisation beinhaltet die Regeln, die sich auf die auszuführenden Aufgaben des Unternehmens beziehen. Diese aufgabenbezogene Strukturierung im Sinne einer Verteilung von Funktionen auf Aufgabenträger wird in der Steuerungssicht des ARIS-Hauses behandelt. Die Organisationssicht umfasst somit hauptsächlich die Betrachtung der aufbauorganisatorischen Gestaltung.

Die Gestaltung der optimalen Organisation eines Unternehmens im Sinne der Erreichung eines möglichst geringen Koordinationsaufwands, richtet sich nach ihren Umfeldbedingungen und Zielsetzungen. Es können daher keine allgemeingültigen optimalen Organisationsstrukturen im Sinne von Referenzstrukturen definiert werden.

Die Strukturierung der organisatorischen Einheiten kann sich an verschiedenen Kriterien orientieren.

Sehr verbreitet ist die funktionale Gliederung. Hierbei wird jeweils eine Unternehmensfunktion (Beschaffung, Produktion, Finanz- und Rechnungswesen, Absatz) für alle Produkte und Gebiete zuständig gemacht. Dem Vorteil der hohen Spezialisierung der eingesetzten Mitarbeiter steht jedoch der Nachteil des hohen Kommunikations- und Koordinierungsaufwandes zwischen den Funktionalbereichen gegenüber.

Auch der Entwurf und Einsatz von Informationssystemen hat sich lange an dieser funktionalen Zergliederung der Unternehmen orientiert. Bei der Betrachtung ganzheitlicher Prozessketten im Sinne der zusammenhängenden Bearbeitung gleichartiger Datenobjekte ist aber bei einer solchen Gliederung der Zusammenhang der einzelnen Funktionen nur schwierig herzustellen.

Dies führte im Umfeld der Diskussion um die integrierte Datenverarbeitung zu der Forderung nach einer einheitlichen Datenbasis zur Unterstützung der unterschiedlichen Funktionen. Die angestrebte Integration der Funktionen führt jedoch dazu, dass der durch die funktionale Gliederung angestrebte Zweck der Komplexitätsreduzierung wieder aufgehoben wird.

Deswegen werden häufig bei der Zielsetzung der Funktionsintegration andere Kriterien der organisatorischen Zerlegung angewendet.

Dies kann z. B. eine Zerlegung nach Kriterien wie Gebiete oder Produkte sein. In folgender Abbildung wird eine Zergliederung nach Produkten schematisch dargestellt (vgl. Scheer, Wirtschaftsinformatik 1994, S. 26 f).

Bei einer gebietsorientierten Aufbauorganisation werden die Organisationseinheiten entsprechend der örtlichen Verteilung des Unternehmens oder Unternehmensbereiches festgelegt. Sie eignet sich insbesondere für Vertriebsfunktionen, da regionale Faktoren gezielter berücksichtigt werden können, z. B. unterschiedliche Gesetzgebungen.

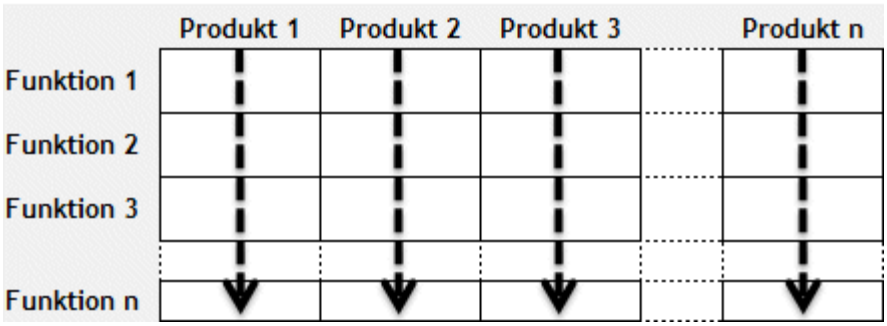


Abbildung 55: Organisatorische Zergliederung nach Produkten

Eine produktbezogene Aufbauorganisation definiert für Produkte bzw. Produktgruppen Organisationseinheiten. Innerhalb einer Produktgruppe werden dann möglichst viele Funktionen, die für die entsprechende Produktgruppe relevant sind, integriert. Zielsetzung ist hierbei, den durch eine funktionale Gliederung auftretenden Kommunikationsaufwand zu verringern. Auf der anderen Seite entsteht jedoch nun ein Vermittlungsaufwand zwischen den produktgruppen-orientierten Teilsystemen.

Um diesen Auswirkungen entgegenzuwirken werden oft gemischte Organisationsformen gebildet. Für das Beispiel Einkauf wird dies in der folgenden Abbildung dargestellt (vgl. Scheer, Wirtschaftsinformatik 1994, S. 26 f).

	Produktgruppe 1	Produktgruppe 2	Produktgruppe 3
Zentraler Einkauf		Lieferantenauswahl	
		Vertragsgestaltung	
Disposition			
Bestellung			
Rechnungsprüfung			

Abbildung 56: Gemischte Organisationsformen

Eine rein funktionale Gliederung würde bedeuten, dass ein zentraler Einkauf für alle Produktgruppen zuständig wäre. Hierbei könnten zwar alle Synergieeffekte, die zwischen den Produktgruppen bestehen, genutzt werden; die Durchführung eines Einkaufsvorgangs durch alle Teilfunktionen würde aber zu erheblichen Abstimmungsproblemen führen. Eine Aufteilung der Einkaufsfunktionen auf die unterschiedlichen Produktgruppen führt dazu, dass für jede Produktgruppe eine Einkaufsabteilung eingerichtet wird, die alle Einkaufsfunktionen wahrnimmt. Synergieeffekte z. B. bei der Lieferantenauswahl und bei Verhandlungen über Rahmenverträge könnten hierbei nur mit einem hohen Abstimmungsaufwand erzielt werden.

Bei der oben dargestellten Zerlegung werden die Einkaufsfunktionen, bei denen hohe Synergieeffekte erwartet werden, funktional zerlegt, d. h. sie werden von einer zentralen Einkaufsstelle wahrgenommen. Die Funktionen, die spezielle Bedürfnisse und Restriktionen der einzelnen Produktgruppen zu berücksichtigen haben, werden objektorientiert nach Produktgruppen zergliedert. Sie können somit zeitnah in die Prozessabläufe der einzelnen Produktgruppen eingebunden werden. Damit wird die operative Abwicklung der Prozesse in den dezentralen Einheiten durchgeführt, während die zwischen den dezentralen Einheiten bestehenden Beziehungen auf der übergeordneten, zentralen Koordinationsebene berücksichtigt werden.

Im ARIS-Konzept werden diesen flexiblen Organisationsformen aufgrund der betont prozessorientierten Betrachtung besondere Beachtung geschenkt. Die Bildung von Aufbauorganisationen unter paralleler Betrachtung mehrerer Zerlegungskriterien wird insbesondere auch von abrechnungsorientierten Ansätzen wie dem Profit-Center-Konzept gefordert.

3.3.1.2 Organigramm

Eine typische Darstellungsart für Organisationsstrukturen ist das Organigramm. In ihm werden je nach den gewählten Strukturierungskriterien die gebildeten Organisationseinheiten (als Aufgabenträger) und ihre Verknüpfungen untereinander dargestellt.

Organisationseinheiten sind Träger der zur Erzielung der Unternehmensziele durchzuführenden Aufgaben.

Die Verknüpfungen stellen die Verbindungen zwischen den Organisationseinheiten dar. In folgender Abbildung wird hierfür ein Beispiel gezeigt.

Zur genaueren Spezifizierung der Unterstellungsverhältnisse werden verschiedene Kantentypen zwischen Organisationseinheiten unterschieden. Eine Kante kann hierbei bedeuten:

- ist fachlich vorgesetzt
- ist disziplinarisch vorgesetzt
- ist bildend für

Werden die funktionalen Zuständigkeiten erfasst, bildet das Organigramm die Verteilung der betrieblichen Aufgaben ab.

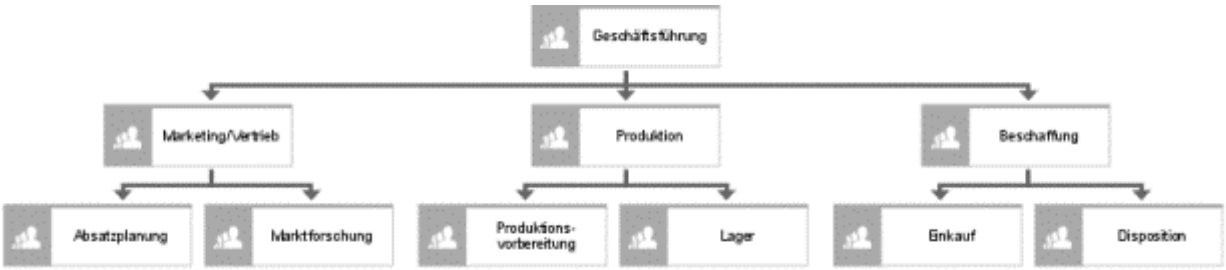


Abbildung 57: Organigramm

Zur Darstellung einzelner Stellen im Unternehmen, für die z. B. Stellenbeschreibungen existieren, steht ein eigener Objekttyp **Stelle** zur Verfügung. Dieser wird wie in der nächsten Abbildung dargestellt. Einer Organisationseinheit können mehrere Stellen zugeordnet werden. Die Bedeutung der Kanten entspricht der zwischen Organisationseinheiten.

Den Stellen und Organisationseinheiten können natürliche Personen, die diese Stellen besetzen, zugeordnet werden. Auch für Personen sind eigene Objekte in ARIS vorhanden, die ebenfalls in folgender Abbildung dargestellt werden. Die Zuordnung einer natürlichen Person zu einer Organisationseinheit drückt aus, dass diese Person als Mitarbeiter der Organisationseinheit zugeordnet ist; die Zuordnung zu einer einzelnen Stelle definiert die aktuelle Stellenbesetzung im Unternehmen. Ein Beispiel zeigt diese Abbildung.

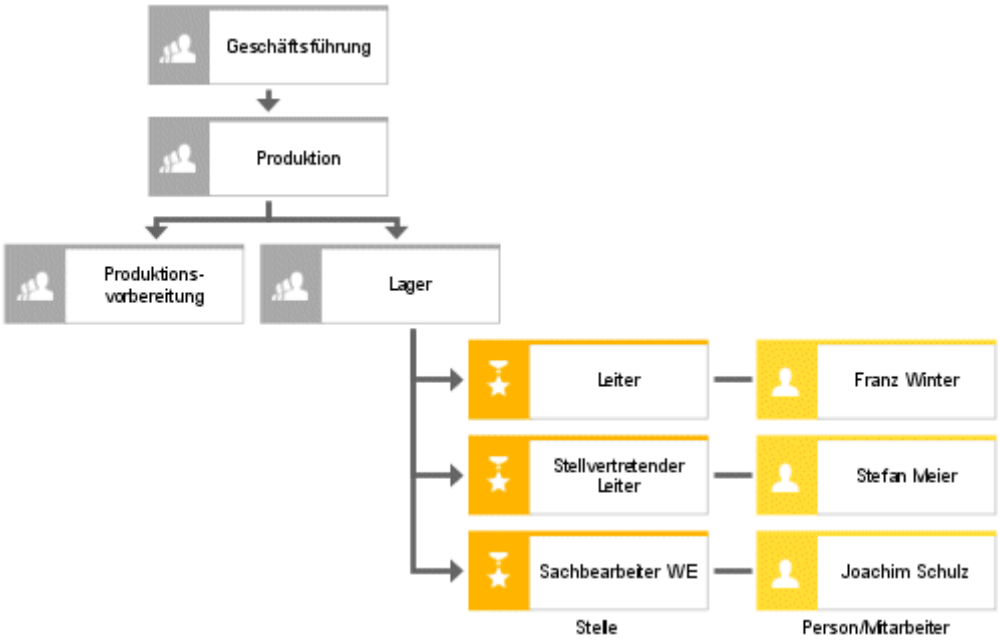


Abbildung 58: Organigramm mit Stellen- und Personenzuordnung

Organisationseinheiten und Personen können auch typisiert werden. Damit kann für eine Organisationseinheit z. B. definiert werden, ob es sich um eine Abteilung, Hauptabteilung oder Gruppe handelt; Personen können z. B. den Rollen **Abteilungsleiter**, **Gruppenleiter** oder **Projektleiter** zugeordnet werden.

Zur Darstellung dieser Typisierung existieren die Objekte **Organisationseinheitstyp** und **Rolle**. Ein Beispiel der erläuterten Typisierung von Organisationseinheiten und Personen zeigt diese Abbildung.

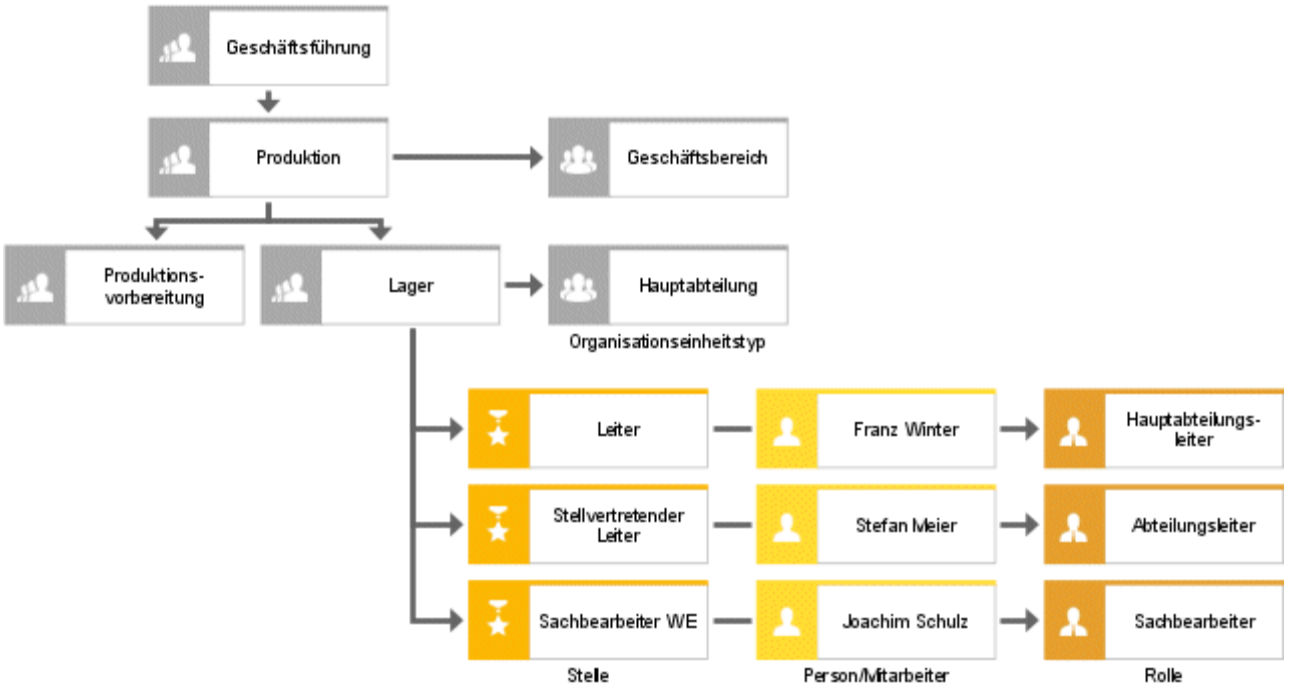


Abbildung 59: Personentypen

Die Verwendung dieser Objekttypen ermöglicht es, allgemeine Geschäftsregeln abzubilden, die von konkreten Organisationseinheiten oder Personen des Unternehmens abgeleitet sind. So besteht in Prozessketten die Möglichkeit zu definieren, dass nur bestimmte Rollen eine Funktion ausführen dürfen oder nur bestimmte Rollen Zugriff auf ein Informationsobjekt haben dürfen.

Die Modellierung der Aufbauorganisation des Unternehmens ist Ausgangspunkt für die in der DV-Konzept-Ebene zu definierenden Netztopologien, die diese Aufbauorganisation bestmöglich unterstützen sollen. Zur Definition der Netztopologie gehören Netzkanten und Netzknoten, die sich an bestimmten Standorten des Unternehmens befinden. Der Standort einer Organisationseinheit ist somit das wichtigste Bindeglied zwischen Fachkonzept und DV-Konzept der Organisationssicht. Jeder Organisationseinheit ist deshalb im Fachkonzept bereits der Standort zuordenbar, an dem sie sich befindet. Ein Beispiel hierzu zeigt diese Abbildung.

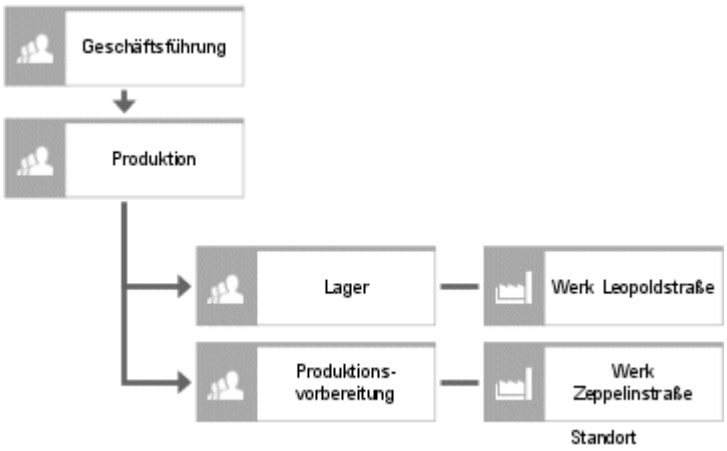


Abbildung 60: Standortzuordnungen

Standorte sind beliebig hierarchisierbar. Ein Standort kann ein gesamtes Werk, ein Gebäude, oder auch bei einer detaillierteren Betrachtung ein Büro bis hin zu einem einzelnen Arbeitsplatz in einem Zimmer sein. Damit ist es im DV-Konzept möglich, Netzknoten eines Netzes den einzelnen Arbeitsplätzen der organisatorischen Einheit zuzuordnen, also z. B. zu definieren, dass in einem Büro (Zimmer 102) insgesamt 3 Netzknoten zur Verfügung stehen müssen.

Diese Abbildung zeigt das Beispiel einer Standorthierarchisierung.

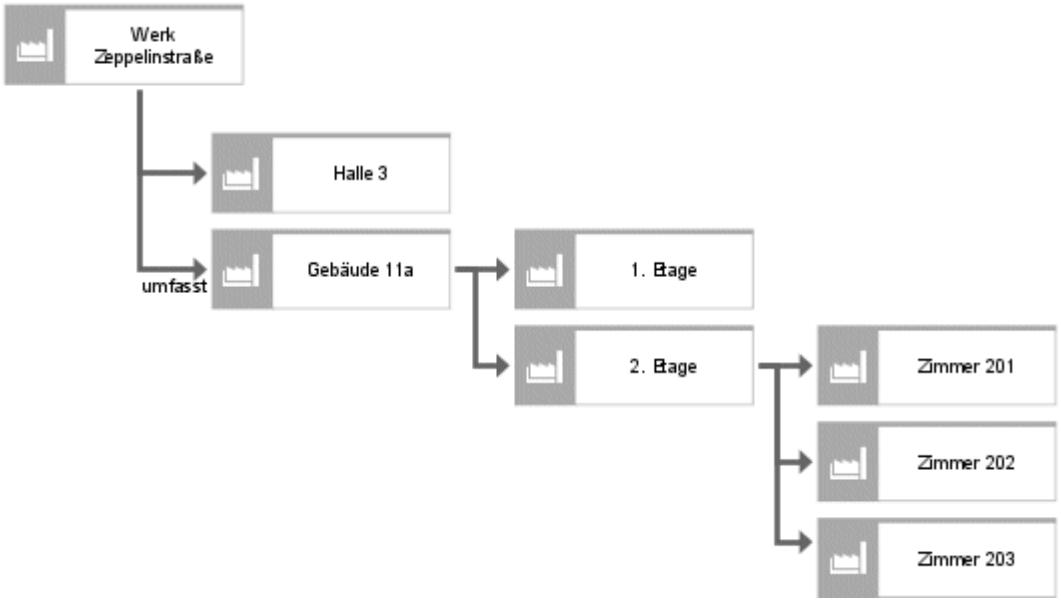


Abbildung 61: Standorthierarchien

3.3.2 DV-Konzept - Netztopologie

Die im Organigramm dargestellte Aufbauorganisation des Unternehmens kann nun durch Kommunikations- und Informationssystemlandschaften unterstützt werden. Die Anforderungen, die an den strukturellen Aufbau dieser Informationssysteme gestellt werden, sind im DV-Konzept zunächst allgemein in Form von Netztopologien definierbar.

Im Modell **Netztopologie** können zunächst unterschiedliche Netztypen aufgenommen werden.

Ein Netztyp stellt eine Typisierung einzelner Netzexemplare dar, die sich auf exakt der gleichen technologischen Basis befinden.

Ein Beispiel für einen Netztyp zeigt diese Abbildung:

Abbildung 62: Grafische Darstellung eines Netztyps

Netztypen können untereinander verbunden sein und sind als logische Konstrukte auch hierarchisierbar.

Jedem Netztyp sind Des Weiteren mögliche Netzknotentypen und Netzkantentypen zuordenbar. Damit können technologische Beschränkungen, die sich aus der Auswahl eines bestimmten Netztyps für ein Unternehmen ergeben, sofort erkannt werden. Für jeden Netzkantentyp kann dargestellt werden, in welchen Netzknotentypen er enden kann.

Bei den Hardwarekomponententypen kann es sich zum einen um Netz-Hardware zur Realisierung der definierten Netzstrukturen oder um Hardwarekomponententypen handeln, die an Netzknotentypen angeschlossen werden können.

Wie schon bei Anwendungssystemtypen oder Netztypen handelt es sich auch bei den Hardware-Komponententypen nicht um einzelne z. B. durch Inventarisierungsnummern des Unternehmens identifizierbare Exemplare von Hardwarekomponenten. Es handelt sich stattdessen um Typisierungen auf der gleichen technologischen Basis Hardwarekomponententypen sind beliebig hierarchisierbar.

Ein Hardwarekomponententyp stellt eine Typisierung einzelner Exemplare von Hardwarekomponenten dar, die sich auf exakt der gleichen technologischen Basis befinden.

In Verbindung mit den Netzknoten- und -kantentypen kann nun eine Art Referenzmodell der Netztopologie aufgebaut werden. Hierbei ist darstellbar, welche Hardwarekomponententypen zur Realisierung bestimmter Netzkantentypen oder Netzknotentypen verwendet werden können. Bei den Kantentypen kann es sich dabei z. B. um einen bestimmten Typ eines Übertragungskabels handeln. Es kann aber auch dargestellt werden, welcher Hardwarekomponententyp an welchen Netzknotentyp anschließbar ist. Netzknotentypen

können auch mit Hardwarekomponententypen, die zur Realisierung von Knotentypen verwendet werden, in Beziehung stehen. Ein Beispiel zeigt die folgende Abbildung.

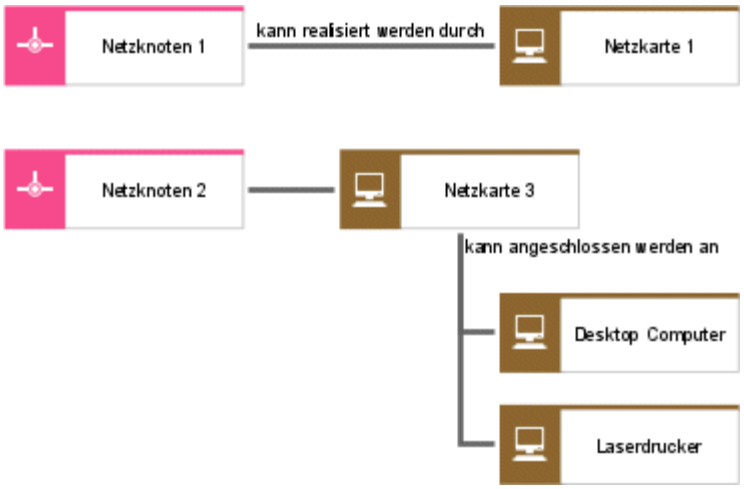


Abbildung 63: Netztopologie

Die Verbindung der Netztopologie zu den Objekten des Fachkonzeptes wird über zwei Konstrukte hergestellt.

Zum einen kann für jeden Hardwarekomponententyp festgelegt werden, welche Organisationseinheit oder Stelle für ihn verantwortlich sein kann.

Zum anderen kann für jeden Netztyp, jeden Netzknotentyp und Netzkantentyp als auch für jeden Hardwarekomponententyp definiert werden, an welchem Standort des Unternehmens er sich befinden kann. Somit ist der Standort das zentrale Bindeglied zwischen dem Fachkonzept und dem DV-Konzept der Organisationssicht.

Eine Zusammenstellung der Objekt- und Beziehungstypen des Netztopologie-Modells finden Sie in der **ARIS-Methode Reference**-Hilfe.

3.3.3 Implementierung

3.3.3.1 Netzdiagramm

Im Netzdiagramm erfolgt die Darstellung der konkreten Umsetzungen der im DV-Konzept spezifizierten Netztopologie.

Mit Hilfe des Objektes **Netz** werden die im Unternehmen konkret existierenden Netze aufgenommen. Für jedes Netz kann spezifiziert werden, aus welchen Netzknoten und Netzkanten es besteht.

Sowohl für jedes Netz als auch für jeden Netzknoten und jede Netzkante kann der genaue Standort im Unternehmen angegeben werden. Ein Standort kann hierbei ein ganzes Werk, ein bestimmtes Gebäude, ein Gebäudekomplex, ein Büro oder ein bestimmter Arbeitsplatz sein.

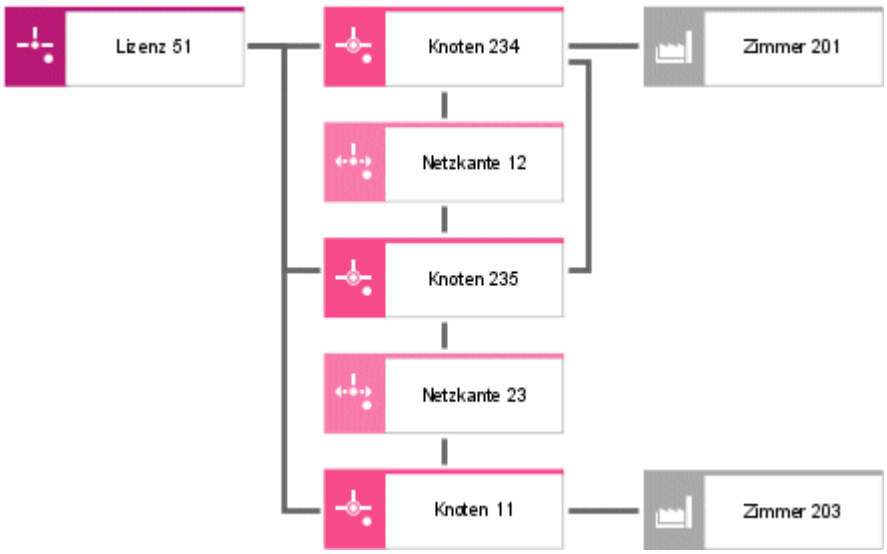


Abbildung 64: Netzdiagramm mit Standortzuordnung

Für jede Netzkante und jeden Netzknoten kann die Hardwarekomponente mit denen sie realisiert sind, aufgenommen werden. Daneben ist auch der strukturelle Aufbau jeder Hardwarekomponente darstellbar. Hardwarekomponenten dienen zum einen der Realisierung von Netzkanten und Netzknoten, können zum anderen aber auch an Netzknoten angeschlossen sein. Auch diese Beziehung ist im Netzdiagramm darstellbar. Für jedes Objekt auf Exemplar-Ebene ist auch die Beziehung zum entsprechenden Objekt der DV-Konzept-Ebene modellierbar. So kann gezeigt werden, dass es sich bei dem Netz im Werk **Zeppelinstraße** um ein Netz des Typs **FDDI ANSI X3.139** handelt.

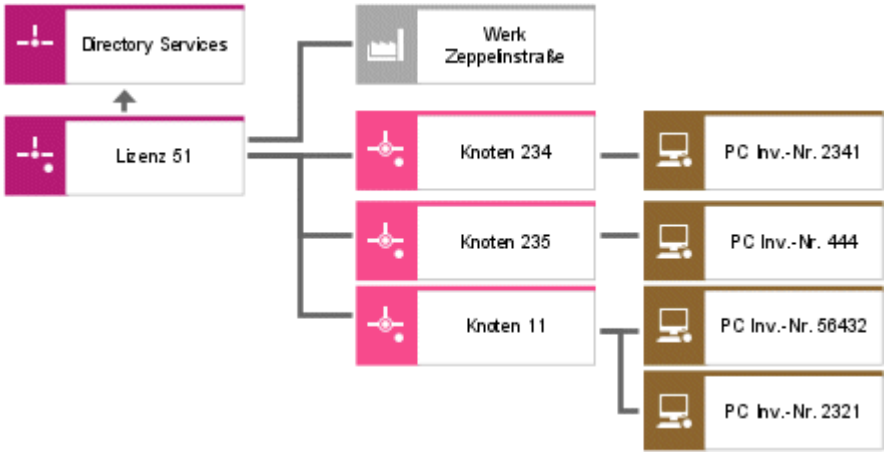


Abbildung 65: Netzdiagramm mit Hardwarekomponenten und Standortzuordnung

Im Netzdiagramm sind somit über die Typzuordnungen die Verbindungen zum DV-Konzept und über die Zuordnung der Netzkomponenten zu konkreten Standorten die Verbindungen zum Fachkonzept hergestellt.

Eine Zusammenstellung der Objekt- und Beziehungstypen des Netzdiagramms finden Sie in der **ARIS-Methode Reference**-Hilfe.

3.3.3.2 Modellierung des Materialflusses - Technische Ressourcen

Zur Darstellung des Materialflusses in Prozessmodellen (EPK (Materialfluss)), werden den einzelnen Funktionen des Geschäftsprozesses die Materialtypen als Funktionsinput oder Funktionsoutput zugeordnet. Ähnlich der Zuordnung von Informationsobjekten zu Funktionen (Darstellung der Transformation von Informationen durch Funktionen) wird mit dieser Zuordnung die Transformation der Inputmaterialtypen zu Output-Materialtypen dargestellt. Zusätzlich kann in den Prozessketten aufgenommen werden, welche technischen Ressourcen zur Materialtransformation benötigt werden. Hierbei werden Betriebsmittel, Lagereinrichtungen, Transportsysteme und technische Hilfsmittel unterschieden.

Im Modelltyp **Technische Ressourcen** können diese Ressourcen hierarchisiert, typisiert und klassifiziert werden. Hierzu stehen folgende Objekttypen zur Verfügung:

BETRIEBSMITTEL

Betriebsmittel stellen die Exemplare verschiedener Betriebsmitteltypen dar, die einer Unternehmung zur Wahrnehmung ihrer Aufgaben zur Verfügung stehen. Betriebsmittel werden oft durch die Vergabe von Inventarisierungsnummern, z. B. Nummer einer Produktionsanlage identifiziert.

BETRIEBSMITTELTYP

Ein Betriebsmitteltyp stellt die Typisierung einzelner Betriebsmittel dar, die sich auf der exakt gleichen technologischen Basis befinden.

BETRIEBSMITTELKLASSE

Gleichartige Betriebsmitteltypen können zu einer Betriebsmittelklasse zusammengefasst werden. Die Gleichartigkeit kann sich hierbei hinsichtlich verschiedener Klassifizierungsaspekte ergeben. Ein Betriebsmitteltyp kann somit auch mehreren Betriebsmittelklassen zugeordnet werden.

LAGEREINRICHTUNG

Lagereinrichtungen stellen die Exemplare verschiedener Lagereinrichtungstypen dar, die einer Unternehmung zur Wahrnehmung ihrer Aufgaben zur Verfügung stehen. Lagereinrichtungen werden oft durch die Vergabe von Lagernummern identifiziert.

LAGEREINRICHTUNGSTYP

Ein Lagereinrichtungstyp stellt die Typisierung einzelner Lagereinrichtungen dar, die sich auf der exakt gleichen technologischen Basis befinden.

LAGEREINRICHTUNGSKLASSE

Gleichartige Lagereinrichtungstypen können zu einer Lagereinrichtungsklasse zusammengefasst werden. Die Gleichartigkeit kann sich hierbei hinsichtlich verschiedener Klassifizierungsaspekte ergeben. Ein Lagereinrichtungstyp kann somit auch mehreren Lagereinrichtungsklassen zugeordnet werden.

TECHN. HILFSMITTEL

Ein techn. Hilfsmittel ist ein einzelnes Exemplar eines technischen Hilfsmitteltyps. Es kann i. d. R. über eine Inventarisierungsnummer identifiziert werden.

TECHN. HILFSMITTELTYP

Ein techn. Hilfsmitteltyp stellt die Typisierung einzelner techn. Hilfsmittel dar, die sich auf der exakt gleichen technologischen Basis befinden.

TECHN. HILFSMITTELKLASSE

Gleichartige techn. Hilfsmitteltypen können zu einer techn. Hilfsmittelklasse zusammengefasst werden. Die Gleichartigkeit kann sich hierbei hinsichtlich verschiedener Klassifizierungsaspekte ergeben. Ein techn. Hilfsmitteltyp kann somit auch mehreren techn. Hilfsmittelklassen zugeordnet werden.

TRANSPORTSYSTEM

Ein Transportsystem ist ein einzelnes Exemplar eines Transportsystemtyps. Es kann i. d. R. über eine Inventarisierungsnummer oder Anlagennummer identifiziert werden.

TRANSPORTSYSTEMTYP

Ein Transportsystemtyp stellt die Typisierung einzelner Transportsysteme dar, die sich auf der exakt gleichen technologischen Basis befinden.

TRANSPORTSYSTEMKLASSE

Gleichartige Transportsystemtypen können zu einer Transportsystemklasse zusammengefasst werden. Die Gleichartigkeit kann sich hierbei hinsichtlich verschiedener Klassifizierungsaspekte ergeben. Ein Transportsystemtyp kann somit auch mehreren Transportsystemklassen zugeordnet werden.

Die Hierarchisierungsmöglichkeiten des Modelltyps **Techn. Ressourcen** ermöglichen es, den Aufbau komplexer techn. Anlagen zu beschreiben. So können z. B. die Komponenten einer komplexen Produktionsanlage und ihre Beziehungen untereinander dargestellt werden.

Neben den oben beschriebenen Modellierungsmöglichkeiten können sowohl Standortzuordnungen als auch organisatorische Verantwortlichkeiten für techn. Ressourcen definiert werden. Hierzu stehen die aus dem Modelltyp **Organigramm** bekannten Objekttypen

Standort, **Organisationseinheit**, **Gruppe**, **Stelle** und **Person** zur Verfügung. Diese Objekttypen können mit den Objekttypen **Betriebsmittel**, **Lagereinrichtung**, **techn. Hilfsmittel** und **Transportsystem** verbunden werden.

Ein Beispiel für den Modelltyp **Techn. Ressourcen** zeigt diese Abbildung.

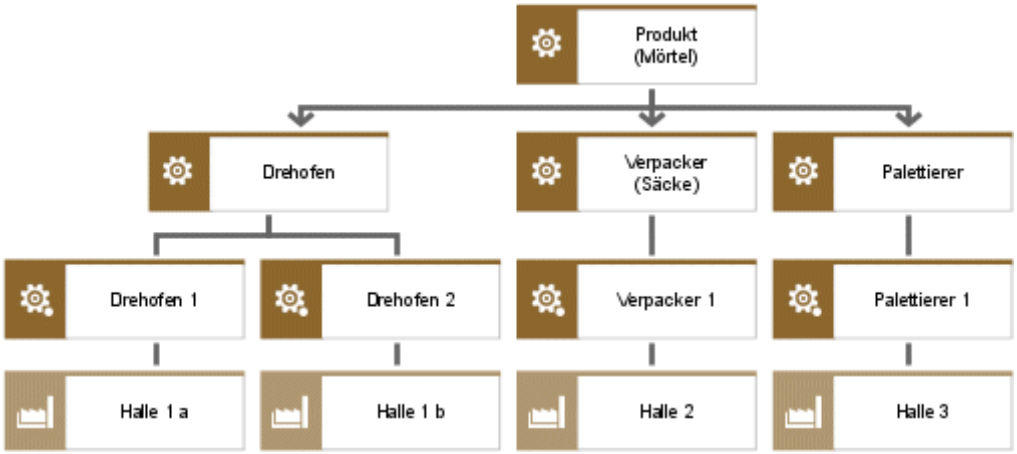


Abbildung 66: Beispiel für ein Modell "Technische Ressourcen"

3.4 Prozesssicht

3.4.1 Fachkonzept

In der Steuerungs-/Prozesssicht werden die Beziehungen zwischen den Objekten der Daten-, Organisations- und Funktionssicht betrachtet. Die Beziehungen, die zu betrachten sind, ergeben sich aus den Kanten zwischen den Sichten.

Zunächst werden die Beziehungen zweier Sichten zueinander betrachtet, danach werden Diagramme für die Darstellung der Beziehungen zwischen allen drei Sichten vorgestellt.

3.4.1.1 Funktionen mit Organisation - EPK

Die Verbindung der Funktionssicht mit der Organisationssicht dient dazu, die im Funktionsbaum definierten Funktionen den Aufgabenträgern (Organisationseinheiten) des Organigramms zuzuordnen. Diese Zuordnung definiert die Verantwortung und die Entscheidungskompetenz einer Organisationseinheit für die zugeordneten Funktionen. Betrachtet man diese organisatorischen Zuordnungen entlang einer Prozesskette (Geschäftsprozesse), wird der Grad der Funktionsintegration, d. h. die von einer Organisationseinheit zu bearbeitenden Funktionsschritte eines Geschäftsprozesses, definiert.

Ein Beispiel für die Zuordnung von Organisationseinheiten zu Funktionen zeigt folgende Abbildung. Hierbei ist der links platzierten Funktion die für ihre Ausführung verantwortliche Organisationseinheit zugeordnet. Die Über-/Unterordnungen der Funktionen wurden in der Funktionssicht (Funktionsbaum) und die Beziehungen zwischen den Organisationseinheiten in der Organisationssicht (Organigramm) dargestellt. Sie müssen somit an dieser Stelle nicht mehr definiert werden.

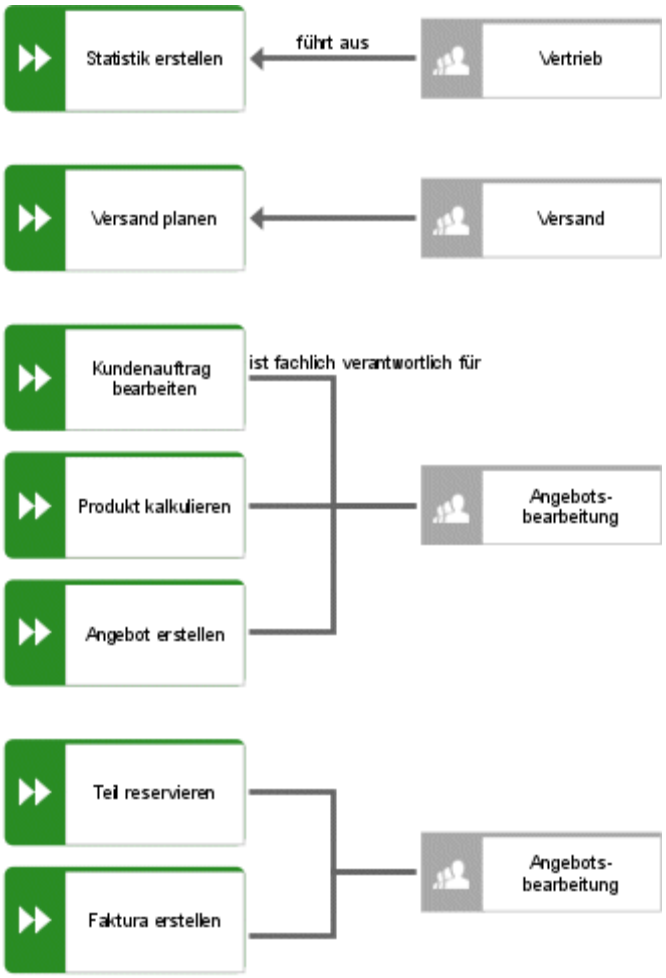


Abbildung 67: Zuordnung der Organisationseinheiten zu Funktionen

3.4.1.2 Ereignissteuerung - ereignisgesteuerte Prozesskette (EPK)

Die Ablauffolge von Funktionen im Sinne eines Geschäftsprozesses wird in Prozessketten dargestellt. Dabei können für jede Funktion die Start- und Endereignisse angegeben werden. Ereignisse sind Auslöser von Funktionen und Ergebnisse von Funktionen.

Ein Ereignis ist das Eingetretensein eines betriebswirtschaftlich relevanten Zustandes eines Informationsobjektes, der den weiteren Ablauf des Geschäftsprozesses steuert oder beeinflusst. Ereignisse lösen Funktionen aus und sind Ergebnisse von Funktionen. Im

Gegensatz zu einer Funktion, die ein zeitverbrauchendes Geschehen darstellt, ist ein Ereignis auf einen Zeitpunkt bezogen.

Die Zustandsänderung eines Informationsobjektes kann sich zum einen auf das erste Aufgetretensein dieses Informationsobjektes beziehen (Bsp.: **Kundenanfrage ist eingetroffen**) oder auf eine Zustandsänderung im Sinne einer Statusänderung, die in einer Attributausprägung festgehalten wird (Bsp.: **Angebot ist abgelehnt**). Da Informationsobjekte und Attribute in der Datensicht von ARIS beschrieben werden, ist die ereignisgesteuerte Darstellung von Prozessketten eine Verbindung zwischen der Daten- und Funktionssicht und ist somit der Steuerungssicht von ARIS zugeordnet.

Grafisch werden Ereignisse durch Sechsecke dargestellt. Die Bezeichnung sollte sowohl das Informationsobjekt (**Auftrag**) als auch die Zustandsänderung dieses Informationsobjektes (**ist eingetroffen**) beinhalten. In der nächsten Abbildung sind Ereignisse abgebildet.

Abbildung 68: Ereignisse (grafische Darstellung)

Ereignisse lösen Funktionen aus und sind Ergebnisse von Funktionen. Durch das Hintereinanderschalten dieses Ereignis-Funktionswechsels entstehen so genannte ereignisgesteuerte Prozessketten. Eine ereignisgesteuerte Prozesskette (EPK) zeigt den logisch-zeitlichen Ablauf eines Geschäftsprozesses.

Ein Beispiel für eine EPK zeigt die folgende Abbildung. Da Ereignisse definieren, durch welchen Zustand oder Bedingung eine Funktion gestartet wird und welcher Zustand den Abschluss einer Funktion definiert, sind die Start- und Endknoten einer solchen EPK immer Ereignisse. Von einem Ereignis können mehrere Funktionen gleichzeitig ausgehen, andererseits kann eine Funktion mehrere Ereignisse als Ergebnis haben. Um diese Verzweigungen und Bearbeitungsschleifen in einer EPK darstellen zu können, wird eine Regel in Form eines Kreises verwendet. Die Verbindungen dienen jedoch nicht nur als grafische Verknüpfungen, sondern sie definieren die logischen Verknüpfungen der Objekte, die sie verbinden.

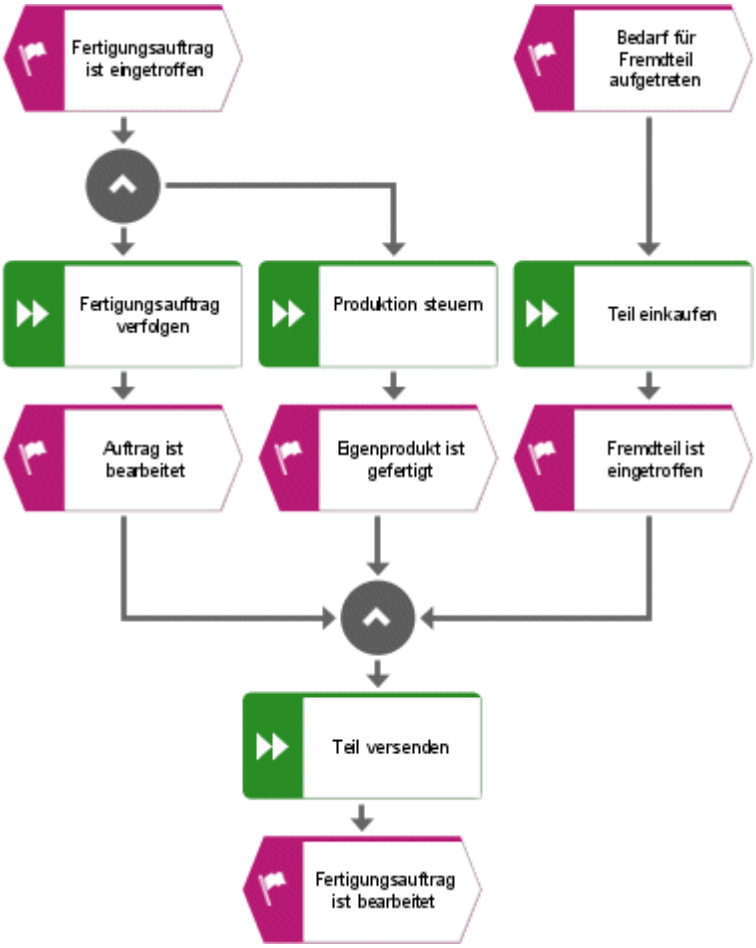


Abbildung 69: Beispiel für eine EPK

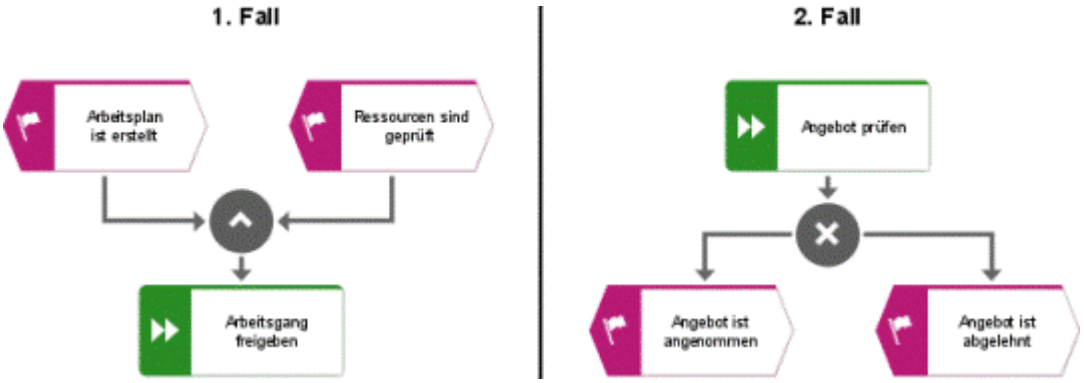


Abbildung 70: Beispiele für Regeln

Im ersten Fall dieser Abbildung liegt eine UND-Verknüpfung durch eine UND-Regel der Startereignisse vor. Dies bedeutet, dass der Vorgang **Arbeitsgang freigeben** nur dann gestartet wird, wenn sowohl der Arbeitsplan vorhanden ist als auch die benötigten Ressourcen geprüft sind. Beide Ereignisse müssen also eingetreten sein, damit der Vorgang starten kann. Der zweite Fall stellt eine Exklusiv-oder-Verknüpfung (exklusives oder) mit Hilfe einer XOR-Regel dar. Ergebnis der Funktion **Lieferantenangebot prüfen** kann sein, dass das Angebot entweder angenommen oder abgelehnt wird. Beides kann jedoch nicht gleichzeitig eintreten. Neben diesen beiden Fällen und der Verknüpfung im Sinne eines „einschließenden oder“ sind auch komplexere Beziehungen denkbar.

Grundsätzlich können somit zwei Arten von Verknüpfungen unterschieden werden:

1. Ereignisverknüpfungen und
2. Funktionenverknüpfungen.

Eine Übersicht der möglichen Ereignis- und Funktionsverknüpfungen zeigt folgende Abbildung (vgl. Hoffmann, Kirsch, Scheer, Modellierung mit Ereignisgesteuerten Prozessketten, 1993, S. 13).

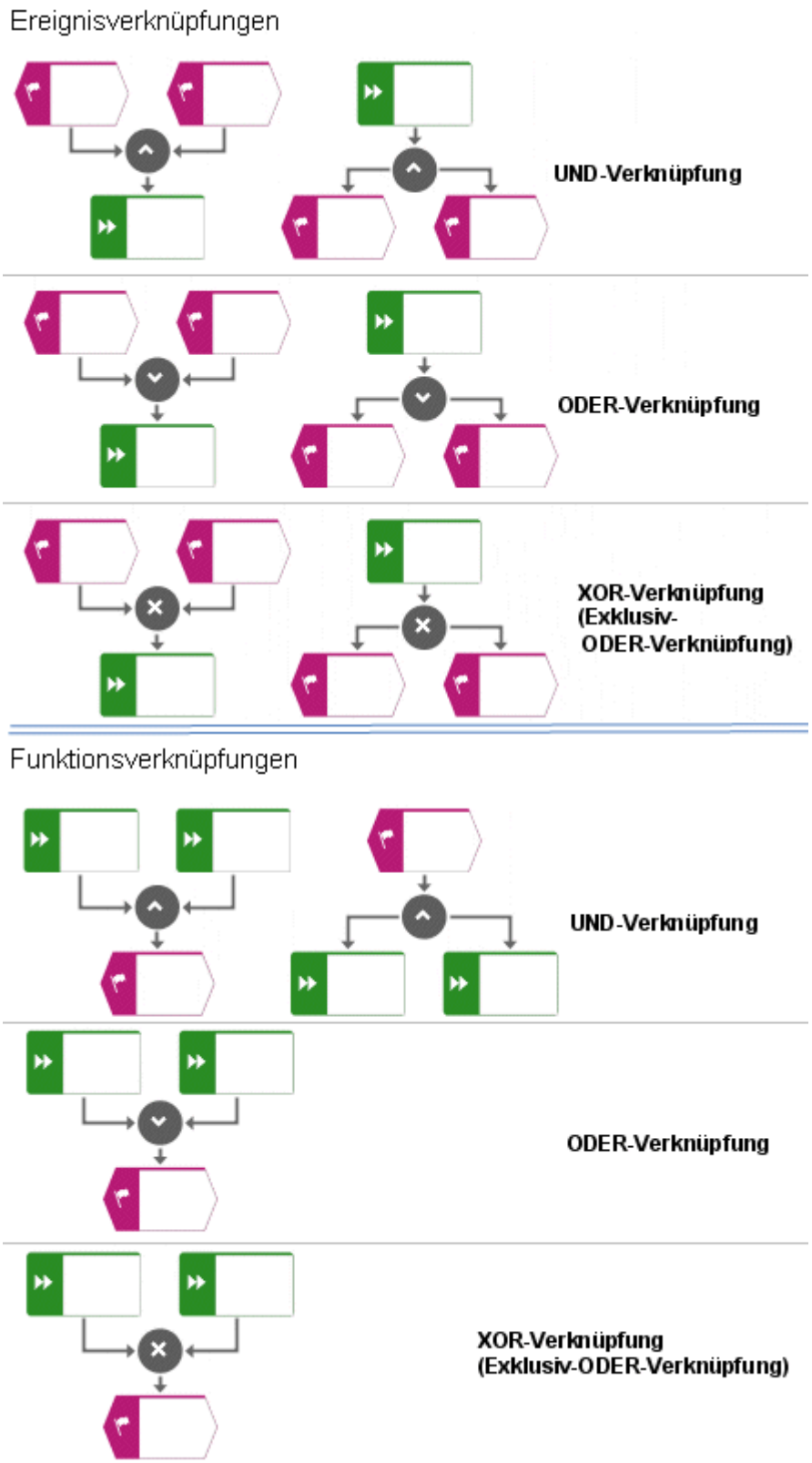


Abbildung 71: Verknüpfungsoperatoren (Regeln)

Zu beachten sind hierbei vor allem die Einschränkungen bezüglich der Funktionenverknüpfungen. Da Ereignisse keine Entscheidungen treffen können (dies können nur Funktionen), ist die Verknüpfung eines auslösenden Ereignisses mit ODER- und Exklusiv-ODER-Regel nicht erlaubt!

Nachfolgend werden mögliche Verknüpfungen durch Beispiele erläutert.

DIE VERKNÜPFUNG VON AUSLÖSENDEN EREIGNISSEN

UND-REGEL

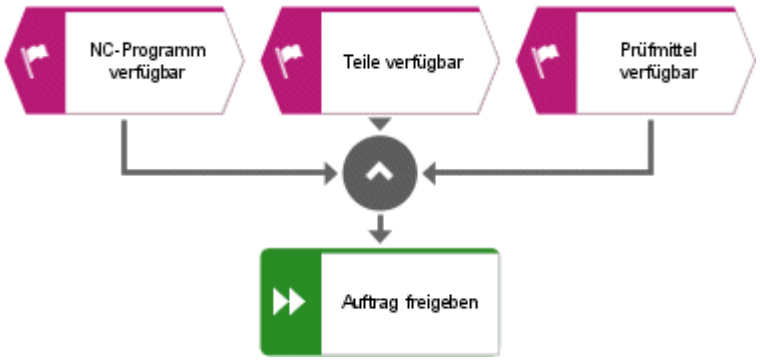


Abbildung 72: Und-Verknüpfung auslösender Ereignisse

Die Funktion kann erst gestartet werden, wenn alle Ereignisse eingetreten sind.

ODER-REGEL

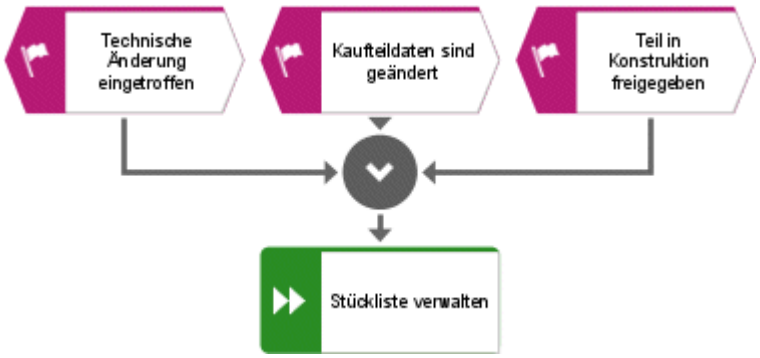


Abbildung 73: Oder-Verknüpfung auslösender Ereignisse

Die Funktion wird ausgeführt, wenn mindestens eines der Ereignisse eingetreten ist.

EXKLUSIV-ODER-REGEL (ENTWEDER-ODER-REGEL)

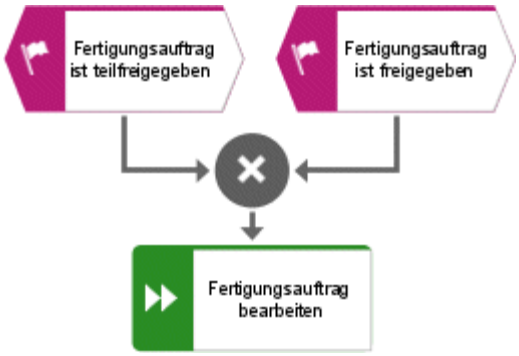


Abbildung 74: Exklusiv-oder-Verknüpfung auslösender Ereignisse

Die Funktion wird gestartet, wenn genau ein und nur ein Ereignis eingetreten ist.

DIE VERKNÜPFUNG VON ERZEUGTEN EREIGNISSEN

UND-REGEL

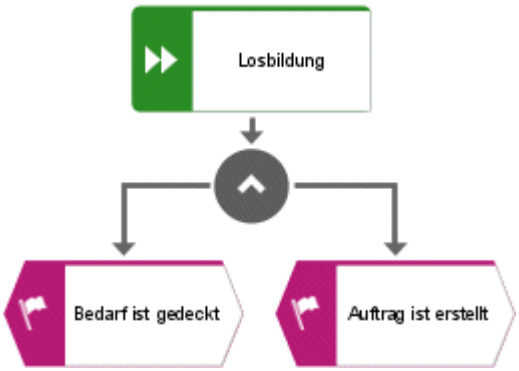


Abbildung 75: Und-Verknüpfung erzeugter Ereignisse

Die Funktion führt dazu, dass alle Ereignisse eintreten.

ODER-REGEL

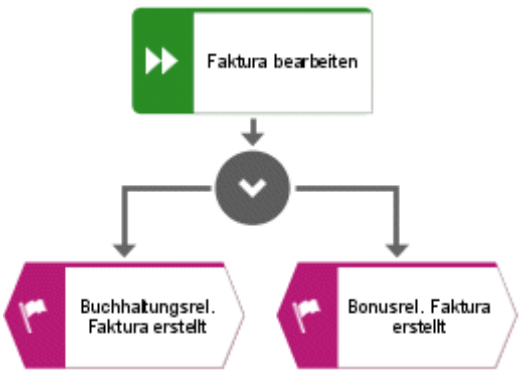


Abbildung 76: Oder-Verknüpfung erzeugter Ereignisse

Die Ausführung der Funktion führt dazu, dass mindestens eines der Ereignisse eintritt.

EXKLUSIV-ODER-REGEL (ENTWEDER-ODER-REGEL)

Abbildung 77: Exklusiv-oder-Verknüpfung erzeugter Ereignisse

Die Ausführung der Funktion führt dazu, dass höchstens eines der Ereignisse eintritt.

DIE VERKNÜPFUNG VON FUNKTIONEN MIT ERZEUGTEN EREIGNISSEN

UND-REGEL

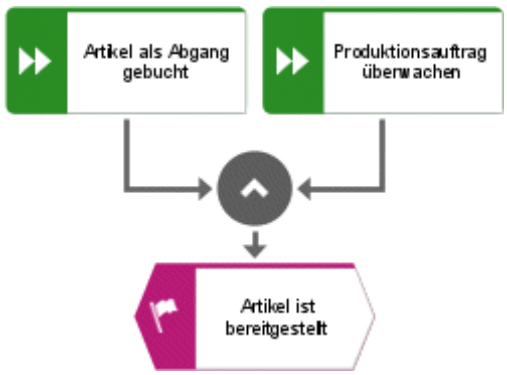


Abbildung 78: Und-Verknüpfung von Funktionen mit erzeugten Ereignissen

Das Ereignis ist erst eingetreten, wenn alle Funktionen ausgeführt sind.

ODER-REGEL

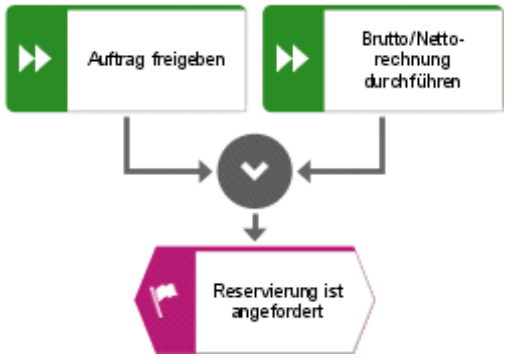


Abbildung 79: Oder-Verknüpfung von Funktionen mit erzeugten Ereignissen

Das Ereignis tritt ein, wenn mindestens eine der Funktionen durchgeführt ist.

EXKLUSIV-ODER-REGEL (ENTWEDER-ODER-REGEL)

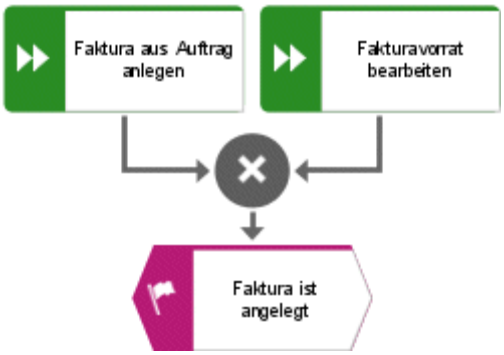


Abbildung 80: Exklusiv-oder-Verknüpfung von Funktionen mit erzeugten Ereignissen

Das Ereignis tritt ein, wenn genau eine und höchstens eine der Funktionen ausgeführt wurde.

DIE VERKNÜPFUNG VON FUNKTIONEN MIT AUSLÖSENDEN EREIGNISSEN

UND-REGEL

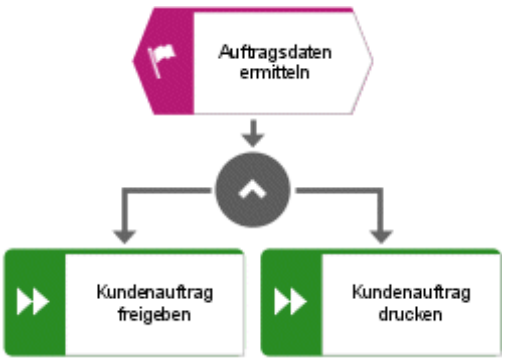


Abbildung 81: Und-Verknüpfung von Funktionen mit auslösenden Ereignissen

Alle Funktionen werden durch das Ereignis ausgelöst.

ODER-REGEL

Ereignisse haben keine Entscheidungskompetenz! Diese Verknüpfung ist nicht möglich!

EXKLUSIV-ODER-REGEL (ENTWEDER-ODER-REGEL)

Ereignisse haben keine Entscheidungskompetenz! Diese Verknüpfung ist nicht möglich!

Neben der Darstellung in Form von ereignisgesteuerten Prozessketten können diese Verzweigungen auch in tabellarischer Form in den Ereignis- und Funktionsspalten eines Vorgangskettendiagramms gezeigt werden. Die sequentielle Anordnung der Funktionen in einem Vorgangskettendiagramm führt jedoch dazu, dass Verzweigungen und Bearbeitungsschleifen nur sehr unübersichtlich darstellbar sind.

3.4.1.3 Funktionszuordnungsdiagramm (I/O)

Neben der in Kapitel **Ereignissteuerung- ereignisgesteuerte Prozesskette (EPK)** (Seite 66) erläuterten Darstellung der Ereignissteuerung stellen auch die Transformation von Eingabedaten zu Ausgabedaten und die Darstellung der Datenflüsse zwischen Funktionen eine Verbindung zwischen der Datensicht und Funktionssicht des ARIS-Konzeptes dar. Die Transformation von Input-Daten zu Output-Daten kann in sogenannten Funktionszuordnungsdiagrammen (I/O) dargestellt werden. Ein Beispiel für ein Funktionszuordnungsdiagramm (I/O) zeigt die folgende Abbildung. Die Funktion **Liefertermin bestimmen** hat als Input-Daten **Teiledaten**, **Bestandsdaten**, **Stücklistendaten** und **Versanddaten**. **Anfragedaten** sind sowohl Input-Daten als auch Output-Daten. Bestandteile eines Funktionszuordnungsdiagramms (I/O) sind somit Funktionen der Funktionssicht und Informationsobjekte der Datensicht. Die Pfeile definieren, ob ein Informationsobjekt nur Inputdatum, nur Output-Datum oder Input-/Output-Datum ist. Darüber hinaus sind genauere Spezifizierungen möglich, die z. B. aussagen, dass ein Informationsobjekt durch die Funktion erstellt oder gelöscht wird. Je nach dem Detaillierungsgrad kann es sich bei den Informationsobjekten um Cluster/Datenmodell, Entity-/Beziehungstypen oder auch Attribute der Datensicht handeln.

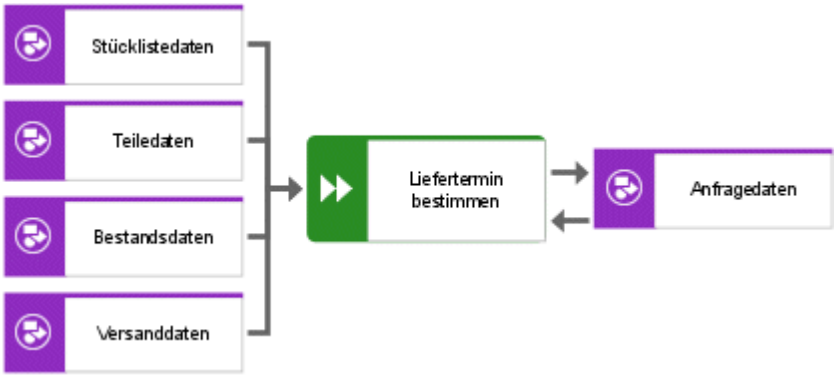


Abbildung 82: Beispiel für ein Funktionszuordnungsdiagramm (I/O)

Das hier gezeigte Beispiel verdeutlicht die eigentliche Zielsetzung der Funktionszuordnungsdiagramme (I/O), nämlich die Darstellung der Input-/Output-Daten einer Funktion.

Neben der Aufnahme der Input-/Output-Daten einer Funktion stehen außer den Ereignissen auch alle anderen Objekte zur Verfügung, die man in einer EPK den Funktionen zuordnen kann. Damit ist dem Anwender die Möglichkeit gegeben, bei der Modellierung von Prozessketten in EPK-Diagrammen nur die Ereignisse und Funktionen aufzunehmen und hinter jede Funktion ein Funktionszuordnungsdiagramm (I/O) mit den weiteren Beziehungen der Funktion zu legen. Dies ermöglicht eine wesentlich übersichtlichere Darstellung der Geschäftsprozesse und erklärt auch die Verwendung einer neuen Bezeichnung für diesen

Modelltyp. Ein Beispiel für eine solche erweiterte Darstellung in einem Funktionszuordnungsdiagramm zeigt diese Abbildung.

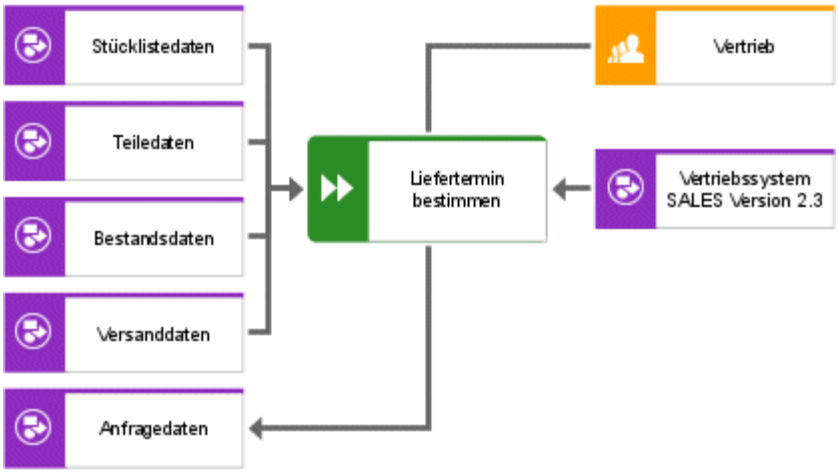


Abbildung 83: Erweiterte Darstellung des Funktionszuordnungsdiagramms

Neben der getrennten Darstellung der Datentransformation in Form von Funktionszuordnungsdiagrammen (I/O) können diese Informationen auch in einer EPK aufgenommen werden. Ein Beispiel zeigt die folgende Abbildung. Die Verbindungen zwischen den Funktionen und Informationsobjekten haben hierbei die gleiche Bedeutung wie die des Funktionszuordnungsdiagramms (I/O). Ihre Aufnahme in eine stark verzweigte Prozesskette führt jedoch unter Umständen zu unübersichtlichen Darstellungen.

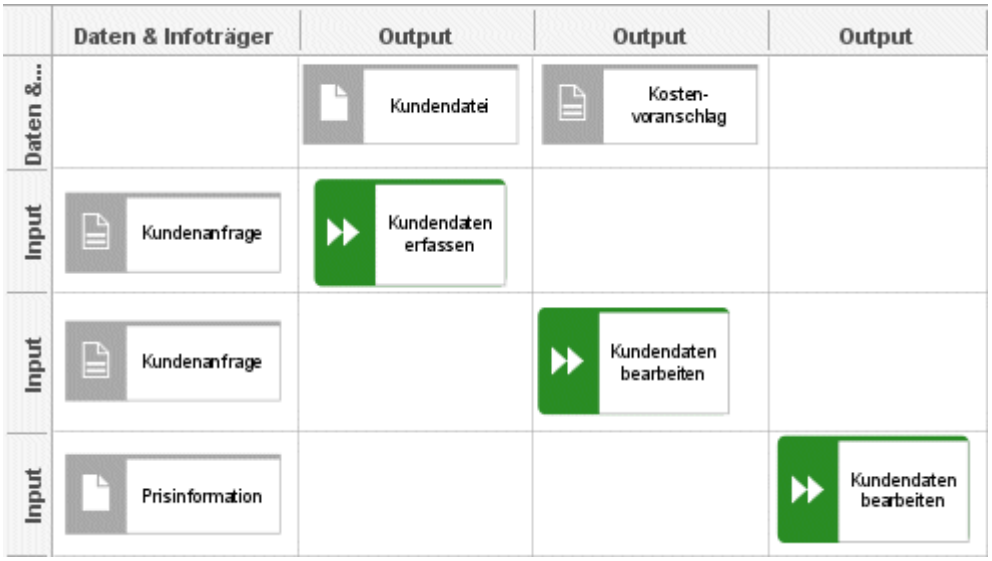


Abbildung 84: EPK mit Input-/Output-Daten

3.4.1.4 Ereignisdiagramm

Ereignisse definieren das Eingetretensein von Zustandsänderungen von Informationsobjekten. Jedes Ereignis referenziert somit auf bestimmte Informationsobjekte des Datenmodells und definiert den Status des jeweiligen Informationsobjektes zu einem bestimmten Zeitpunkt.

Ereignisse werden in einer Top-down-Vorgehensweise zunächst grob spezifiziert (Bsp. **Auftrag ist bearbeitet**). Auf der nächsten Detaillierungsstufe der Prozessmodellierung werden nun die detaillierteren Ereignisse definiert, die in bestimmter Kombination dazu führen, dass das Ereignis auf grober Ebene eingetreten ist. So kann z. B. das Eintreten der Ereignisse **Machbarkeit ist geprüft**, **Auftragskopf ist erfasst** und **Auftragspositionen sind erfasst** in der Summe den Zustand **Auftrag ist bearbeitet** definieren.

Diese Ereigniszusammenhänge auf grober und feiner Modellierungsebene können mit Hilfe des Ereignisdiagramms dargestellt werden. Hierzu kann hinter ein Ereignis auf grober Ebene ein Ereignisdiagramm gelegt werden (Hierarchisierung!!), in dem die Ereignisse auf feiner Ebene und ihre Verknüpfungen (auch über Regel-Verknüpfungen) untereinander dargestellt werden. Zusätzlich können in diesem Modelltyp auch Informationsobjekte des Datenmodells aufgenommen werden und mit den Ereignissen verknüpft werden. Damit wird definiert, welches Ereignis die Zustandsänderung welches Informationsobjektes definiert.

Folgende Abbildung zeigt ein Beispiel.

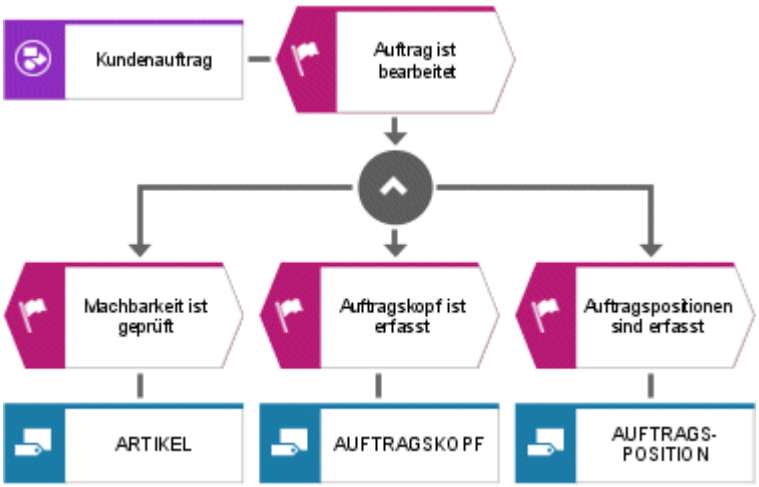


Abbildung 85: Beispiel für ein Ereignisdiagramm

3.4.1.5 Wertschöpfungskettendiagramm

Das Wertschöpfungskettendiagramm dient in erster Linie dazu, die Funktionen im Unternehmen zu spezifizieren, die direkt an der Wertschöpfung des Unternehmens beteiligt sind. Diese Funktionen können in Form einer Funktionsfolge miteinander verbunden werden und bilden damit eine Wertschöpfungskette. Ein Beispiel für eine Wertschöpfungskette zeigt diese Abbildung.

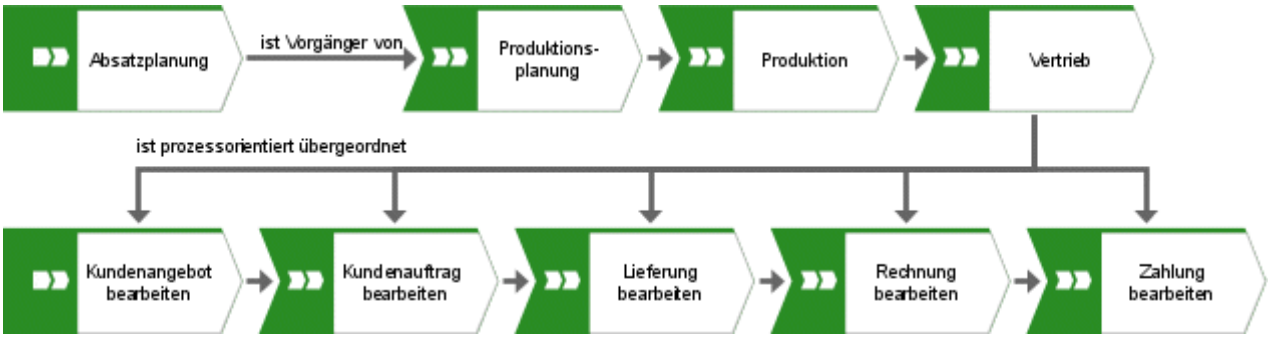


Abbildung 86: Wertschöpfungskette

Funktionen können im Wertschöpfungskettendiagramm ähnlich einem Funktionsbaum hierarchisiert werden. Hierbei wird immer die prozessorientierte Über-/Unterordnung dargestellt.

Neben der Über-/Unterordnung von Funktionen können in einem Wertschöpfungskettendiagramm auch die Verbindungen der Funktionen zu Organisationseinheiten und zu Informationsobjekten dargestellt werden. Bei der Zuordnung der Organisationseinheiten zu Funktionen werden wie auch schon in den Prozessketten die fachliche Verantwortung, die DV-Verantwortung und die eigentliche Ausführung einer Funktion unterschieden.

Weitere Beziehungen, die im Wertschöpfungskettendiagramm zur Verfügung stehen, finden Sie in der **ARIS-Methode Reference**-Hilfe.

3.4.1.6 Objektorientierte Modellierung

Für die Modellierung im UML-Umfeld steht UML zur Verfügung. Alle methodischen Informationen zu UML-Diagrammen und UML-Elementen werden direkt in der Oberfläche von ARIS UML Designer angeboten.

3.4.1.7 Prozessauswahlmatrix

In der Prozessauswahlmatrix werden verschiedene Prozessszenarien dargestellt, indem Hauptprozesse einzelnen Szenarien zugeordnet werden.

Der Anwender kann danach festlegen, welche Funktionen der Szenarioprozesse im Unternehmen auftreten. Hierzu sollten immer alle Hauptfunktionen (Szenariofunktionen) eines Anwendungssystems oder eines Branchenreferenzmodells als Prozesse enthalten sein.

Für die Modellierung einer Prozessauswahlmatrix stehen folgende Symboltypen zur Verfügung:

- Szenario
- Prozess
- Hauptprozess

Ein Szenario repräsentiert einen Szenarioprozess in der Auswahlmatrix der verschiedene Hauptprozesse gruppiert.

Der Prozess repräsentiert Funktionen des Szenarioprozesses, die im Referenzmodell durch Prozessmodelle näher beschrieben werden.

Der Hauptprozess repräsentiert die Hauptfunktionen in den Funktionsbäumen, denen die Prozesse (Funktionen aus den Szenarioprozessen) zugeordnet sind.

Ein Beispiel für eine Prozessauswahlmatrix zeigt die folgende Abbildung.

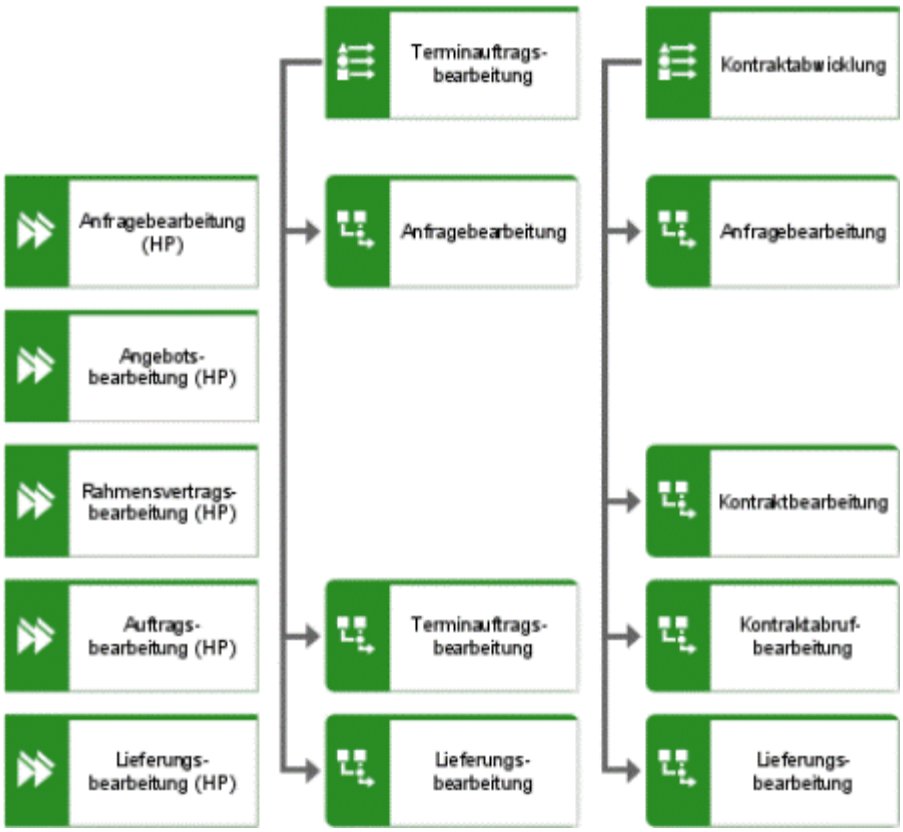


Abbildung 87: Prozessauswahlmatrix (Auszug aus dem Referenzmodell R/3 der SAP AG)

3.4.1.8 Modellierung des Materialflusses

Neben der Darstellung des Informationsflusses eignen sich EPKs auch zur Darstellung der Materialtransformation. Zur Darstellung des Materialflusses innerhalb von Geschäftsprozessen wird in ARIS ein eigener Modelltyp angeboten - EPK (Materialfluss) -, der eine Erweiterung des Modelltyps **EPK** darstellt.

3.4.1.9 EPK (Materialfluss)

Neben den Objekttypen der EPK stehen zusätzlich in der EPK (Materialfluss) folgende Objekttypen zur Verfügung:

- Materialtyp
- Packmitteltyp
- Betriebsmitteltyp
- Betriebsmittel
- techn. Hilfsmitteltyp
- techn. Hilfsmittel
- Lagereinrichtungstyp
- Lagereinrichtung
- Transportsystemtyp
- Transportsystem

Der Objekttyp **Materialtyp** kann mit einer eingehenden oder einer ausgehenden Kante mit dem Objekttyp **Funktion** verknüpft werden. Im Falle einer eingehenden Kante wird definiert, welche Materialien von einer Funktion als Input benötigt werden. Hierbei kann durch Auswahl des entsprechenden Kantentyps definiert werden, ob das Material durch die Funktion nicht, teilweise oder ganz verbraucht wird. Eine ausgehende Kante spezifiziert die Materialtypen, die durch die Funktion erzeugt werden.

Zur Materialtransformation werden techn. Ressourcen benötigt. Diese können ebenfalls in den Prozessketten mit dem Objekttyp **Funktion** verbunden werden. Hierzu wird neben dem Kantentyp **benötigt** auch ein Kantentyp **benötigt alternativ** angeboten, um eventuell zur Verfügung stehende Alternativressourcen spezifizieren zu können.

Werden Materialien innerhalb einer Funktion verpackt, werden bestimmte Packmitteltypen benötigt. Zur Spezifikation der entsprechenden Packmitteltypen kann eine Beziehung zwischen der Funktion und den benötigten Packmitteltypen modelliert werden.

In folgender Abbildung ist eine EPK (Materialfluss) und den dazugehörigen techn. Ressourcentypen und Packmitteltypen dargestellt.

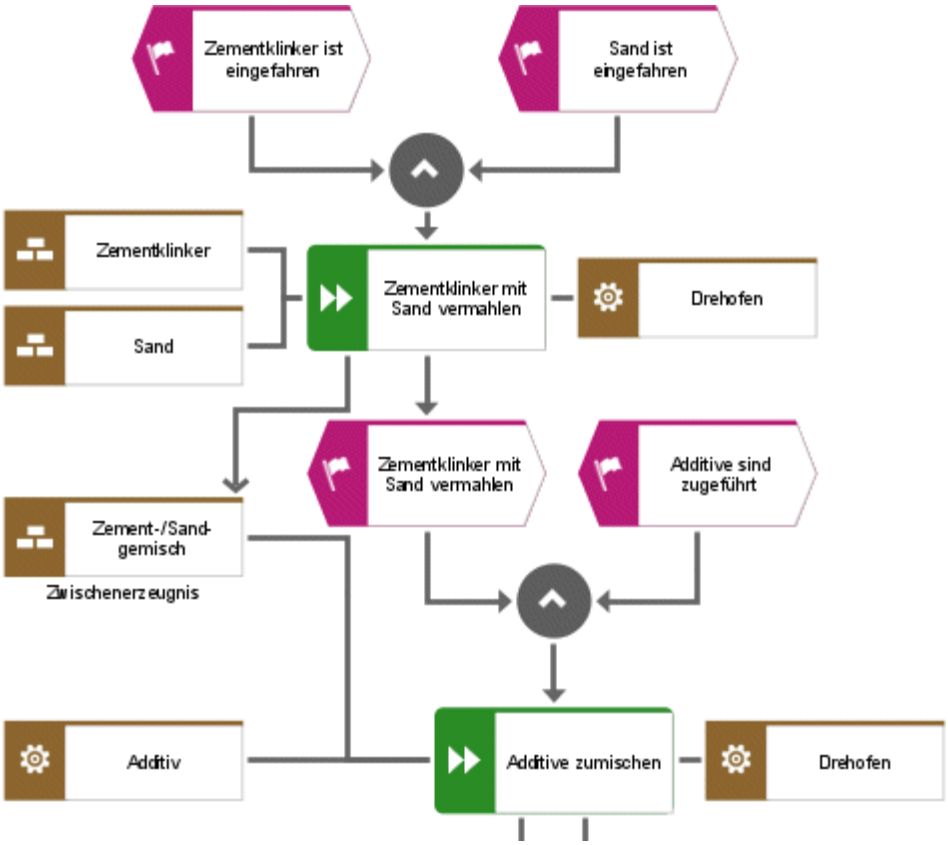


Abbildung 88: Auszug einer EPK (Materialfluss)

3.4.1.10 EPK (Spalten-/Zeilendarstellung)

Die folgende Beschreibung gilt analog für die EPK (Zeilendarstellung).

Für den Modelltyp **EPK (Spaltendarstellung)** gelten im Wesentlichen die Ausführungen zur EPK mit dem Unterschied, dass in diesem Modell alle Symbole in verschiedenen Spalten verteilt werden. Das hat den Vorteil, dass die Übersichtlichkeit der EPK durch diese Darstellung deutlich verbessert wird. Organisatorische- und Anwendungssystem-Elemente werden in der Kopfzeile des Diagramms platziert. Alle anderen Symbole werden in der zweiten Zeile der Spalten positioniert.

Besonderes Merkmal sämtlicher Lane-Modelle, also von Modellen, die in Spalten und/oder Zeilen modelliert werden, ist die automatische Erstellung von nicht sichtbaren (impliziten) Beziehungen. Beispielsweise wird in den standardmäßig vorgegebenen Spalten der EPK (Spaltendarstellung) beim Modellieren von Anwendungssystemen und Funktionen automatisch die implizite Beziehung unterstützt angelegt. Zwischen organisatorischen Elementen und Funktionen werden führt aus-Beziehungen implizit angelegt. Darüber hinaus hat der Anwender die Möglichkeit, folgende weiteren Spalten hinzuzufügen, die nach den impliziten Beziehungen benannt wurden:

- Wirkt mit bei
- Entscheidet über
- Ist dv-verantwortlich für
- Ist fachlich verantwortlich für
- Muss bei Abbruch informiert werden
- Muss informieren über Ergebnis von
- Muss informiert werden über
- Stimmt zu
- Wirkt beratend mit

Die folgende Abbildung zeigt exemplarische eine EPK (Spaltendarstellung).

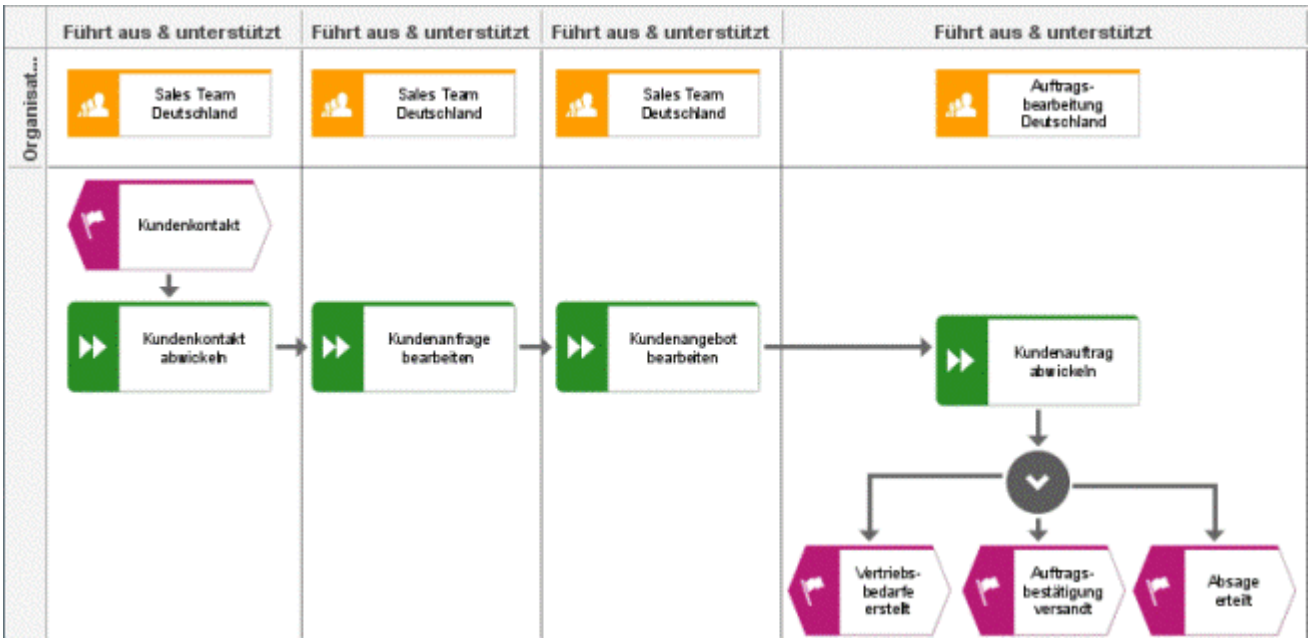


Abbildung 89: EPK (Spaltendarstellung)

Ein Unterschied zwischen der EPK (Spaltendarstellung) und der EPK (Zeilendarstellung) ist lediglich durch die sich unterscheidende Richtung der Modellierung gegeben. In der EPK (Spaltendarstellung) wird von oben nach unten, bei der EPK (Zeilendarstellung) von links nach rechts modelliert.

3.4.1.11 Business Controls Diagram

In einem Business Controls Diagram werden zu einem Prozess bzw. einer Funktion mögliche Risiken und Risikobewältigungen abgebildet.

Ein Risiko (Risk) stellt eine mögliche Gefahr für einen Prozess dar, das angestrebte Prozessziel nicht zu erreichen.

Eine Risikobewältigung bzw. -bekämpfung (Control) stellt einen allgemeinen Weg dar, Risiken zu eliminieren oder zu vermindern.

Eine Risikolösung (Solution) stellt die Umsetzung einer Risikobekämpfung (Control) bezogen auf ein Risiko (Risk) dar.

Das Layout des Business Controls Diagram entspricht einer Matrix bzw. Tabelle. Auf der Abszisse werden die möglichen Prozessrisiken abgetragen und auf der Ordinate mögliche Risikobewältigungen. Risikolösungen werden nun als Verknüpfungen eines Risikos mit einer Risikobekämpfung eingefügt. Zusätzlich können dem Modell noch Organisationseinheiten (im Sinne von Benutzeranforderungen) und Dokumente hinzugefügt werden, die ebenfalls eine Umsetzung einer Risikobewältigung hinsichtlich eines Risikos unterstützen.

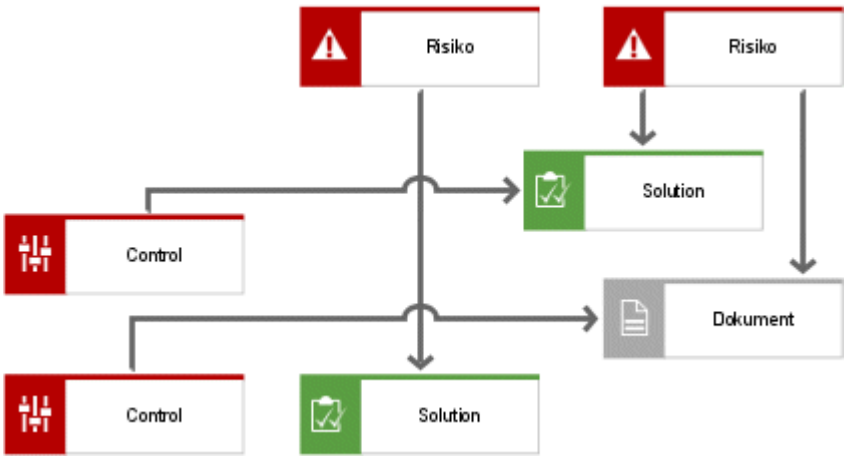


Abbildung 90: Beispiel eines Business Controls Diagram

Angewendet wird der Modelltyp üblicherweise zur Beschreibung von SAP®-Standardprozessen. Hierbei zeigt das Modell, welche Risiken und Risikobewältigungen eine SAP®-Lösung bezüglich des betrachteten Prozesses aufweist.

3.4.1.12 E-Business scenario diagram

Die reibungslose Abfolge von unternehmensübergreifenden Geschäftsprozessen gewinnen ständig an Bedeutung. Dabei steht auf der einen Seite der Ablauf bestimmter Vorgänge an den Schnittstellen zwischen den Unternehmen, auf der anderen Seite die Schnittstelle zwischen Unternehmen und ihren Kunden im Blickpunkt. Die Kontakte sollen übersichtlich, schnell, einheitlich und direkt erfolgen.

Auch das schnelle Finden von adäquaten Geschäftspartnern aus Unternehmenssicht und Anbietern aus Verbrauchersicht wird ständig wichtiger. Eine optimale Gestaltung dieser Prozesse bedeutet einen Wettbewerbsvorteil. Die geeignete Plattform zur Unterstützung dieser bilateralen Beziehungen ist das Internet. Da die Vorgänge innerhalb des oben beschriebenen Umfeldes sehr vielschichtig sind, muss definiert werden, was unter **E-Business** verstanden wird.

Der Begriff E-Business bezeichnet alle computergestützten Prozesse zwischen zwei Wirtschaftssubjekten und den Versuch durch neue Medien einen Mehrwert zu erzielen.

Dies kann somit der simple Kauf eines Artikels via Internet sein, ein hoch komplexes Projekt zweier Unternehmen oder die Erstellung einer Web-Site zur Unternehmenspräsentation.

Beziehungen Unternehmen - Unternehmen werden als Business-To-Business (kurz: B2B) bezeichnet, Beziehungen Kunde - Unternehmen als Business-To-Customer (kurz: B2C).

Zur Unterstützung des E-Business wurde das E-Business scenario diagram entwickelt.

Durch die Möglichkeit, eine Wertschöpfungskette ganzheitlich zu betrachten, d.h. vom Endkunden bis zu allen an einem Vorgang beteiligten Unternehmen, soll Optimierungspotenzial entwickelt werden. Ziel hierbei ist beispielsweise die Verbesserung der Supply Chain, Senkung von Beschaffungs- und Vertriebskosten oder Optimierung der Architektur des Informationssystems. Die durch die Ziele dargestellten Inhalte werden durch diese Methode modellierbar.

Die Wirtschaftsobjekte werden im Diagramm in der oberen Zeile eingeordnet und als **Business participants** bezeichnet. Die beteiligten Unternehmen können durch ein Organigramm hinterlegt werden. Zentrale Objekte sind hierbei die am Gesamtprozess beteiligten Einzelprozesse der Wirtschaftsobjekte und ihre Schnittstellen zueinander. Ein Einzelprozess, auch Business Process, ist ein wesentlich an unternehmensübergreifenden Kooperationen beteiligter Unternehmensprozess, der dem Prozessmodell hinterlegt werden kann. Der Business Process wird von Anwendungssystemen (Business Components) unterstützt, z. B. durch das R/3-System.

Auch die Rollen der am Prozess beteiligten Mitarbeiter können definiert werden. Sie werden im Modell als **Employee role** bezeichnet.

Die Schnittstellen sind durch die Übertragung prozessspezifischer Informationen charakterisiert. Sie sind in Business Documents zusammengefasst und können die Form eines

XML- oder eines HTML-Dokuments annehmen. Auch das Business Document kann als Datenmodell hinterlegt werden. Alternativ zu diesem Objekt können auch die Objekte **Money Transaction** (zur Darstellung des Geldstroms), **Goods Shipment** (zur Darstellung des Güterstroms), **E-Mail**, **Internet**, **Intranet**, **Extranet** und **Handy** (zur Spezifizierung des technologischen Aspektes der Datenübertragung) verwendet werden.

Alle ein Unternehmen betreffenden Abläufe werden im Modell in der Zeile unter dem Business Participant, aber in derselben Spalte modelliert.

Daraus folgt, dass die Spaltenränder die abstrakten Schnittstellen bilden. Diesen gilt besonderes Augenmerk, da sie das Hauptoptimierungspotenzial beinhalten und deshalb für Modellierungszwecke besonders interessant sind.

Begriffserläuterung: Im folgenden Beispielmodell steht OEM für **Original Equipment Manufacturer** und MRP für **Material Ressource Planning Controller**.

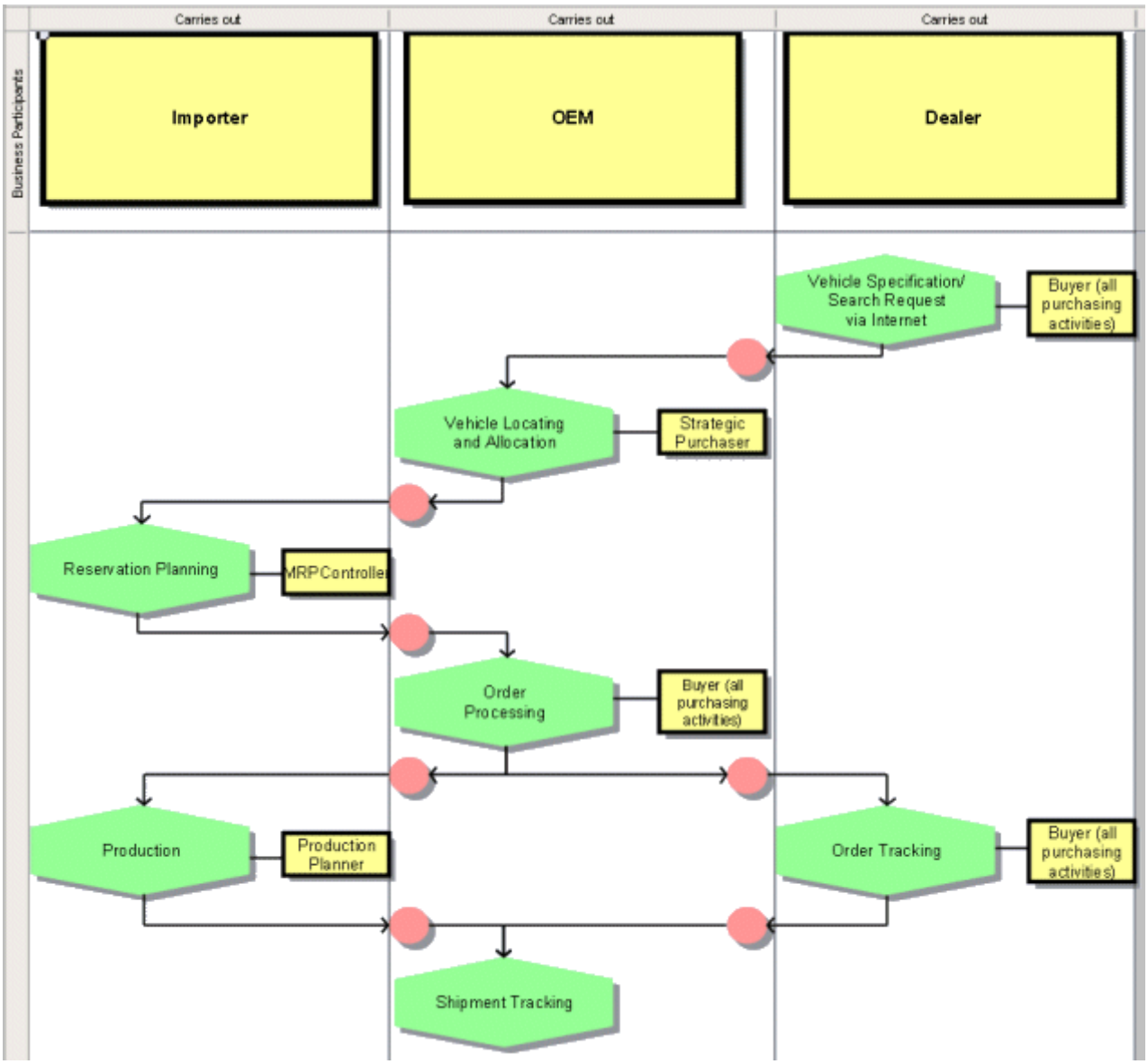


Abbildung 91: eBusiness Szenario Diagram-Beispiel: Automobilbranche

Im Beispielmodell wird das Zusammenspiel eines Herstellers, eines Importeurs und eines Händlers dargestellt. Jeder hat im Gesamtgebilde seine spezifischen Prozesse, die an den Schnittstellen zu den Prozessen anderer Geschäftspartner via Business Dokuments Informationen austauschen. Auch die an den Geschäftsprozessen beteiligten Personen werden mit ihren Rollen erfasst und zugeordnet.

3.4.1.13 Gliederungsmodell

Das Gliederungsmodell dient allgemein der Hierarchisierung bzw. Systematisierung eines Sachverhaltes (Spezialisierung bzw. Generalisierung von Sachverhalten).

Ein Gliederungselement stellt einen Sachverhalt (in Richtung der beabsichtigten Systematisierung) dar.

Hinter den einzelnen Gliederungselementen der Sachverhaltshierarchie können Modelle hinterlegt werden, die dem Sachverhalt zugeordnet werden.

Gebräuchlichstes Anwendungsgebiet des Gliederungsmodells ist das Qualitätsmanagement, speziell die Zertifizierung. Dort wird durch das Gliederungsmodell eine Norm in ihre einzelnen Komponenten untergliedert, und bei den einzelnen Gliederungselementen werden die Modelle hinterlegt, die zur Erfüllung des Qualitätskriteriums beitragen.

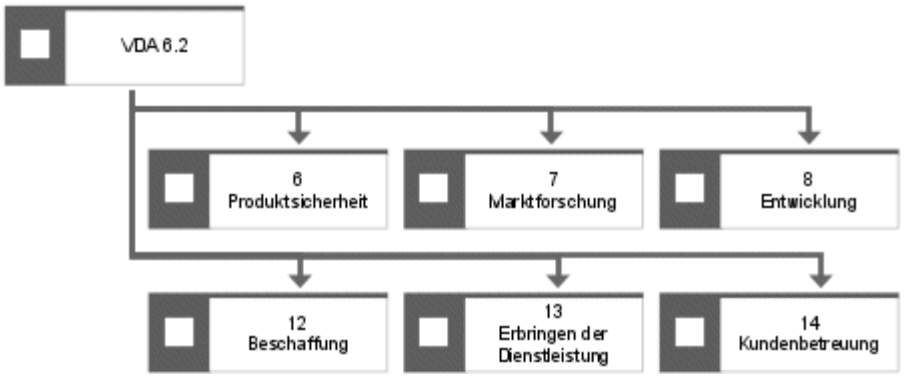


Abbildung 92: Beispiel eines Gliederungsmodells (Auszug aus der Norm VDA 6.2)

Durch einen Report können diese Sachverhalte leicht ausgewertet werden bzw. für Dokumentationszwecke benutzt werden.

3.4.1.14 Rollendiagramm

Das Rollendiagramm dient generell der näheren Beschreibung von Prozessen. Im Vordergrund stehen die Organisationseinheiten, die an den Prozessen beteiligt sind, und deren Rollen. Charakterisiert werden die Objekte und ihre Beziehungen durch folgende Eigenschaften:

Eine Rolle ist unter Berücksichtigung von Berechtigungen an Prozessen beteiligt. Nicht nur die Durchführungsmöglichkeit ist bedeutend, sondern auch die Spezifizierung der Berechtigungsart im Prozess (eine Rolle ist „durchführend beteiligt“). Bei der Durchführung eines Prozesses mit einer bestimmten Berechtigung wird also die Beziehungskette Rolle - Beteiligung - Prozess (mit Beteiligung - Berechtigungsbedingung sowie Beteiligung - Berechtigungswert) aufgebaut.

Eine Rolle kann von Personen, Stellen oder Informationssystemen besetzt werden. Die Rolle bildet das Bindeglied zwischen den Prozessen und den an ihnen beteiligten Ressourcen. Sie wird durch eine Aggregation von Erwartungen an die an den Prozessen beteiligten Ressourcen definiert.

Die Durchführung eines Prozesses erfordert Fertigkeiten (Skills), die die beteiligte Rolle bzw. die zugewiesene Ressource aufweisen muss Um Rollen überhaupt prozessorientiert definieren zu können, müssen die Prozesse begutachtet und die Anforderungen der Prozesse an die beteiligten Personen/Systemen zu bestimmt werden. Anforderungen an Personen/Systeme heißt genauer: Unter Anforderungen wird hier das Wissen und die Fähigkeiten (Skills) der Personen/Systeme verstanden. Die Bewertung einer Fähigkeit wird durch eine zugeordnete Bewertungsskala standardisiert.

Im Rollendiagramm besteht damit die Möglichkeit, Prozesse und bestimmte Elementarprozesse abzubilden, die daran beteiligten Ressourcen darzustellen, ihre Fähigkeiten bzw. benötigte Fähigkeiten festzuhalten und ihre Berechtigungen wiederzugeben.

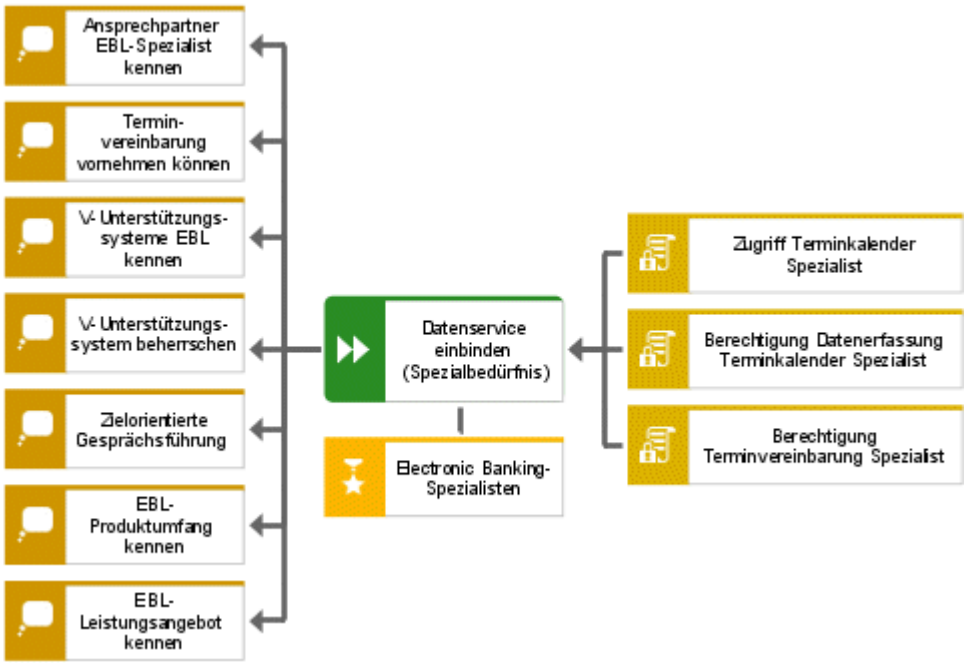


Abbildung 93: Rollendiagramm

Im Beispielmodell werden die Anforderungen des Elementarprozesses an die Rolle (Fähigkeiten und Berechtigungen) und ebenso die Anforderungen der Elementarrolle an die Ressource hinsichtlich der Fähigkeiten und Berechtigungen gezeigt.

Das Diagramm wird dem jeweiligen Elementarprozess und der jeweiligen Elementarrolle hinterlegt. Durch die Hinterlegung des Diagramms am Elementarprozess können die Anforderungen aus der zugrunde liegenden Prozess-EPK (entspricht dem Prozessreferenzmodell) eingesehen werden. Durch die Hinterlegung an der Elementarrolle kann die Anforderung der Elementarrolle an die Ressource hinsichtlich ihrer Fähigkeiten und Berechtigungen aus dem Rollenstrukturdiagramm eingesehen werden.

3.4.1.15 Casual model

Das Modell des Typs **Casual Model** bietet Ihnen die Möglichkeit ohne methodische Einschränkungen zu modellieren. Innerhalb des Casual Model ist der Objekttyp **Casual Object** verfügbar, für den über 30 verschieden Symbole angeboten werden. Zwischen den Casual Objects können Beziehungen des Typs **steht in Beziehung mit** angelegt werden. Es sind mehrere Kanten dieses Typs zwischen zwei Objekten erlaubt.

Für Modell, Objekt und Kante können die jeweiligen Standardattribute gepflegt werden.

Jedem Objekt eines beliebigen Objekttyps der ARIS-Methode. können mehrere Casual Models hinterlegt werden. Einem Casual Object können wiederum beliebige viele Modelle der ARIS-Methode-Methode hinterlegt werden, unabhängig vom Modelltyp.

3.4.1.16 Maskendesign

In einem Maskendesign können Sie im Rahmen des Software-Entwurfs mit ARIS die fachlichen Anforderungen spezifizieren, die an einen Dialog oder ein Webformular gestellt werden.

In der Spalte **Layout** bestimmen Sie die Struktur des Dialogs oder Webseiten-Formulars. Das Vorgehen beim Entwurf z. B. eines Dialogs ähnelt dem Arbeiten mit einem Ressource-Editor in einer Entwicklungsumgebung.

Zu den grafischen Bestandteilen, die in der Spalte **Layout** platziert werden können, gehören Textfelder, Drehfelder, Optionsfelder und Kontrollkästchen, Kombinationsfelder, Schaltflächen, Tree- und Listen-Steuerelemente sowie Bitmaps und statischer Text. Mit Hilfe des Attributtyps **Tabindex** können Sie die Reihenfolge festlegen, in der die einzelnen Maskenelemente mit der Tabulatortaste angesprochen werden können.

In den Spalten **Daten** und **Funktionen** können verschiedene Datenelemente bzw. Funktionsobjekte platziert werden. Mit einer Kante des Typs **stellt dar** können die Objekte mit den Datenelementen und Funktionen in Beziehung gesetzt werden, die sie editieren.

Jedes Maskendesign kann dem entsprechenden Maskenobjekt hinterlegt werden, das z. B. in einer EPK oder einem Modell des Typs **Maskennavigation** verwendet wird. Darüber hinaus kann ein Maskendesign aber auch dem Entitytyp, dem Cluster, dem komplexen Objekttyp, der Klasse oder der Funktion/DV-Funktion hinterlegt werden, welche mit der Maske bearbeitet werden.

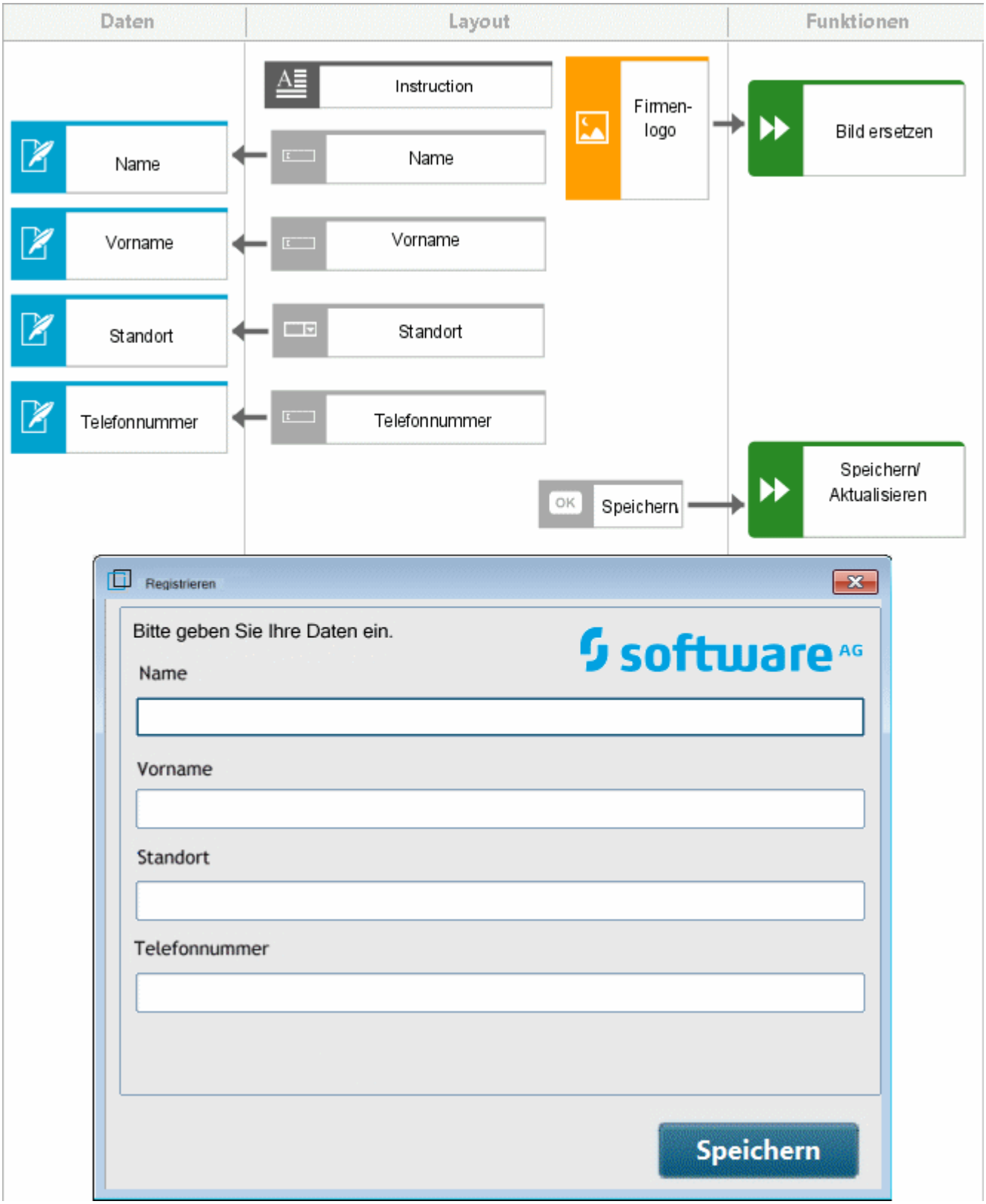


Abbildung 94: Beispiel für das Maskendesign eines Registrierungsdialogs und die Umsetzung in C++

3.4.1.17 Maskennavigation

In einem Modell des Typs **Maskennavigation** können Sie entweder den Aufbau einer Maske beschreiben, die aus mehreren Teilmasken besteht (z. B. eine Web-Site mit mehreren Formularfeldern oder Frames) oder die Übergänge zwischen verschiedenen Masken. Die Übergänge zwischen den Masken können detailliert werden.

Beispiel

Sie möchten verdeutlichen, dass ein Maskenelement betätigt werden muss, um in eine andere Maske zu wechseln. Ordnen Sie das auslösende Maskenelement (des Modells **Maskendesign**) mit Hilfe der Kante **beinhaltet** der Maske zu. Ziehen Sie danach eine Kante des Typs **ruft auf** vom Maskenelement zur nachfolgenden Maske.

Ebenso ist es möglich, die Abhängigkeit der Navigation von Ereignissen darzustellen. Beim Verlassen einer Maske können unterschiedliche Ereignisse eintreten. Wenn z. B. ein Benutzer die Registrierungsseite eines Online-Shops ausgefüllt hat, kann das Registrieren erfolgreich durchgeführt werden oder fehlschlagen. Abhängig davon wechselt er entweder auf die Inhaltsseite des Katalogs oder er kehrt zur Registrierungsseite zurück.

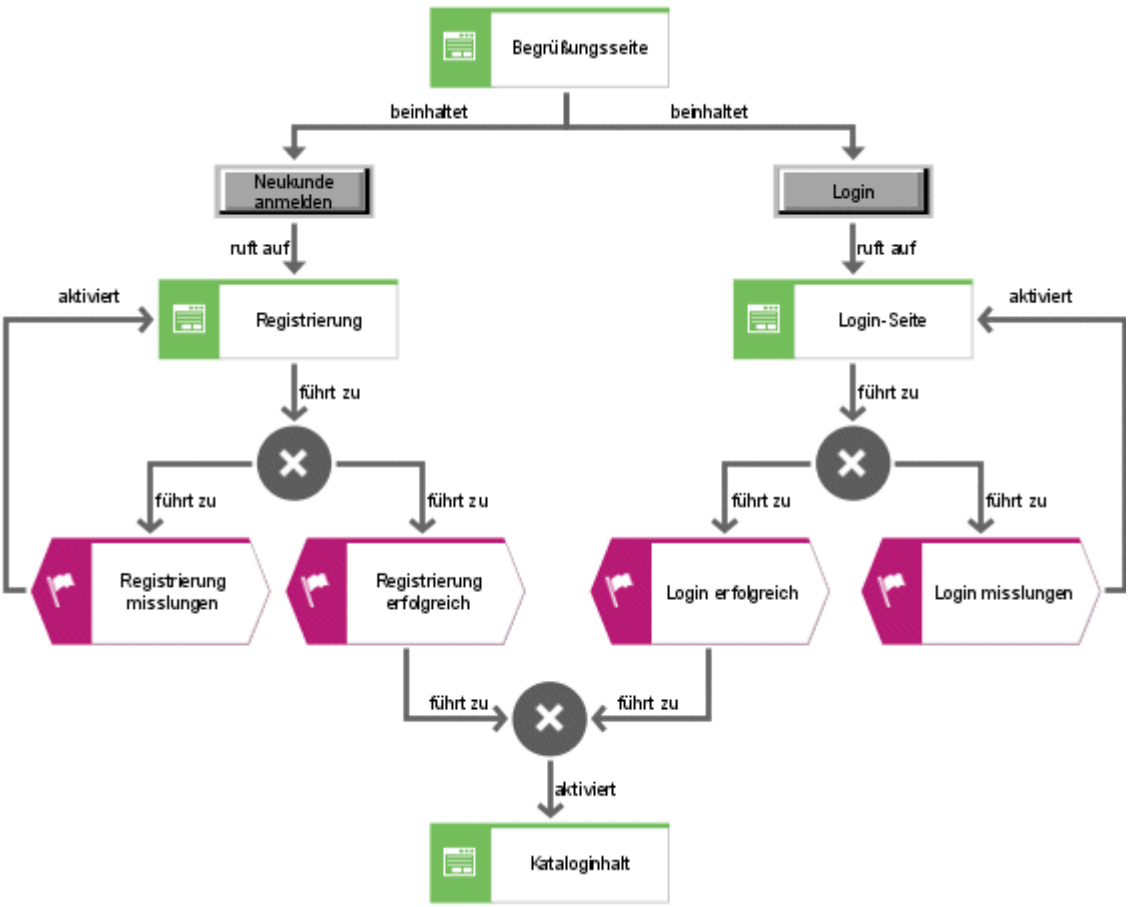


Abbildung 95: Beispiel für die Maskennavigation mit Ereignissen

3.4.1.18 Geschäftsfelder-Matrix

In der Geschäftsfelder-Matrix werden die verschiedenen Märkte, in denen ein Unternehmen tätig ist, im Überblick dargestellt und ihre Bedeutung für den Unternehmenserfolg visualisiert.

Jeder Markt wird beschrieben

- durch das angebotene Produkt oder die angebotene Dienstleistung und
- die Kundengruppe, an die sich das Angebot richtet.

Produkte und Dienstleistungen (Objekte des Typs **Leistung**) werden in den Zellen der ersten Spalte der Geschäftsfeldermatrix platziert. Die Zielgruppe (verschiedene Organisationselemente) in den Zellen der ersten Zeile. Sie definieren den Markt, indem Sie ein Geschäftsfeld-Objekt in der Zelle platzieren, in der sich Produktzeile und Zielgruppenspalte treffen. Dabei werden zwischen dem Produkt/der Dienstleistung und dem Organisationselement implizite Beziehungen des Typs **gehört zu Geschäftsfeld** aufgebaut.

Um die Bedeutung eines Geschäftsfeldes gegenüber den anderen hervorzuheben, stehen fünf verschiedene Symbole zur Verfügung, mit denen die Wichtigkeit von **fast unbedeutend** bis **sehr bedeutend** zum Ausdruck gebracht werden kann.

Beim Modellieren müssen Sie beachten, dass jedes Geschäftsfeld nur einmal in der Matrix platziert werden darf.

Für jedes Geschäftsfeld kann angegeben werden, im Rahmen welcher Unternehmensstrategie es von Bedeutung ist. Eine Strategie beschreibt langfristige Verfahrensweisen, mit denen sich das Unternehmen gegenüber seinem Umfeld zur Verwirklichung seiner langfristigen Ziele zu behaupten versucht.

Die folgende Abbildung zeigt eine Geschäftsfelder-Matrix aus dem pharmazeutischen Bereich.

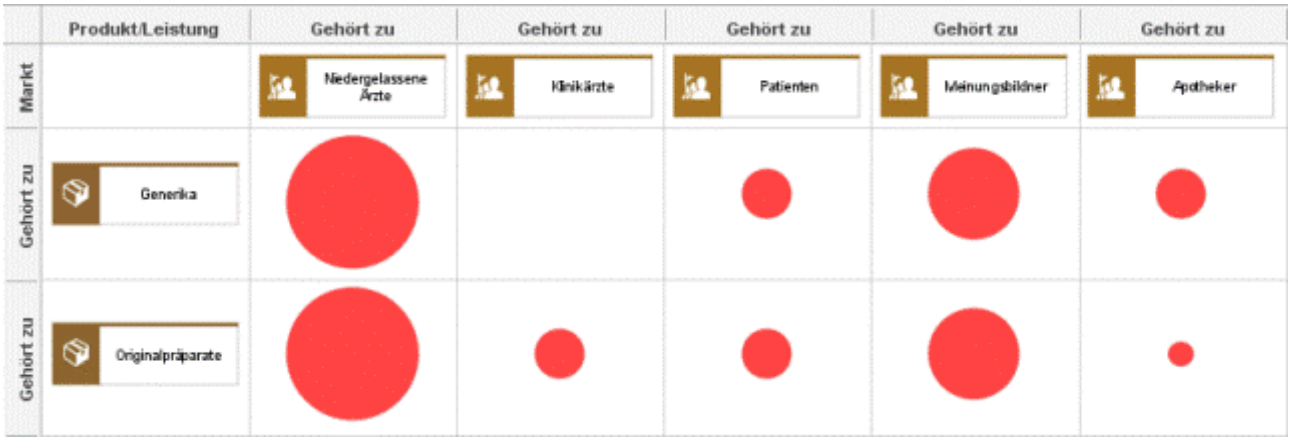


Abbildung 96: Beispiel Geschäftsfelder-Matrix

Geschäftsfelder können auch mit einem Zieldiagramm hinterlegt werden. Das Zieldiagramm enthält die für das Geschäftsfeld gesetzten Ziele sowie die Prozesse und Erfolgsfaktoren, die die Zielerreichung unterstützen.

Die Erfolgsfaktoren im Zieldiagramm können Basis für eine Erfolgsfaktorenanalyse sein, wenn in der gleichnamigen Attributtypgruppe die Attribute **Erfolg - Ist**, **Erfolg - Soll** und **Erfolg - Wettbewerber** gepflegt sind. Der Erfolg kann anhand einer 5-stufigen Skala von **sehr gering** bis **sehr groß** beurteilt werden.

Vorgehen

Um eine Erfolgsfaktorenanalyse durchzuführen,

1. starten Sie mit dem Kontextmenü des Geschäftsfeldes den ARIS-Report (**Auswerten > Report**)
2. und wählen im Report-Assistenten das Reportskript **HEF-Analyse (Objekt).rso** der Gruppe **BPM** im Standardpfad.

Die Reportausgabe erfolgt im Format HTML.

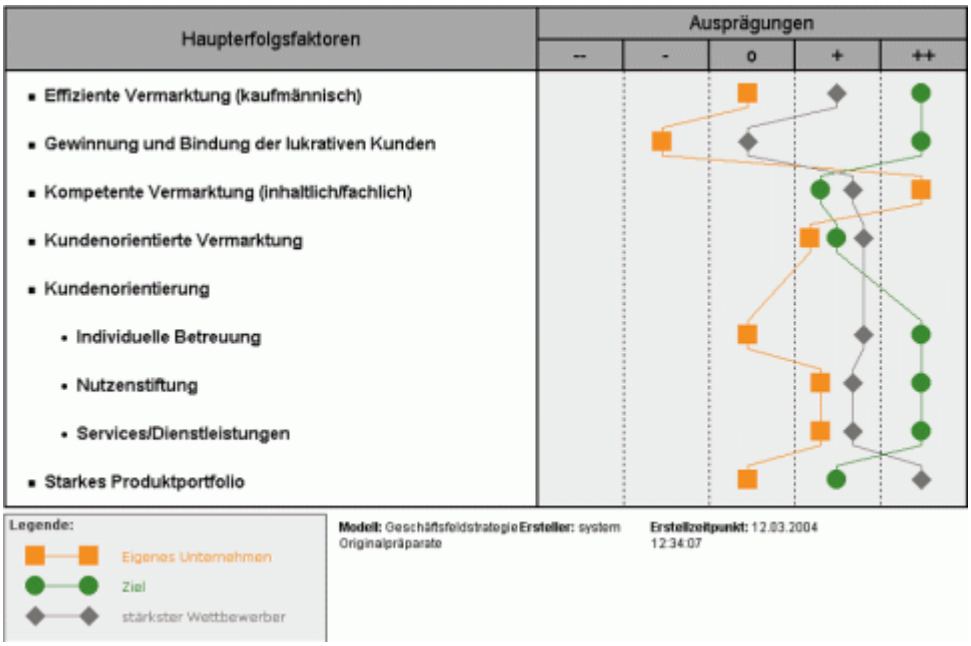


Abbildung 97: Report

Alternativ kann die Erfolgsfaktorenanalyse auch mit dem Kontextmenü des Zieldiagramms gestartet werden. Wählen Sie dann das Reportskript **HEF-Analyse (Modell).rsm**.

3.4.2 DV-Konzept

3.4.2.1 Zugriffsdiagramm

Die nachfolgend dargestellten Beziehungen zwischen den Objekten, die in den DV-Konzept-Beschreibungen der anderen Sichten erläutert wurden, können im Zugriffsdiagramm der Steuerungssicht aufgenommen werden. Zur besseren Übersichtlichkeit der Darstellung werden die einzelnen Zweierbeziehungen getrennt behandelt.

3.4.2.2 Funktionen mit Daten

Hier können zunächst die Informationsflüsse die zwischen Anwendungssystemtypen, Modultypen oder DV-Funktionen bestehen, definiert werden. Zu diesem Zweck wird zwischen den jeweiligen Anwendungssystemtypen oder Modultypen ein Informationsflussobjekt angelegt. Um den Informationsfluss zwischen den Systemtypen näher spezifizieren zu können, wird mit dem Informationsflussobjekt ein eERM-Diagramm oder ein Tabellendiagramm verbunden. Die Objekte des Informationsflusses können sich somit auf der Fachkonzept-Ebene, der DV-Konzept-Ebene oder der Implementierungsebene befinden.

Ein Beispiel zeigt die folgende Abbildung.

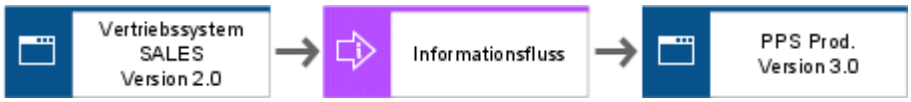


Abbildung 98: Datenfluss zwischen Anwendungssystemtypen

Neben den Informationsflüssen können auch für jeden Anwendungssystemtyp, Modultyp und jeden DV-Funktionstyp die Input- und Output-Daten in Form von Datenobjekten des Fachkonzeptes oder des DV-Konzeptes dargestellt werden. Anhand der Pfeilrichtung kann dabei spezifiziert werden, ob es sich um einen eingehenden (Input) oder einen ausgehenden (Output) Datenstrom handelt.

Ein Beispiel zeigt die folgende Abbildung.

Abbildung 99: I/O-Daten auf DV-Konzept-Ebene

3.4.2.3 Organisation mit Funktion

Die Verbindungen der organisatorischen Aspekte mit den auf DV-Konzept-Ebene definierten funktionalen Aspekten dient im Wesentlichen zur Beantwortung folgender Fragestellungen:

- Wer (welche Organisationseinheiten, Stellen, Personen,...) ist für die in der Funktionssicht auf DV-Konzept-Ebene spezifizierten Anwendungssystemtypen und Modultypen verantwortlich oder ist Anwender dieser Systeme?
- An welchen Standorten (Organisationssicht) im Unternehmen werden Anwendungssystemtypen oder Modultypen eingesetzt?
- Auf welchen zur Verfügung stehenden Plattformen des Unternehmens (Hardware-Komponententypen (Organisationssicht)) können Anwendungssystemtypen laufen?

Zur Beantwortung der ersten Fragestellung können im Zugriffsdiagramm Kanten zwischen den organisatorischen Einheiten des Organigramms (Organisationseinheiten, Stellen und Personen) und den Objekten des Anwendungssystemtypdiagramms (Anwendungssystemtyp, Modultyp, DV-Funktion,...) gezogen werden. Hierbei kann nochmals genauer spezifiziert werden, welche Bedeutung diese Beziehung haben soll. Es werden unterschieden:

- Eine Organisationseinheit **kann fachlich** für einen Anwendungssystemtyp **verantwortlich** sein.
- Eine Organisationseinheit **kann für die Entwicklung** eines Anwendungssystemtyps **verantwortlich** sein.
- Eine Organisationseinheit kann **Anwender** eines Anwendungssystemtyps sein.

Zur Beantwortung der Standortfrage können Standorte der Organisationssicht den Anwendungssystemtypen, Modultypen und DV-Funktionstypen zugeordnet werden.

Da es sich im DV-Konzept nicht um einzelne Anwendungssysteme im Sinne einer einzelnen Lizenz, sondern um Anwendungssystemtypen handelt, werden mit dieser Beziehung nicht die konkreten Standorte von Anwendungssystemen definiert (diese Zuordnungen erfolgen auf Implementierungsebene), sondern es wird eine Aussage darüber getroffen, an welchen Standorten sich ein Anwendungssystemtyp befinden kann.

Im DV-Konzept der Organisationssicht wird definiert, welche Hardwarekomponententypen im Unternehmen zur Verfügung stehen. In der Steuerungssicht können nun diese Hardware-Komponententypen mit den Anwendungssystemtypen in Beziehung gesetzt werden. Dadurch wird festgelegt, auf welchen Hardware-Plattformen Anwendungssystemtypen, Modultypen oder DV-Funktionstypen lauffähig sind. An dieser Stelle können auch die in der Funktionssicht aufgenommenen Benutzeroberflächentypen, Betriebssystemtypen und DBMS-Typen den Hardware-Komponententypen zugeordnet werden.

Eine Zusammenfassung der möglichen Beziehungen des ERM-Attributzuordnungsdiagramms finden Sie in der **ARIS-Methode Reference**-Hilfe.

In folgender Abbildung sind beispielhaft Beziehungen dargestellt.

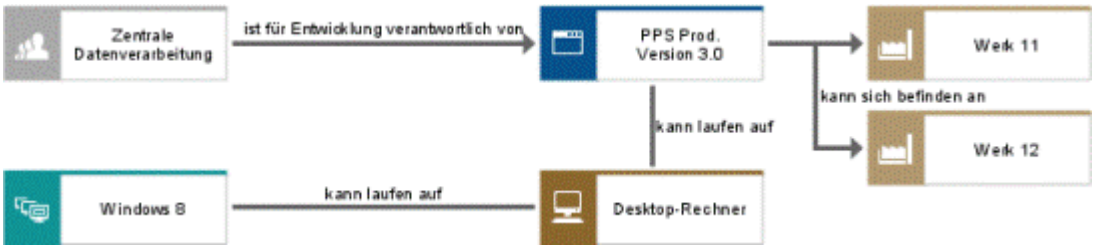


Abbildung 100: Zugriffsdiagramm (Auszug)

3.4.2.4 Programmablaufdiagramm

Im Zugriffsdiagramm können für die im Anwendungssystemtypdiagramm spezifizierten Anwendungssystemtypen, Modultypen und DV-Funktionstypen die Beziehungen zu den Objekttypen der Organisationssicht und Datensicht hergestellt werden (vgl. Kapitel **Zugriffsdiagramm** (Seite 96)). Die direkte Darstellung der Zuordnung von Funktionen des Fachkonzeptes ist in diesem Modelltyp nicht möglich. Diese Zuordnung erfolgt im Anwendungssystemtypdiagramm. Ebenfalls nicht direkt darstellbar sind die möglichen zeitlich-logischen Ablauffolgen der Anwendungssystemtypen, Modultypen und DV-Funktionstypen. Der ARIS-Architektur streng folgend, sind diese Verknüpfungen nur mit Hilfe der Navigation über verschiedene Modelltypen hinweg nachvollziehbar.

Gerade im Umfeld des Systemdesigns haben sich jedoch Modelltypen etabliert (z. B. Programmablaufpläne (PA) (Seite 99), die eine integrierte Sicht auf alle Aspekte des Systemdesigns zulassen.

Aus diesem Grunde wird in **ARIS** der Modelltyp **Programmablaufplan** angeboten, in dem ungeachtet der Sichtenzerlegung von **ARIS** alle Beziehungen zu Anwendungssystemtypen, Modultypen und DV-Funktionstypen, die in den anderen Modelltypen von **ARIS** zur Verfügung gestellt werden, modellierbar sind. Daneben können auch die zeitlich-logischen Ablauffolgen der genannten Objekttypen dargestellt werden. Hierzu werden auch Ereignisse in diesem Modelltyp zur Verfügung gestellt. Ähnlich der Zuordnung von Funktionen und Ereignissen in der EPK können hiermit im Programmablaufdiagramm Modulfolgen definiert werden. Das Ereignis ist hierbei im Sinne eines Triggers zu verstehen, das Modultypen oder Anwendungssystemtypen anstößt. Verzweigungen können mit den aus der EPK bekannten Regeln dargestellt werden. Im Gegensatz zur EPK können im Programmablaufdiagramm jedoch auch Ablauffolgen ohne das Zwischenschalten von Ereignissen definiert werden.

3.4.2.5 Programmablaufplan (PA)

Der Programmablaufplan (PA) dient zur Darstellung von Verarbeitungsfolgen in einem Programm. Die Beziehungen zwischen den Objekten zeigen dabei die Reihenfolgen der Verarbeitung auf. Daten werden durch dieses Diagramm nicht wiedergegeben.

Die folgende Abbildung zeigt ein vereinfachtes Beispiel für die Verarbeitungsfolgen eines Geldautomaten. In der Darstellung der Verarbeitungsfolgen wird eine starke Orientierung an die Implementierung sichtbar.

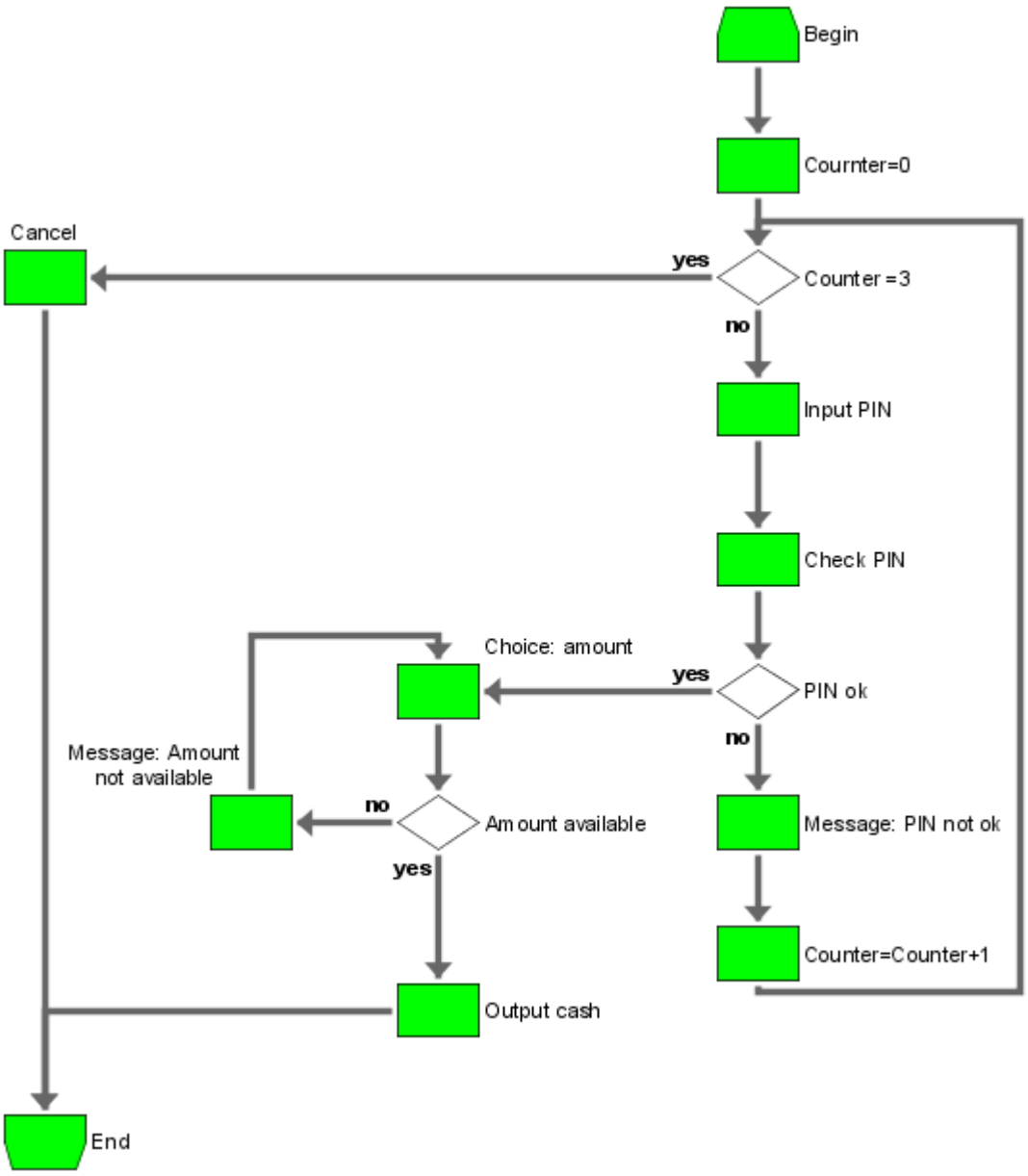


Abbildung 101: Beispiel eines Programmablaufplans (PA)

3.4.2.6 Maskendiagramm

Anwendungsgebiet des Maskendiagramms ist die Beschreibung von Bildschirmmasken bei der Software-Entwicklung. Ziel ist dabei eine automatische Ableitung der Bildschirmmasken aus dem Maskendiagramm.

Folglich bildet das Maskendiagramm den Aufbau und in gewissem Grad die Funktionalität von Bildschirmmasken ab. Es gilt dabei: der Aufbau des Maskendiagramms von links nach rechts und von oben nach unten entspricht der Geometrie der beschriebenen Oberfläche.

Zentrales Symbol ist die Maske; es stellt entsprechend der Windows-Terminologie ein Fenster dar. Dieses Fenster kann mehrere Registerkarten aufweisen (Symbol **Seite**). Generell kann die Oberfläche mittels einer Tabellenform geographisch in Bereiche eingeteilt werden (Symbol **Sektion** für eine Zeile und Symbol **Spalte** für eine Spalte). Die Symbole **Sektion** und **Spalte** sind zur Bildung komplexer Oberflächen beliebig schachtelbar. Auf der Oberfläche können Tabellen (Symbol **Maskentabelle**), Texteingabefelder (Symbol **COT-Attribut**) sowie Grafiken (Symbol **Bitmap**) und Erläuterungstext (Symbol **Text**) platziert werden. Mit dem Symbol **Layout** können den Objekten **Maske**, **Seite**, **Sektion**, **Spalte**, **Maskentabelle**, **COT-Attribut** und **Text** Darstellungseigenschaften zugewiesen werden.

Weitere Symbole können für die Beschreibung der Bildschirmoberfläche verwendet werden.

In folgender Abbildung ist beispielhaft ein Maskendiagramm abgebildet. Die zweite Abbildung zeigt die daraus abgeleitete Bildschirmmaske.

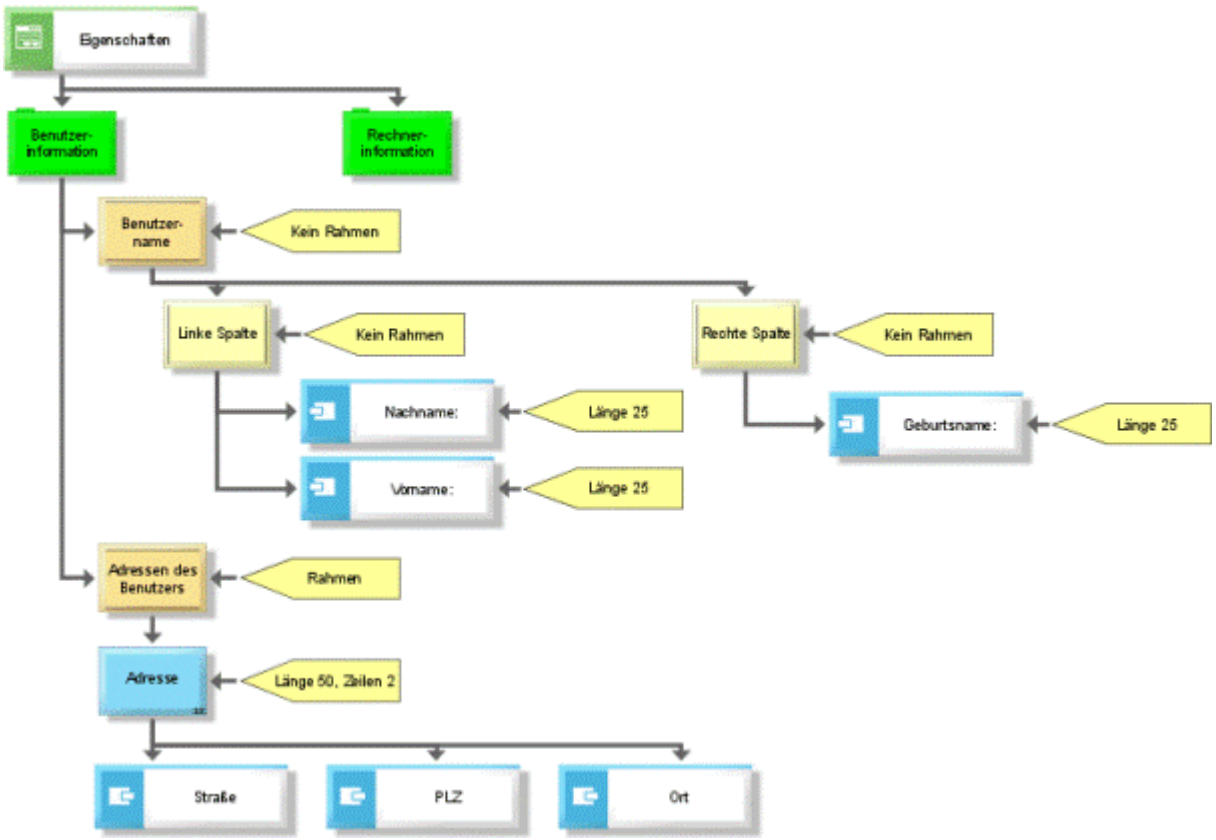


Abbildung 102: Beispiel eines Maskendiagramms

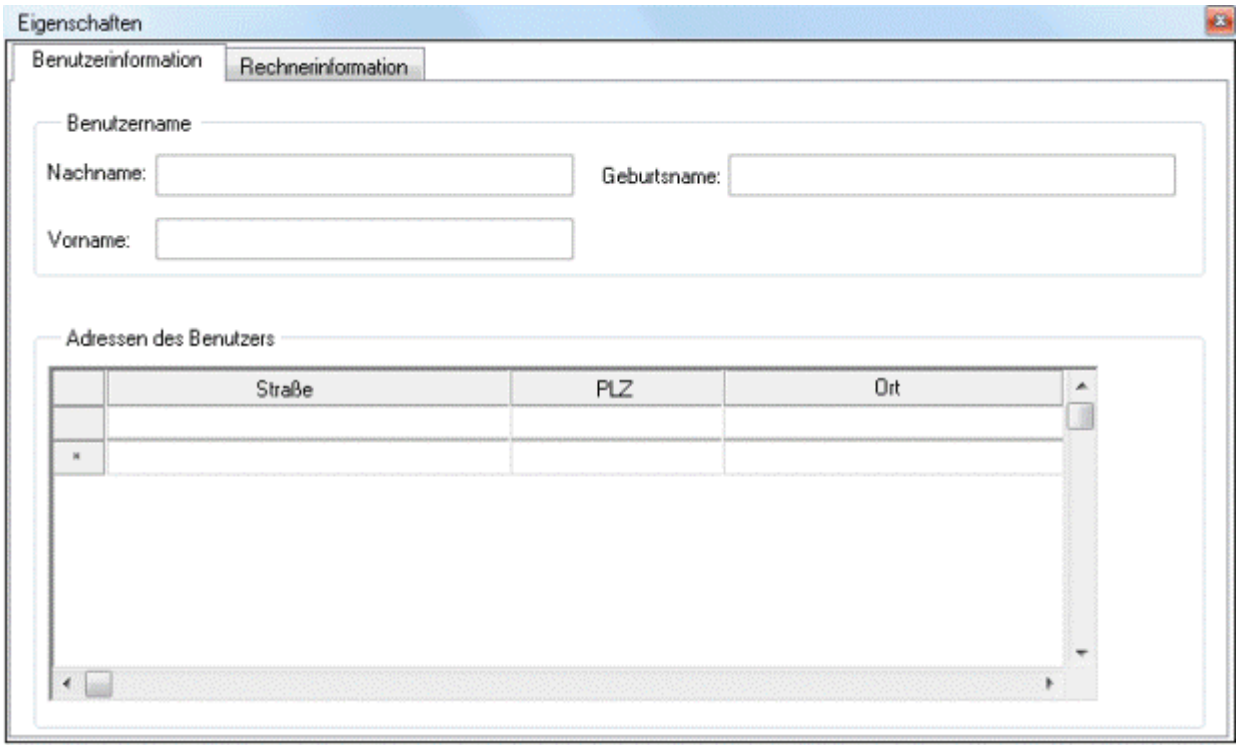


Abbildung 103: Aus dem Maskendiagramm der vorherigen Abbildung abgeleitete Bildschirmmaske

3.4.3 Implementierung - Zugriffsdiagramm (physikalisch)

Die im DV-Konzept der Steuerungssicht behandelten Fragestellungen sind auch Gegenstand der Implementierungsebene. Im Unterschied zur DV-Konzept-Ebene werden hier jedoch nicht die Objekttypen betrachtet, sondern die konkreten Exemplare der einzelnen Objekte. So werden z. B. die Beziehungen von konkreten Anwendungssystemen zu Organisationseinheiten betrachtet und nicht die Beziehungen von Anwendungssystemtypen zu Organisationseinheiten.

Die im Folgenden dargestellten Beziehungen werden im Zugriffsdiagramm (physikalisch) modelliert.

3.4.3.1 Funktionen mit Daten

Zur Beantwortung der Frage, welche Daten zwischen Anwendungssystemen fließen, können Informationsflussobjekte zwischen den Anwendungssystemobjekten der Funktionssicht angelegt werden. Im Unterschied zur DV-Konzept-Ebene handelt es sich bei diesen Anwendungssystemobjekten jedoch nicht um Anwendungssystemtypen, sondern um konkrete Exemplare (einzelne Lizenzen). Es können also Anwendungssysteme, Module und Programmteiltypen über Datenflusskanten miteinander verbunden werden. Wurde auf der DV-Konzept-Ebene definiert, dass der Modultyp **Vertriebssystem SD Version 2.1** Daten mit dem Modultyp **Materialwirtschaftssystem MM Version 1.2** austauschen kann, wird auf der Implementierungsebene dargestellt, dass das konkret installierte Modul **SD Lizenznummer 1234** Daten mit dem Modul **MM, Lizenznummer 2352** und mit dem Modul **MM, Lizenznummer 34234** austauscht. Beide Module MM sind vom Modultyp **Materialwirtschaftssystem MM Version 1.2**. Dieser Sachverhalt wird auch in folgender Abbildung dargestellt.

Abbildung 104: Datenfluss

Um die Datenobjekte, die zwischen den Systemen ausgetauscht werden, näher spezifizieren zu können, werden den Informationsflussobjekten entsprechende Modelltypen der Datensicht hinterlegt.

Neben den Datenflüssen zwischen Anwendungssystemen können auch für jedes Anwendungssystem die Input-/Output-Daten spezifiziert werden. Dabei gibt es zwei Gründe dafür, dass die Beziehungen im Zugriffsdiagramm (physikalisch) dargestellt werden. Im ersten Fall handelt es sich bei den Datenobjekten um Objekte des Tabellendiagramms (Tabelle, Feld, View (phys.)), welches sich in der Datensicht auf Implementierungsebene befindet. Diese Datenobjekte können über Input-/Output-Beziehungen mit Anwendungssystemobjekten der DV-Konzept-Ebene oder der Implementierungsebene verbunden werden. Im zweiten Fall handelt es sich bei den Anwendungssystemobjekten um die konkreten Anwendungssysteme oder Module der Implementierungsebene, die mit den Objekten der Datensicht verbunden werden.

Als allgemeine Regel kann somit definiert werden:

Stammt einer der Objekttypen bei Input-/Output-Beziehungen aus der Implementierungsebene der entsprechenden Sicht, werden auch die Beziehungen in der Prozesssicht auf Implementierungsebene (Zugriffsdiagramm (physikalisch)) dargestellt.

Ein Beispiel zeigt die folgende Abbildung.

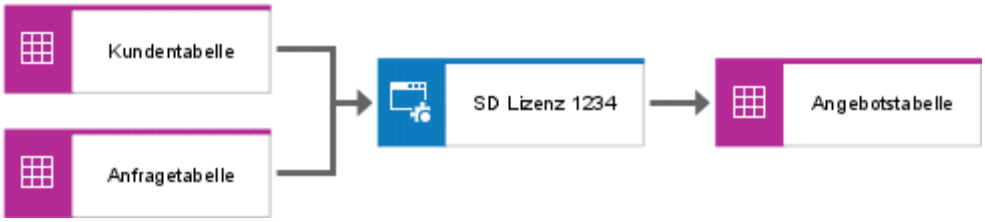


Abbildung 105: Input-/Outputbeziehungen

3.4.3.2 Organisation mit Daten

Im Vordergrund stehen hier die gleichen Fragestellungen, die auch im DV-Konzept betrachtet wurden:

- Welche organisatorischen Einheiten sind für Datenobjekte verantwortlich?
- Wer darf auf welche Datenobjekte zugreifen?
- Auf welchen Hardwarekomponenten sind Datenobjekte gespeichert?

Im Gegensatz zum DV-Konzept werden hier die Beziehungen jedoch zu den Datenobjekten hergestellt, die auf der Implementierungsebene der Datensicht dargestellt werden.

Die Verantwortlichkeit für Datenobjekte wird also für Relationen und Attribute physischer Strukturen, also Tabellen, Felder und deren Exemplare [Tabelle (Exemplar), Feld (Exemplar)] definiert.

Zur Darstellung dieser Abhängigkeiten werden im Zugriffsdiagramm (physikalisch) Kanten zwischen den Objekten der Organisationssicht (Organisationseinheit, Stelle, Person, ...) und den bereits erwähnten Objekten des Tabellendiagramms (Tabelle, Feld, View [phys.], ...) angelegt.

Beim Anlegen der Kante zwischen den organisatorischen Einheiten und den Tabellen und Feldern muss dann jeweils die Bedeutung der Beziehung definiert werden. **Ist verantwortlich für** bedeutet, dass die organisatorische Einheit für den Inhalt der jeweiligen Tabelle oder des jeweiligen Feldes verantwortlich ist; **greift zu auf** bedeutet, dass die Stelle oder Person die Zugriffsberechtigung auf die dargestellten Datenobjekte hat.

Neben der Definition der Zugriffsrechte und Verantwortlichkeiten kann nun mit dem Objekt Hardwarekomponente (Organisationssicht/Implementierung) definiert werden, auf welchen konkret vorhandenen Hardwarekomponenten - die z. B. über die Inventarisierungsnummer des Unternehmens eindeutig bestimmt werden - sich bestimmte Informationsobjekte des Unternehmens befinden. Hierzu können im Zugriffsdiagramm (physikalisch) mit dem Objekt **Hardwarekomponente** Informationsobjekte der Implementierungsebene (Tabellen, Felder...), der DV-Konzept-Ebene (Relationen, Attribute) oder auch der Fachkonzept-Ebene (Entitytypen, Cluster/Datenmodell...) verbunden werden.

Ein Beispiel zeigt die folgende Abbildung.

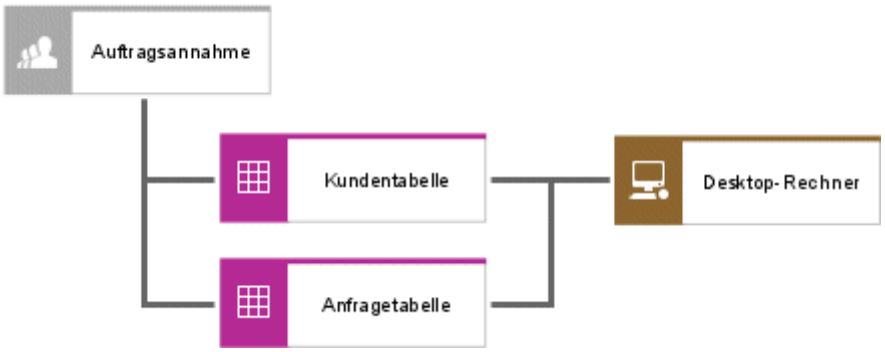


Abbildung 106: Zuordnungen zur Hardwarekomponente

3.4.3.3 Organisation mit Funktionen

Die im Zugriffsdiagramm (physikalisch) definierten Beziehungen zwischen den Objekten der Organisationssicht und der Funktionssicht erlauben die Beantwortung der folgenden Fragen:

AUF WELCHEN HARDWAREKOMPONENTEN LAUFEN WELCHE ANWENDUNGSSYSTEME UND WELCHE ANWENDUNGSSYSTEMTYPEN KÖNNEN DARAUF LAUFEN?

Zur Darstellung dieser Abhängigkeiten können die Beziehungen **ist Plattform von** und **kann Plattform sein von** zwischen den Anwendungssystemobjekten der Implementierungsebene (Anwendungssystem, Modul, Programmteil, ...) oder der DV-Konzept-Ebene (Anwendungssystemtyp, Modultyp, ...) und dem Objekttyp **Hardwarekomponente** aus der Organisationssicht modelliert werden.

Ein Beispiel zeigt die folgende Abbildung.

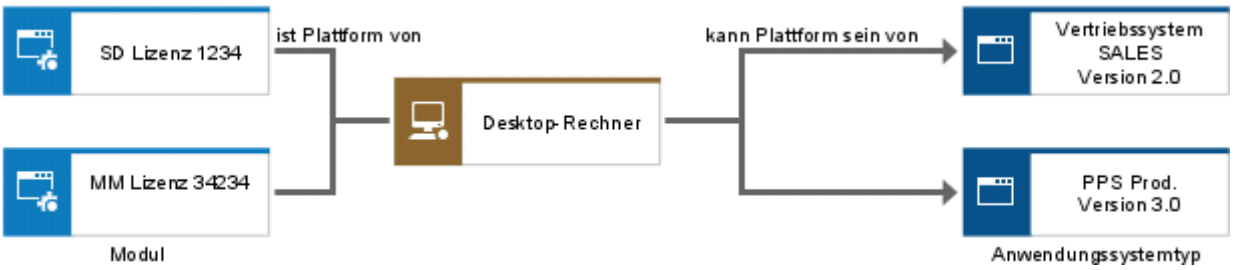


Abbildung 107: Hardwarekomponente als Plattform

WELCHE ORGANISATORISCHE EINHEIT IST ANWENDER EINES KONKRETEN ANWENDUNGSSYSTEMS?

Werden auf DV-Konzept-Ebene die Anwender definiert, die auf Anwendungssystemtypen zugreifen, kann nun auf Implementierungsebene diese Beziehung für konkrete Anwendungssysteme (einzelne Lizenzen) definiert werden. So können zum Beispiel von einer

Version des Anwendungssystemtyps **ARIS Architect** mehrere Lizenzen in unterschiedlicher Konfigurierung im Unternehmen zur Verfügung stehen. Im Zugriffsdiagramm (physikalisch) kann nun dargestellt werden, welcher Anwender mit welcher Lizenz arbeitet. Hierzu können die Objekttypen **Organisationseinheit**, **Stelle** und **Person** über die Kante **ist Anwender** mit den Objekttypen **Anwendungssystem** und **Modul** verbunden werden. Ein Beispiel zeigt die folgende Abbildung.

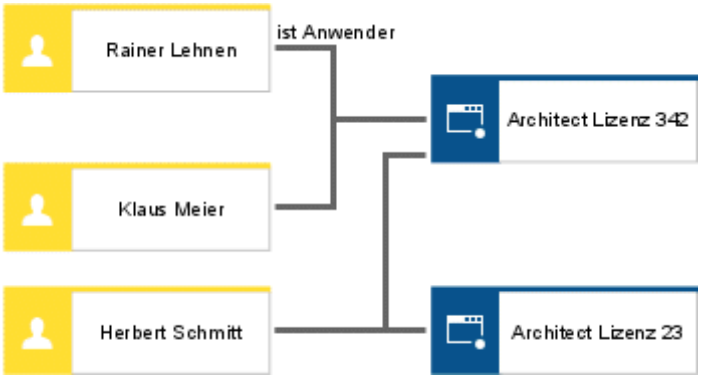


Abbildung 108: Anwender und Anwendungssystem

AN WELCHEN STANDORTEN DES UNTERNEHMENS SIND ANWENDUNGSSYSTEME INSTALLIERT?

Im DV-Konzept konnte über die Beziehung **Anwendungssystemtyp - Standort** definiert werden, welche Anwendungssystemtypen sich an bestimmten Standorten des Unternehmens befinden können. Um nun auch konkret definieren zu können, an welchen Standorten des Unternehmens sich die einzelnen Lizenzen, die von einem Anwendungssystemtyp erworben wurden, eingesetzt werden, bietet das Zugriffsdiagramm (physikalisch) die Möglichkeit, Standorte auch mit den Objekttypen **Anwendungssystem**, **Modul** und **DV-Funktion** zu verbinden.

Ein Beispiel zeigt die folgende Abbildung.

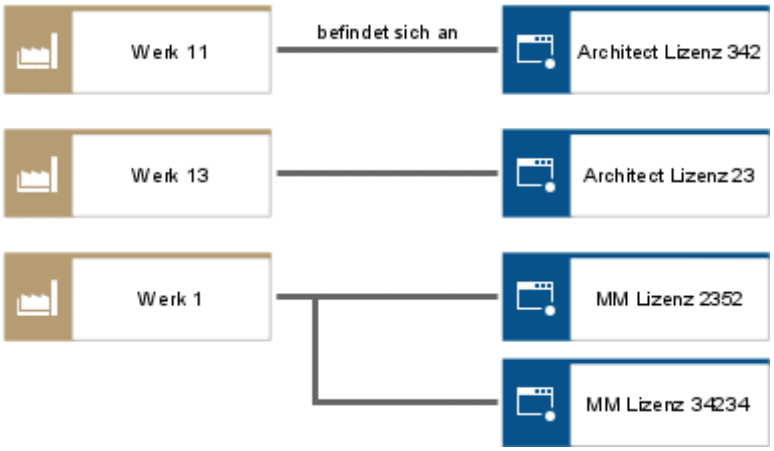


Abbildung 109: Standortzuordnungen

In der **ARIS-Methode Reference**-Hilfe sind alle Beziehungen zusammengefasst, die im Zugriffsdiagramm (physikalisch) zur Verfügung stehen.

3.5 Leistungsmodellierung

ARIS stellt zur Beschreibung von Leistungen in einer Unternehmung verschiedene Modelltypen zur Verfügung.

Eine Leistung wird im Verlauf eines Wertschöpfungsprozesses erbracht. Sie ist das Ergebnis einer menschlichen Handlung oder eines technischen Vorgangs. Eine Leistung kann eine Dienstleistung oder eine Sachleistung sein.

Sachleistungen können einem Verbrauchsprodukt, Materialtyp, Betriebsmitteltyp, technischem Hilfsmitteltyp oder einem Packmitteltyp entsprechen. Auslöser zur Erstellung einer Leistung ist immer der Bedarf einer Organisationseinheit oder eines Kunden. Eine Sachleistung wird dem Kunden in Form eines materiell existierenden Guts angeboten.

Eine Dienstleistung ist ein immaterielles Gut, für das Produktion und Verbrauch zeitlich zusammenfallen.

Anbieter von reinen Dienstleistungen sind z. B. Banken, Versicherungen oder die öffentliche Verwaltung.

Je stärker die Kundenorientierung im Marktsegment eines Anbieters von Sachleistungen betont wird, desto wichtiger es für ihn die Dienstleistung im Produktumfeld genauer zu betrachten und weiterzuentwickeln.

Die verschiedenen Modelltypen von ARIS sind daher sowohl für die Beschreibung reiner Sach- oder Dienstleistungen als auch für die Beschreibung kombinierter Leistungen geeignet.

Folgende Modelltypen stehen Ihnen für die Leistungsmodellierung zur Verfügung:

- Leistungsaustauschdiagramm
- Leistungsbaum
- Produktzuordnungsdiagramm
- Produktbaum
- Produktauswahlmatrix

3.5.1 Leistungsaustauschdiagramm

Das Leistungsaustauschdiagramm dient dazu, die Leistungserstellung sowie den Leistungsaustausch im Unternehmen abzubilden. Eine Leistung kann entweder eine Dienstleistung oder ein Produkt sein und wird durch entsprechende Symbole repräsentiert. Produkte können Materialtypen, Betriebsmitteltypen, technischen Hilfsmitteltypen und/oder Packmitteltypen entsprechen, die man z. B. aus der EPK (Materialfluss) kennt. Leistungen als Input und/oder Output von Funktionen können mit den Anfangs- und/oder Endereignissen dieser Funktionen in Verbindung gesetzt werden.

Dieser Leistungsaustausch zwischen betriebswirtschaftlichen Funktionen kann bereits auf einer Abstraktionsebene sinnvoll eingesetzt werden, die zwischen Wertschöpfungskettendiagramm und EPK liegt. Neben den Austauschbeziehungen aus funktionaler Sicht sind die Austauschbeziehungen von Leistungen aus organisatorischer Sicht abzubilden. Hierzu werden im Leistungsaustauschdiagramm ebenfalls Modellierungsmöglichkeiten geboten.

Die folgende Abbildung zeigt ein Beispiel für ein Leistungsaustauschdiagramm.

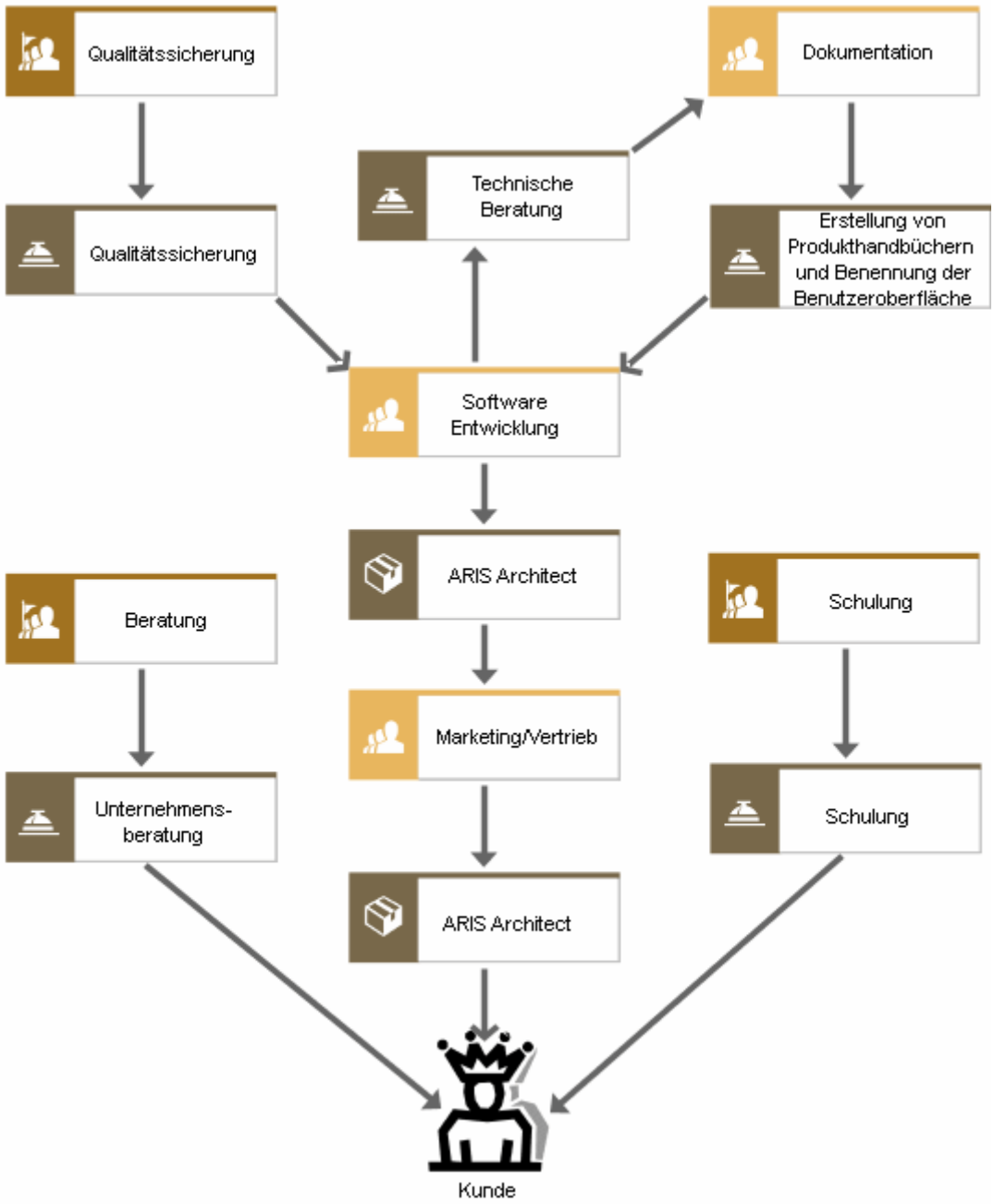


Abbildung 110: Exemplarischer Leistungsaustausch in einem Softwarehaus

3.5.2 Leistungsbaum

Leistungen können auf verschiedenen Abstraktionsebenen betrachtet werden. Dazu ist es sinnvoll, diese Beziehungen in einem Modell abzulegen, in dem dargestellt wird, aus welchen Teilleistungen sich eine Leistung zusammensetzt. Dieser statische Aspekt wird im Leistungsbaum abgebildet. So enthält beispielsweise ein komplexes Produkt häufig

verschiedene Baugruppen, die wiederum aus diversen Einzelteilen bestehen. Jedes dieser Elemente kann als Leistung verstanden werden.

Mit Hilfe der Kante **steht in Beziehung mit**, die auch zwischen Leistungen im Leistungsbaum erlaubt ist, können Abhängigkeiten anderer Art beschrieben werden. Hierzu gehört z. B. die Beziehung zwischen Konsumentenkredit und Girokonto, über das die Zahlungen abgewickelt werden können.

Zudem können auch Substitutionsbeziehungen zu anderen Leistungen wie (potentiellen) Ersatzprodukten bzw. -dienstleistungen dargestellt werden.

In diesem statischen Modell werden auch die Beziehungen der Leistungen zu den (Unternehmens-) Zielen abgebildet.

Die folgende Abbildung zeigt beispielhaft einen Leistungsbaum.

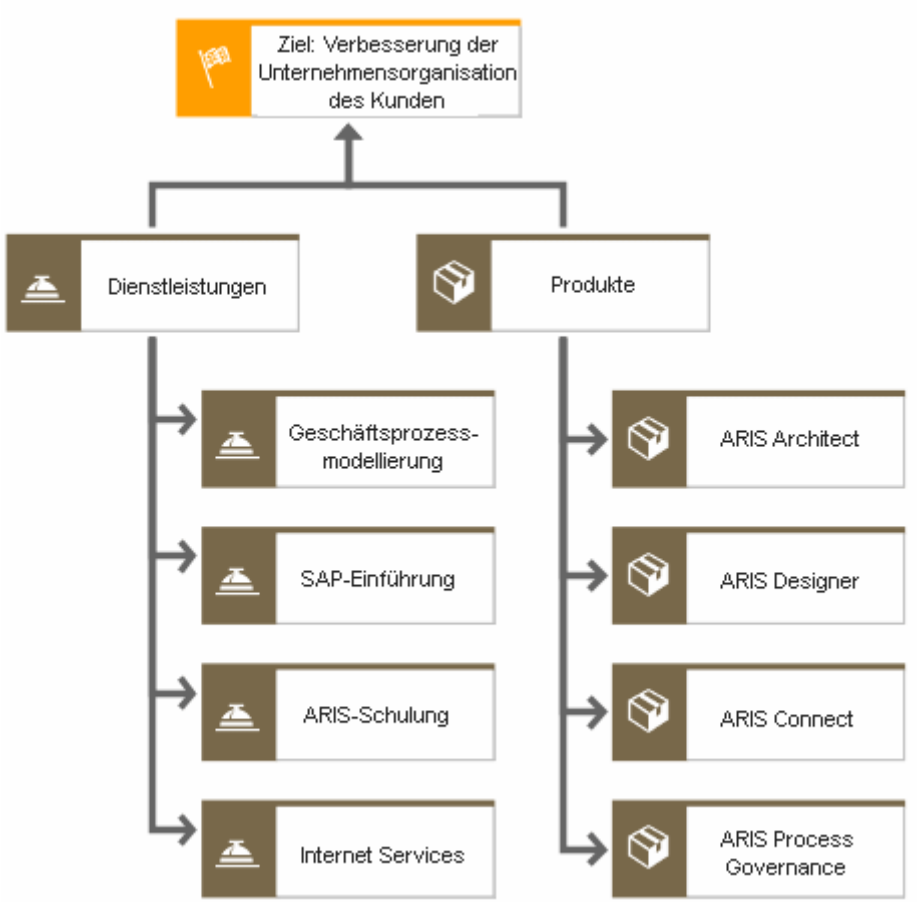


Abbildung 111: Leistungsbaum

3.5.3 Produktzuordnungsdiagramm

Neben den allgemeinen Leistungsdiagrammen, die zu den bildhaften Modellen gehören, bieten sich die Produktmodelle an, um eine eher abstrakte Darstellung zu ermöglichen. Das Produktzuordnungsdiagramm dient in erster Linie der Analyse der Produkterstellung in öffentlichen Verwaltungen. Ähnlich wie im Leistungsaustauschdiagramm kann mit diesem

Modelltyp aufgezeigt werden, welche organisatorischen Einheiten Produkte bereitstellen oder nutzen und welche Funktionen zur Erstellung der Produkte benötigt werden oder für welche Funktionen die Produkte einen Input liefern. Ergänzend werden hier die (gesetzlichen) Auftragsgrundlagen der jeweiligen Produkte aufgezeigt. Daneben können auch die Ziele, die mit den Produkten erreicht werden sollen, dargestellt werden.

Die folgende Abbildung zeigt einen Ausschnitt aus einem Produktzuordnungsdiagramm der öffentlichen Verwaltung.

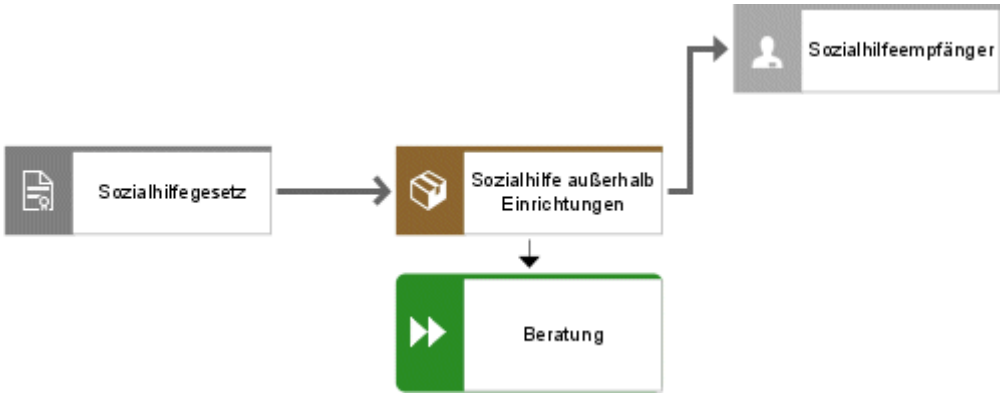


Abbildung 112: Beispiel eines Produktzuordnungsdiagramms

Der Modelltyp bietet aber auch die Möglichkeit Aspekte zu beschreiben, die mit der Vermarktung des Produktes in Zusammenhang stehen.

Diese Aspekte sollen am vereinfachten Beispiel von Bankprodukten verdeutlicht werden.

Mit der Verbreitung des Internets und der steigenden Anzahl von Internet-Anschlüssen in privaten Haushalten der letzten 10 Jahre stieg auch die Verbreitung des Online-Banking. Gleichzeitig stieg auch die Finanzkraft der Jugendlichen, ihre Bedeutung als Zielgruppe stieg.

Dies hat dazu geführt, dass die ursprüngliche Leistung **Girokonto** in unterschiedlichen Formen angeboten wird:

Es kann z. B. als Seniorenkonto angeboten werden, dessen Inhaber von den Mitarbeitern einer Geschäftsstelle betreut werden. Dieses Produkt wendet sich u. a. an ältere Kunden, die weniger mit neuen Technologien vertraut sind, Wert auf persönliche Betreuung und Beratung durch ihnen bekannte Personen legen und aufgrund altersbedingter Gebrechlichkeiten nicht mehr sehr mobil sind. Die für ein solches Konto anfallenden Gebühren dürfen über dem Durchschnitt liegen.

Das Girokonto kann aber auch als günstiges Teeny-Konto im Rahmen des Online-Banking angeboten werden. Dieses Produkt wendet sich an Jugendliche zwischen 12 und 20, die mit der Internet-Technologie vertraut sind, aber über eher geringe finanzielle Mittel verfügen. Die Gebühren müssen sich daher im unteren Bereich bewegen.

Folgende Grafiken zeigen Produktzuordnungsdiagramme für diese beiden Produktvarianten:

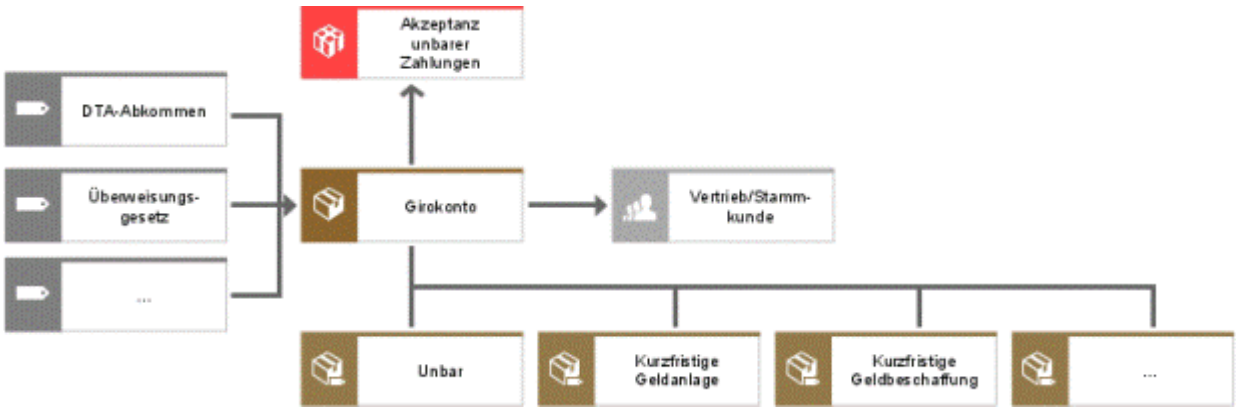


Abbildung 113: Produktzuordnungsdiagramm - Girokonto

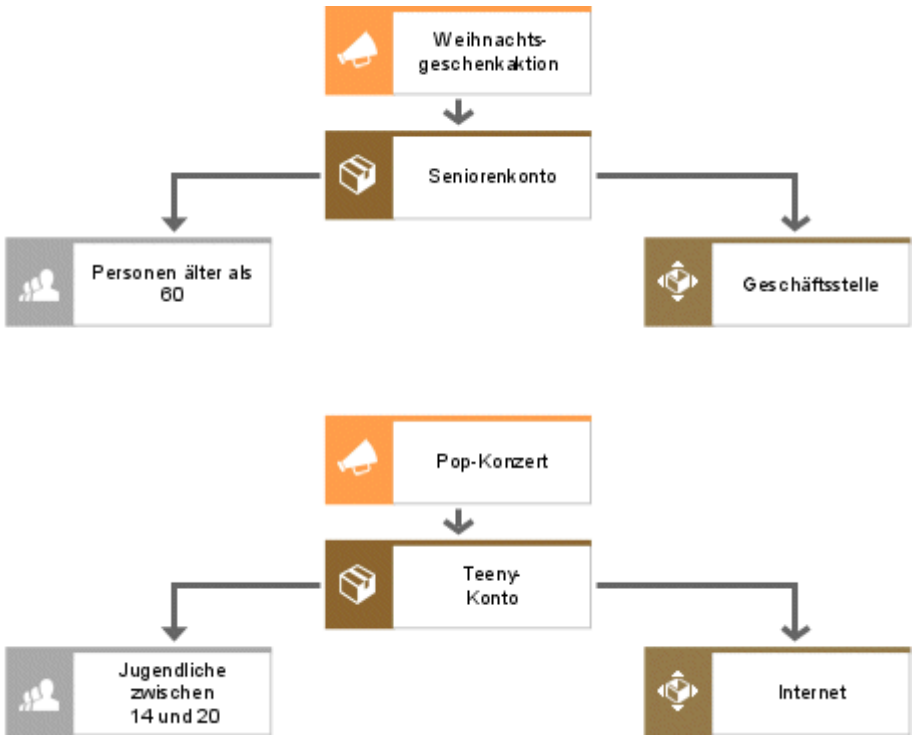


Abbildung 114: Produktzuordnungsdiagramme - Verkaufsprodukte

Die Leistungen **Teeny-** und **Seniorenkonto** wurden als Objektvarianten des Girokontos angelegt und durch das Attribut **Verkaufsprodukt** gekennzeichnet. Ein Verkaufsprodukt ist eine vom Unternehmen erstellte Leistung, die unter verschiedenen Namen in verschiedenen Marktsegmenten angeboten wird. Für die verschiedenen Verkaufsprodukte werden in der Regel jeweils unterschiedliche marketingpolitische Instrumente verwendet.

Mit Hilfe der Varianten-Komponente ARIS können ausgehend von einem Produkt beliebig viele Verkaufsprodukte angelegt werden.

3.5.4 Produktbaum

Zweck des Produktbaumes ist es, die Zusammensetzung von Produkten in der öffentlichen Verwaltung zu analysieren. Dieses Modell entspricht im Wesentlichen dem Leistungsbaum, wobei auf die Möglichkeit der Modellierung von Ersatzprodukten verzichtet wurde. Der Produktbaum ist in der Fachkonzeptebene der Leistungssicht angesiedelt.

In folgender Abbildung wird ein Produktbaum beispielhaft dargestellt.

Abbildung 115: Klassifizierung der Produktgruppe "Einwohner- und Staatsangehörigkeitsangelegenheiten" mit Hilfe eines Produktbaumes

3.5.5 Produktauswahlmatrix

In der Produktauswahlmatrix steht eine Organisationseinheit mit den von ihr zu verantworteten Produkten im Vordergrund. Diesen Produkten können die für ihre Erstellung notwendigen Funktionen zugeordnet werden. Das Modell eignet sich als Einstiegspunkt, von dem zu Organigrammen, zu Produktbäumen und zu den für die Erstellung der Leistungen relevanten Prozessen navigiert werden kann. Ein Beispiel für eine Produktauswahlmatrix zeigt diese Abbildung.

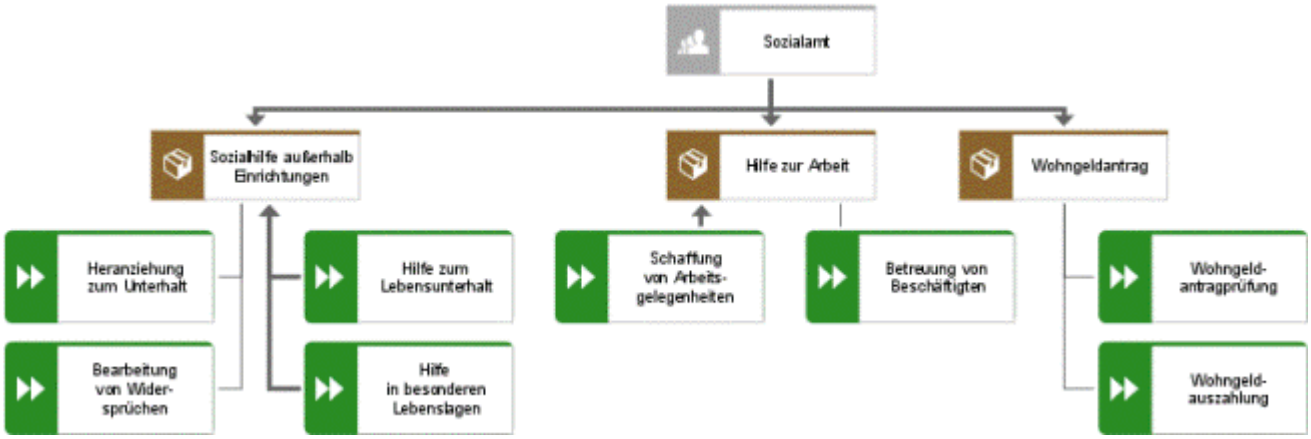


Abbildung 116: Produktauswahlmatrix des Sozialamts

4 Unified Modeling Language (UML) in ARIS

4.1 Einleitung

UML (Unified Modeling Language) ist eine objektorientierte Modellierungssprache, deren Sprachkonstrukte durch eine Arbeitsgruppe der OMG (Object Management Group) standardisiert werden. UML basiert auf den objektorientierten Ansätzen von OMT, Booch und OOSE.

4.2 ARIS UML Designer - Unterstützter UML-Standard

ARIS UML Designer 9.x unterstützt den kompletten UML-Standard 2.5.

Die UML-Spezifikation finden Sie unter **https://www.omg.org/spec/UML/2.5/Beta2/PDF/**.

5 Methoden für das Wissensmanagement

5.1 Einleitung

Wissensmanagement bezweckt die systematische Bewirtschaftung der zunehmend wichtigen Unternehmensressource Wissen. Es umfasst die Entwicklung, Überwachung, Unterstützung und Weiterentwicklung von Strategien, Prozessen, Organisationsstrukturen und Technologien für die effektive Wissensverarbeitung im Unternehmen. Hierzu gehören alle Aktivitäten, die den Erwerb, die Aufbereitung, die Übertragung und die Nutzung von Wissen betreffen. Diese Aktivitäten der Wissensverarbeitung finden in der Regel nicht isoliert statt, sondern hauptsächlich in den operativen und dispositiven Geschäftsprozessen des Unternehmens. Es ist daher eine integrierte Betrachtung von Geschäftsprozessen, Wissensverarbeitung, Organisationsstrukturen, Informationssystemen usw. erforderlich.

Ein Großteil dieser Aspekte lässt sich mit bereits seit längerem existierenden ARIS-Methoden abbilden, z. B. EPKs, Organigrammen, Funktionszuordnungsdiagrammen, eERMs usw. Ist jedoch eine genaue Darstellung, Analyse und Verbesserung der Wissensverarbeitung gewünscht, sind zusätzliche Darstellungsmittel notwendig, z. B. zur Identifikation und inhaltlichen Strukturierung relevanter Wissenskategorien, zur Beschreibung der Verteilung von Wissen in der Organisation und zur Modellierung der Wissensentstehung und -nutzung in den Geschäftsprozessen.

Aus diesem Grunde wurden die zwei neuen Objekttypen **Wissenskategorie** und **Dokumentiertes Wissen** sowie die zwei neuen Modelltypen **Wissensstrukturdiagramm** und **Wissenslandkarte** aufgenommen. Außerdem wurden die Modelltypen zur Darstellung von Geschäftsprozessen (EPK etc.) um Konstrukte für die Wissensentstehung und -nutzung erweitert. Die neuen Objekt- und Modelltypen sind methodisch voll in den wichtigsten sonstigen Modelltypen des Fachkonzeptes wie eERM, Organigramm, Funktionsbaum integriert, sodass eine durchgängige Betrachtung gewährleistet wird. Hierdurch können beispielsweise Modelle aus einem Projekt zur Geschäftsprozessoptimierung auch zur Analyse und Verbesserung der Wissensverarbeitung herangezogen werden.

Das Wissensstrukturdiagramm befindet sich in der Datensicht des Fachkonzeptes, die Wissenslandkarte - ebenso wie die erweiterten Modelltypen für die Geschäftsprozessmodellierung - in der Steuerungssicht des Fachkonzeptes.

5.2 Objekttypen zur Modellierung der Wissensverarbeitung

5.2.1 Wissenskategorie

Der Objekttyp **Wissenskategorie**, dargestellt durch eine ovale Denkblase (vgl. Abbildung **Wissensstrukturdiagramm** (Seite 121)) repräsentiert einen inhaltlichen Gegenstand, auf den sich bestimmtes Wissen bezieht. Beispiele für **Wissenskategorien** sind: Wissen über Projektmanagement, Wissen über eine bestimmte Branche, Wissen über eine bestimmte Technologie, Wissen über Kunden und Wettbewerber usw. Mit Hilfe derartiger Kategorien lässt sich das im Unternehmen vorhandene oder benötigte Wissen einteilen.

Bei dem in eine bestimmte Wissenskategorie eingeteilten Wissen kann es sich sowohl um implizites Wissen handeln, also um nicht vollständig dokumentierbares, als Fähigkeiten von Mitarbeitern oder Gruppen vorhandenes Wissen, als auch um explizites Wissen, das - etwa in Form einer Beschreibung oder einer technischen Zeichnung - dokumentiert sein kann. Häufig beinhalten Wissenskategorien beides. Beispielsweise kann Wissen über Projektmanagement sowohl die Erfahrungen von Projektleitern als auch Hinweise in einem Handbuch zum Projektmanagement umfassen.

Zur näheren Beschreibung von Wissenskategorien dienen - neben allgemeinen Attributen wie Beschreibung, Bemerkung, Quelle usw. - folgende spezifischen Attribute:

Attributname	Wertebereich	Beschreibung/Beispiel
Umschlaghäufigkeit	Aufzählungstyp: stündlich, täglich, wöchentlich, monatlich, jährlich, selten, nie	Die Umschlaghäufigkeit gibt an, in welchen Zeitabständen das Wissen der entsprechenden Kategorie erneuert werden muss, um aktuell zu sein. Beispielsweise sind Grundkenntnisse der Trigonometrie nur selten oder - für praktische Zwecke - nie zu erneuern, wogegen Wissen über bestimmte Börsenkurse ggf. täglich oder gar stündlich zu erneuern ist.
Bedeutung	Prozentzahl: 0..100	Die Bedeutung der Wissenskategorie für das Unternehmen kann zwischen 0% (völlig unwichtig) und 100% (extrem wichtig) angegeben werden.

Attributname	Wertebereich	Beschreibung/Beispiel
Abdeckungsgrad	Prozentzahl: 0..100	Der aktuelle Abdeckungsgrad des betreffenden Wissens im Unternehmen kann zwischen 0% (überhaupt nicht abgedeckt) und 100% (maximal denkbare Abdeckung) angegeben werden. Soll der Abdeckungsgrad der Wissenskategorie durch eine bestimmte Organisationseinheit oder Person dargestellt werden, kann dies in einem entsprechenden Attribut der Kante des Typs **verfügt über** in einer Wissenslandkarte erfolgen.
Wissensvorsprung	Prozentzahl: 0..100	Der relative Wissensvorsprung gegenüber dem Wettbewerb kann zwischen 0% (Wettbewerb verfügt über maximal vorstellbaren Wissensvorsprung) und 100% (eigenes Unternehmen verfügt über maximalen Wissensvorsprung) angegeben werden.
Wissensnutzung	Prozentzahl: 0..100	Der Nutzungsgrad einer speziellen Wissenskategorie kann zwischen 0% (betreffendes Wissen wird überhaupt nicht genutzt) und 100% (betreffendes Wissen wird optimal genutzt) angegeben werden.
Angestrebter Abdeckungsgrad	Prozentzahl: 0..100	Der anzustrebende Abdeckungsgrad des betreffenden Wissens durch das Unternehmen kann zwischen 0% (überhaupt nicht abgedeckt) und 100% (maximal denkbare Abdeckung) angegeben werden.
Künftige Bedeutung	Aufzählungstyp: stark sinkend, sinkend, gleich bleibend, steigend, stark steigend	**Künftige Bedeutung** stellt die erwarteten Veränderungstendenzen bzgl. der Bedeutung einer Wissenskategorie für das Unternehmen dar.
Strukturelle Änderungsgeschwindigkeit	Prozentzahl: 0..100	Die strukturelle Änderungsgeschwindigkeit gibt an, wie schnell sich die angewandten Methoden zum Erwerb des entsprechenden Wissens ändern müssen (0%: keine Änderung, 100%: maximale Änderungsgeschwindigkeit).

Diese Attribute dienen der Beurteilung der jeweiligen **Wissenskategorie** in Bezug auf das eigene Unternehmen. Sie können daher als Grundlage für die Identifizierung von wichtigen und dringenden Maßnahmen zur Verbesserung des eigenen Wissensmanagements herangezogen werden. Häufig ist es nützlich, derartige Werte grafisch darzustellen. Eine

einfache Möglichkeit hierfür besteht darin, sie per Kopieren und Einfügen aus dem Fenster **Attribute** in ein Tabellenkalkulationsprogramm zu übertragen und dort entsprechende Modelle zu erstellen. Beispielsweise können für die betrachteten **Wissenskategorien** in einem Balkendiagramm jeweils der derzeitige und der angestrebte **Abdeckungsgrad** gegenübergestellt werden.

5.2.2 Dokumentiertes Wissen

Im Gegensatz zum Objekttyp **Wissenskategorie**, der implizites und explizites Wissen umfassen kann, bezieht sich der Objekttyp **Dokumentiertes Wissen** ausschließlich auf solche **Wissenskategorien**, die explizit dokumentiert sind oder zumindest prinzipiell dokumentierbar wären. Ein Beispiel hierfür ist das in einem Handbuch dokumentierte Wissen über die Bedienung einer Software. Bei der Einteilung von Wissen in Wissenskategorien hilft die Unterscheidung zwischen allgemeinen Wissenskategorien und dokumentiertem Wissen dabei, Möglichkeiten und Grenzen einer Informationssystemunterstützung der Wissensverarbeitung zu identifizieren, da nur dokumentiertes Wissen elektronisch gespeichert, übertragen und verarbeitet werden kann.

Der Objekttyp **Dokumentiertes Wissen** wird durch eine rechteckige Denkblase dargestellt. Er beinhaltet die gleichen spezifischen Attributtypen wie der Objekttyp **Wissenskategorie** (Seite 118).

5.3 Modelltypen zur Modellierung der Wissensverarbeitung

5.3.1 Wissensstrukturdiagramm

Mit Hilfe eines Wissensstrukturdiagramms lassen sich Wissenskategorien inhaltlich untergliedern. Dies ist in folgender Abbildung an einem Beispiel dargestellt. Eine Wissenskategorie kann sowohl andere Wissenskategorien umfassen, als auch dokumentiertes Wissen. Dokumentiertes Wissen kann wiederum in verschiedene Unterkategorien dokumentierten Wissens unterteilt werden. Es kann jedoch keine allgemeinen Wissenskategorien umfassen.

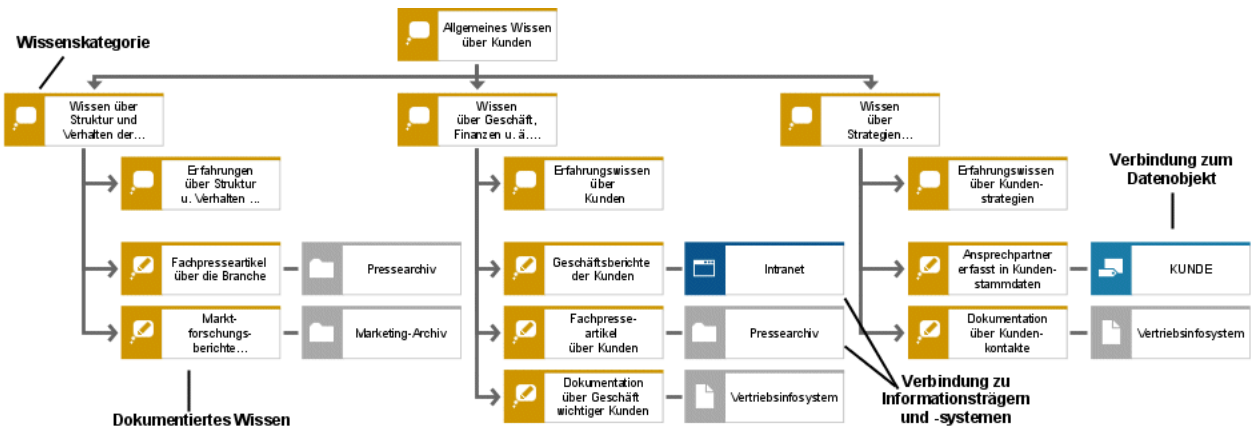


Abbildung 117: Wissensstrukturdiagramm

Für dokumentiertes Wissen kann im Wissensstrukturdiagramm dargestellt werden, auf welchen Informationsträgern es dokumentiert ist, bzw. welche Informationsobjekte eines Datenmodells oder welche Klassen eines objektorientierten Systems der Dokumentation dieses Wissens dienen. Schließlich lässt sich modellieren, mit welchen Typen oder Klassen von Anwendungssystemen dokumentiertes Wissen verwaltet wird.

5.3.2 Wissenslandkarte

In einer Wissenslandkarte lässt sich die organisatorische Verteilung der unterschiedlichen Wissenskategorien abbilden. Hierzu lassen sich unterschiedliche Objekttypen der Organisationssicht (z. B. Organisationseinheit, Stelle, Person, Standort, Gruppe) mit Hilfe von Kanten des Typs **verfügt über** mit Wissenskategorien verbinden. Neben der Tatsache, dass eine bestimmte Person oder Organisationseinheit über Wissen einer bestimmten Kategorie verfügt, lässt sich hierfür auch noch ein Abdeckungsgrad angeben. Die Kante **verfügt über** enthält das Attribut **Abdeckungsgrad**, mit dessen Hilfe in Prozentzahlen ausgedrückt werden kann, in welchem Maße das Wissen der gewählten Kategorie von der entsprechenden Organisationseinheit abgedeckt wird. Ein Wert von 100% steht für eine maximale Abdeckung; ein Wert von 0% bedeutet, dass keinerlei Wissen der genannten Kategorie vorhanden ist. Dies ist gleichbedeutend mit dem völligen Fehlen der Kante. Neben dieser quantitativen Angabe ist auch eine grafisch darstellbare qualitative Bewertung möglich. Hierzu dient das Kantenattribut **Abdeckung qualitativ**, das die Werte **wenig**, **mittel**, **hoch** und **maximal** annehmen kann. Diese Angaben können durch grafische Symbole an den Kanten visualisiert werden, wie dies in folgender Abbildung dargestellt ist. Zwischen den Werten der Attribute **Abdeckungsgrad** und **Abdeckung qualitativ** besteht kein unmittelbarer Zusammenhang. Sollen beide Attribute gemeinsam verwendet werden, ist es sinnvoll, die Qualifikation **wenig** bei einem Abdeckungsgrad bis zu 25% zu verwenden, entsprechend **mittel** von 26% und 50%, **hoch** von 51 und 75% und **maximal** zwischen 76% und 100%.

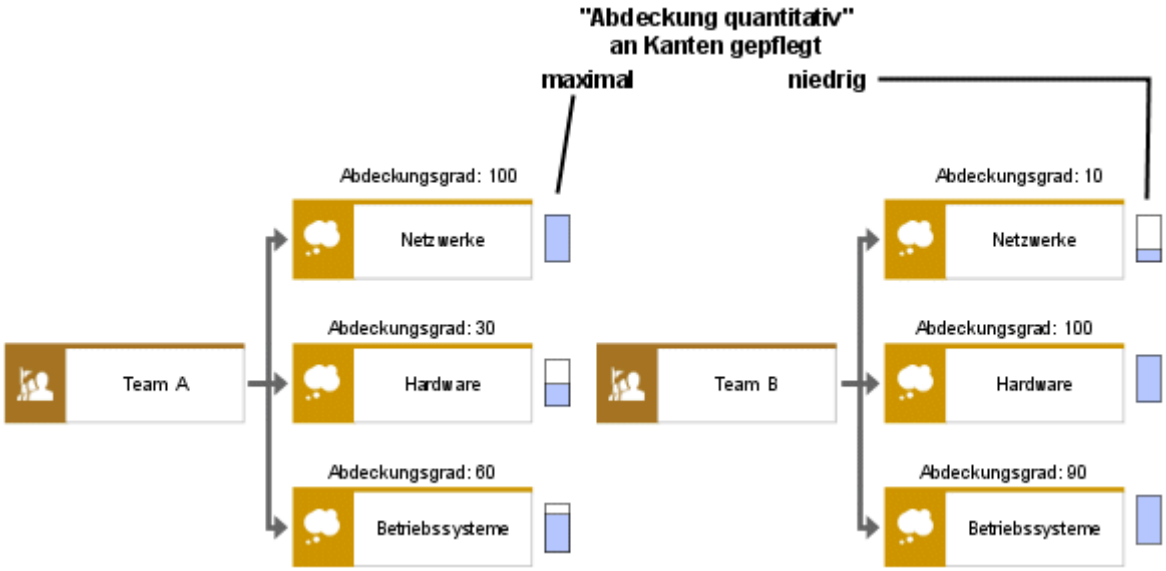


Abbildung 118: Wissenslandkarte - organisationseinheitenbezogen

Die Wissenslandkarte im Beispiel dieser Abbildung ist organisationseinheitenbezogen aufgebaut, d. h. für jede Organisationseinheit sind alle relevanten **Wissenskategorien** angegeben. Umgekehrt ist es natürlich auch möglich, die **Wissenseinheiten** als zentralen Betrachtungsgegenstand zu wählen und alle betreffenden Organisationseinheiten hinzu zu modellieren. Über die Navigationsmöglichkeiten von ARIS (Registerkarte **Beziehungen** des

Dialogs **Eigenschaften - Objekt**) lassen sich in beiden Fällen leicht die sonstigen Verbindungen einer Organisationseinheit oder einer Wissenskategorie finden. Häufig wird für Wissenslandkarten eine Matrixdarstellung verwendet. Diese kann über die spaltenförmige Anordnung mehrerer Ausprägungen der gleichen Wissenskategorie erreicht werden, wie dies in der nächsten Abbildung dargestellt ist. In diesem Beispiel wird der Name nur für die - ähnlich wie in der Kopfzeile einer Tabelle -oben dargestellten **Wissenskategorien** angezeigt, für die anderen Ausprägungen wurde der Name über die Attributplatzierung entfernt. Hier wird außerdem eine alternative optische Darstellungsart für unterschiedliche Abdeckungsgrade gezeigt, indem die **Wissenskategorien** unterschiedlich groß skaliert wurden.

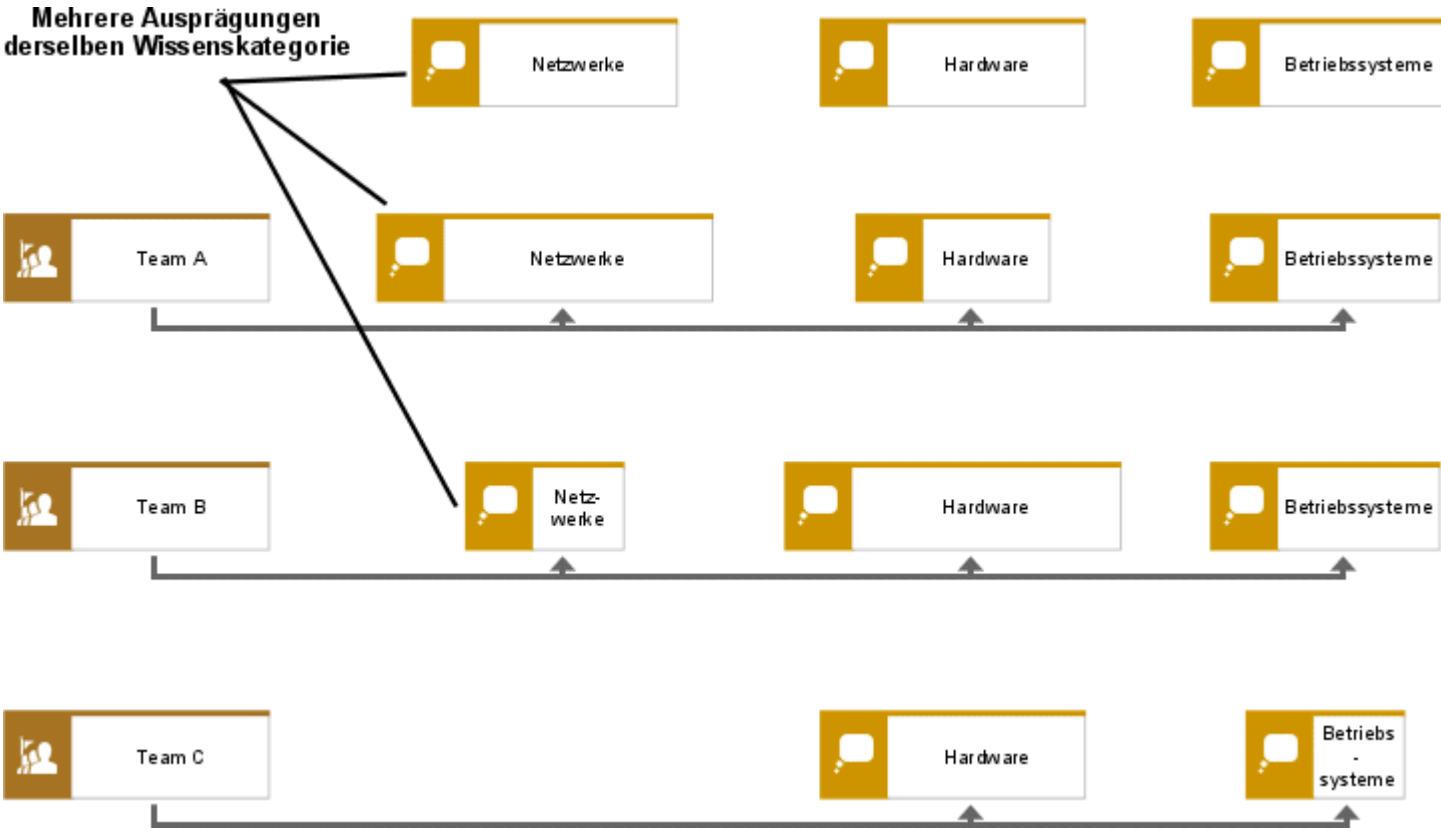


Abbildung 119: Wissenslandkarte - Matrixdarstellung

5.3.3 Darstellung der Wissensverarbeitung in Geschäftsprozessen

Die Nutzung und das Entstehen von Wissen in den Geschäftsprozessen des Unternehmens wird in den Modelltypen für die Geschäftsprozessdarstellung (EPK, EPK (Materialfluss)) modelliert. In diesen Modelltypen stehen nun auch die Objekttypen **Wissenskategorie** und **Dokumentiertes Wissen** zur Verfügung. Zu einer Funktion kann nun einerseits angegeben werden, welches (allgemeine oder dokumentierte) Wissen für ihre Durchführung benötigt, andererseits welches Wissen bei ihrer Durchführung entsteht bzw. dokumentiert wird. Mit

Hilfe einer derartigen Darstellung können Geschäftsprozesse im Hinblick auf ihre Wissensverarbeitung untersucht werden. Beispielsweise können nicht gedeckte Wissensbedarfe aufgedeckt werden. Ebenso kann z. B. das benötigte Qualifikationsprofil zur Durchführung einer Funktion ermittelt werden.

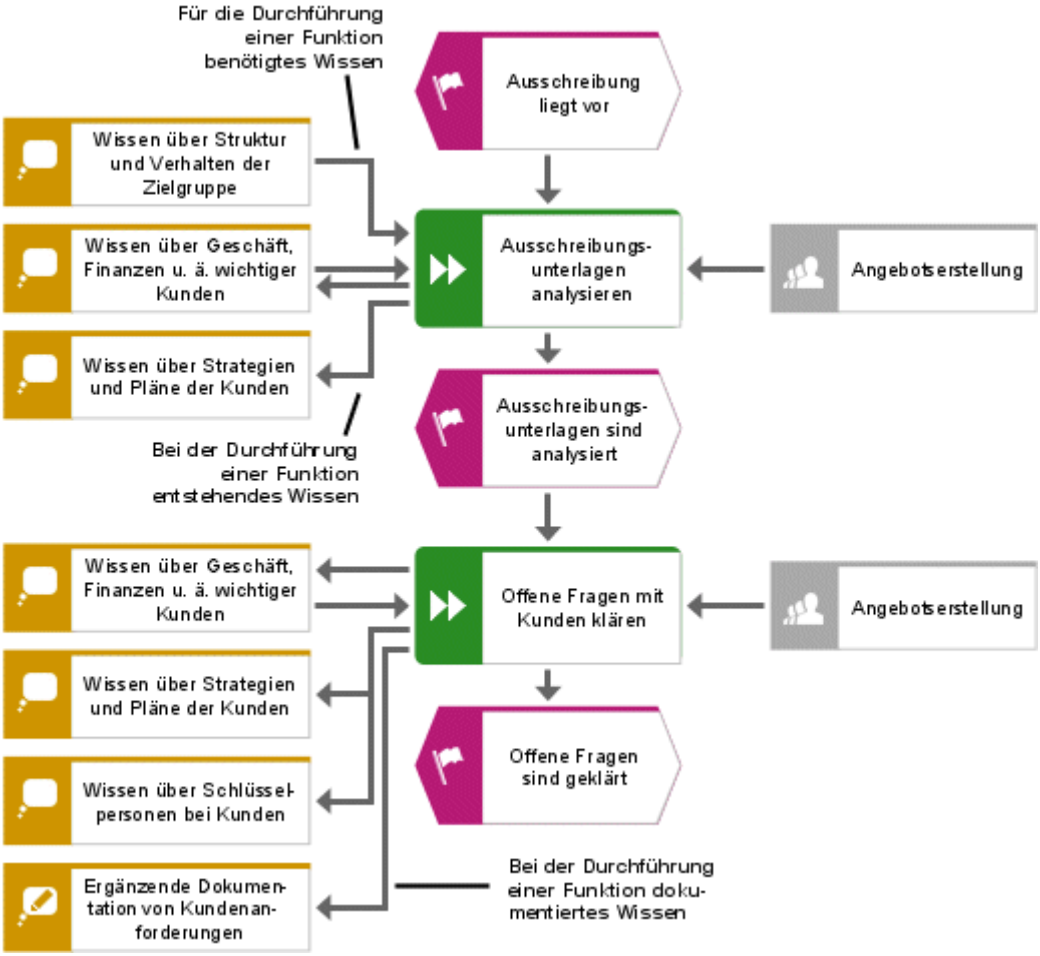


Abbildung 120: Wissensverarbeitung in einer EPK

6 Balanced Scorecard-Methode

6.1 Einleitung

Um im turbulenten Unternehmensumfeld und dem globalen Wettbewerb heutiger Märkte bestehen zu können, benötigen Unternehmen optimal gestaltete Geschäftsprozesse. Ebenso bedeutsam ist aber auch die Fähigkeit, schnell und, in Bezug auf die strategischen Unternehmensziele, auf neue Entwicklungen im Unternehmensumfeld reagieren zu können. Dazu sind effiziente Managementprozesse notwendig, die eine konsequente Realisierung der Unternehmensstrategien und der strategischen Ziele sowie deren Umsetzung im Tagesgeschäft anhand operativer Maßnahmen erlauben.

Viele der klassischen Managementansätze schaffen keine Verbindung zwischen der Formulierung von Unternehmensstrategien und deren Umsetzung durch strategieorientierte Maßnahmen sowie einer konsequenten Kontrolle der Zielerreichung strategischer Ziele.

Zusätzlich werden viele Unternehmen noch mit reinen Finanzkennzahlen geführt, deren Aussagefähigkeit jedoch begrenzt ist, da sie größtenteils vergangenheitsorientiert sind und deshalb nur wenig zukunftsrelevante Steuerungsinformationen liefern können. Erst die Berücksichtigung der Ursachen des finanziellen Erfolgs (Kunden, Prozesse, Innovationen usw.) ermöglichen es, frühzeitig Abweichungen bezüglich der Erreichung strategischer Ziele zu erkennen.

Der Balanced Scorecard-Ansatz stellt eine übersichtlich strukturierte und einfach zu implementierende Methodik zur Verfügung, mit der diese Schwächen ausgeglichen werden können.

6.2 Das Konzept der Balanced Scorecard

6.2.1 Kernaussagen des BSC-Ansatzes

Der Balanced Scorecard-Ansatz ist ein strategisches Managementsystem, das erstmals von Robert Kaplan und David Norton im Jahre 1992 (The Balanced Scorecard-Measures that drive Performance, Harvard Business Review Januar/Februar 1992) vorgestellt wurde. Es entstand als Weiterentwicklung aus den Ergebnissen einer Studie zum Thema **Performance-Measurement-Ansätze**. Im Rahmen der Studie wurde festgestellt, dass die rein auf monetäre Kennzahlen ausgerichteten Performance-Measurement-Systeme viele Unternehmen an wertschöpfenden Tätigkeiten eher hindern. Ausgehend von dieser Erkenntnis versuchten Kaplan und Norton in Zusammenarbeit mit innovativen Unternehmen ein Kennzahlensystem aufzubauen, das es ermöglichen sollte, die Umsetzung von Visionen und Strategien der Unternehmen möglichst optimal zu messen.

Beim Balanced Scorecard-Ansatz werden Kennzahlen verschiedenen Sichtweisen auf das Unternehmen zugeordnet (sog. Perspektiven). Diese beinhalten sowohl interne (z. B. Lern- und Entwicklungsperspektive, Prozessperspektive) als auch externe Performance-Perspektiven (z. B. Kundenperspektive, finanzwirtschaftliche Perspektive). Durch diese Anordnung der Kennzahlen wird eine gewisse Ausgewogenheit hinsichtlich kurzfristiger und langfristiger Ziele, monetärer und nicht-monetärer Kennzahlen, Früh- und Spätindikatoren sowie interner und externer Sichtweisen erreicht. Mit dem Einbezug von branchenspezifischen Kennzahlen wird zusätzlich noch eine Benchmarking-Komponente in das Konzept integriert.

Aus dem reinen Performance-Measurement-Ansatz entwickelte sich ein umfassendes Führungssystem, das eine zielorientierte Unternehmensführung enthält, ausgehend von der Unternehmensvision über die einzelnen Wettbewerbsstrategien bis hin zur Formulierung und Überwachung von Maßnahmen anhand ausgewogener Kennzahlen. Der Balanced Scorecard-Ansatz ist daher mehr als nur ein Kennzahlen-System zum Performance Measurement. Er unterstützt Unternehmen bei der Kommunikation und Umsetzung der Unternehmensstrategie und dem daraus resultierenden strategischen Lernprozess (double loop learning).

6.2.2 Der strategische Managementprozess und die Balanced Scorecard

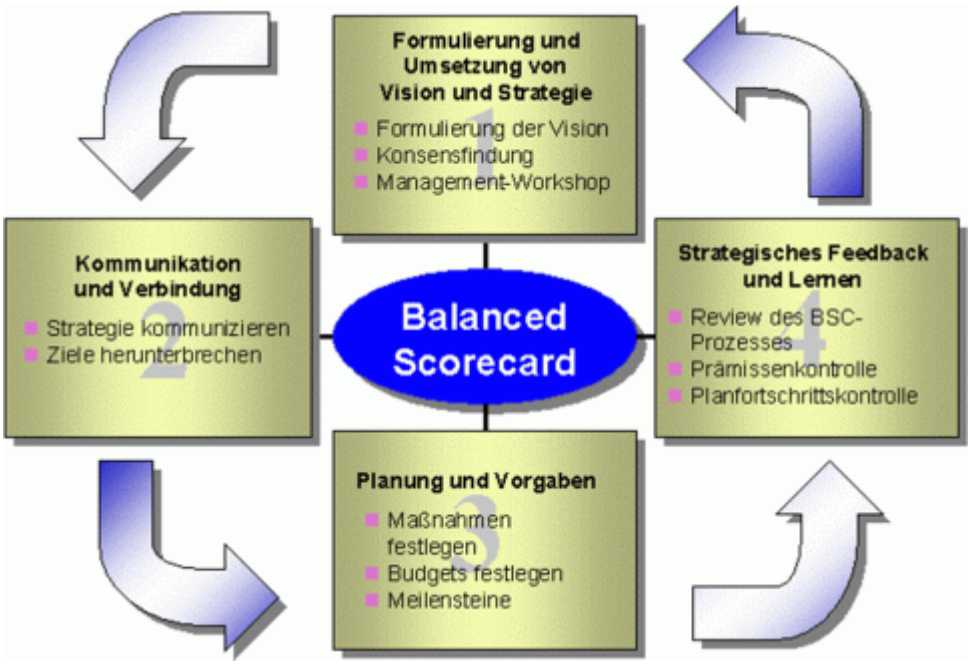


Abbildung 121: Die BSC als Handlungsrahmen für das strategische Management

Der strategische Managementprozess im Rahmen des Balanced Scorecard-Ansatz besteht aus zwei Phasen:

1. In der ersten Phase muss im Rahmen einer strategischen Analyse die Strategie des Unternehmens gefunden werden. Im Rahmen der strategischen Analyse werden alle wettbewerbsrelevanten Informationen erfasst. Ziel der Analyse ist es, Trends, Chancen und Risiken der Entwicklung des Unternehmensumfelds zu identifizieren und zu bewerten sowie unternehmenseigene Kernkompetenzen zu erfassen. Am Ende dieser Phase wird die individuelle Unternehmensstrategie definiert.
2. Die Umsetzung der Unternehmensstrategie findet in der zweiten Phase statt. Für diese Phase bietet die BSC-Methodik Unterstützungspotenzial. Die Unternehmensstrategie kann anhand von Teilstrategien (z. B. geschäftsfeldspezifische Strategien) verfeinert werden. Daraus leiten sich wiederum strategische Ziele ab. Die strategischen Ziele werden anhand von Vorgaben für bestimmte Maßgrößen konkretisiert. Wie die Zielvorgaben erreicht werden sollen, wird durch ein Aktionsprogramm in Form von Handlungsanweisungen bestimmt. Das Aktionsprogramm wird in der Unternehmensplanung auf unterschiedliche Unternehmensbereiche bzw. Abteilungen „heruntergebrochen“. Durch eine Budgetierung können Aktionen weiter konkretisiert werden. Mit Hilfe einzelner strategiebezogener Scorecards wird die Zielerreichung anhand der generierten Kennzahlen gemessen. Aufbauend auf den definierten Scorecards findet ein Review-Prozess statt, indem auf der Grundlage möglicher Abweichungen weitere Maßnahmen festgelegt werden oder die Strategie „überarbeitet“ wird.

6.2.2.1 Formulierung und Übersetzen der Vision und der Strategie

Die Strategie eines Unternehmens resultiert aus der Unternehmensvision.

Die Vision stellt ein übergeordnetes Leitbild für das betrachtete Unternehmen dar, sie ist als pauschale, aber sehr plakative Aussage wenig konkret, gilt aber als oberste Handlungsmaxime und stellt so eine Art Kurzfassung der Unternehmensphilosophie dar.

Für unterschiedliche strategische Geschäftseinheiten eines Unternehmens werden entsprechende Strategien festgelegt. Diese Strategien müssen sich am Erfolgsziel des Unternehmens orientieren. Daher ist es notwendig, dass zu Beginn der Einführung einer BSC das Top-Management des Unternehmens in einem Workshop die Vision und die daraus resultierenden Strategien für die strategischen Geschäftseinheiten bestimmt. In den meisten Fällen wird das Ergebnis eine finanzielle Zielsetzung (finanzielle Perspektive) sein. Der Schwerpunkt kann unterschiedlich sein: Mögliche Ziele sind bestimmte Werte für Kapitalrentabilitäten, ROI, Shareholder Value, Umsatzerlöse oder Cash Flow. Die finanziellen Ziele müssen durch ein spezifisches Verhalten auf dem zu bearbeitenden Markt realisiert werden (Kundenperspektive) Daher werden nach der Zielsetzung die entsprechenden Markt- und Kundensegemente ausgewählt. Auch für diese Kundenperspektive werden strategische

Ziele definiert und die relevanten Kennzahlen erfasst. Sie können sich z. B. auf Marktanteile oder Wachstumsraten bestimmter Kundensegmente beziehen.

Zur Realisierung spezifischer marktorientierter Strategien sind entsprechende Unternehmensressourcen notwendig. Bei der Festlegung einer Balanced Scorecard werden die Ressourcen auf zwei Arten erfasst:

1. Nachdem die Finanz- und Kundenziele festgelegt sind, werden die bedeutenden Geschäftsprozesse (Prozessperspektive) betrachtet und Zielsetzungen festgelegt sowie Maßnahmen und Kennzahlen generiert. Im Mittelpunkt der Betrachtung stehen hier in der Regel Prozesszeiten und -kosten.
2. Im Rahmen der so genannten Lern- und Entwicklungsperspektive werden aus den strategischen Zielvorgaben der Finanzperspektive, Kundenperspektive und Prozessperspektive die strategischen Ziele für Personalentwicklung, Informationstechnologie und Innovationen abgeleitet.

Alle definierten strategischen Ziele stehen durch diese Vorgehensweise bei der Definition einer Balanced Scorecard in einem gegenseitigen Wirkungsgefüge, das als Ursache-Wirkungs-Kette beschrieben wird.

Wichtig für diesen ersten Schritt ist, dass auf Managementebene ein allgemeiner Konsens über die Vision und die abgeleiteten Strategien und strategischen Zielen gefunden wird.

6.2.2.2 Die Standardperspektiven einer Balanced Scorecard

Kaplan und Norton schlagen vier Standardperspektiven für den Aufbau einer Balanced Scorecard vor:

1. Finanzielle Perspektive - Erwartungen der Shareholder: „Wie wirkt sich die Strategie auf die Finanzen aus?“
2. Kundenperspektive - Erwartungen der Kunden: „Wie positionieren wir uns auf den Zielmärkten?“
3. Prozessperspektive - Anforderungen an Prozesse: „Welche Prozesse sind strategisch bedeutend?“

4. Lernperspektive - Anforderungen an das organisatorische Lernen und Innovationen:
   „Wie entwickeln wir uns zu einer lernenden Organisation? Wie fördern wir das Wachstum?“

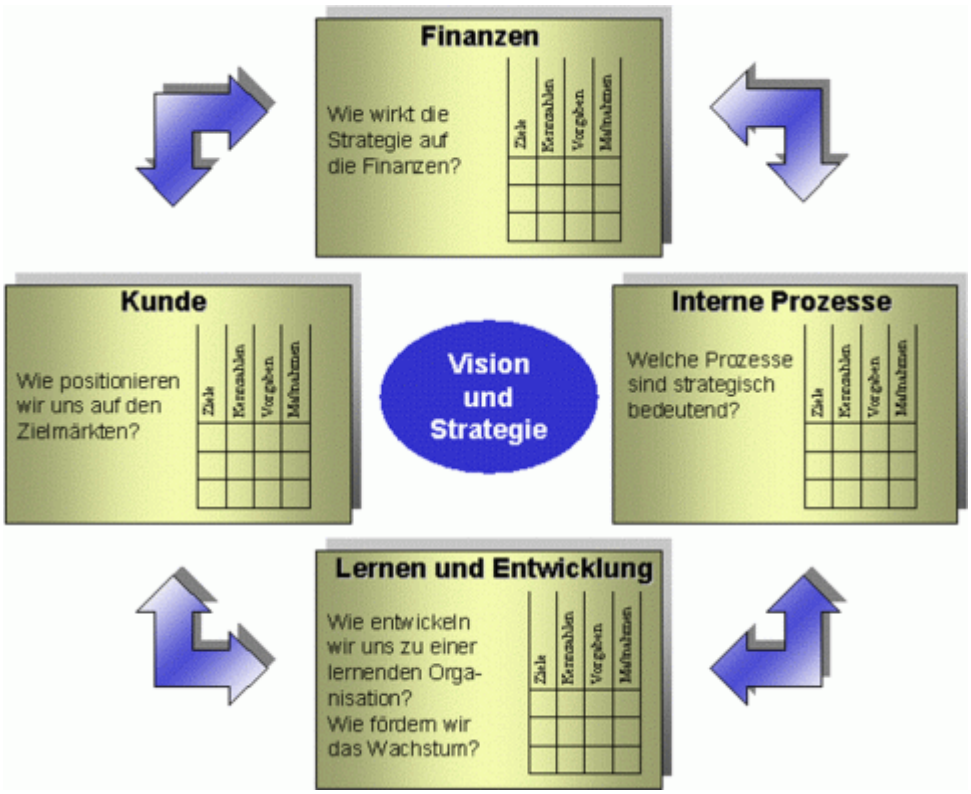


Abbildung 122: Die Perspektiven der BSC

Die Standardperspektiven bilden eine implizite Logik, die bei der Strategieumsetzung und der Formulierung von konkreten Ursache-Wirkungsbeziehungen hilfreich ist. Es besteht jedoch grundsätzlich die Möglichkeit, weitere für ein Unternehmen relevante Perspektiven zu definieren (z. B. Umweltperspektive), die strategierelevant sind und deren Ziele und Kennzahlen in Beziehung zu den anderen Perspektiven stehen können.

6.2.2.3 Ursache-Wirkungs-Kette

Die Ursache-Wirkungs-Kette verdeutlicht die Ursache-Wirkungsbeziehungen zwischen den einzelnen Teilzielen einer Strategie, also über die definierten Perspektiven der Kennzahlen hinweg. D. h., sie beschreibt, wie die Ziele der Perspektiven erreicht werden können. Dabei bilden die Perspektiven selbst sozusagen die Rahmenbedingung für die Ursache-Wirkungs-Kette („Kraftfeld des Unternehmens“). Man geht davon aus, dass Verbesserungen im Umfeld der Lern- und Entwicklungsperspektive einen direkten und positiven Einfluss auf Ziele und Kennzahlen der internen Prozessperspektive haben. Weiterhin wirken sich die Entwicklungen der Prozessperspektive auch positiv auf die Ziele und Kennzahlen der Kundenperspektive aus, was letztlich zu einer Verbesserung der Finanzziele führt.

Während des Aufbaus eines unternehmensweiten Balanced Scorecard-Systems werden für jede Strategie Ziele und Kennzahlen definiert und deren Wirkungsbeziehungen innerhalb der Perspektiven sowie perspektivenübergreifend dargestellt. Man geht von den Zielen und Kennzahlen der finanziellen Perspektive aus und sucht entsprechende „positive Verursacher“ (Leistungstreiber) in der Kundenperspektive. Diese werden dann über die interne Prozessperspektive bis auf die Lern- und Entwicklungsperspektive heruntergebrochen. Diese Vorgehensweise ermöglicht es, Strategien eines Unternehmens oder einer Division in Teilziele und Kennzahlen aufzuspalten und auch entsprechende operative Maßnahmen abzuleiten.

Durch die Festlegung der Ursache-Wirkungs-Kette und die Definition der Kennzahlen ergibt sich implizit ein Gleichgewicht zwischen Frühindikatoren und Spätindikatoren.

6.2.2.4 Definition der Früh- und Spätindikatoren

Durch den Aufbau von Ursache-Wirkungs-Ketten wird auch ein zeitlicher Zusammenhang zwischen den einzelnen Zielen und Kennzahlen vorgegeben. So gehören den Perspektiven **Lern- und Entwicklungsperspektive** und **Prozessperspektive** in der Regel Leistungstreiber, sog. Frühindikatoren, an, die den Wirkungen der Kundenperspektive und der finanziellen Perspektive zeitlich vorangehen. In der Kunden- und finanziellen Perspektive sind die meisten der Kennzahlen Ergebniskennzahlen, die den Geschäftserfolg messen; sie sind als sog. Spätindikatoren in die Vergangenheit gerichtet.

Es sollten in jeder Perspektive der Balanced Scorecard sowohl Früh- als auch Spätindikatoren definiert sein, damit der Zusammenhang zwischen den Maßnahmen und dem Erreichen der Ziele dargestellt werden kann. Durch die Frühindikatoren können frühzeitig Zielabweichungen erkannt werden.

6.2.2.5 Kommunikation und Ableiten weiterer Scorecards

Die Wirksamkeit einer Unternehmensstrategie und ihrer Umsetzung anhand des Balanced Scorecard-Konzepts ist unter anderem abhängig von der unternehmensweiten Bekanntheit und Akzeptanz der Unternehmens-Scorecard. Aus diesem Grund ist es überaus wichtig, dass die Unternehmensstrategien und die Unternehmens-Scorecard durch ein umfassendes Kommunikationsprogramm auf allen Hierarchiestufen des Unternehmens publiziert werden.

Aus der übergeordneten Unternehmensstrategie (Unternehmens-Scorecard) werden mit Hilfe eines Top-Down-Ansatzes Zielvorgaben für die unteren Hierarchiestufen des Unternehmens abgeleitet sowie übergeordnete strategische Ziele an die bereichsspezifischen Ziele angepasst. Für sie werden ebenfalls Kennzahlen generiert, die den Grad der Zielerreichung messen. Um die Ziele erreichen zu können, sind in den Funktionsbereichen eines Unternehmens verschiedene Maßnahmen notwendig, die ebenfalls in den Scorecards der darunter liegenden Hierarchiestufen erfasst werden. Alle Zielvorgaben müssen auf der Basis langfristiger strategischer Überlegungen gemacht werden und dürfen nicht auf kurzfristige Ergebnisse ausgerichtet sein, die nicht im Einklang mit der übergeordneten Strategie stehen. Ein Beispiel für ein kurzfristiges Ergebnis wäre eine Kostenreduzierungen aufgrund einer kurzfristigen Erhöhung der Produktionsintensität.

6.2.2.6 Planung und Vorgaben

Die BSC kann dazu genutzt werden, den Prozess der Strategieumsetzung auf die unterschiedlichen Hierarchiestufen der Unternehmung mit dem Budgetierungsprozess zu verknüpfen. Zweck der Kopplung ist die Ausrichtung aller Ressourcen auf die Unternehmensstrategie.

In vier Schritten wird die BSC in den langfristigen strategischen Planungs- und Budgetierungsprozess integriert.

1. Herausfordernde Zielvorgaben werden gemacht, die allgemein kommuniziert und von den Mitarbeitern akzeptiert werden. Durch die Ursache-Wirkungsbeziehungen zwischen den Kennzahlen werden die relevanten Kennzahlen identifiziert, die auf die Ergebniswerte der Kunden- und Finanzperspektive bezogen sind.
2. Die strategischen Maßnahmen müssen genau an den Kennzahlen ansetzen, deren aktueller Wert sich am deutlichsten von den Vorgabewerten unterscheidet. Dieses Vorgehen bündelt langfristig die Kapitalinvestitionen und Aktionsprogramme konsequent auf strategisch bedeutsame Zielwerte.
3. Kritische, unternehmensweite Initiativen werden identifiziert, sodass Synergieeffekte mit den strategischen Zielen realisiert werden können.

4. Die Planwerte für die Ergebniskennzahlen werden mit der Unternehmensplanung verknüpft, indem sie an Budgetvorgaben mit unterschiedlichen Fristigkeiten (5-, 3- und 1-Jahresplanung) gekoppelt werden.

6.2.2.7 Strategisches Lernen und Feedback

Das Balanced Scorecard-Konzept trägt dazu bei, dass im Unternehmen strategisches Lernen stattfindet und Feedback zur Strategieerreichung gegeben wird.

Das strategische Lernen bezieht sich auf jeden einzelnen Mitarbeiter sowie auf das gesamte Unternehmen als lernende Organisation. Mitarbeiter sollen die Strategie kennen und selbstständig dafür sorgen, dass sie ihre Handlungen an ihr ausrichten. Über einen definierten Feedback-Prozess werden Leistungsdaten gesammelt, an denen die Strategieerreichung gemessen wird. Auf diesem Weg findet eine Prämissenkontrolle statt, indem Hypothesen über die vorab getroffenen Ursache-Wirkungsbeziehungen strategischer Ziele geprüft werden.

Auf diesem Weg kommt es zum so genannten Double-Loop-Lernen, wenn die strategischen Zielvorgaben in einem Bottom-Up-Prozess einer kritischen Revision unterzogen und danach neue Ursache-Wirkungsbeziehungen generiert werden, die aufgrund einer ständig erweiterten Informationsbasis zu einer effizienteren Strategieverfolgung führen.

Dem Feedback kommt dabei eine große Bedeutung zu: es repräsentiert den Prozess, der dem „Herunterbrechen“ von BSCs entgegen läuft. Das Feedback beinhaltet nicht nur Rückmeldungen von Kennzahlen in Form von Planfortschrittszahlen, sondern auch Vorschlägen für weitere Maßnahmen sowie neue Wirkzusammenhänge. Es weist damit auf eventuelle Inkonsistenzen in der Strategieverfolgung hin.

6.2.3 Vorteile und Nutzen der Balanced Scorecard

Der Vorteil der Balanced Scorecard-Methode liegt in der konsequenten strategieorientierten Unternehmensführung, welche die untersten Unternehmensebenen und die strategiekonforme Ableitung operativer Maßnahmen auf diesen Unternehmensebenen umfasst. Alle Unternehmensressourcen und alle Mitarbeiter werden auf die Unternehmensstrategie ausgerichtet. Die Unternehmensstrategie wird durch Balanced Scorecards konsequent kommuniziert und kontrolliert.

Die Balanced Scorecard-Methode überwindet die Nachteile traditioneller Kennzahlensysteme zur Unternehmenssteuerung:

- Keine Beschränkung auf finanzielle Kennzahlen
- Keine reine Vergangenheitsorientierung
- Berücksichtigung auch qualitativer Daten
- Komplexitätsreduktion und Konsensfindung
- Strategische Ziele als Ausgangspunkt des Balanced Scorecard-Ansatzes
- Orientierung an Führungsengpässen, nicht an vorhandenen Daten

- Beschleunigung notwendiger Veränderungen

Die Balanced Scorecard-Methode eignet sich hervorragend zur strategieorientierten Unternehmensführung, da sie die folgenden Barrieren überwindet:

- Konkretisierungsbarriere: Bei klassischen Managementansätzen sind Vision und Strategie Wunschformulierungen und die Ableitung konkreter (operativer) Aktionsformulierungen bleibt offen.
- Visionsbarriere: Die Strategie wird oftmals nicht von jenen Mitarbeitern verstanden, die sie umsetzen müssen.
- Commitment-Barriere: Strategien sind nicht mit konkreten Abteilungs- oder Individualzielen verbunden.
- Implementierungsbarriere: Das Berichtswesen ist an operativ-monetären Zielen, statt an strategischen Zielen ausgerichtet.
- Operative Barriere: Der Budgetierungsprozess ist vom strategischen Planungsprozess getrennt.

6.3 Entwicklung einer Balanced Scorecard mit ARIS BSC

6.3.1 Begriffe und Abkürzungen

Vision: Eine Vision gibt die zukünftige strategische Stoßrichtung und den Auftrag eines Unternehmens an und wird durch verschiedene Strategien realisiert.

Strategie: Ausgehend von der Vision werden Strategien entwickelt. Die Strategien können in spezifische Teilstrategien aufgespaltet werden.

Strategisches Ziel: Jede Strategie besteht aus einer bestimmten Anzahl strategischer Ziele. Die strategischen Ziele werden in der Regel vom verantwortlichen Management in Workshops festgelegt. Zwischen den Zielen bestehen eine Ursache-Wirkungsbeziehung und eine zeitliche Logik.

Erfolgsfaktor: Ein Erfolgsfaktor ist ein für die Geschäftsabwicklung des Unternehmens erfolgskritischer Einflussfaktor, der in einem Wirkungsgefüge zu anderen Erfolgsfaktoren stehen kann und diese in eine bestimmte Wirkungsrichtung mit einem bestimmten Wirkungsgrad beeinflusst.

Sicht/Perspektive: Eine Perspektive ist die Konkretisierung einer bestimmten Sicht auf das Unternehmen. Grundsätzlich sollte die Auswahl von Perspektiven folgende Sichtweisen auf ein Unternehmen beinhalten: humanorientierte Sicht, interne Sicht, prozessorientierte Sicht und externe Sicht.

Kennzahl: Jedem strategischen Ziel oder Erfolgsfaktor werden Kennzahlen zur Leistungsmessung und Zielerreichung zugeordnet.

Istwert: Der aktuelle Wert den eine Kennzahl, ein strategisches Ziel oder ein Erfolgsfaktor annehmen kann. Aus dem Istwert kann durch einen Plan-Ist-Vergleich der aktuelle Zielerreichungsgrad abgeleitet werden.

Plan-Wert: Der für diesen Zeitraum geplante und zu erreichende Planwert einer Kennzahl, eines strategischen Zieles oder eines Erfolgsfaktors. Bei strategischen Zielen handelt es sich in der Regel um einen Prozentwert, der sich aus der gewichteten Zielerreichung einzelner Kennzahlen ergibt. Der Planwert kann auf einzelne Perioden heruntergebrochen werden und kann dann auch periodisch bedingte Schwankungen einbeziehen. Der Planwert entspricht einem Zielerreichungsgrad von 100 %.

Zielwert: Der für einen zukünftigen Zielzeitraum geplante und zu erreichende Wert. Er wird in der Regel zum Planwert, wenn der Zielzeitraum erreicht ist.

Minimalwert: Der kleinste Wert, den ein Ziel, ein kritischer Erfolgsfaktor oder eine Kennzahl annehmen kann. In ARIS wird als Vorgabewert für den Minimalwert 0 vorgegeben. Er kann auch geändert werden.

Maximalwert Der größte Wert (Obergrenze), den ein strategisches Ziel, ein Erfolgsfaktor oder eine Kennzahl annehmen kann. Er wird in ARIS in der Regel nur dazu verwendet, um die Darstellung mehrerer Kennzahlen zu normieren und vergleichbar zu machen.

Warngrenze: Die Warngrenze entspricht dem Planwert, d. h., der Planwert entspricht der Schwelle, an welcher der Wert einer Kennzahl, eines strategischen Ziels oder eines Erfolgsfaktors unter eine gewünschte Vorgabe fällt.

Toleranzbereich: Der Toleranzbereich ist die negative Abweichung vom Planwert, die für ein strategisches Ziel, eine Kennzahl oder einen Erfolgsfaktor noch als tolerierbar gilt.

Alarmgrenze: Die Alarmgrenze entspricht dem Planwert abzüglich des Toleranzbereichs. Alle Werte eines strategischen Ziels, einer Kennzahl oder eines Erfolgsfaktors, die unterhalb der Alarmgrenze liegen, werden nicht mehr toleriert.

Maßnahme: Eine Maßnahme wirkt auf ein Ziel oder auf mehrere strategische Ziele. Maßnahmen haben in der Regel einen Verantwortlichen, ein Anfangs- und einen Endzeitpunkt, Ressourcen usw.

Indikatortyp: Eine Kennzahl kann vom Indikatortyp **Frühindikator** oder vom Indikatortyp **Spätindikator** sein. Ein Frühindikator misst einen Leistungstreiber und ist eine in die Zukunft gerichtete Kennzahl. Ein Spätindikator misst ein Ergebnis (Ergebniskennzahl) und ist retrospektiv. Finanzwirtschaftliche Ziele sind in der Regel Spätindikatoren, Ziele der Prozess-, Lern- und Entwicklungsperspektive sind zunehmend Frühindikatoren. Zusätzlich sollten auch Spätindikatoren innerhalb der Perspektiven erfasst werden, um die Ursache-Wirkungsbeziehung zwischen den Kennzahlen darzustellen.

Datenherkunft Jede Kennzahl besitzt eine Datenquelle, aus der sie in das Balanced Scorecard-System extrahiert wird.

6.3.2 Aufbau der Balanced Scorecards mit ARIS BSC

6.3.2.1 Perspektiven festlegen

Zu Beginn eines Balanced Scorecard-Projektes sollten im Rahmen der strategischen Planung die Perspektiven festgelegt werden. Diese Perspektiven werden für alle Scorecards eines Unternehmens verwendet.

Dafür steht der Objekttyp **Perspektive** zur Verfügung. Perspektiven können im BSC-Ursache-Wirkungsdiagramm modelliert werden.

Perspektiven werden mit Hilfe der Kante **wird beeinflusst von** verbunden. Die Festlegung der Reihenfolge ist methodisch nicht notwendig. Durch die logische Festlegung der Perspektivenstruktur (d. h. die Reihenfolge der einzelnen Perspektiven) wird allerdings eine automatische Modellgenerierung wesentlich vereinfacht und der logische Aufbau der Ursache-Wirkungs-Kette verdeutlicht.

6.3.2.2 Struktur des Balanced Scorecard-Systems festlegen

Den Rahmen für den Aufbau des Balanced Scorecard-Systems bilden folgende Modelle:

- Organigramm

  Ein Balanced Scorecard-System kann in Anlehnung an die Organisationsstruktur eines Unternehmens aufgebaut werden. Dabei können den Objekten des Organigramms beliebig viele BSC-Ursache-Wirkungsdiagramme hinterlegt werden (z.B. zur Variantenbildung, Historienverwaltung usw.). Diese Hinterlegung ordnet die für die Strategieumsetzung erforderlichen Ziele den entsprechenden Organisationseinheiten zu. Das Balanced Scorecard-System eines Unternehmens kann somit durch die Orientierung am Organigramm von der obersten Unternehmensebene bis auf die Ebene einzelner Abteilungen oder Mitarbeiter heruntergebrochen werden.

- Gliederungsmodell

  Möchte ein Unternehmen den Aufbau des Balanced Scorecard-Systems nicht am Organigramm, sondern beispielsweise anhand von strategischen Geschäftsfeldern festlegen, bietet sich das Gliederungsmodell als Ausgangsmodell für die Balanced Scorecard-Struktur an. In diesem Fall können beliebig viele BSC-Ursache-Wirkungsdiagramme einem Gliederungsobjekt hinterlegt werden.

- Wertschöpfungskettendiagramm oder Funktionsbaum

  Da sich die Balanced Scorecard als Instrument zum Performance Management und Performance Measurement von Unternehmen versteht, kann in ARIS das Balanced Scorecard-System auch wertschöpfungsorientiert aufgebaut werden. Dazu können das Wertschöpfungskettendiagramm oder der Funktionsbaum genutzt werden.

6.3.2.3 Ursache-Wirkungs-Beziehungen festlegen

Den Objekten des Organigramms, den Funktionen oder den Gliederungselementen im Gliederungsmodell, die den Aufbau des unternehmensweiten Balanced Scorecard-Systems festlegen, werden BSC-Ursache-Wirkungsdiagramme hinterlegt.

Im BSC-Ursache-Wirkungsdiagramm werden die zur Strategieumsetzung festzulegenden strategischen Ziele und Erfolgsfaktoren definiert und deren gegenseitige Beeinflussung über die verschiedenen Perspektiven der Balanced Scorecard dargestellt. Dazu werden zuerst die einzelnen Perspektiven als Objekte in ARIS angelegt, die in allen Ursache-Wirkungsketten des Unternehmens genutzt werden.

Beim BSC-Ursache-Wirkungsdiagramm handelt es sich um ein Lane-Modell, das in der Spalte **relevante Perspektiven** die Ablage der Objekttypen **Perspektive** und in der Zeile **Strategie** die Ablage der Objekttypen **Strategie** erlaubt. In den Ursache-Wirkung-Spalten können die Objekttypen **Ziel**, **Erfolgsfaktor** und **Kennzahl** modelliert werden. In der Ursache-Wirkung-Spalte werden die Wirkungsstärke der strategischen Ziele und ihre Erfolgsfaktoren beschrieben. Die jeweiligen Pfeile werden mit zunehmender Einflussstärke dicker dargestellt.

Im BSC-Ursache-Wirkungsdiagramm können weiterhin Erfolgsfaktoren für eine Balanced Scorecard definiert werden. Durch die Verwendung des Objekttyps **Erfolgsfaktoren** lassen sich vom Unternehmen nicht direkt beeinflussbare Faktoren und deren Kennzahlen in die Balanced Scorecard mit aufnehmen, z. B. Marktwachstum.

Das BSC-Ursache-Wirkungsdiagramm finden Sie auf der Fachkonzeptebene der Steuerungssicht.

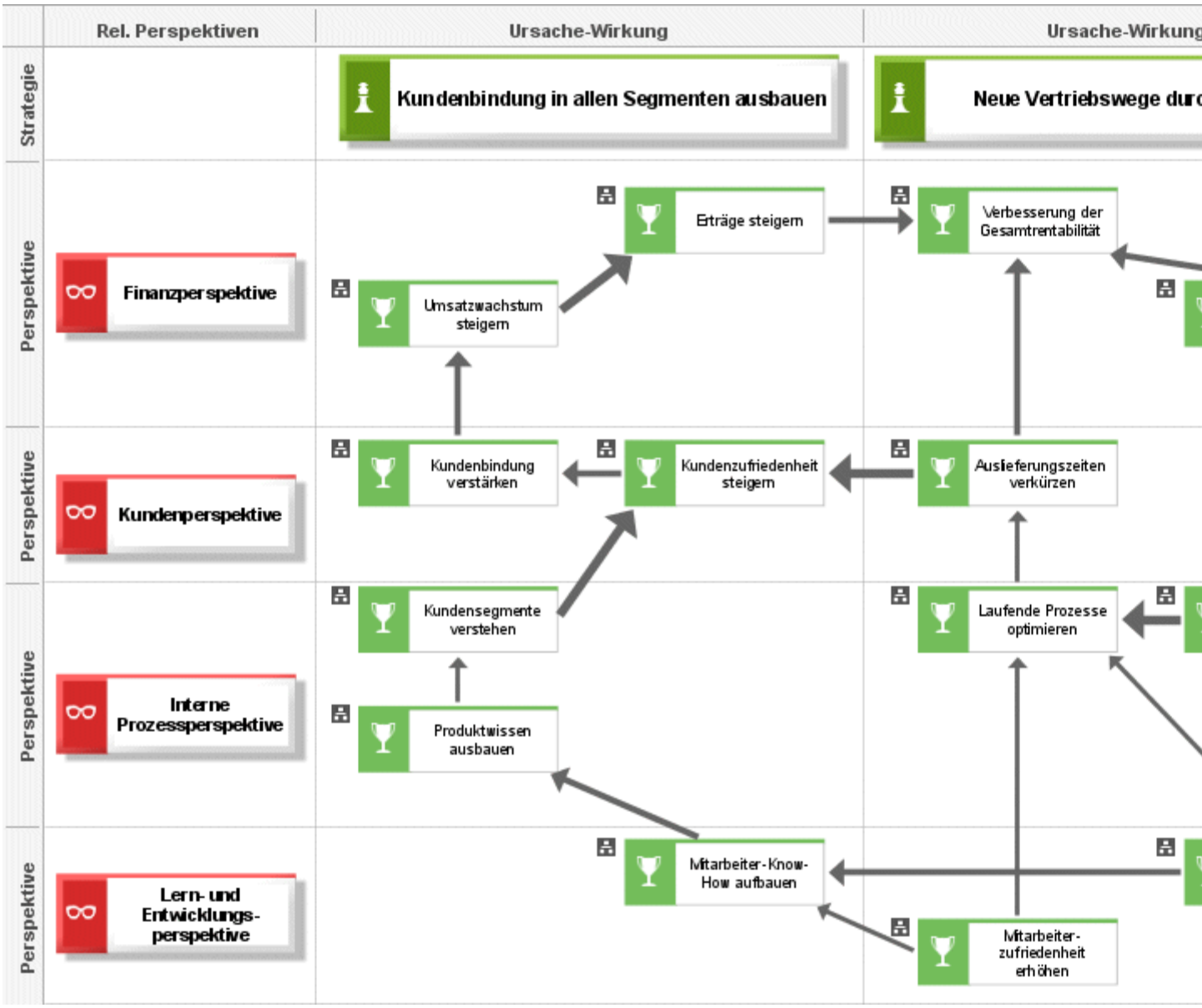
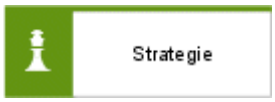
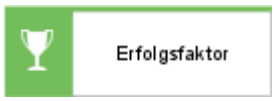


Abbildung 123: BSC-Ursache-Wirkungsdiagramm

Die Objekttypen **Ziel** und **Erfolgsfaktor** in der Modellierungsspalte des BSC-Ursache-Wirkungsdiagramms können ausschließlich mit einem BSC-Kennzahlenzuordnungsdiagramm hinterlegt werden.

Folgende Symbole werden im BSC-Ursache-Wirkungsdiagramm verwendet:

Symbol	Objekttyp
	Perspektive
	Strategie
	Ziel
	Erfolgsfaktor
	Kennzahlinstanz

6.3.2.4 Maßnahmen und Kennzahlen zur Zielkontrolle festlegen

Nach der Definition der strategischen Ziele und der Erfolgsfaktoren im BSC-Ursache-Wirkungsdiagramm wird den einzelnen Objekten je ein BSC-Kennzahlenzuordnungsdiagramm hinterlegt. Darin werden den Zielen und Erfolgsfaktoren die Kennzahlen zugeordnet, die als Maßstab für das jeweilige Ziel oder den Erfolgsfaktor dienen sollen. Diese Kennzahlen werden anschließend ermittelt und kanalisiert. Werden einem Objekt mehrere Kennzahlen zugeordnet, kann mit der Kante des Typs **wird gemessen durch/misst** eine Gewichtung der jeweiligen Kennzahl festgelegt werden. Diese gibt an, mit welchem Gewicht die Kennzahl in den Zielerreichungsgrad des Zieles oder Erfolgsfaktors eingeht. Zur erfolgreichen Ausführung der Analyseskripte von ARIS BSC muss das Attribut **Gewichtung** bei allen Kanten des Typs **wird gemessen durch/misst** gepflegt sein.

Neben Kennzahlen können den Zielen oder Erfolgsfaktoren im BSC-Kennzahlenzuordnungsdiagramm auch Objekte für die Definition der Datenherkunft zugeordnet werden, z. B. Entitytyp, Datei, Dokument, Datenbank, Datenträger, etc. Durch diese Zuordnung wird eine Verbindung zur Data Warehouse-Methode von ARIS geschaffen. Hier kann genau definiert werden, mit welchen Datenelementen der Data Warehouse-Methode die jeweilige Balanced Scorecard-Kennzahl verbunden ist.

Aufgrund der Möglichkeit, mit dem Kantentyp **wirkt auf/wird beeinflusst von** auch das Wirkungsgefüge von Kennzahlen zu beschreiben, können die Wirkungsweisen von Frühindikatoren auf Spätindikatoren im BSC-Kennzahlenzuordnungsdiagramm beschrieben werden. Wir empfehlen dafür ein zusätzliches BSC-Kennzahlenzuordnungsdiagramm zu entwerfen, in dem nicht die Zuordnung einzelner Kennzahlen zu Zielen oder Erfolgsfaktoren zueinander im Vordergrund steht, sondern das Zusammenführen und die Definition der genutzten Kennzahlen aus den anderen BSC-Kennzahlenzuordnungsdiagrammen und die Beschreibung ihrer Wirkungen.

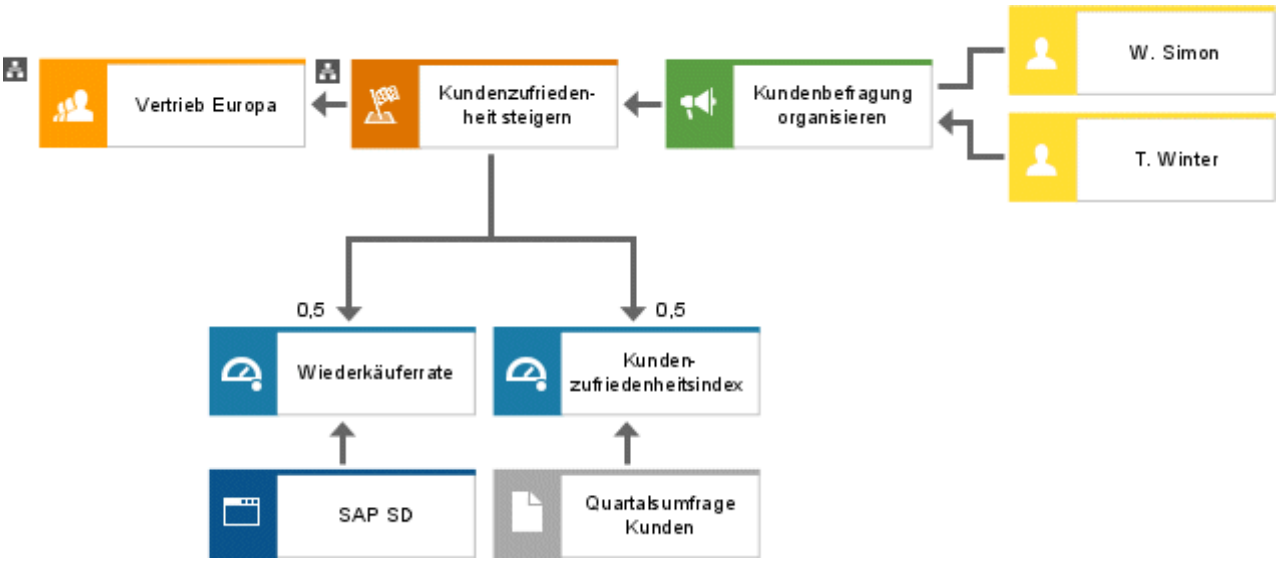
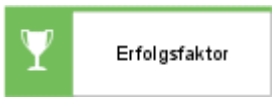
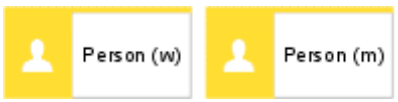
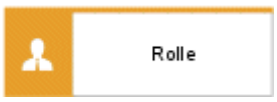


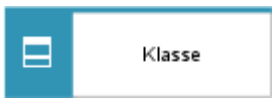
Abbildung 124: BSC-Kennzahlenzuordnungsdiagramm

Das BSC-Kennzahlenzuordnungsdiagramm finden Sie auf der Fachkonzeptebene der Prozesssicht.

Folgende Symbole werden im BSC-Kennzahlenzuordnungsdiagramm verwendet:

Symbol	Objekttyp
	Ziel
	Erfolgsfaktor
	Kennzahlinstanz
	Aufgabe

Symbol	Objekttyp
	Organisationseinheit
	Organisationseinheitstyp
	Stelle
	Person
	Rolle
	Gruppe
	Anwendungssystemtyp
	Fachbegriff
	Entitytyp
	Beziehungstyp
	ERM-Attribut
	Cluster/Datenmodell
	Wissenskategorie

Symbol	Objekttyp
	Dokumentiertes Wissen
	Klasse
	Informationsträger

6.3.2.5 Kennzahlen und deren Zusammenhänge beschreiben

Nach der Festlegung der Kennzahlen zur Messung strategischer Ziele und kritischer Erfolgsfaktoren können diese Kennzahlen und ihre Zusammensetzung durch den Modelltyp **Kennzahlenbaum** detaillierter definiert werden.

In einem Kennzahlenbaum werden verschiedenen Kennzahlen mit Hilfe des Kantentyps **beeinflusst** hierarchisch strukturiert. Für diese Kanten kann das Attribut **Gewichtung** gepflegt werden, sodass innerhalb eines Kennzahlenbaums die Berechnung einer Gesamtkennzahl anhand von Gewichtungen möglich wird. Der Kennzahlenbaum wird der Kennzahlinstanz hinterlegt, die die Gesamtkennzahl repräsentiert. Die Hinterlegung wird durch die BSC-Auswertungen **Management View erzeugen** und **Plan-Ist-Vergleich durchführen** berücksichtigt.

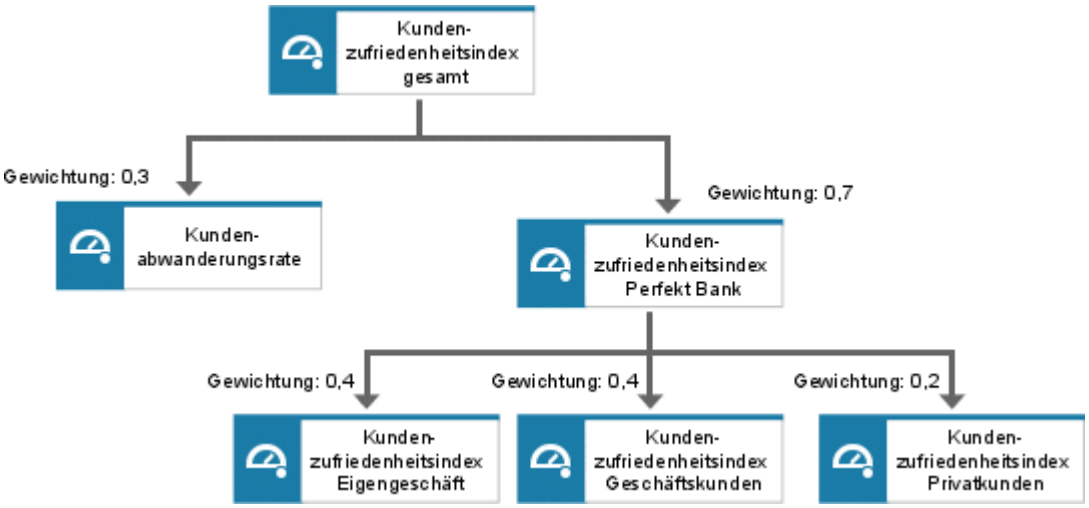


Abbildung 125: Kennzahlenbaum

Der Modelltyp **Kennzahlenbaum** verwendet lediglich den Objekttyp **Kennzahlinstanz**.

Den Kennzahlenbaum finden Sie auf der Fachkonzeptebene in der Datensicht.

6.3.3 Beziehungen zu anderen Modellen

Die Objekte der Balanced Scorecard-Methode können zu folgenden ARIS-Modellen in Beziehung stehen:

- EPK
- Zieldiagramm
- DW-Struktur

7 eBusiness Scenario Diagram

7.1 Einleitung

Die reibungslose Abfolge unternehmensübergreifender Geschäftsprozesse gewinnt zunehmend an Bedeutung. Dabei steht auf der einen Seite der Ablauf bestimmter Vorgänge an den Schnittstellen zwischen den Unternehmen, auf der anderen Seite die Schnittstelle zwischen Unternehmen und ihren Kunden im Blickpunkt. Die Kontakte sollen übersichtlich, schnell, einheitlich und direkt erfolgen.

Größte Relevanz besitzt auch das schnelle Finden von adäquaten Geschäftspartnern - aus Unternehmenssicht - und Anbietern - aus Verbrauchersicht. Eine optimale Gestaltung dieser Prozesse bedeutet einen Wettbewerbsvorteil. Die geeignete Plattform zur Unterstützung der multilateralen Beziehungen ist das Internet. Da die Vorgänge innerhalb des oben beschriebenen Umfeldes sehr vielschichtig sind, muss definiert werden, was unter E-Business verstanden wird.

E-Business ist ein Oberbegriff für die Unterstützung der Geschäftstätigkeit eines Unternehmens durch Informations- und Kommunikationstechnologien. Es umfasst die Unterstützung der Beziehungen und Prozesse zwischen Geschäftspartnern, Mitarbeitern und Kunden durch elektronische Medien (Herrmans, Sauter, 1999).

Somit kann E-Business die Erstellung einer Web-Site zur Unternehmenspräsentation, der Kauf eines Artikels via Internet, ein hoch komplexes Projekt zweier Unternehmen oder auch vielschichtige Beziehungen unter beliebig vielen Partnern bedeuten, die auf einem Marktplatz zusammenkommen.

Daraus leiten sich folgende Benennungen ab:

B2B (BUSINESS TO BUSINESS)

Business to Business bezeichnet die Abwicklung von Transaktionen zwischen Unternehmen. Dies wird beispielsweise durch die Vernetzung der Supply Chains der Unternehmen ermöglicht.

B2C (BUSINESS TO CONSUMER)

Business to Consumer bezeichnet die Abwicklung von Transaktionen zwischen Unternehmen und Ihren Kunden. Beispiel hierfür sind die Einkäufe von Kunden in Online-Shops.

B2A/C2A (BUSINESS/CONSUMER TO ADMINISTRATION)

Business/Consumer to Administration bezeichnet alle Transaktionen zwischen Unternehmen oder Einzelpersonen und der öffentlichen Verwaltung. Besonders im Bereich der Kontakte zwischen Unternehmen und Verwaltung wird ein enormes Potenzial zur Kosteneinsparung gesehen.

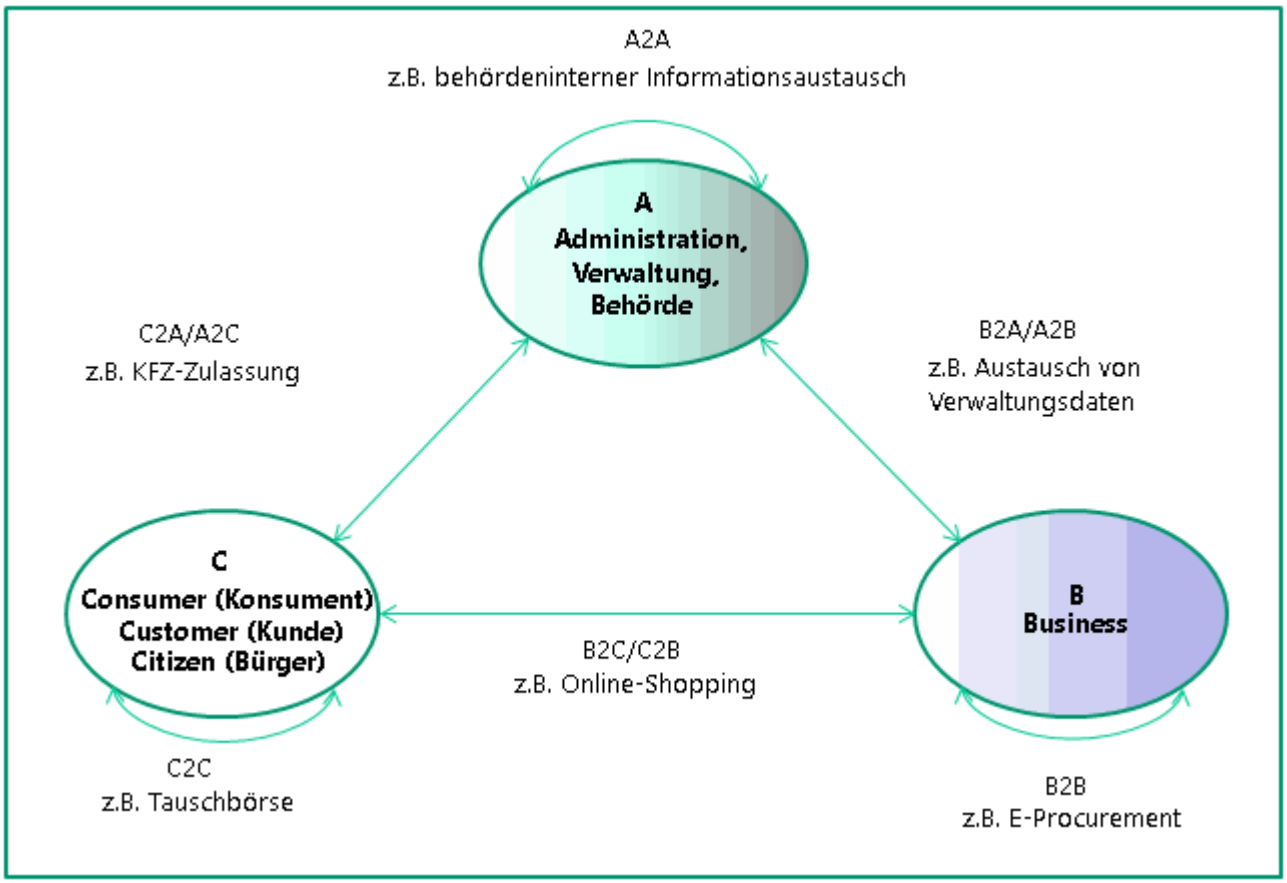


Abbildung 126: Transaktionsmöglichkeiten im E-Business

Neben der Unterscheidung in verschiedene Partner ist auch eine Unterscheidung im Umfang der Beziehungen zwischen den verschiedenen Partnern möglich: one to one, one to many, many to many. Große Bedeutung hat vor allem das Marktplatz-Szenario erlangt.

MARKTPLÄTZE

Elektronische Marktplätze sind virtuelle Plätze, auf denen eine (beliebige) Zahl Käufer und Verkäufer Waren und Dienstleistungen (offen) handeln und Informationen tauschen.

Zur Unterstützung dieser denkbaren Szenarien wurde das eBusiness Scenario Diagram entwickelt. Gemeinsam mit anderen Methoden und verschiedenen Komponenten, die ARIS zur Verfügung stellt, kann die Realisierung von E-Business-Projekten optimal unterstützt werden. Dieses Kapitel zu den eBusiness Scenario-Diagrammen beschreibt zunächst die Methode mit allen Objekten und Auswertungsmöglichkeiten und geht danach auf die Verbindung zu anderen Methoden ein. Abschließend wird ein Anwendungsfall vorgestellt, der die komplexen Möglichkeiten aufzeigt.

7.2 Die Methode des eBusiness Scenario Diagrams

7.2.1 Die Idee

Durch die Möglichkeit, eine Wertschöpfungskette ganzheitlich zu betrachten, d.h. vom Endkunden bis zu allen an einem Vorgang beteiligten Unternehmen, soll Optimierungspotenzial entwickelt werden. Ziel hierbei ist beispielsweise die Verbesserung der Supply Chain, Senkung von Beschaffungs- und Vertriebskosten oder Optimierung der Architektur des Informationssystems. Die Darstellung mittels des eBusiness Scenario Diagram ermöglicht die Visualisierung der Inhalte, die zur Erreichung der genannten Ziele untersucht werden müssen. Durch die gewählte Form der Spaltendarstellung werden die Schnittstellen zwischen unterschiedlichen Prozesspartnern abstrahiert über die Spaltenränder abgebildet. Verschiedene Reporte ergänzen die Modelle und bieten wichtige Analysemöglichkeiten.

7.2.2 Das Modell und seine Objekte

Die im Modell betrachteten Wirtschaftssubjekte werden in der Kopfzeile eingeordnet und als **Business Participant** bezeichnet. Sie entstammen der Organisationssicht und können mit Organigrammen hinterlegt werden, die beispielsweise die Unternehmensstruktur oder auch die Beziehungen zwischen den verschiedenen Objekten der einzelnen Spalten verdeutlichen können.

Zentrale und strukturbildende Objekte des Modells sind die am Gesamtprozess beteiligten Einzelprozesse der Wirtschaftssubjekte und ihre Schnittstellen zueinander. Ein Einzelprozess, auch **Business Process**, ist ein wesentlich an unternehmensübergreifenden Kooperationen beteiligter Unternehmensprozess. Die genauere Darstellung und Analyse eines solchen Einzelprozesses erfolgt durch Hinterlegung eines Prozessmodells. Alle Abläufe eines Unternehmens werden im Modell in der Zeile unter dem Business Participant aber in der gleichen Spalte modelliert. Die unternehmensübergreifende Abstimmung bedarf auch der genauen Analyse der **Anwendungssysteme** und Hardware, die von den verschiedenen Wirtschaftsobjekten zur Unterstützung ihrer Einzelprozesse eingesetzt werden, z.B. der ERP-Systeme. Sie werden durch **Business Components** dargestellt. Zur Abstimmung dieser unterschiedlichen Komponenten ist es notwendig, die genauen Verantwortlichkeiten für die Systeme zu bestimmen. Dazu können Objekte vom Typ **Organizational Unit Type** verwendet werden. Auch die Rollen der am Prozess beteiligten Mitarbeiter können definiert werden. Sie werden im Modell als **Employee role** bezeichnet. Eine besondere Anforderung an eBusiness-Modellierung im Allgemeinen stellt die Einbeziehung der Schnittstellen dar. Dabei kommt den Spaltenrändern eine zentrale Bedeutung zu, da sie die Schnittstellen zwischen den Prozessteilnehmern symbolisieren. Ihre Betrachtung erfolgt unter verschiedenen Gesichtspunkten.

Ein Schwerpunkt kann die Übertragung prozessspezifischer Informationen sein. Dazu dienen die **Business Documents**, die die Form eines XML- oder eines HTML-Dokuments annehmen können. Das Business Document kann mit einem Modell der Datensicht hinterlegt werden, beispielsweise mit einer Document Type Definition. Alternativ dazu kann auch der Geld- oder Güterstrom durch die Objekte **Money Transaction** bzw. **Goods Shipment** abgebildet werden.

Ein weiterer Aspekt ist von großer Bedeutung: Die Sicherheit der via Internet übermittelten Daten und besonders auch der elektronisch abgewickelten Zahlungen muss gewährleistet sein. Dazu werden verschiedene Verschlüsselungstechniken verwendet, z. B. SET (Secure Electronic Transaction) oder SSL (Secure Socket Layer). Dieser Sicherheitsaspekt wird durch die Einbeziehung des Objektes **Security Protocol** berücksichtigt. Auch für die Sicherung von Transaktionen können Verantwortliche bestimmt werden, die ebenfalls durch den **Organizational Unit Type** dargestellt werden. Zum Schwerpunkt kann auch die Analyse eines eher technischen Aspekts, nämlich die Betrachtung der technischen Ausgestaltung der Datenübertragung an den Schnittstellen bestimmt werden. Hierzu werden im Modell verschiedene Informationsträger verwendet. Die Einzelprozesse können über **Intranet**, **Extranet** oder **Internet** verbunden werden. Die Datenübertragung kann per **E-Mail** erfolgen. Als Übertragungsmedium erlangt das Handy immer größere Bedeutung.

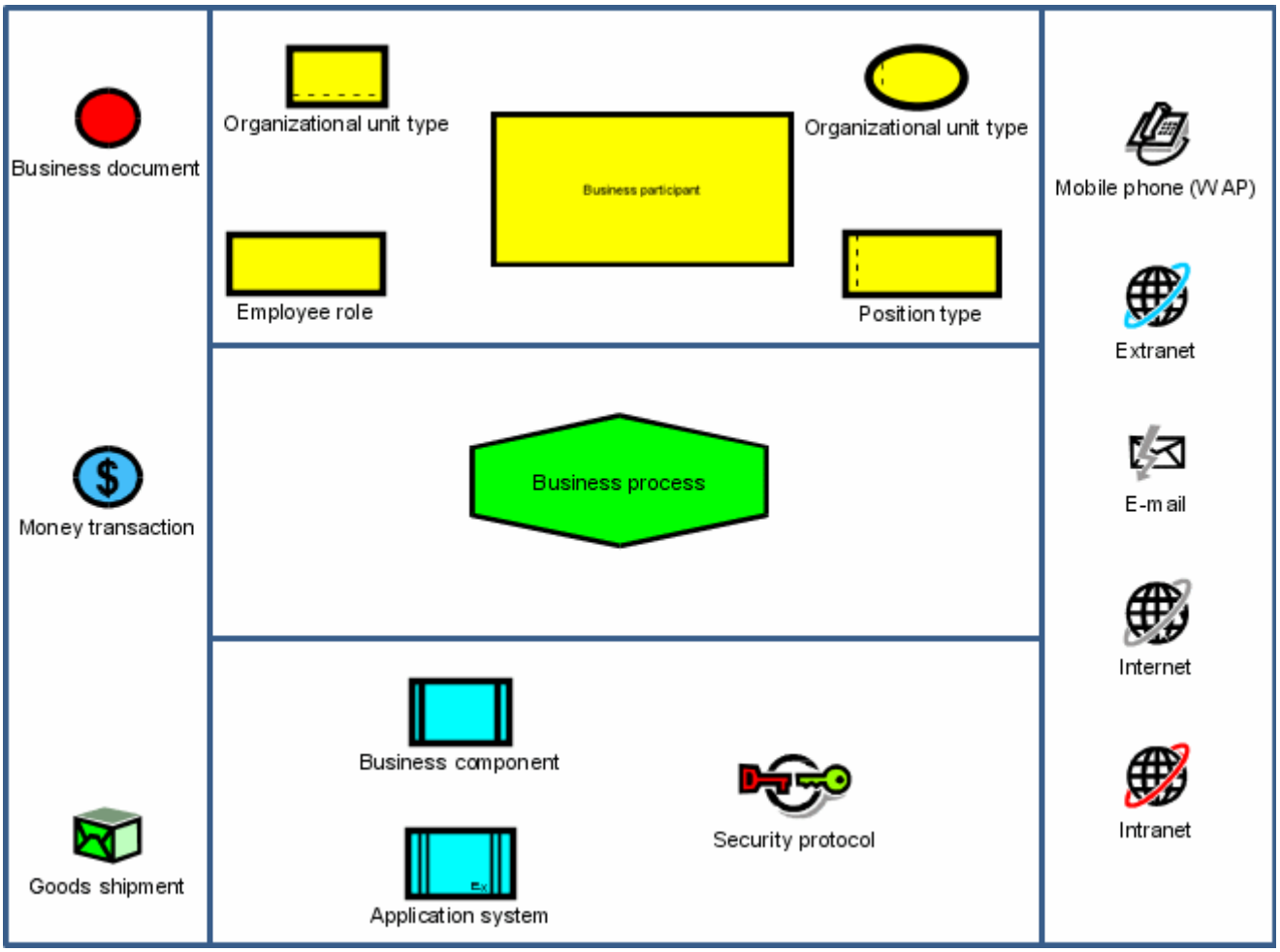


Abbildung 127: Die Objekte im eBusiness Szenario Diagram

7.2.3 Attributgruppe Übertragungsart

Die modellierten Objekte können durch die Pflege ihrer Attribute weiter spezifiziert werden. Ein Attribut ist besonders auf die Anforderungen des E-Business zugeschnitten.

Die Attributgruppe **Übertragungsart** der Objekte **Business Document**, **Money Transaction** und **Goods Shipment** ist hervorzuheben. Durch Pflege eines der Übertragungsart-Attribute wird nicht nur der Übertragungsweg gekennzeichnet, es kann zugleich die Notwendigkeit der Sicherung der Transaktion zum Ausdruck gebracht werden. Handelt es sich beispielsweise um eine Online-Übertragung, darf die bereits angesprochene notwendige Sicherung vertraulicher Informationen und Daten nicht fehlen.

7.3 Auswertungen durch Reporte

Verschiedene Auswertungsmöglichkeiten unterstützen die Modellierung der eBusiness Scenarios. Diese Auswertungen erfolgen in Form von Reporten. Mehrere vordefinierte Auswertungen werden durch ARIS angeboten, aber auch eigene, benutzerspezifische sind denkbar. Für die eBusiness Scenarios werden die im Folgenden erläuterten Reporte ausgeliefert.

7.3.1 Überprüfung der Datensicherheit

Die Sicherheit online übertragener Daten ist einer der wichtigsten Punkte, die die Akzeptanz von E-Business beeinflussen. Persönliche Daten oder gar Zahlungen vor den Zugriffen unberechtigter Personen zu schützen, ist eine Aufgabe, die gelöst werden muss, will man das Vertrauen der Kunden und Partner nicht verlieren. Mittels eines Reports lassen sich alle ausgetauschten Leistungen (Money Transaction und Goods Shipment) und Daten (Business Documents) diesbezüglich überprüfen. Dazu wird die bereits angesprochene Attributgruppe **Übertragungsart** ausgewertet und für den Fall, dass es sich um eine Online-Übertragung handelt, überprüft, ob eine Verschlüsselung der Daten erfolgt. Somit werden eventuelle Sicherheitslücken identifiziert, aber auch veraltete Verschlüsselungsmethoden aufgezeigt und damit wird eine leichtere Aktualisierung ermöglicht.

7.3.2 Systemunterstützung

Ein zweiter wichtiger Aspekt bei der Realisierung von E-Business-Projekten ist die Abstimmung der Anwendungssysteme. Dadurch ergeben sich für Unternehmen viele Fragen. Welche Prozesse müssen durch welche Systeme unterstützt werden? Wer zeichnet sich für die Bedienung welcher Systeme verantwortlich? Wo entsteht gegebenenfalls Schulungsaufwand? Welche Anpassungen der bestehenden Systeme müssen gemacht werden? Auch hierzu lassen sich mittels eines Reports Aussagen treffen. Es werden die einzelnen Prozesse mit den dazugehörigen Systemen und den dafür Verantwortlichen aufgelistet

7.3.3 Informationsfluss

Im Gegensatz zu anderen Prozessmodellen steht bei den eBusiness Scenarios die Transaktion im Vordergrund. Besonderes Augenmerk liegt auf den Daten und Leistungen, die ausgetauscht werden. Daher werden auch hierzu Auswertungen angeboten. Wichtig ist die Frage, welche Daten und Leistungen an welcher Stelle erzeugt werden und wo sie Verwendung finden. Hierzu gibt ein Report die modellierten Daten und Leistungen sowie die jeweiligen Prozesse aus, denen sie als Input oder Output zuzurechnen sind.

7.3.4 Collaborative Business Maps

Eine spezielle Form der Modelle stellen die von der SAP verwendeten Collaborative Business Maps dar. Sie stellen die Sicht auf verschiedene Partner besonders in den Vordergrund. Es werden zwei Sichten unterschieden, die auf verschiedene Zielgruppen ausgerichtet sind. Eine „aggregated view“, die nur die Business Participants sowie die Prozesse enthält und die an das führende Management adressiert ist, und eine "detailed view" mit den Dokumenten und Rollen der Prozessverantwortlichen für die Fachabteilungen. Die Reporte ermöglichen es jederzeit, die im eBusiness Scenario Diagram dargestellten Informationen in beide Sichten zu übertragen.

7.4 Verbindung zu anderen Methoden und Komponenten

Die Verwendung der verschiedenen Modellierungsmethoden bietet die Möglichkeit, einen Sachverhalt aus unterschiedlichen Perspektiven heraus zu betrachten und mit einem jeweils speziellen Fokus verschiedenen Zielgruppen zugänglich zu machen. Das E-Business Scenario ist Ausgangspunkt dieser Betrachtungen. Seine Objekte bieten die Möglichkeit, zielgruppenspezifisch Detaillierungen vorzunehmen. Hierdurch lässt sich die ganzheitliche

Darstellung eines E-Business-Projektes erreichen. Außerdem ermöglichen die unterschiedlichen Komponenten von ARIS Auswertungen für die erzeugten Modelle, so dass Projekte im E-Business Umfeld optimal unterstützt werden können.

BEISPIEL: EINFÜHRUNG EINES ONLINE-SHOPS

In einem ersten Schritt werden die Ziele definiert, die durch die geplanten E-Business-Aktivitäten erreicht werden sollen. Dieser Schritt kann mit Hilfe der ARIS-Komponente Balanced Scorecard (siehe Kapitel **Balanced Scorecard-Methode** (Seite 126)) ausgeführt werden. Die Prozesse, die zur Erreichung dieser Ziele optimiert werden sollen, werden dabei identifiziert. In vorliegendem Beispiel lässt sich die Erschließung eines neuen Vertriebsweges, nämlich über Internet, als Ziel identifizieren. Um diesen neuen Weg optimal zu beschreiten, ist eine genaue Dokumentation und Planung notwendig.

Nicht nur der Prozessablauf als solcher, sondern auch die Organisation der Verantwortlichkeiten, der Schnittstellen zwischen verschiedenen Systemen und der Datensicherheit ist in die Überlegungen einzubeziehen.

Ausgangspunkt der Betrachtung ist das E-Business Scenario Diagram. Die Geschäftsbeteiligten sind in unserem Beispiel das Unternehmen, welches das Shop-System implementiert, und der Kunde, der dieses Angebot nutzen wird. Der gesamte Prozess vom „Betreten des Shops“ bis zum „Verlassen“ wird in seine wichtigsten Teilschritte zerlegt. Die Darstellung beinhaltet die Sicht des Kunden und des Unternehmens. Das E-Business Scenario Diagram dient als Einstieg in das Einführungsprojekt. Diese Abbildung zeigt die Aufteilung des Gesamtprozesses.

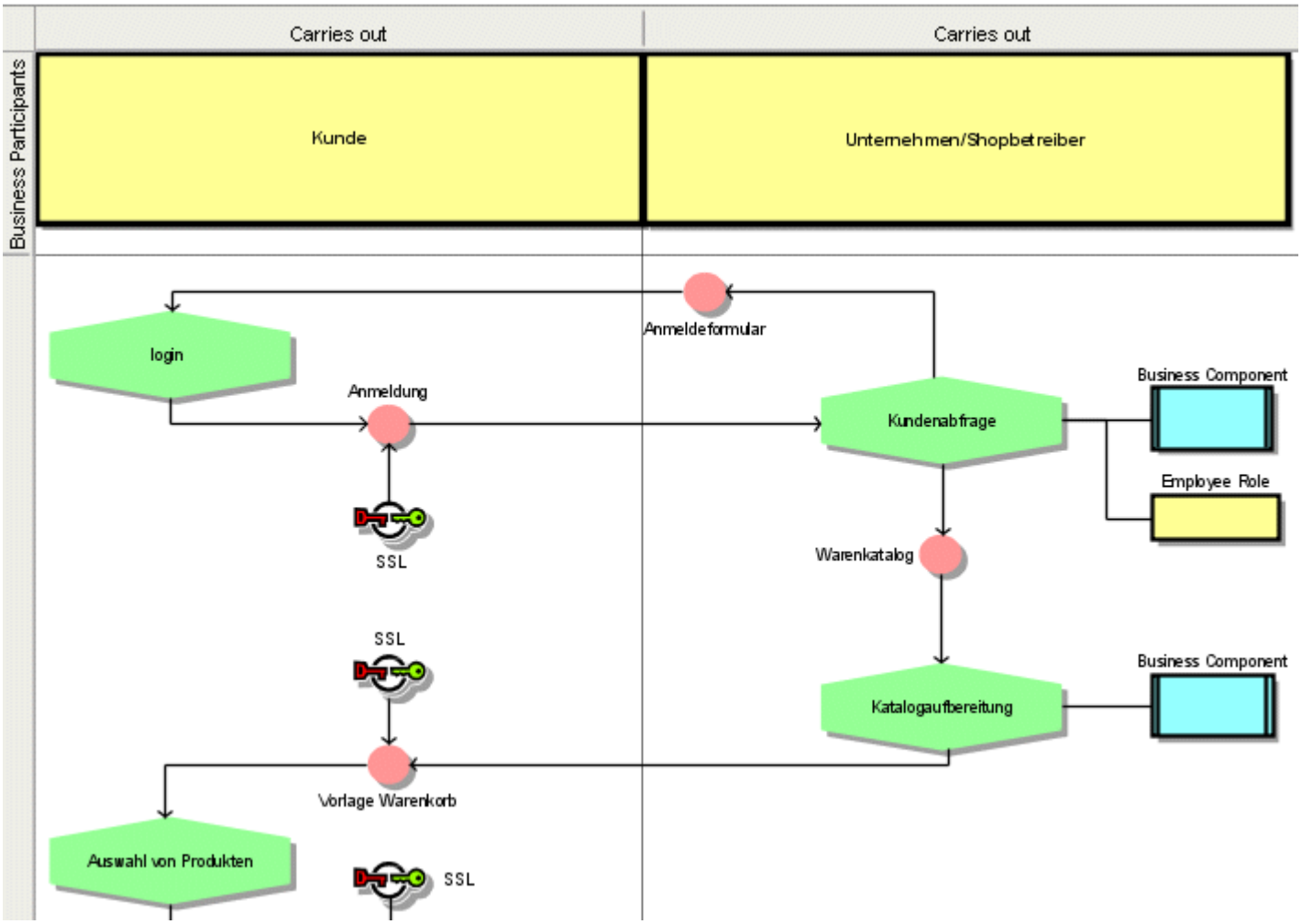


Abbildung 128: Ausschnitt aus eBusiness Scenario "Online-shop"

Die verschiedenen Prozessschritte lassen sich nun durch EPKs verfeinern, z. B. mit der Simulations-Komponente überprüfen, nach ihrer Optimierung durch Pipeline-Diagramme darstellen und durch Intershop Enfinity in ein fertiges Shop-System umwandeln und weiter verbessern.

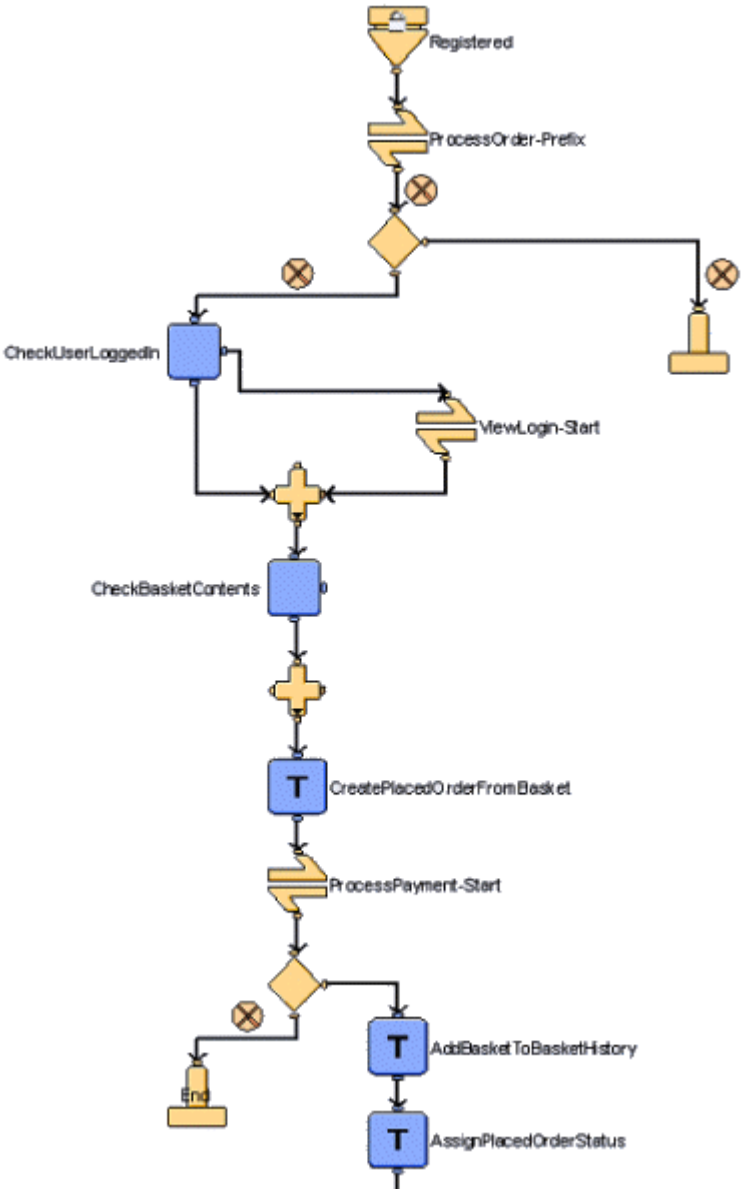


Abbildung 129: Ausschnitt aus dem Pipeline Diagram

Soll eine Anbindung eines Shops an ein externes ERP-System (Enterprise Resource Planning-System) erfolgen, benötigt man die korrekten Formate der zu übergebenden Daten. Um dies zu ermöglichen, gibt es verschiedene Wege, die Dokumente und Daten zu standardisieren. Eine Möglichkeit der Standardisierung bietet die Verwendung der Extensible Markup Language (XML). Da XML eine Sprache ist, die völlig unterschiedlich weiterentwickelt wird, ist auch hier eine einheitliche Grundlage notwendig. Unterschiedliche Organisationen, bestehend aus Unternehmen und wissenschaftlichen Instituten, bemühen sich um die Vereinheitlichung von XML für verschiedene Bereiche.

Die Verwendung standardisierter XML-Dokumente erleichtert die Anbindung von ERP-Systemen.

Die Problematik, verschiedene Komponenten aufeinander abstimmen zu müssen, wurde bereits angesprochen. Eine Visualisierung der Systeme kann im Anwendungssystemtypdiagramm erfolgen, das den Anwendungssystemen oder auch Geschäftskomponenten als Modell hinterlegt werden kann, um die Zusammenhänge der Systeme zu verdeutlichen.

Die neuen E-Business-Aktivitäten werden auch die Organisationsstruktur beeinflussen. Eventuell müssen neue Verantwortlichkeiten definiert oder Zuordnungen neu getroffen werden. Im E-Business Scenario Diagram können Rollen und Organisationseinheiten für einzelne Prozessschritte beschrieben werden. Welche Position sie in der Organisation des Unternehmens oder auch im Prozess einnehmen, kann durch Organigramme weiter analysiert werden.

Der Schritt der Implementierung beginnt mit der Umsetzung modellierter Inhalte. Bei Verwendung der Intershop-Methode werden die modellierten Inhalte mit Intershop Enfinity in ein laufendes System umgewandelt.

8 IT City Planning

8.1 Enterprise Architecture und IT City Planning

IT City Planning ist ein in den 80er Jahren entwickelter Architekturansatz des Franzosen Jacques Sassoon. Ziel des IT City Planning ist es, Stimmigkeit in eine heterogene Systemlandschaft zu bringen, indem die Interaktionen, also der Informationsaustausch zwischen den Applikationen innerhalb eines Systems genauer betrachtet werden.

In Anlehnung an die Städteplanung ist das Vorgehen beim Erstellen eines IT City Plan von dem Gedanken bestimmt, ein langfristiges, strategisches IT-Management zu ermöglichen, das nicht nur die Gegenwart berücksichtigt, sondern auch die Aspekte der Vergangenheit (Legacy-Systeme) und die der Zukunft einbezieht.

Dabei wird kein Redesign für das gesamte System durchgeführt. Vielmehr wird inkrementell von Projekt zu Projekt vorgegangen.

Ähnlich wie bei der Model Driven Architecture (MDA), die durch die ARIS-Produkte unterstützt wird, erfolgt der Einstieg über Modelle, die das Informationssystem losgelöst von technologiebezogenen Informationen beschreiben. Allerdings wird im Rahmen des IT City Planning nicht auf UML zurückgegriffen, was den Einstieg für Personen mit weniger technischem Hintergrund vereinfacht und die Akzeptanz erhöht.

8.2 An welche Unternehmen richtet sich IT City Planning?

IT City Planning richtet sich an folgende Unternehmen:

- Unternehmen, mit einem großen Bestand an Anwendungen.
- Unternehmen, die schon sehr lange Informationstechnologien einsetzen.
- Unternehmen, in denen die Informationstechnologie von strategischer Bedeutung ist.
- Unternehmen, die fusionieren.

Ziele des IT City Planning:

- Wiederverwendung von Software-Ressourcen, um das Anlegen weiterer Redundanzen zu vermeiden.
- Verringerung von Wartungskosten, indem durch "blockweise" Überarbeitung und Definition neuer Software-Ressourcen die verschiedenen, bisher verwendeten, ersetzt und die verschiedenartigen Anwendungsfälle abgedeckt werden können.
- Konsolidierung der Informationssysteme.
- Vorbereitung des Einsatzes von EAI-Software auf einer höheren Ebene.

Das Erstellen eines IT City Plan ist Aufgabe des Integration Competence Center. Der Plan selbst muss sowohl die Design Pattern als auch die Informations- und Technologiearchitektur ansprechen.

EAI = Enterprise Application Integration. Unternehmensweite Integration von Anwendungen. EAI liefert die E-Business-Infrastruktur. EAI-Software ist die technische Middleware und Voraussetzung für die Realisierung einer E-Business-Strategie.

8.3 IT City Planning mit ARIS

ARIS betrachtet folgende Sichten eines Informationssystems:

- Datensicht
- Funktionssicht
- Organisationssicht
- Leistungssicht
- Prozesssicht

Jede dieser Sichten ist in die Beschreibungsebenen **Fachkonzept**, **DV-Konzept** und **Implementierung** untergliedert. Sie orientieren sich am Lebenszyklus eines Informationssystems und der Nähe zur Informationstechnik.

Die DV-Konzept- und Implementierungsebenen beschreiben im Wesentlichen das Software-System. Die Begriffswelten dieser Ebenen sind eng miteinander verwandt und die "Übersetzung" ist unproblematisch.

Anders verhält es sich mit dem Übergang zwischen Fachkonzept und DV-Konzept. Bei der DV-Konzepterstellung muss die betriebswirtschaftliche Sicht mit der Standardsoftware abgeglichen werden. Dies erfordert betriebswirtschaftliches Know-how ebenso wie DV-nahes Wissen (vgl. Scheer, A.-W., ARIS - Vom Geschäftsprozess zum Anwendungssystem, 1998, 3. Auflage, S. 7).

Die Informationssystem-Sicht (IS-Sicht) des IT City Planning gibt hier als Vermittler zwischen den Ebenen Hilfestellung. Die Objekttypen der IS-Sicht sind in ARIS zwischen Funktion und Anwendungssystemen anzusiedeln und erweitern somit die Funktionssicht in ARIS. Wie Funktionen werden IS-Elemente mit den verschiedenen Konstrukten aus den bekannten Sichten des ARIS-Hauses in Beziehung gesetzt. Diese Erweiterungen betreffen im Wesentlichen die Prozesssicht und Datensicht. Wenn im Folgenden von IS-Sicht gesprochen wird, sind darunter die Modelltypen aus Funktions- und Prozesssicht des ARIS-Hauses zu verstehen, in denen Beziehungen zwischen IS-Elementen beschrieben werden oder die einer Detailbeschreibung von IS-Elementen im Kontext der anderen ARIS-Sichten dienen.

Anwendungssystemtypen, DV-Funktionstypen und Sockets werden im Folgenden als Elemente der so genannten IT-Sicht (IT-Sicht = Informationstechnologie-Sicht) betrachtet. In Analogie zur IS-Sicht gehören alle Modelltypen der IT-Sicht an, in denen Beziehungen zwischen Anwendungssystemtypen, DV-Funktionstypen und dem neuen Objekttyp **Socket** beschrieben werden oder die einer Detailbeschreibung eines dieser Elemente dienen.

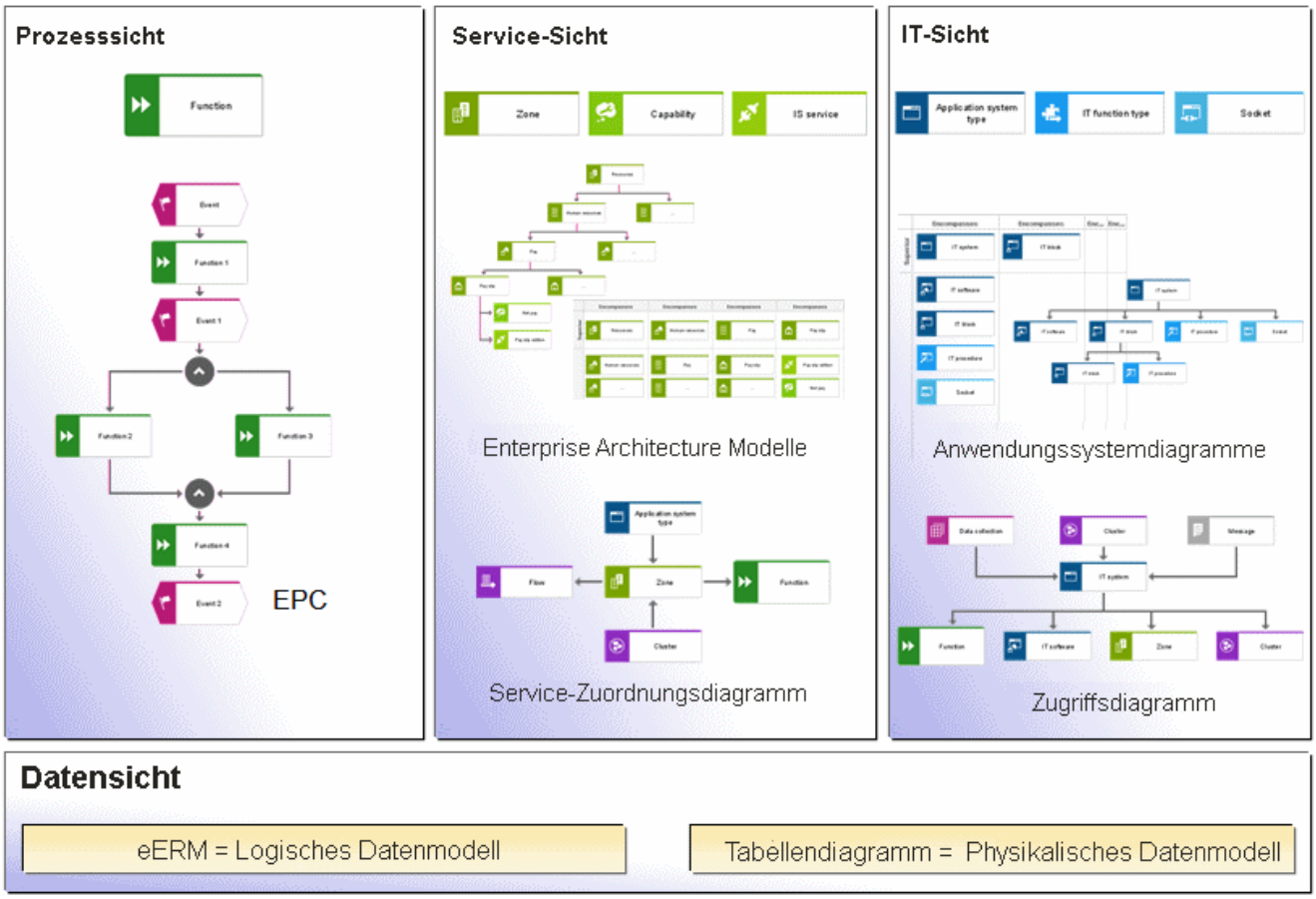


Abbildung 130: Prozess-, IS- und IT-Sicht

8.4 Service-Sicht

Zum Beschreiben der Service-Sicht stellt ARIS folgende Modelltypen zur Verfügung:

- Service-Architekturdiagramm
- Service-Zuordnungsdiagramm
- Service-Kollaborationsdiagramm

In den beiden Service-Architekturdiagramm-Modellen wird das Informationssystem in hierarchischer Form gegliedert

Eine IS-Hierarchie kann folgende Ebenen umfassen:

- Zone
- District
- Building cluster
- Functional block
- Fähigkeit
- IS Service
- Geschäftsservice

Zone, District, Building Cluster, Functional Block und Geschäftsservice sind vom Objekttyp **Servicetyp**. Servicetypen dienen der funktionalen Gliederung eines Informationssystems in unabhängige Einheiten/Blöcke.

Jeder Servicetyp zeichnet sich dadurch aus, dass es "Eigner" der von ihm verwendeten Daten und der zugehörigen Verarbeitungsmethoden ist. Andere Servicetypen können auf diese Daten und Verarbeitungsmethoden nur zugreifen, wenn sie einen Dienst des "Eigner-Servicetyp" aufrufen.

In einem Servicetyp werden ähnliche Daten verwendet und gleiche Aktivitäten und Geschäftsfunktionen ausgeführt.

Auf der höchsten Ebene wird das Informationssystem in Zonen gegliedert. Eine Zone entspricht z. B. einem Arbeits- und Entwicklungsbereich.

Folgende Abbildung zeigt die Zonen, in die ein Informationssystem eines Unternehmens typischerweise untergliedert werden kann.

Abbildung 131: Zonen eines Unternehmens-Informationssystems

Jede Zone kann wiederum einen Distrikt oder mehrere Distrikte enthalten.

Distrikte einer Zone zeichnen sich durch ähnliche Prozesse und große Ähnlichkeit bzgl. zeitlicher Merkmale aus, z. B. ähnliche Lebens- und Informationsbearbeitungszyklen. Distrikte sind beispielsweise Zahlungskonditionen, Preiskonditionen, Personalverwaltung, Reiserichtlinien usw.

Die Zone **Ressourcen** kann u. a. die Distrikte **Human Resources** und **Rechnungswesen** umfassen:

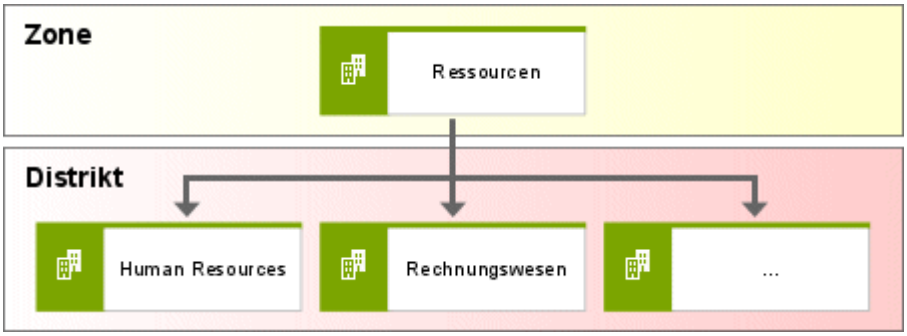


Abbildung 132: Zone in Distrikte untergliedert

Innerhalb eines Distrikts gibt es einen oder mehrere Building Cluster, die einen funktionalen Zweck erfüllen, z. B. Lohnzahlungen, Fakturierung usw.

Der Distrikt Human Resources umfasst das Building Cluster Bereichsadministration, Recruiting, Personalentwicklung und Personalbetreuung.

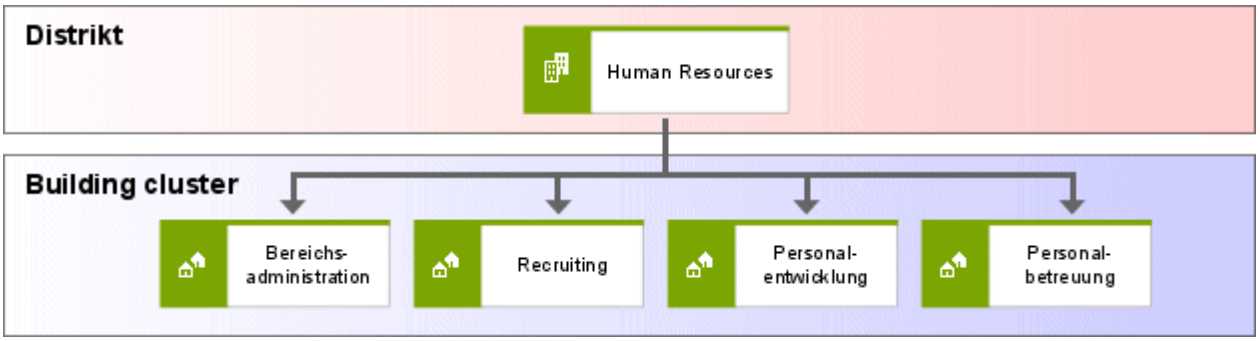


Abbildung 133: Distrikt in Building Cluster untergliedert

Jedes Building Cluster kann wiederum ein oder mehrere Functional Blocks umfassen. Die Functional Blocks eines Building Cluster zeichnen sich durch hohe Ähnlichkeit der von ihnen verwalteten Business-Objekte und Ereignisse aus.

Ein Functional Block stellt eine unabhängige, wieder verwendbare funktionale Komponente dar. In einem Functional Block werden Fähigkeiten und IS Services nach folgenden Regeln zusammengefasst:

- Sie weisen einen starken Zusammenhang zwischen den von ihnen verwalteten Objekten und unterstützten Funktionen auf.
- Der Austausch mit anderen funktionalen Blöcken ist minimal.

Das Building Cluster **Personalbetreuung** unseres Beispiels umfasst die Functional Blocks **Stammdatenpflege**, **Zeugnisse**, **Controlling** und **Gehälter**.

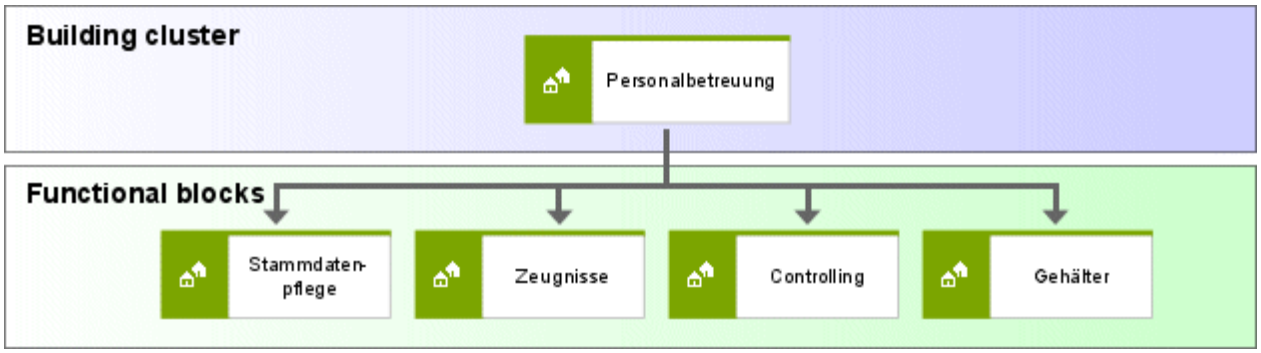


Abbildung 134: Aufteilung des Building Cluster "Personalbetreuung" in Functional Blocks

Eine Fähigkeit beschreibt eine elementare Funktionalität innerhalb eines Systems. Sie unterstützt die Realisierung einer Aktivität innerhalb eines Prozesses.

Ein IS Service beschreibt eine Schnittstelle eines Servicetyps oder einer Fähigkeit. Sie ermöglichen anderen IS-Elementen einen kontrollierten Zugriff auf Daten und Verarbeitungsmethoden des IS-Elements, das den Service zur Verfügung stellt.

Über diese Schnittstellen können Nachrichten mit anderen Elementen der IS-Sicht ausgetauscht werden.

Folgende Abbildung zeigt die Fähigkeiten und IS Services des Functional Block **Gehälter**.

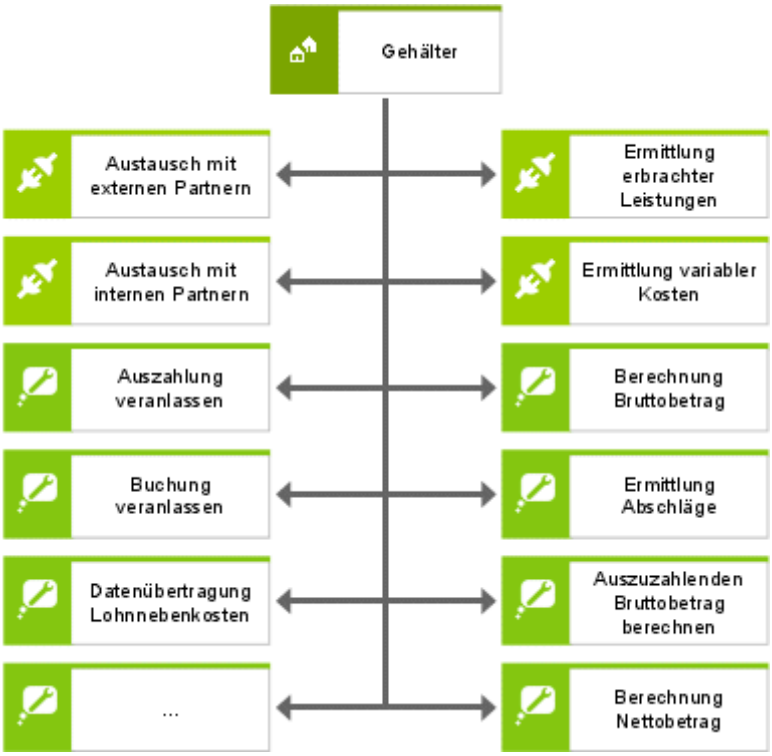


Abbildung 135: Fähigkeiten und IS Services des Functional Block "Gehälter"

Für die Beschreibung der IS-Hierarchie ist es nicht notwendig alle hier beschriebenen Ebenen auszumodellieren. Die IS-Elemente **Fähigkeit** und **IS Service** werden im Rahmen des IT City

Planning nicht als Elemente des Städteplans gesehen. Das Aufgabenfeld des Städteplaners endet auf der Ebene der Building Blocks. Fähigkeiten und IS Services gehören in das Aufgabengebiet des Architekten (vgl. Longépé, Christoph: Le projet d'urbanisation du système d'information, S. 18).

8.5 Servicetypen und ihre Daten

Um zu beschreiben, welche Daten innerhalb eines Servicetyps oder einer Fähigkeit gekapselt sind, wird das eERM verwendet. Im Rahmen des City Planning werden im eERM die Symbole der IS-Sicht angeboten. Mit Hilfe einer Kante des Typs **ist Eigner von** können diese Objekte mit Beziehungstypen und Entitytypen in Beziehung gesetzt werden.

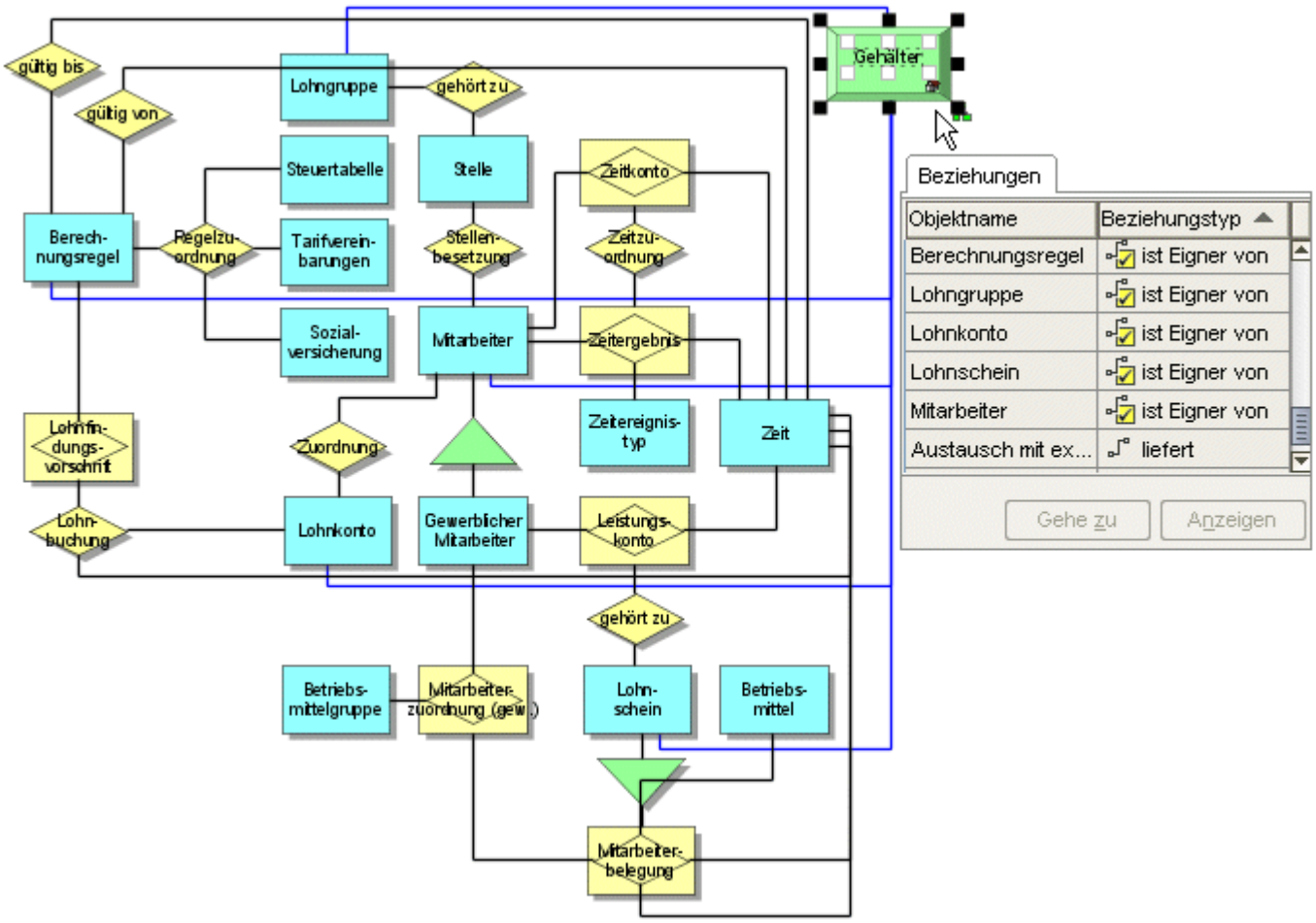


Abbildung 136: Kante "ist Eigner von" zwischen Symbolen der IS-Sicht und Beziehungs- und Entitytypen

8.6 Detailbeschreibung der Servicetypen

Eine detaillierte Beschreibung der Servicetypen und Fähigkeiten eines Informationssystems wird im Service-Zuordnungsdiagramm vorgenommen. Hierzu gehören

- die Schnittstellen eines Blocks,
- die Interaktionen zwischen den Blöcken,
- die Anwendungssysteme, die den Block unterstützen, und
- betriebswirtschaftliche Funktionen, die durch den Block unterstützt werden.

Zonen, Distrikte, Building Cluster, Functional Block und Fähigkeit können mittels der Kante **liefert** mit einem IS Service verbunden werden.

Mit Hilfe von Input/Output-Kanten zwischen IS- und Datenelementen können die Informationsflüsse zwischen den Servicetypen beschrieben werden.

Die verschiedenen Anwendungssystem- und DV-Funktionstypobjekte können durch eine Kante des Typs **unterstützt** den Objekten der IS-Sicht zugeordnet werden. Interpretiert man den City Plan nun wie den Bebauungsplan einer Stadt, kommt durch diese Kante zum Ausdruck, welche Informationssystembereiche von welchen Anwendungssystemen "besiedelt" werden. Die Kante **unterstützt** steht auch zwischen IS-Elementen und der Funktion zur Verfügung.

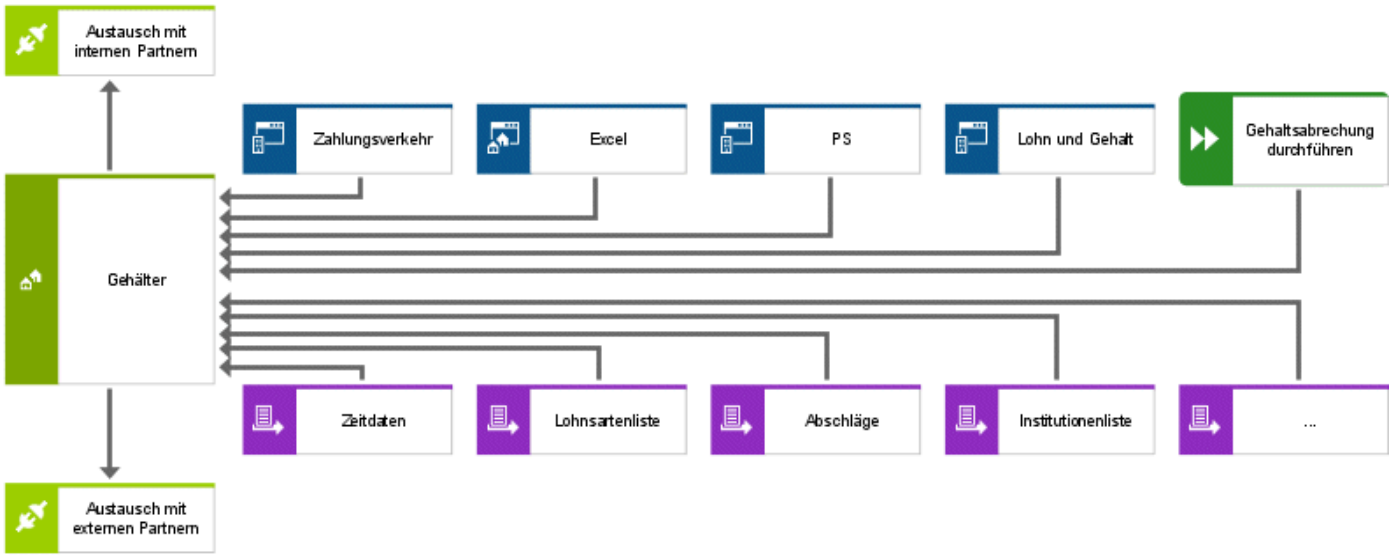


Abbildung 137: Kante "unterstützt" zwischen IS-Elementen und Funktion

8.7 Zeitlich-logische Abläufe zwischen IS-Elementen

Im Service-Zuordnungsdiagramm können für die im Service-Architekturdiagramm spezifizierten Servicetyp, Fähigkeit und IS Services die Beziehungen zu den Objekttypen der Organisations-, Daten- und Prozesssicht hergestellt werden. Die möglichen zeitlich-logischen Ablauffolgen der IS-Elemente können nicht dargestellt werden.

Das Service-Kollaborationsdiagramm wird im Rahmen des IT City Planning verwendet, um zeitlich-logische Ablauffolgen der IS-Elemente wiederzugeben, also den dynamischen Aspekte innerhalb des Informationssystems zu beschreiben. Es stellt das Pendant zum Programmablaufdiagramm (vgl. Kapitel **Programmablaufdiagramm** (Seite 98)) der IT-Sicht

dar. Durch diesen Modelltyp werden zur Darstellung der Ablauffolge Ereignisse zur Verfügung gestellt. Ähnlich der Zuordnung von IT-Elementen und Ereignissen im Programmablaufdiagramm können hiermit im Service-Kollaborationsdiagramm Folgen funktionaler Module definiert werden. Das Ereignis ist hierbei im Sinne eines Triggers zu verstehen, das Servicetypen, Fähigkeiten oder IS Services anstößt. Verzweigungen können mit den aus der EPK oder dem Programmablaufdiagramm bekannten Regeln dargestellt werden. Diese Ablauffolgen können auch ohne das Zwischenschalten von Ereignissen definiert werden.

8.8 IT-Sicht

In Analogie zur IS-Sicht stehen in der IT-Sicht folgende Modelltypen zur Verfügung:

- Anwendungssystemtypdiagramm
- Zugriffsdiagramm
- Programmablaufplan

Anwendungssystemhierarchie

Im Rahmen des City Planning wird die aktuelle Anwendungssystemhierarchie im Unternehmen mit Hilfe des Anwendungssystemtypdiagramms abgebildet.

Folgende Ebenen einer Anwendungssystemtyphierarchie können abgebildet werden:

- IT System
- Subsystem
- IT Software
- IT Block
- IT Procedure
- Socket

IT System, Subsystem, IT Software und IT Block sind Symbole des Objekttyps **Anwendungssystemtyp**. Die Hierarchie wird durch den Beziehungstyp **umfasst** hergestellt.

Auf der obersten Ebene der Anwendungssystemtyphierarchie befinden sich die IT-Systeme. Ein IT-System beschreibt eine strukturierte Menge von IT-Elementen, i. d. R. Subsystemen. Verwaltung und Betrieb eines IT-Systems unterliegen einer eindeutig bestimmten Organisationseinheit.

Ein Subsystem beschreibt eine Komponente eines IT-Systems. Die Bausteine eines Subsystems werden IT-Software genannt.

Eine IT-Software unterstützt einen homogenen Satz von Funktionen. Sie ist benutzerorientiert und unterstützt einen oder mehrere Geschäftsprozesse. Bestandteile einer IT-Software sind IT-Blöcke.

In einem IT Block werden in der Regel solche IT Procedures zusammengefasst, die auf gleiche Daten (Datenbanken, Tabellen, Dateien etc.) zugreifen.

IT Procedures sind Objekte des Typs **DV-Funktionstyp**. Jede IT Procedure unterstützt eine bestimmte Funktionalität.

Ein Socket entspricht dem IS Service, d. h., er beschreibt eine Schnittstelle, die ein IT-Element anderen IT-Elementen zur Verfügung stellt, um ihnen den Zugriff auf seine Daten und Verarbeitungsmethoden zu ermöglichen.

Folgende Abbildung zeigt beispielhaft die Subsystemstruktur des Systems DATEV:

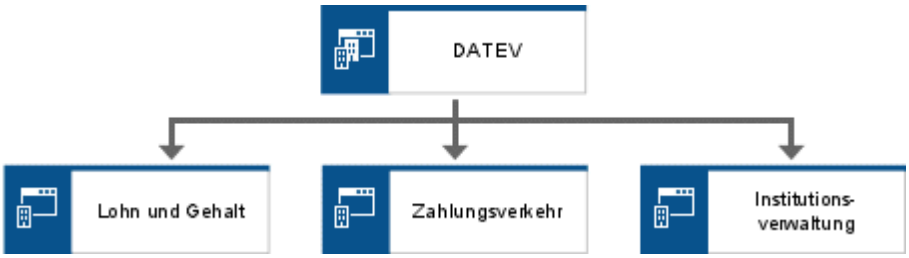


Abbildung 138: Subsystemstruktur des DATEV-Systems

8.9 IT-Elemente und ihre Daten

Um zu beschreiben, welche Daten innerhalb eines IT-Elements gekapselt sind, steht wie für die die IS-Elemente im eERM die Kante **ist Eigner von** zwischen Anwendungssystemtyp, DV-Funktionstyp oder Socket und Entity- oder Beziehungstyp zur Verfügung.

8.10 Detailbeschreibung der IT-Elemente

Eine detaillierte Beschreibung der IT-Elemente innerhalb des IT City Plan wird im Zugriffsdiagramm vorgenommen. Es entspricht dem Service-Zuordnungsdiagramm der IS-Sicht.

Beschrieben werden können

- In- und Output-Beziehungen des betrachteten IT-Elements
- unterstützte Geschäftsfunktionen
- unterstützte IS-Elemente
- Aufruf anderer IT-Elemente durch das betrachtete
- Plattform, auf der das IT-Element läuft

- Anwender des IT-Elements

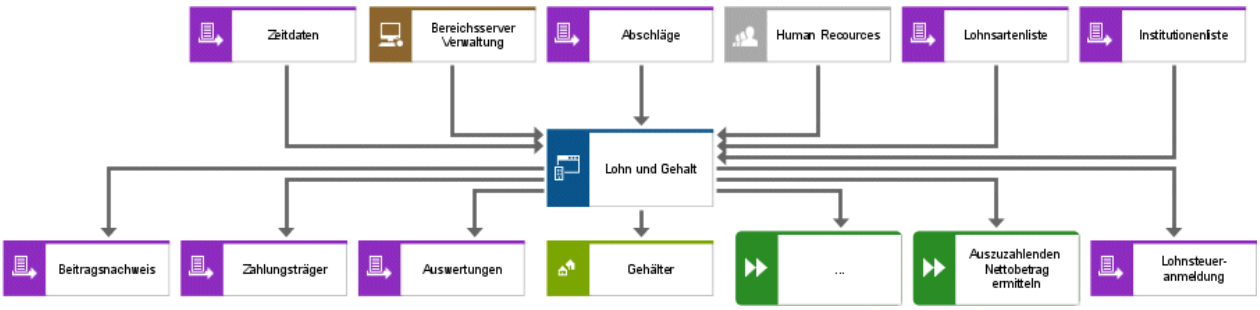


Abbildung 139: Detaillierte Beschreibung von IT-Elementen im Zugriffsdiagramm

8.11 Organisatorische Aspekte

In die Detailbeschreibung eines IT-Elements fließen auch Informationen der Organisationssicht ein. Hierzu zählen nicht nur die Informationen darüber, welche organisatorischen Elemente Anwender eines IT-Elements sein können. Mit Hilfe des Netzdiagramms können auch Einflüsse und Auswirkungen bzgl. der technischen Infrastruktur verdeutlicht werden.

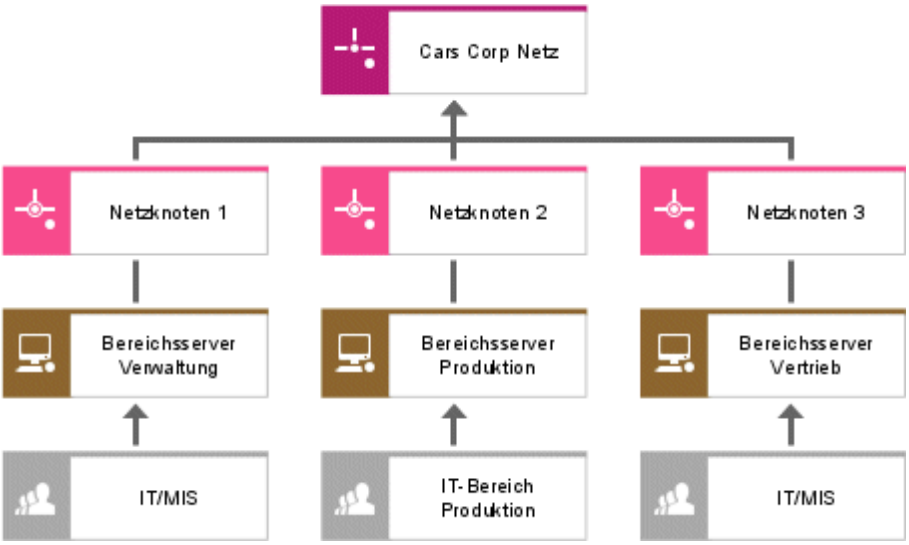


Abbildung 140: Einflüsse und Auswirkungen der technischen Infrastruktur

8.12 Zeitlich-logische Abläufe zwischen IT-Elementen

In Analogie zum Service-Kollaborationsdiagramm dient das Programmablaufdiagramm der Beschreibung der zeitlich-logischen Abläufe zwischen den verschiedenen IT-Elementen **Anwendungssystemtyp**, **DV-Funktionstyp** und **Socket**.

8.13 Zeitlich-logische Abläufe innerhalb der Architektur

Um die Einbindung von IS- und IT-Elementen in einen zeitlich-logischen Ablauf darzustellen, stehen entsprechende Objekte in den verschiedenen Prozessmodellen (alle Variationen der EPK) und im Programmablaufdiagramm zur Verfügung.

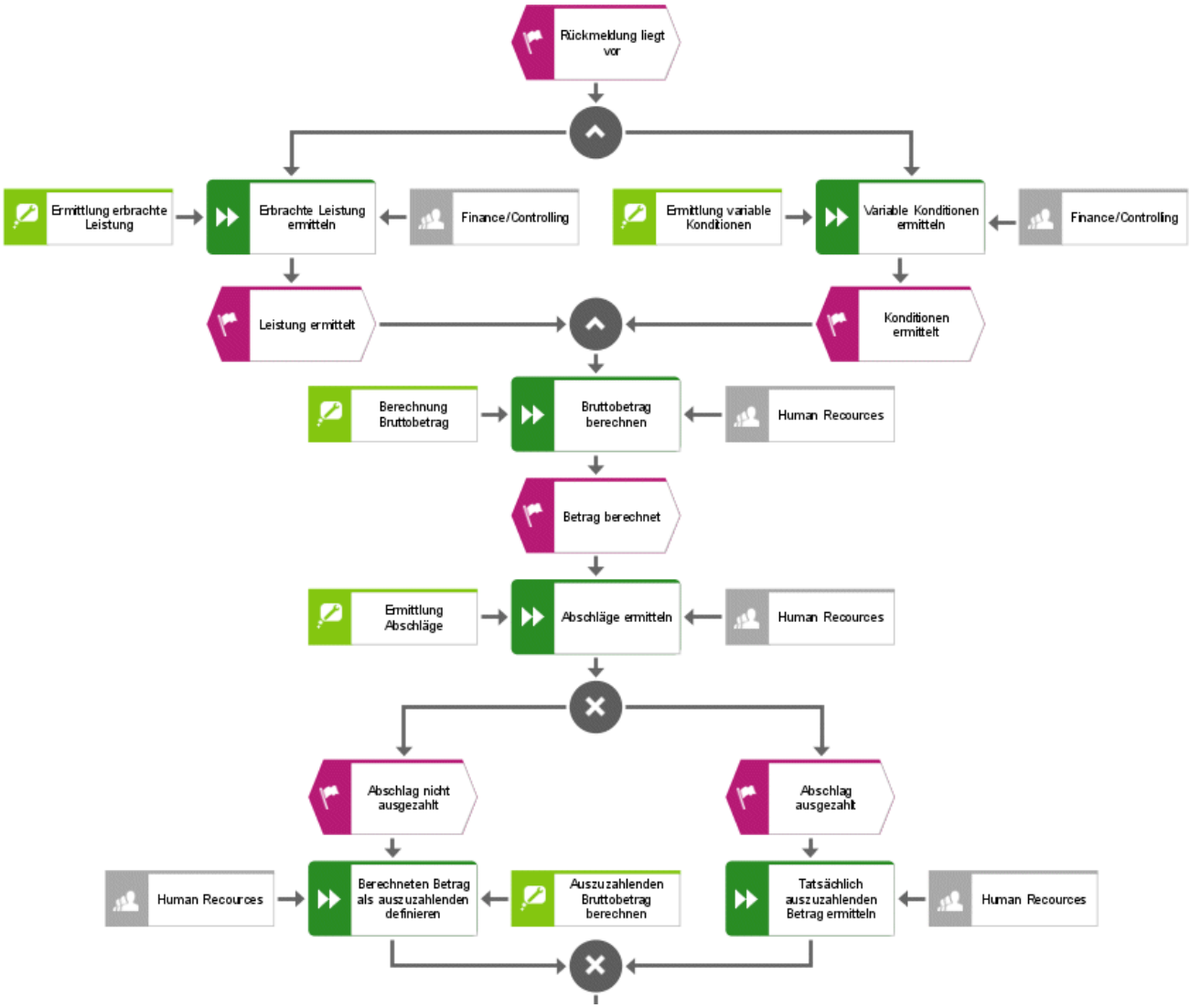


Abbildung 141: Einbindung von IS- und IT-Elementen in einen zeitlich-logischen Ablauf

8.14 Mögliche Auswertungen

Basierend auf der in den vorhergehenden Abschnitten beschriebenen Modellierung sind Auswertungen möglich, die folgende Fragen beantworten und beim Erstellen des zukünftigen Informationssystems hilfreich sind:

- Welche Daten werden von einem bestimmten IS-Element verwaltet?
- Welche Anwendungssysteme unterstützen ein IS-Element?
- Welche Funktionen werden von einem bestimmten IS-Element unterstützt?
- Welche Daten werden von den IT-Elementen eines bestimmten IS-Elements verwendet?
- Welche Daten werden von den IT-Elementen eines bestimmten IS-Elements erzeugt?
- Welche IS Services werden von einem IS-Element zur Verfügung gestellt und in welchen Prozessen werden sie genutzt?
- Auf welchen Hardwarekomponenten laufen die Anwendungssysteme eines bestimmten IS-Elements?

Für gewählte Anwendungssystemtypen, DV-Funktionstypen und Sockets werden folgende Auswertungen angeboten:

- Daten, die von einem IT-Element verwendet werden.
- IS-Elemente, die von einem IT-Element unterstützt werden.
- Funktionen, die von einem IT-Element unterstützt werden.
- Daten, die von den durch ein IT-Element unterstützten IS-Elementen verwendet werden.
- Daten, die von den durch ein IT-Element unterstützten IS-Elementen erzeugt werden.
- Hardwarekomponenten, auf denen ein IT-System läuft.

9 Business Process Modeling

Neue Informations- und Kommunikationstechnologien führen zu immer komplexeren Interaktionen und Transaktionen zwischen Unternehmen, ihren Partnern, Lieferanten und Kunden. Zunehmend zeigt sich, dass die weitere Entwicklung und Leistungsfähigkeit der Geschäftsprozesse von der engen Zusammenarbeit der einzelnen Geschäftspartner abhängt.

Zum einen möchte man sicherstellen, dass ein Unternehmen sein eigenes Handeln und das seiner Geschäftspartner besser verstehen kann, zum anderen sollen Organisationen die Fähigkeit erhalten, sich schneller an interne und marktbedingte Veränderungen anzupassen. Dabei kann eine standardisierte Prozessmodellierungssprache den Unternehmen helfen, ihre internen und externen Geschäftsabläufe transparent und flexibel zu beschreiben. Die Unternehmen sollen außerdem in der Lage sein, die modellierten Prozesse in einer angemessenen, anschaulichen und verständlichen Weise den beteiligten Geschäftspartnern kommunizieren zu können. Alle Beteiligten sollen die gleiche „Prozesssprache“ sprechen.

Die Business Process Management Initiative (BPMI.org) bietet zur Realisierung dieser Ziele die eigens entwickelte, standardisierte Modellierungssprache "Business Process Model and Notation (BPMN)" an. BPMN ist eine grafische Notation zum Beschreiben von Geschäftsprozessen.

Die Notation soll für alle Anwender leicht verständlich sein. Damit ist sie für Geschäftsprozess-Analysten und Personen geeignet, die die Prozesse überwachen und verwalten, aber auch für Entwickler, die Technologien zur Prozessausführung implementieren.

Daneben soll sichergestellt sein, dass XML-basierte Sprachen für die Geschäftsprozessautomation mit dieser Notation visualisiert werden können, z. B. Business Process Execution Language for Web-Services (BPEL4WS).

9.1 Prozessklassen und das Business process diagram

Business Process Model and Notation (BPMN) sieht zur Prozessbeschreibung den Modelltyp **Business process diagram (BPD)** vor. Dieses Modell bildet drei Klassen von Geschäftsprozessen und deren Beziehung untereinander ab:

- **Private Business Processes** (interne Geschäftsprozesse)
- **Abstract Business Processes** (öffentliche Geschäftsprozesse)
- **Collaboration Processes** (globale Geschäftsprozesse)

Private Business Processes sind Geschäftsprozesse, die nur innerhalb einer Organisation ablaufen. Sie sind allgemein bekannt unter dem Namen Workflow oder BPM-Prozesse.

Verschiedene interne Geschäftsprozesse werden als Sequence flow innerhalb der Grenzen von einzelnen Pools (siehe Kapitel Realisierung der BPMN in ARIS (Seite 171)) modelliert, deren Interaktion untereinander mittels Message flow dargestellt wird.

BPMN verwendet anstelle des Begriffs Kontrollfluss die Begriffe Sequence flow und Message flow, da nicht nur die Ereignisse den Prozess steuern, sondern auch die ausgetauschten Nachrichten.

Abstract Business Processes beschreiben Interaktionen zwischen Private Processes unterschiedlicher Pools, zwischen Objekten verschiedener Pools oder Kombinationen aus beiden. Hier ist - neben dem Sequence flow innerhalb des Private Process - der Message flow zwischen den einzelnen Prozessen von besonderer Bedeutung. Die Interaktionen werden über Message flows modelliert.

Abstract Business Processes sind in einzelne Pools eingebunden und können separat oder innerhalb eines ganzen BPMN-Diagramms modelliert werden. Wenn ein Abstract Business Process im gleichen Modell erscheint wie sein entsprechender Private Business Process, können sie miteinander assoziiert werden.

Collaboration Processes beschreiben ausschließlich die Interaktionen zwischen zwei oder mehreren Business Entities (Geschäftspartnern). Dabei wird eine Abfolge von Aktivitäten modelliert, die das Nachrichtenaustauschmuster zwischen den verschiedenen Geschäftspartnern beschreibt. Der Sequence flow spielt dabei keine Rolle mehr.

Relevante Sprachen für Collaborations sind beispielsweise bXML BPSS, RosettaNet, W3C Choreography Working Group. Die Mapping-Spezifikation ist für weitere Versionen der BPMN-Spezifikation geplant.

Collaboration Processes können in Pools eingebunden werden. Die Interaktionen der beteiligten Partner werden in einzelnen Lanes beschrieben. Die Prozesse können auf diese Weise separat oder innerhalb eines umfassenden BPMN-Diagramms modelliert werden. Wenn eine Collaboration im gleichen Diagramm erscheint wie einer ihrer internen Prozesse, können Aktivitäten, die in beiden gebräuchlich sind, miteinander assoziiert werden.

Aus diesen drei Prozessklassen können wiederum verschiedene Arten von Geschäftsprozessen abgeleitet werden:

- Private Business Processes auf höherer Ebene
- Private Business Processes auf Detailebene (Soll- oder Ist-Prozesse)
- Abläufe zwischen Detailprozessen und externen Entities
- Abläufe zwischen Detailprozessen
- Abläufe zwischen Detailprozessen und abstrakten Prozessen
- Abläufe zwischen Detailprozessen und Collaboration-Prozessen
- Abläufe zwischen abstrakten Prozessen
- Abläufe zwischen abstrakten Prozessen und Collaborations
- Abläufe zwischen Collaborations
- Abläufe zwischen mehreren Detailprozessen durch ihre abstrakten Prozesse
- Abläufe zwischen mehreren Detailprozessen durch eine Collaboration

- Abläufe zwischen mehreren Detailprozessen durch ihre abstrakten Prozesse und einer Collaboration

Folgende Abbildung zeigt beispielhaft ein BPMN Collaboraton Diagram mit zwei Geschäftspartnern, denen ein eigener Prozess zugeordnet ist. Beide Detailprozesse bestehen aus einem Startereignis, Aktivitäten, Sequence flow-Kanten und einem Endereignis. Zwischen den Aktivitäten der beiden Detailprozesse sind Message flow-Kanten modelliert.

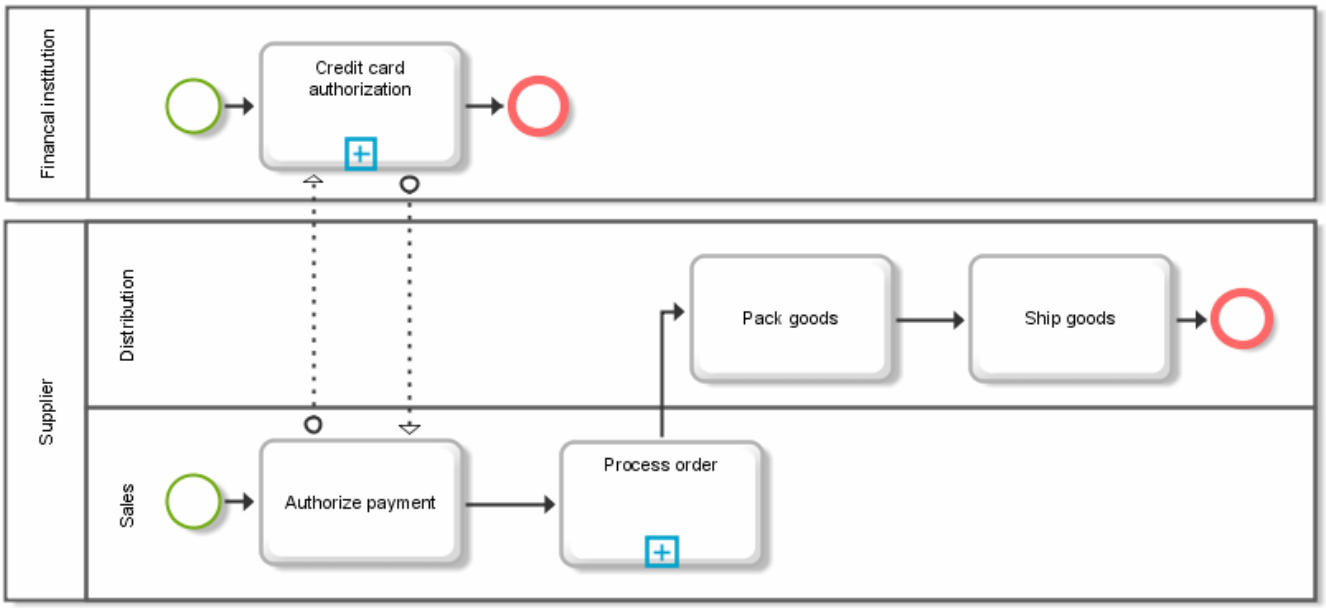


Abbildung 142: Zwei Pools mit Sequence- und Message flow

Da innerhalb eines BPMN Collaboraton Diagram Prozesse mehrerer Geschäftspartner dargestellt werden können und jeder Geschäftspartner eine andere Sicht auf den gleichen Prozess hat, ist es sinnvoll einen "Standpunkt" (Point of view) anzugeben. Die BPMN schreibt nicht vor, wie der Standpunkt in einem BPMN Collaboraton Diagram hervorgehoben wird. Am einfachsten ist es, die Namen der zugeordneten Business Entities (Geschäftspartner) im Attribut **Beschreibung/Definition** zu pflegen (vgl. Abbildung).

9.2 Realisierung der BPMN in ARIS

Obwohl die BPMN nur den Modelltyp **Business process diagram (BPD)** vorsieht, können in ARIS zwei Modelltypen genutzt werden: die EPK und der neue Modelltyp **Business process diagram (BPD)**. Auf diese Weise können bereits vorliegende Prozesse in ARIS als Private Processes wieder verwendet werden. Die EPK besitzt alle Modellattribute, die laut BPMN für das Business process diagram vorgesehen sind. Indem Sie jedoch den Modelltyp Business process diagram (BPD) verwenden, bleiben existierende Modelle des Typs EPK frei von Aspekten des B2B-Kontextes. Damit wird die Komplexität der EPK-Modelle nicht durch zusätzliche Beziehungstypen erhöht.

Das neue Business process diagram erbt wiederum alle BPMN-relevanten Modellattribute der EPK und alle Sequence flow-relevanten Objekte, Kanten und Symbole. Im neuen Modelltyp Business process diagram (BPD) können Sequence flow-relevante Konzepte der EPK wieder verwendet werden. Zusätzlich können aber auch Pools, Lanes und Message flows abgebildet werden.

9.3 Die Elemente des Business process diagram

9.3.1 Pools und Lanes

Ein Business process diagram durch Pools ist strukturiert.

Ein Pool ist ein grafischer Container, in dem ein Satz von Aktivitäten eines Business Entity zusammengefasst wird.

Ein Business Entity kann entweder eine Funktion, ein Anwendungssystem, ein Organisationselement (dazu zählen Organisationseinheit, Organisationseinheitstyp, Gruppe, Rolle, Stelle, Person, Standort, Systemorganisationseinheit und Systemorganisationseinheitstyp) **oder ein Datenelement** (dazu zählen Fachbegriff, Cluster/Datenmodell, Entitytyp, Beziehungstyp, ERM-Attribut, Business-Objekt, Komplexer Objekttyp, COT-Attribut, Class und Informationsträger.) sein.

In BPMN repräsentieren zwei Pools zwei verschiedene Business Entities. Die Strukturierung eines Modells in Pools wird typischerweise im B2B-Kontext verwendet.

Ein Pool bindet die unterschiedlichen Aktivitäten eines Prozesspartners, die mittels Lanes strukturiert und organisiert werden. Auf diese Weise wird eine Abgrenzung zu den Aktivitäten anderer Prozesspartner deutlich (vgl. vorherige Abbildung **Zwei Pools mit Sequence- und Message flow** (Seite 169)).

In einem BPD muss ein Pool nicht notwendigerweise Prozesselemente enthalten. Man kann auch einen leeren Pool („Black Box“) in ein Modell einfügen, wenn man beispielsweise die Zusammenhänge eines beteiligten, aber in den Details unbekannten Teilprozesses (z. B. eines Geschäftspartners) in einen Gesamtprozess integrieren möchte. Vielleicht möchte man aber auch nur aus Komplexitätsgründen auf die Darstellung von Details eines Teilprozesses verzichten (vgl. Abbildung **E-Mail Voting Process** (Seite 184)).

Pools bestehen aus mindestens einer Lane. Eine Lane kann wiederum weitere Lanes enthalten, die ineinander geschachtelt sind, oder als Matrix definiert werden. Besteht ein Pool aus einer einzigen Lane, erhält der Pool den gleichen Namen wie die Lane. Wenn ein Pool mehr als eine Lane umfasst, müssen die verschiedenen Lane-Namen und ein spezieller Pool-Name angegeben werden.

Abbildung 143: Pool mit zwei Lanes nach BPMN

In ARIS sind Pools und Lanes eigene Objekttypen, die zunächst im Modell platziert werden. Innerhalb des Pools kann der Prozess vergleichbar einer EPK modelliert werden. Alle Funktionen, Ereignisse und Regeln des Prozesses werden auf dem Pool-Objekt platziert. Um die Zugehörigkeit dieser Objekte zu einem Pool zu beschreiben, kann die Kante **gehört zu** verwendet werden. Es empfiehlt sich, diese als implizite Beziehung anzulegen. Das Pool-Objekt wird durch die Beziehung des Typs **bildet ab** mit einem Organisationselement, einem Anwendungssystemtyp-Objekt, einem Datenelement oder einer Funktion verbunden. Hierbei ist zu beachten, dass jeder Pool datenbankweit nur eine Kante dieses Typs besitzen darf. Auch diese Beziehungen sollen implizit angelegt werden

Ein Pool muss gemäß der BPMN nicht durch ein Symbol im Modell repräsentiert werden. Auch dürfen die Grenzen **eines** Pools ausgeblendet sein, insbesondere wenn das Diagramm nur einen Pool enthält (vgl. Abbildung **E-Mail Voting Process** (Seite 184)). Um die Übersichtlichkeit eines Modells mit mehreren Pools sicherzustellen, wird empfohlen, auf den Gebrauch dieser Optionen zu verzichten.

9.3.2 Modellierungsrichtlinien für Pools und Lanes

- Innerhalb eines Diagramms darf es nur einen Pool geben, dessen Grenzen nicht sichtbar sind.
- Wenn das Attribut **Pool type** auf **Collaboration**gesetzt ist, soll kein Owner (Attribut **Verantwortlicher**) gepflegt sein.
- Jede Lane darf nur einen übergeordneten Pool besitzen.

9.3.3 Sequence flow

Ein Prozess in Form eines Sequence flow beschreibt die Reihenfolge, in der die Aktivitäten eines Prozesses ausgeführt werden. Der Sequence flow verbindet die Objekttypen **Ereignis**, **Aktivität** und **Gateway** miteinander. Sequence flows sind nur innerhalb der Grenzen von Pools erlaubt und dürfen diese nicht überschreiten (vgl. nächste Abbildung).

Der Sequence flow wird durch eine durchgezogene Linie mit einer schwarzen Pfeilspitze dargestellt:

Abbildung 144: Sequence Flow-Kante

Abhängig von der Art des Quell- und Zielobjekts der Kante werden entsprechende Kantentypen gepflegt, z. B. aktiviert, wird ausgewertet von, erzeugt, verknüpft, aktiviert oder führt zu.

9.3.4 Modellierungsrichtlinien für Sequence flow-Kanten

- Für Sequence flows, die auf ein XOR-databased Gateway oder ein Inclusive Gateway folgen, muss für das Attribut **Condition** ein Wert gesetzt sein.
- Wenn im Attribut **Condition** der Wert **Expression** gesetzt ist, soll am Kantenanfang der Diamant als Symbol platziert werden.
- Wenn das Attribut **Condition** den Wert **Default** hat und das Quellobjekt eine Funktion ist, soll am Kantenanfang das Zeichen **** (umgekehrter Schrägstrich) als Symbol platziert werden.
- Das Zeichen **** (umgekehrter Schrägstrich) darf nicht platziert werden, wenn das Quellobjekt ein Gateway ist.
- Wenn es sich bei dem Quellobjekt um eines der folgenden Symbole handelt, soll keine Condition gesetzt werden:

  - Event-based Gateway
  - Complex Gateway
  - Parallel Gateway
  - Start Event
  - Intermediate Event
- Wenn für eine Sequence flow-Kante der Wert **Default** des Attributs **Condition** aktiviert ist, darf keine Condition gepflegt sein.
- Das Attribut **Condition** darf den Wert **Default** haben, wenn das Quellobjekt eine Funktion oder ein XOR-databased Gateway ist.
- Wenn für das Attribut **Condition** der Wert **Expression** gesetzt ist, muss auch das Attribut **Expression** gepflegt sein.

9.3.5 Message flow

Ein Message flow beschreibt den Austausch von Informationen zwischen zwei Pools. Der Message flow kann sich entweder direkt zwischen den beiden Pool-Objekten befinden oder zwischen Objekten im Sequence flow der Prozesse des jeweiligen Pools. Nur Message flows dürfen Pool-Grenzen überschreiten und eine Message flow-Kante darf nicht zwischen zwei Objekten des gleichen Pools angelegt werden (vgl. Abbildung **Zwei Pools mit Sequence- und Message flow** (Seite 169)).

Die Kante wird durch eine gestrichelte Linie dargestellt. Am Anfangspunkt der Linie befindet sich ein Kreis, am Endpunkt eine weiße Pfeilspitze.

Abbildung 145: Message flow-Kante

Jeder Message flow besteht aus einem Sender-Objekt, einer Kante des Typs **sendet**, einer Kante des Typs **wird empfangen von** und dem Empfänger. Von einem Startereignis und einem Zwischenereignis gehen keine Message flow-Kanten aus. Ein Endereignis empfängt hingegen keine Message flows, kann aber selbst Sender sein. Lanes, Gateways, Datenobjekte und Textannotationen sind von Message flows ausgeschlossen.

9.3.6 Modellierungsrichtlinien für Message flow-Kanten

Quell- und Zielobjekt müssen unterschiedlichen Pools angehören.

9.3.7 Association

Eine Association wird verwendet, um die Bestandteile von Sequence- oder Message flow mit Informationen zu versehen. Diese Informationen können textueller oder grafischer Art sein. Wenn mehrere verschiedene Prozesse im gleichen Diagramm erscheinen, können ihre einzelnen Prozesselemente miteinander über Kanten assoziiert werden.

Die Association wird durch eine gepunktete Linie dargestellt, die bei Bedarf mit offenen Pfeilspitzen versehen werden können. Dies ist insbesondere dann der Fall, wenn es sich um Zuordnungen von Artefakten des Typs **Data Object** handelt.

Abbildung 146: Association-Kanten

Abhängig von der Art des Quell- und Zielobjekts der Kante werden entsprechende Kantentypen gepflegt, z. B. **hat als Ausgabe**, **ist Eingabe für**, **liefert Input für** oder **erzeugt Output auf**.

Von besonderer Bedeutung ist in der BPMN die Zuordnung von Artefakten des Typs Data Object zu Aktivitäten.

Diese Zuordnung ist gerichtet und beschreibt, wie Informationen innerhalb eines Prozesses verwendet und verändert werden. Sie wird im BPD (BPMN) durch folgende Beziehungen vorgenommen:

Funktion - erzeugt Output auf - Datenelemente (insbesondere Informationsträger)

Datenelement (insbesondere Informationsträger) - liefert Input für - Funktion

9.3.8 Ereignisse

Ein Ereignis (Event) ist ein Zustand, der während des Ablaufs des Geschäftsprozesses eintritt. Ereignisse beeinflussen den Ablauf des Prozesses. Gewöhnlich stellen sie Auslöser oder Wirkungen innerhalb der Prozesse dar. In Abhängigkeit davon, wann ein Ereignis eintritt, werden drei Ereigniskategorien differenziert: Startereignis, Zwischenereignis und Endereignis. Die drei Kategorien von Events werden in BPMN durch unterschiedliche Symbole dargestellt:

Abbildung 147: Ereigniskategorien

Innerhalb dieser Kategorien werden weitere Spezialisierungen vorgenommen. Die Symbole der drei Ereigniskategorien werden mit der Pflege des Attributs **Event Type** durch zusätzliche Symbole erweitert, wie die drei folgenden Beispiele zeigen:

Abbildung 148: Beispiele für Ereignistypen

Alle für den Objekttyp **Ereignis** relevanten Attribute werden in der Attributtypgruppe **BPMN** zusammengefasst.

9.3.9 Modellierungsrichtlinien für Ereignisse

- Für Start events darf der Attributtyp **Event type** nur einen der folgenden Werte besitzen: **Message**, **Timer**, **Rule**, **Link**, oder **Multiple**.
- Für End events darf der Attributtyp **Event type** nur einen der folgenden Werte besitzen: **Message**, **Exception**, **Cancel**, **Compensation**, **Rule**, **Link**, **Multiple**, oder **Terminate**.
- Für Intermediate events darf der Attributtyp **Event type** nur einen der folgenden Werte besitzen: **Message**, **Timer**, **Exception**, **Cancel**, **Compensation**, **Rule**, **Link**, und **Multiple**.
- In Abhängigkeit vom gesetzten Ereignistyp müssen entsprechende Attribute mit Zusatzinformationen gepflegt werden.
- Ein Start event darf mehrere ausgehende Sequence flow-Kanten besitzen. Für das Attribut **Condition** dieser Kanten darf kein Wert gesetzt sein.
- Intermediate events, die eine Exception oder eine Compensation anzeigen, sollen am Rand der Funktion platziert werden.
- Ist ein Intermediate event am Rande einer Funktion platziert, muss ein Wert, jedoch nicht **Link**, gepflegt sein.
- Die Werte **Multiple**, **Rule** und **Cancel** dürfen nicht für Intermediate events gesetzt werden, die sich im normale Sequence flow eines Prozesses befinden.
- Der Wert **Cancel** darf nicht gesetzt werden, wenn
  - das Intermediate event am Rande einer Funktion platziert und das Attribut **Transaction** der Funktion nicht aktiviert ist oder
  - das Ereignis nicht zu einem Prozess gehört, der eine Transaktion beschreibt.
- Wenn ein Intermediate event am Rande einer Funktion platziert ist, darf es nicht Zielobjekt einer Sequence flow-Kante sein.
- Wenn sich ein Intermediate event innerhalb des normalen Sequence flow eines Prozesses befindet, d. h., es nicht am Rande einer Funktion platziert ist, darf es genau eine eingehende Sequence flow-Kante besitzen. Für das Attribut **Event type** des Ereignisses muss dann kein Wert oder darf einer der folgenden Werte gepflegt sein: **Message**, **Timer**, **Exception**, **Link**, oder **Compensation**.
- Der Wert **Link** darf für Intermediate events im normalen Sequence flow nur dann gesetzt werden, wenn das Quellobjekt ein Gateway ist, für dessen Attribut **Gateway type** der Wert **XOR (Event-based)** gepflegt ist.
- Jedes Intermediate event muss genau eine ausgehende Sequence flow-Kante besitzen.
- Ein Intermediate event, für dessen Attribut **Event type** der Wert **Message** gepflegt ist, darf einen eingehenden Message flow besitzen (eingehende Kante des Typs **wird empfangen von**).

- Ein Intermediate event darf keinen ausgehenden Message flow besitzen (ausgehende Kante des Typs **sendet**).

9.3.10 Aktivitäten

Eine Aktivität ist eine Tätigkeit, die innerhalb eines Prozesses ausgeführt wird. Sie kann atomar oder nichtatomar (zusammengesetzt) sein. BPMN kennt drei Kategorien von Aktivitäten: Process, Subprocess und Task.

Folgende Symbole sind gemäß BPMN für Aktivitäten vorgesehen:

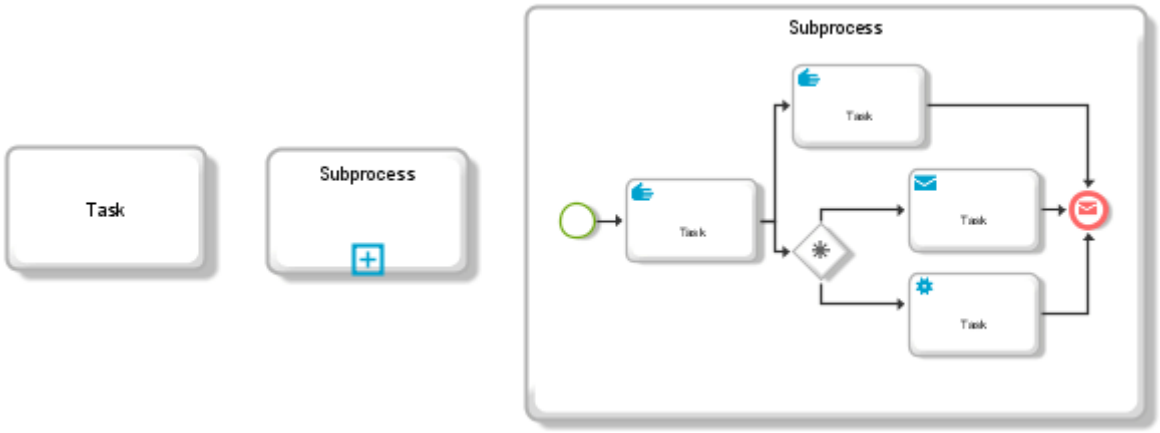


Abbildung 149: Aktivitäten nach BPMN

In ARIS werden Aktivitäten standardmäßig als Funktionen abgebildet:

Abbildung 150: Hinterlegte Funktion als Aktivität in ARIS

Die Funktion erhält alle Attribute, die seitens der BPMN für Processes, Subprocesses und Tasks vorgesehen sind. Wie für Ereignisse wird die Attributtypgruppe **BPMN** genutzt, die zusätzliche Untergruppen für die Activity-Typen enthält.

Im Sinne der BPMN beschreibt ein Process eine Aktivität, die innerhalb eines Unternehmens oder einer Organisation durchgeführt wird. Ein Prozess wird durch einen Graphen mit Flussobjekten beschrieben, die einen Satz von anderen Aktivitäten und Steuerungsobjekten darstellen. Prozesse sind hierarchisch strukturiert und können somit auf jeder Detaillierungsstufe definiert werden. Im Gegensatz zum Prozess beschreibt ein Business Process (Geschäftsprozess) in der BPMN einen Satz von Aktivitäten, die über Unternehmens-/Organisationsgrenzen hinweg ausgeführt werden.

Im Sinne der BPMN ist ein Sub-Process eine zusammengesetzte Aktivität, die eine Detailbeschreibung besitzt. Ein Sub-Process erscheint als Objekt innerhalb eines Prozessablaufs (Process flow).

Ein Sub-Process ist in der Regel mit einem detaillierten Prozess hinterlegt. In ARIS wird eine hinterlegte Aktivität nicht wie in BPMN durch ein Pluszeichen markiert, sondern auch durch das Hinterlegungssymbol gekennzeichnet.

Neben der Kennzeichnung einer hinterlegten Funktion sieht BPMN auch vor, den Detailprozess auf der nächst höheren Prozessebene einzublenden. Dies geschieht durch Klicken auf das Pluszeichen.

9.3.11 Modellierungsrichtlinien für Aktivitäten

PROCESS

- Wenn das Attribut **Ad hoc = True** ist, dann muss das Attribut **Completion Condition** gepflegt sein.
- Wenn ein Ad hoc-Prozess verfeinert ist, dürfen innerhalb des hinterlegten Modells keine Sequence flows modelliert sein.

SUB-PROCESS

- Wenn für das Attribut **Subprocess Type** der Wert **Independend** verwendet wird, muss auch das Attribut **Process Reference** gepflegt werden.
- Wenn für einen Subprozess das Attribut **Transaction** aktiviert ist, muss auch das Attribut **Transaction ID** gepflegt sein.
- Wenn das Attribut **Loop Type** gepflegt ist, muss auch das Attribut **Loop Condition** gepflegt sein.
- Wenn die Modelle in BPEL4WS übertragen werden sollen, sollte auch geprüft werden, ob für Prozesse mit dem Wert **Standard** für das Attribut **Loop Type** das Attribut **Maximum** gepflegt ist.
- Wenn für das Attribut **Loop Type** der Wert **Standard** gepflegt ist, muss auch das Attribut **Test before** gepflegt sein. Das Attribut **Test before** soll standardmäßig deaktiviert sein.
- Wenn für das Attribut **Loop Type** der Wert **Multi Instance** gepflegt ist, soll auch das Attribut **Parallel Instance Generation** gepflegt sein. Das Attribut **Parallel Instance Generation** soll standardmäßig deaktiviert sein.
- Wenn das Attribut **Loop Type** eines Subprozesses den Wert **Multi Instance** besitzt und zugleich das Attribut **Parallel Instance Generation** aktiviert ist, muss auch das Attribut **Loop Flow Condition** gepflegt sein.

- Wenn in einem Prozess für das Attribut **Loop Flow Condition** der Wert **Complex** gesetzt ist, muss für das Attribut **Complex** ein Ausdruck gepflegt werden, der bestimmt, wann und wie viele Prozessmarken nach dem Subprozess weitergegeben werden.

TASK

- Wenn für das Attribut **Task Type** der Wert **Receive** gepflegt ist, soll die Funktion keine ausgehenden Message flow-Kanten besitzen.
- Wenn für das Attribut **Task Type** der Wert **Send** gepflegt ist, soll die Funktion keine eingehenden Message flow-Kanten besitzen.
- Wenn das Attribut **Task Type** nicht gepflegt ist oder die Werte **Script** oder **Manual** gesetzt sind, soll die Funktion keine ausgehenden und keine eingehenden Message flow-Kanten besitzen.
- Für Funktionen, für deren Attribut **Task Type** der Wert **Abstract** gepflegt ist, muss auch das Attribut **Abstract Type** gepflegt werden. Darüber hinaus dürfen sie nur in Pools des Typs **Abstract** oder in **Collaborations** verwendet werden.

9.3.12 Gateway

Gateways beschreiben, wie Sequence flows innerhalb eines Prozesses zusammenlaufen oder verzweigen. Sie bestimmen das Verhalten einkommender und ausgehender Kanten. In ARIS werden sie als Objekte des Typs **Regel** dargestellt.

Ähnlich den Events können Gateways unterschiedlichen Typs spezifiziert werden. Abhängig vom Typ werden weitere Symbole in der Mitte des Gateway-Symbols angezeigt.

Einige differenzierende Gateway-Symbole:

Abbildung 151: Gateway-Typen

Die Spezifikation der BPMN sieht vor, dass für jedes Gateway eine Anzahl von Gates definiert werden muss. In ARIS wird die Anzahl der Gates durch die Anzahl der ein- und ausgehenden Kanten bestimmt. Gate-abhängige Attribute werden daher an den ein- und ausgehenden Sequence flow-Kanten der Regel gepflegt.

Einen Spezialfall stellt das Complex Gateway dar, bei dem die besonderen Attribute **Incoming Condition** und **Outgoing Condition** angegeben werden. Diese Attribute müssen gepflegt sein, wenn es mehrere ein- oder ausgehende Sequence flow-Kanten am betrachteten Gateway gibt. Der Attributinhalt der Incoming Condition kann Sequence flow-Namen und

Prozesseigenschaften (Daten) enthalten. Die Outgoing Condition enthält Referenzen auf Sequence flow-IDs und Prozesseigenschaften (Data).

9.3.13 Modellierungsrichtlinien für Gateways

- Gateways des Typs XOR (data-based): Für alle ausgehenden Kanten eines XOR-Gateway (data-based) ist für das Attribut **Condition** der Wert **Expression** gesetzt und für das Attribut **Condition Expression** wird ein gültiger Ausdruck verwendet.

Sequence flow, speziell nach Gateways

- Für jedes XOR-Gateway des Typs **XOR (Data-based)** sollte an genau einer ausgehenden Sequence flow-Kante (Kantentyp **aktiviert**) das Attribut **Default-Gateway** gepflegt sein. Es dürfen in keinem Fall mehrere ausgehende Kanten mit diesem Attribut gekennzeichnet sein.
- Für jedes XOR-Gateway des Typs **XOR (Event-based)** muss es mindestens zwei ausgehende Sequence flow-Kanten geben (Typ **aktiviert** oder **führt zu**).
- Für alle ausgehenden Kanten eines Event-based XOR-Gateway darf für das Attribut **Condition** kein Wert gepflegt sein. Das Attribut **Condition Expression** sollte nicht gepflegt werden.
- Folgende Zielobjekte sind für ausgehenden Sequence flow-Kanten eines Event-based XOR-Gateway erlaubt:
  - Funktion, für die der Task Type **Receive** gesetzt ist.
  - Intermediate Events, deren Attributtyp **Event Type** einen anderen Wert besitzen als **Compensation** oder **Multiple**.
- Befindet sich in der Menge der Zielobjekte eine Funktion, darf in dieser Menge kein Event des Typs **Message** vorkommen.
- Besitzt ein Gateway des Typs **OR** keine oder genau eine eingehende Sequence flow-Kante, müssen mindestens zwei ausgehende Sequence flow-Kanten existieren.
- Für alle ausgehenden Sequence flow-Kanten eines OR-Gateway ist für das Attribut **Condition** der Wert **Expression** gesetzt und für das Attribut **Condition Expression** wird ein gültiger Ausdruck verwendet. Der Ausdruck muss auf das aktuelle Gateway bezogen eindeutig sein.
- Besitzt ein OR-Gateway genau eine ausgehende Sequence flow-Kante, darf für das Attribut **Condition** dieser Kante kein Wert gepflegt sein.
- Besitzt ein Gateway des Typs **Complex** keine oder genau eine eingehende Sequence flow-Kante, müssen mindestens zwei ausgehende Sequence flow-Kanten existieren.

- Für alle ausgehenden Kanten eines Complex Gateway muss für das Attribut **Condition** der Wert **None** gepflegt sein, speziell dann, wenn es nur eine ausgehende Kante gibt.
- Wenn ein Complex Gateway mehrere eingehende Sequence flow-Kanten besitzt, muss für das Attribut **Incoming Condition** eine Bedingung gepflegt werden, die Sequence flow-Namen und Prozesseigenschaften (Data) referenziert.
- Wenn ein Complex Gateway mehrere ausgehende Sequence flow-Kanten besitzt, muss für das Attribut **Outgoing Condition** eine Bedingung gepflegt werden, die Sequence flow-Namen und Prozesseigenschaften (Data) referenziert.
- Wenn ein AND-Gateway keine oder genau eine eingehende Sequence flow-Kante besitzt, müssen mindestens zwei ausgehende Sequence flow-Kanten existieren.
- Für alle ausgehenden Sequence flow-Kanten eines AND-Gateway darf für das Attribut **Condition** kein Wert gepflegt sein.

9.3.14 Artifact

Artifacts stellen Informationen über den Prozess zur Verfügung, die nicht zu Sequence Flow und nicht zum Message flow gehören. Es werden insgesamt drei Artifact-Typen differenziert: **Data Objects**, **Groups** und **Annotations** (die Typliste ist beliebig erweiterbar).

Data Objects sind mit dem Informationsträger oder den Datenelementen in ARIS vergleichbar. Im weitesten Sinne könnten sie aber alle Zuordnungen umfassen. Data Objects beeinflussen weder den Sequence flow noch den Message flow, sie liefern aber Informationen darüber, was während des Prozesses geschieht. Sie geben Aufschluss darüber, wie sich Dokumente, Daten und andere Objekte während des Prozesses verändern.

Bei einer **Group** handelt es sich um die grafische Hervorhebung zusammengehörender Prozesselemente. In ARIS bieten sich dazu Grafikobjekte wie Rechteck oder Polygonzug an.

Alternativ können auch Gruppierungen in Betracht gezogen werden. Dies ist allerdings nur dann sinnvoll, wenn die Gruppierung eine Grafik umfasst.

Annotations entsprechen Bemerkungen zu Objekten oder Kanten, wie in folgendem Beispiel **Time out [1week]**. In ARIS werden sie oft mit Hilfe des Attributs **Bemerkung/Beispiel** realisiert. Wichtig ist, dass dieses Attribut im Modell platziert wird, wie in folgendem Beispiel für **Yes** und **No** geschehen.

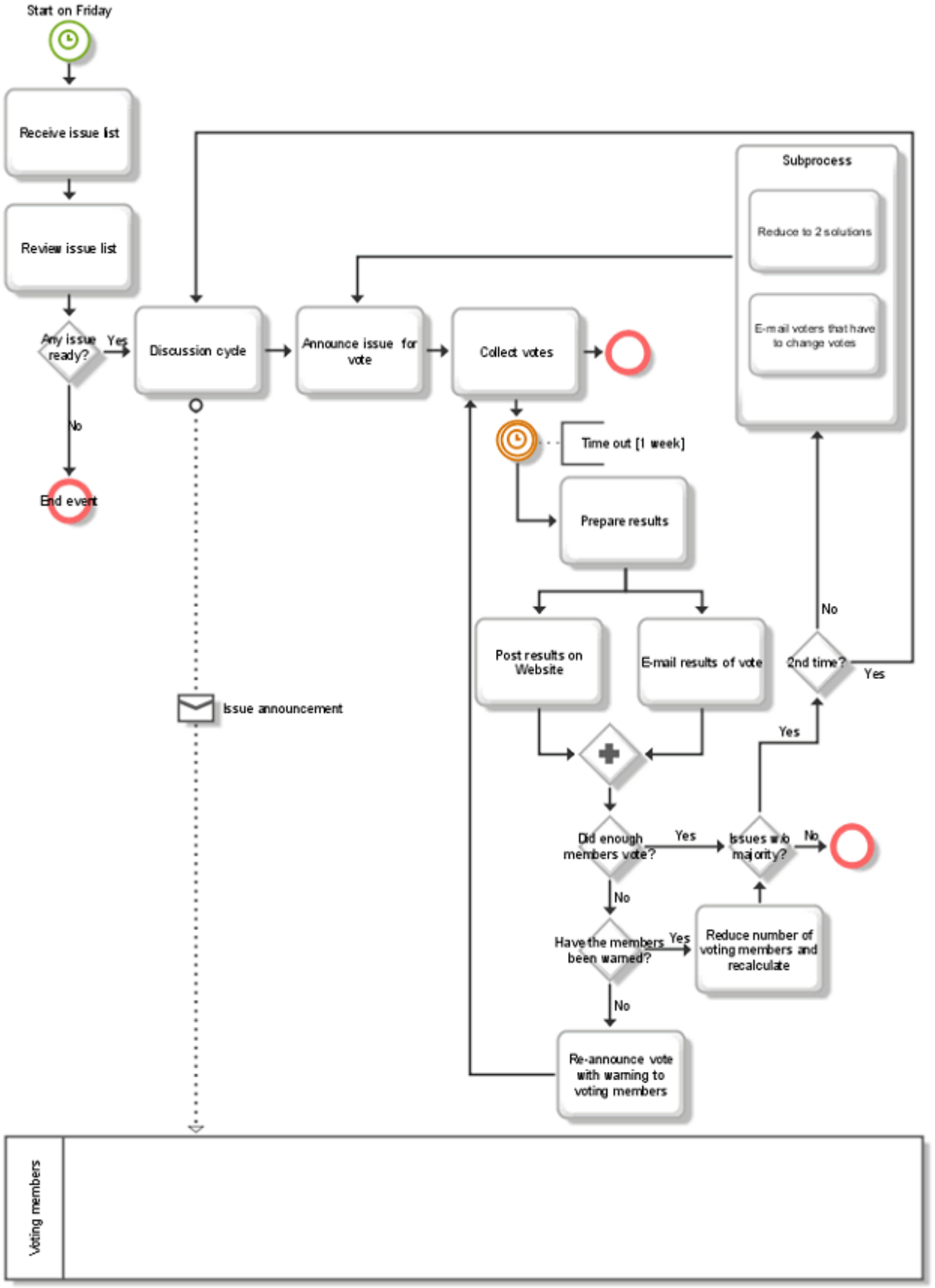


Abbildung 152: E-Mail Voting Process

Diese Abbildung zeigt die beispielhafte Realisierung eines Business Collaboration Diagram gemäß der BPMN 2.0 in ARIS. Das Diagramm umfasst zwei Pools, wobei die Pool-Grenzen des oberen Pools ausgeblendet sind. Bei dem unteren Pool wurde auf die Darstellung der einzelnen Elemente verzichtet.

9.3.15 Quellenangabe zu den Abbildungen

Abbildung **Zwei Pools mit Sequence- und Message flow** (Seite 169):

Business Process Modeling Notation, Working Draft (1.0); BPMI.org; August 25, 2003; Seite 85.

Abbildung **Pool mit zwei Lanes nach BPMN** (Seite 172):

Business Process Modeling Notation, Working Draft (1.0); BPMI.org; August 25, 2003; Seite 87.

Abbildung **Ereigniskategorien** (Seite 177)und Abbildung **Beispiele für Ereignistypen** (Seite 177):

Business Process Modeling Notation, Working Draft (1.0); BPMI.org; August 25, 2003; Seite 27.

Abbildung **Aktivitäten nach BPMN** (Seite 179):

Business Process Modeling Notation, Working Draft (1.0); BPMI.org; August 25, 2003; Seite 28.

Abbildung **Gateway-Typen** (Seite 181):

Business Process Modeling Notation, Working Draft (1.0); BPMI.org; August 25, 2003; Seite 28.

10 Modeling BPMN 2.0

10.1 Introduction

10.1.1 Initial situation and objective

BPMN (Business Process Modeling and Notation) has emerged as a widely adopted standard for process modeling. Its popularity is based on the fact that it is has been developed by the Object Management Group (OMG), a consortium of organizations that also released other important modeling standards like UML.

The primary goal of BPMN is to provide a notation that is understandable by all users: business analysts designing and documenting business processes, developers implementing these business processes, and business end users executing, managing and monitoring their business processes. Now, the OMG released a new version of BPMN 2.0. This standard shall be supported by ARIS. In a first step, the objective is to focus on process modeling conformance, one of four conformance types defined by the OMG.

The four conformance types are described in detail in the BPMN specification: Business Process Model and Notation (BPMN), version 2.0.

10.1.2 Purpose of this chapter

Unfortunately the BPMN specification has increased an order of magnitude in technical complexity and fails to distinguish those elements needed for business process modeling from those required for process execution.

The purpose of this chapter is to describe the ARIS implementation of the BPMN 2.0 elements that are part of business process modeling documenting the process flow. Those parts that are needed for executable design are ignored. The elements relevant for business process modeling are essentially those displayed in a diagram.

The mapping described in the chapters of this document is based on the BPMN specification **Business Process Model and Notation (BPMN)**, **version 2.0** (https://www.bpmn.org).

The attribute and model association tables are also taken from the BPMN 2.0 specification and extended to describe the implementation in ARIS.

10.2 BPMN core elements and their implementation in ARIS

The BPMN core consists of four packages:

- Foundation
- Infrastructure
- Common Elements as well as
- Service

It provides the basis for modeling processes, collaborations, choreographies and conversations. These packages are described in detail in chapter 8 of the BPMN specification.

In the following sections the core constructs and their attributes and associations are mapped to ARIS constructs.

10.2.1 Infrastructure

The infrastructure package consists of two elements which are particularly relevant for import and export. Thus, their attributes and model associations are not included in the current version of the BPMN 2.0 implementation.

See: Business Process Model and Notation (BPMN), Version 2.0.

10.2.2 Foundation

The foundation package contains classes which are shared amongst other packages in the BPMN core. The foundation package consists of eight classes: BaseElement, Documentation, RootElement, Extension, Extension Definition, ExtensionAttributeDefinition, ExtensionAttributeValue and Relationship. See: Business Process Model and Notation (BPMN), version 2.0.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
BaseElement	id: string	The ARIS GUID of the corresponding modeling construct represents the BPMN ID. For imported BPMN elements an attribute type in the attribute type group **Attributes of external systems** will be used.
	documentation: Documentation [0..*]	see below: Documentation
	extensionDefinitions: ExtensionDefinition [0..*]	ARIS-Methode can be enhanced, for example, by user-defined attributes.
	extensionValues: ExtensionAttributeValue [0..*]	The ARIS-Methode can be enhanced, for example, by user-defined attributes.
Documentation	inherits from BaseElement	
	text: string	All ARIS attribute types assigned to model types, object types, and connection types can be used for documentation purposes. The attribute types Description/Definition (AT_DEC) and Remark/Example (AT_REM) should be used to for general information. Specific attribute types should be used to store specific information.

Class	BPMN attribute name	Implementation in ARIS
Extension	mustUnderstand: boolean [0..1] = False	Currently not implemented.
	definition: ExtensionDefinition	
ExtensionDefinition	name: string	Currently not implemented.
	extensionAttributeDefinitions: ExtensionAttributeDefinition [0..*]	
ExtensionAttributeDefinition	name: string	Currently not implemented.
	type: string	
	isReference: boolean [0..1] = False	
ExtensionAttributeValue	value: Element [0..1]	Currently not implemented.
	valueRef: Element [0..1]	
	extensionAttributeDefinition: ExtensionAttributeDefinition	
Relationship	inherits from BaseElement	Currently not implemented.
	type: string	
	direction: RelationshipDirection {none | forward | backward | both}	
	sources: Element [1..*]	
	targets: Element [1..*]	
RootElement	inherits from BaseElement	RootElement is an abstract class, it has no direct representation in ARIS. For example, ARIS object types are root elements, ARIS attribute types are not.

10.2.3 Common Elements

Common Elements are basic elements that may be used in more than one type of diagram, for example, Process, Collaboration, Conversation, and Choreography. The Common Elements are categorized into seventeen different groups.

10.2.3.1 Artifacts

Artifacts are used to depict additional information in a BPMN process diagram (BPMN2.0) or BPMN collaboration diagram (BPMN2.0) that is not directly related to the sequence flow or message flow. BPMN 2.0 provides three standard artifacts:

- Associations,
- Groups, and
- Text annotations

Data objects are no longer artifacts, they are concepts of their own (see chapter Items and Data (Seite 237)).

See: Business Process Model and Notation (BPMN), version 2.0.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Association	inherits from BaseElement	various connection types
	associationDirection: AssociationDirection = None {None | One | Both}	This attribute is represented by the direction and the style of the corresponding ARIS connection type.
	sourceRef: BaseElement	Corresponds to the source object type of the connection type representing the association.
	targetRef: BaseElement	Corresponds to the target object type of the connection type representing the association.

Class	BPMN attribute name	Implementation in ARIS
Group	inherits from BaseElement	Object type: Structural element (OT_STRCT_ELMT) Symbol: Structural element in model type **Structuring model** (MT_STRCT_DGM) Symbol: Group (ST_BPMN_GROUPING_1)
	categoryValueRef: CategoryValue [0..1]	Attribute type **Name** (AT_NAME) of object type Structural element (OT_STRCT_ELMT)
Category	inherits from BaseElement	Object type Structural element (OT_STRCT_ELMT) in model type Structuring model (MT_STRCT_DGM)
	categoryValue: CategoryValue [0..*]	Connection type in model type Structuring model: * Structural element (representing the category) contains structural element (representing the category value).
CategoryValue	inherits from BaseElement	Object type: Structural element (OT_STRCT_ELMT) Symbol: Structural element in model type Structuring model (MT_STRCT_DGM) Symbol: Group (ST_BPMN_GROUPING_1) in BPMN 2.0 diagrams
	value: string	Attribute type **Name** of object type Structural element
	category: Category [0..1]	Connection type in model type Structuring model: * Structural element (representing the category) contains structural element (representing the category value).

Class	BPMN attribute name	Implementation in ARIS
	categorizedFlowElements: FlowElement [0..*]	Connection type **belongs to** [CT_BELONGS_TO_1] in the BPMN process diagram (BPMN 2.0) and BPMN collaboration diagram (BPMN 2.0): Target object type: Structural element (OT_STRCT_ELMT; ST_BPMN_GROUPING_1) Source object types: * Function (OT_FUNC) representing activities * Event (OT_EVT) * Rule (OT_RULE) representing Gateways * Cluster/data model (OT_CLST) representing data objects * Information carrier (OT_INFO_CARR) representing data stores
Text annotation	inherits from BaseElement	

Class	BPMN attribute name	Implementation in ARIS
	text: string	For object types in the BPMN process diagram (BPMN 2.0), BPMN collaboration diagram (BPMN 2.0), and BPMN conversation diagram (BPMN 2.0): * Text annotation (OT_BPMN_ANNOTATION) with symbol **Text annotation** (ST_BPMN_ANNOTATION_1) is associated with <target object type>. Target object types are all object types available in the corresponding model type. For connection types in the BPMN process diagram (BPMN 2.0), BPMN collaboration diagram (BPMN 2.0), and BPMN conversation diagram (BPMN 2.0): three attribute types in the attribute type group BPMN 2.0 attributes/Text annotation attributes: * Text annotation 1 (AT_BPMN_TEXT_ANNOTATION_1) * Text annotation 2 (AT_BPMN_TEXT_ANNOTATION_2) * Text annotation 3 (AT_BPMN_TEXT_ANNOTATION_3)

10.2.3.2 Association

Associations are used to associate information and artifacts with other BPMN elements. Thus, associations are (usually) represented by connection types in ARIS. The relevant connection types are described in the context of the object types being associated.

10.2.3.3 Group

BPMN 2.0 uses three different classes to represent groupings, but there is only one symbol: **Group**. Thus, a group is the graphical representation of a category value.

Categories and their category values are modeled in an auxiliary model of type **Structuring model**.

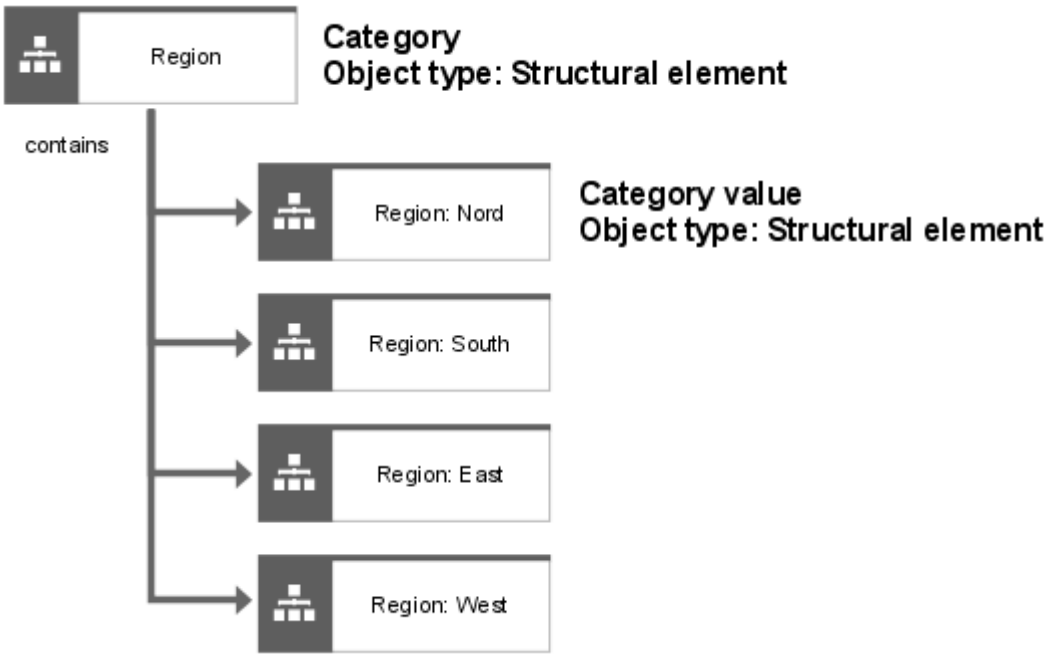


Abbildung 153: Structuring model: Categories and their values

In ARIS the graphical element **Group** is an occurrence copy of a category value object and is depicted by a special symbol in the BPMN 2.0 models. The symbol name is **Group**.

Abbildung 154: Group symbol

10.2.3.4 Text annotation

Text annotations are used to provide additional textual information for the reader of a BPMN model. They can be associated with graphical elements in a model, ARIS objects and connections.

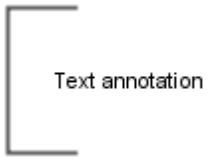


Abbildung 155: Symbol representing text annotations

Text annotations are implemented in ARIS in 2 different ways:

TEXT ANNOTATIONS ASSOCIATED WITH ARIS OBJECTS

The object type **Text annotation** and the connection type **is associated with** is used to annotate objects (occurrences) in a model.

TEXT ANNOTATIONS ASSOCIATED WITH ARIS CONNECTIONS

Objects (here: Text annotation) cannot be assigned to connections. Thus, the program provides a new functionality: The modeler selects the text annotation symbol in the **Symbols** bar, places it on/near by the connection he/she wants to annotate and enters the text. The program draws a line looking like an association and stores the text in a **Text annotation** attribute of the corresponding connection. In the first step three text annotation attributes are provided in the attribute type group **BPMN 2.0 attributes/BPMN text annotations**:

Text annotation 1 (AT_BPMN_TEXT_ANNOTATION_1)

Text annotation 2 (AT_BPMN_TEXT_ANNOTATION_2)

Text annotation 3 (AT_BPMN_TEXT_ANNOTATION_3)

10.2.3.5 Callable Elements

Callable Element is an abstract class and has four specialized classes: Process, Global task, Choreography, and Choreography task. Only processes and global tasks are relevant for business process modeling compliance. They are represented by the object type Function.

See: Business Process Model and Notation (BPMN), version 2.0.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Callable Element	inherits from BaseElement	Object type: Function (OT_FUNC) Symbol: Call activity (ST_BPMN_CALL_ACTIVITY)
	name: string [0..1]	Attribute type Name (AT_NAME) of object type **Function** (OT_FUNC)
	supportedInterfacesRefs: Interface [0..*]	Currently not implemented.
	ioSpecification: InputOutputSpecification [0..1]	Currently not implemented.
	ioBinding: InputOutputBinding [0..*]	Currently not implemented.
InputOutputBinding	inputData: DataInput	Currently not implemented.
	outputData: DataOutput	Currently not implemented.
	operationRef: Operation	Currently not implemented.

10.2.3.6 Event

Events are described in detail in the context of the BPMN process diagram (see chapter Events (Seite 241)).

10.2.3.7 Expression

FormalExpressions belong to the execution design level and are not included in the current version of the BPMN 2.0 implementation.

However, natural-language expressions are used to allow the modeler to specify conditions. They are described in the context of the corresponding BPMN elements (object types and connection types).

10.2.3.8 Flow Element

Flow Elements are described in detail in the context of the BPMN process diagram (see chapter Process (Seite 213)).

See: Business Process Model and Notation (BPMN), version 2.0.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
FlowElement	inherits from BaseElement	No direct representation in ARIS -> abstract class
	name: string [0..1]	Attribute type **Name** (AT_NAME) of the object types representing flow nodes.
	auditing: Auditing [0..1]	Currently not implemented.
	monitoring: Monitoring [0..1]	Currently not implemented.

10.2.3.9 Flow Elements Container

A FlowElementsContainer is an abstract super class for BPMN diagrams (or views). So, Processes and Subprocesses as well as Choreographies and Choreography subprocess are FlowElementsContainers.

The specific attributes and model associations of a process and subprocess are described in detail in the context of the BPMN process diagram.

See: Business Process Model and Notation (BPMN), version 2.0.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
FlowElementsContainer	inherits from BaseElement	Model type **BPMN process diagram (BPMN 2.0)** (MT_BPMN_PROCESS_DIAGRAM)
	flowElements: FlowElement [0..*]	Occurrences of the object types and connection types allowed in a BPMN process diagram (BPMN 2.0).
	artifacts: Artifact [0..*]	Occurrences of the object types and attribute types representing groups and text annotations as well as their connection types allowed in the BPMN process diagram (BPMN 2.0).

10.2.3.10 Gateways

Gateways are described in detail in the context of the BPMN process diagram (see chapter Gateways (Seite 256)).

See: Business Process Model and Notation (BPMN), version 2.0.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Gateway	inherits from FlowElement	Object type: Rule (OT_RULE)
	gatewayDirection: GatewayDirection = unspecified { unspecified | converging | diverging | mixed }	The number of incoming and outgoing sequence flows depends on the modeling context, that is, the position of the gateway in the process. Thus, there is no ARIS attribute type representing the gateway direction. Gateways whose direction is unspecified or mixed should be avoided.

10.2.3.11 Message

Messages normally represent information exchanged between two participants in a BPMN collaboration diagram.

A message is represented by the symbol **Message** of the ARIS object type **Message**.

Abbildung 156: Message symbol

See: Business Process Modeling Notation (BPMN), version 2.0, page 93 and the following.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Message	inherits from BaseElement	Object type: Message (OT_MSG_FLW) Symbol: Message (ST_BPMN_MESSAGE_2)
	name: string	Attribute type **Name** (AT_NAME) of object type Message (OT_MSG_FLW)
	structureRef : ItemDefinition [0..1]	Currently not implemented.

10.2.3.12 Message flow

The message exchange between participants is shown by a message flow that connects two pools or the objects within the pools.

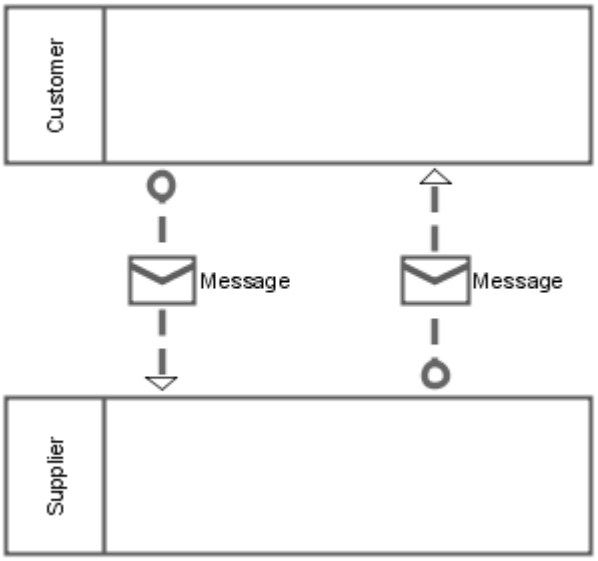


Abbildung 157: Message flow between participants/pools

A message flow is represented in ARIS by the connection type **message flow**. If the message sent from one participant to another should be displayed in the diagram, the connection type **message flow** is replaced by the object type **Message** (symbol **Message**) and two connection types:

- <Source object type> sends message.
- Message is received from <target object type>.

More details can be found in chapter Message flow (Seite 265).

Message flow associations are used to map message flows modeled in two different diagrams, for example, in a conversation and a collaboration diagram. These associations are realized in ARIS by occurrence copies of the message flow connections.

Message flow is also described in the context of the BPMN collaboration diagram (chapter Message flow (Seite 265)) and the BPMN conversation diagram (chapter Message flow in a conversation (Seite 270)).

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
message flow	inherits from BaseElement	Connection type: message flow (CT_BPMN_MESAGE_FLOW)
	name: string	Attribute type **Connection role** of connection type **message flow** (CT_BPMN_MESSAGE_FLOW)
	sourceRef: MessageFlowNode	Source object type of connection type **message flow** (CT_BPMN_MESAGE_FLOW) (Participant, Function, Event)
	targetRef: MessageFlowNode	Target object type of connection type **message flow** (CT_BPMN_MESAGE_FLOW) (Participant, Function, Event)
	messageRef: Message [0..1]	Object type: Message (OT_MSG_FLW) Symbol: Message (ST_BPMN_MESSAGE_2) Connection types in the BPMN collaboration diagram (BPMN 2.0): * Participant sends (CT_SENDS_2) message. * Event sends (CT_SENDS_2) message. * Function sends (CT_SENDS_2) message. * Message is received from (CT_IS_RECEIVED_FROM) participant. * Message is received from (CT_IS_RECEIVED_FROM) function. * Message is received from (CT_IS_RECEIVED_FROM) event.
Flow node		Object types that can be the source or target of **message flow** (CT_BPMN_MESSAGE_FLOW) connection type: Participant (OT_BPMN_POOL), Function (OT_FUNC), Event (OT_EVT)
Message flow association	inherits from BaseElement	This association is used to map message flows modeled in a collaboration and a conversation diagram.
	innerMessageFlowRef: Message Flow	Occurrence copy of a message flow connection in a BPMN collaboration diagram and BPMN conversation diagram.

Class	BPMN attribute name	Implementation in ARIS
	outerMessageFlowRef: Message Flow	Occurrence copy of a message flow connection in a BPMN collaboration diagram and BPMN conversation diagram.

10.2.3.13 Participant

A participant represents a Partner entity and/or a Partner role that participates in a collaboration. Participants may be modeled in a BPMN collaboration diagram or a BPMN conversation diagram.

The assignment of a Partner entity and/or a Partner role to a participant is transferred to the BPMN allocation diagram (BPMN 2.0) assigned to the participant.

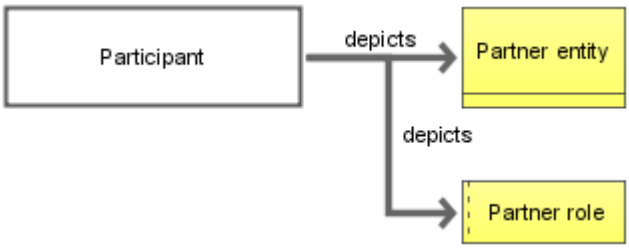


Abbildung 158: BPMN allocation diagram (BPMN 2.0): Participant and partner entity/partner role

The usage of participants is described in the context of the BPMN collaboration diagram (see chapter Pool and participant (Seite 264)) and the BPMN conversation diagram (see chapter Participant (Seite 268)).

Participant, Partner entity and Partner role inherit from base element

See: Business Process Model and Notation (BPMN), version 2.0.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Participant	inherits from BaseElement	Object type: Participant (OT_BPMN_POOL) Symbol: Pool (ST_BPMN_POOL_1)
	name: string [0..1]	Attribute type **Name** (AT_NAME) of object type Participant (OT_BPMN_POOL)

Class	BPMN attribute name	Implementation in ARIS
	processRef: Process [0..1]	BPMN process diagram (BPMN 2.0) assigned to the participant (OT_BPMN_POOL Process displayed within in the pool
	partnerRoleRef: PartnerRole [0..1]	Model type: BPMN allocation diagram (BPMN 2.0): Object type: Role (OT_PERS_TYPE) Symbol: Partner role (ST_BPMN_PARTNER_ROLE) Connection type: depicts (CT_DEPICTS_1) Role
	partnerEntityRef: PartnerEntity [0..1]	Model type: BPMN allocation diagram (BPMN 2.0): Object type: Organizational unit (OT_ORG_UNIT) Symbol: Partner entity (ST_BPMN_PARTNER_ENTITY) Connection type: depicts (CT_DEPICTS_1) organizational unit
	interfaceRef: Interface [0..*]	Currently not implemented.
	participantMultiplicity: participantMultiplicity [0..1]	Attribute type in the attribute type group **BPMN 2.0 attributes/Participant multiplicity attributes** of the object type **Participant** (OT_BPMN_POOL): * Multi-instance participant (AT_BPMN_MI_PARTICIPANT) The mini-symbol (three vertical lines) is displayed by the program if the value of the attribute type **Multi-instance participant** is set to **true**.
	endpointRefs: EndPoint [0..*]	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
Partner entity	inherits from BaseElement	Object type: Organizational unit (OT_ORG_UNIT) Symbol: Partner entity (ST_BPMN_PARTNER_ENTITY)
	name: string	Attribute type **Name** (AT_NAME) of object type **Organizational unit** (OT_ORG_UNIT)
Partner role	inherits from BaseElement	Object type: Role (OT_PERS_TYPE) Symbol: Partner role (ST_BPMN_PARTNER_ROLE)
	name: string	Attribute type **Name** of object type **Role** (OT_PERS_TYPE)
Participant Multiplicity	minimum: integer [0..1] = 2	Attribute type in the attribute type group **BPMN 2.0 attributes/Participant multiplicity attributes** of the object type **Participant** (OT_BPMN_POOL): * Minimum participant multiplicity (AT_BPMN_MINIMUM_MI_PARTICIPANT)
	maximum: integer [0..1] = 2	Attribute type in the attribute type group **BPMN 2.0 attributes/Participant multiplicity attributes** of the object type **Participant** (OT_BPMN_POOL): * Maximum participant multiplicity (AT_BPMN_MAXIMUM_MI_PARTICIPANT)
Participant Association	inherits from BaseElement	
	innerParticipantRef: Participant	Occurrence copy of the relevant participant.
	outerParticipantRef: Participant	Occurrence copy of the relevant participant.

10.2.3.14 Resource

Resources can be human resources as well as any other resource assigned to activities during process execution. A direct mapping of the BPMN resources to ARIS constructs is not possible - due to the semantically different object types representing resources in ARIS. ARIS does not only provide different object types, but also different connection types.

BPMN 2.0 only knows one object type called **Resource**. The BPMN ActivityResource and its specialized sub-classes correspond to ARIS connection types in combination with object types. Therefore, resources are not included in the current version of the BPMN 2.0 implementation.

See: Business Process Model and Notation (BPMN), version 2.0.

10.2.3.15 Sequence flow

The BPMN Sequence flow is mapped to nine different ARIS connection types, which are used to depict the control flow in traditional ARIS process models.

See: Business Process Model and Notation (BPMN), version 2.0.

Source object type	Connection type	Target object type
Event	occurs before	Event
Event	activates	Function
Event	is evaluated by	Rule
Function	creates	Event
Function	is predecessor of	Function
Function	leads to	Rule
Rule	leads to	Event
Rule	activates	Function
Rule	links	Rule

BPMN distinguishes three types of sequence flow:

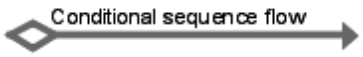
- **Unconditional sequence flow**

The unconditional sequence flow means the **normal** flow, no specific conditions apply. In other words: its condition has always the value **true**. It is depicted by a solid line with a solid arrowhead.

- **Conditional sequence flow**

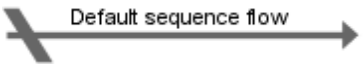
  The conditional sequence flow from an activity is drawn with a little diamond at the beginning of the connector, signifying a data condition. A conditional sequence flow from a gateway shares the same shape as a normal sequence flow.

  Conditional sequence flow from an activity:



- **Default sequence flow**

  The default sequence flow, denoted by a slash marker at the beginning of the connector means **otherwise**, that is, it is enabled if no other sequence flow condition evaluates to **true**.



All connection types used in BPMN diagrams must hold attributes for recording text annotations (Seite 196). Connection types emerging from activities and gateways need additional attributes for recording sequence flow conditions.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Sequence flow	inherits from FlowElement	The sequence flow is depicted by nine different connection types in the model types BPMN process diagram (BPMN 2.0) (MT_BPMN_PROCESS_DIAGRAM) and BPMN collaboration diagram (BPMN 2.0) (MT_BPMN_COLLABORATION_DIAGRAM): * event occurs before (CT_SUCCEED) event * event activates (CT_ACTIV_1) function * event is evaluated (CT_IS_EVAL_BY_1) by rule * function creates (CT_CRT_1) event * function is predecessor of (CT_IS_PREDEC_OF_1) function * function leads (CT_LEADS_TO_1) to rule * rule leads to (CT_LEADS_TO_2) event * rule activates (CT_ACTIV_1) function * rule links (CT_LNK_2) rule
	name: string	Attribute type Connection role of connection type Message flow (CT_BPMN_MESSAGE_FLOW)
	sourceRef: FlowNode	Source object of a sequence flow connection. Object types are: * Function * Event * Rule
	targetRef: FlowNode	Target object of a sequence flow connection. Object types are: * Function * Event * Rule

Class	BPMN attribute name	Implementation in ARIS
	conditionExpression : Expression [0..1]	Attribute type Condition expression (AT_BPMN_CONDITION_EXPRESSION) in attribute type group **BPMN 2.0 attributes** of the following connection types: * activates (CT_ACTIV_1) * creates (CT_CRT_1) * links (CT_LNK_2) * leads to (CT_LEADS_TO_1) * leads to (CT_LEADS_TO_2) * is predecessor of (CT_IS_PREDEC_OF_1) The value of the attribute type **Sequence flow condition** in the attribute type group **BPMN 2.0 attributes** must be set to **Conditional sequence flow**.
	isImmediate: boolean	Currently not implemented.
Flow node	incoming: Sequence Flow [0..*]	Incoming connections representing the sequence flow of the flow node object (object types: function, event, rule)
	outgoing: Sequence Flow [0..*]	Outgoing connections representing the sequence flow of the flow node object (object types: function, event, rule)

10.2.3.16 Elements not included in the current implementation

The following elements belong to the execution design level and are not included in the current version of the BPMN 2.0 implementation.

- Correlations (See: Business Process Modeling Notation (BPMN), version 2.0, page 136 and the following.)
- Conversation Associations (See: Business Process Modeling Notation (BPMN), version 2.0, page 135 and the following.)
- Error (See: Business Process Modeling Notation (BPMN), version 2.0, page 81 and the following.)
- Interaction node (See: Business Process Modeling Notation (BPMN), version 2.0, page 123.)
- Item definition (See: Business Process Modeling Notation (BPMN), version 2.0, page 91 and the following.)
- Services (See: Business Process Modeling Notation (BPMN), version 2.0, page 104 and the following.)

10.3 BPMN diagrams and ARIS model types: An overview

According to the BPMN 2.0 specification three diagram types are required for process modeling conformance: Process diagram, Collaboration diagram and Conversation diagram (See: Business Process Modeling Notation (BPMN), version 2.0. Page 2).

A careful consideration of these BPMN diagrams shows that the modeling constructs of the process diagram are a subset of the modeling constructs used in the collaboration diagram. There are also overlapping constructs in the collaboration and conversation diagram.

The model types listed in the following table are available in ARIS.

The BPMN allocation diagram allows the mapping of BPMN attributes and associations to the semantically richer ARIS-Methode where graphical elements are often used to represent BPMN attributes and associations.

BPMN diagram	ARIS model type
Process diagram	BPMN process diagram (BPMN 2.0)
Process diagram	Enterprise BPMN process diagram
Collaboration diagram	BPMN collaboration diagram (BPMN 2.0)
Collaboration diagram	Enterprise BPMN collaboration diagram
Conversation diagram	BPMN conversation diagram (BPMN 2.0)
	BPMN allocation diagram (BPMN 2.0)

In models of types **Enterprise BPMN process diagram** and **Enterprise BPMN collaboration diagram**, the following object types of the ARIS-Methode are available as lane symbols in addition to the BPMN 2.0 specification:

- Application system type
- Organizational unit
- Position
- Role
- Group

In models of type **Enterprise BPMN process diagram** and **Enterprise BPMN collaboration diagram**, the following additional connections to task objects are available in addition to the BPMN 2.0 specification:

- Application system type **supports** Task
- Organizational unit **supports** Task
- Position **carries out** Task
- Role **carries out** Task

- Group **carries out** Task

10.4 Process

See: Business Process Model and Notation (BPMN), version 2.0.

The BPMN process diagram depicts a BPMN process. A process is a specialization of a FlowElementsContainer. So, it contains the following elements:

- flow nodes (event, activity, and gateway)
- sequence flow
- artifacts (see chapter Artifacts (Seite 191))

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Process	inherits from CallableElement inherits from FlowElementsContainer	Model type: BPMN process diagram (BPMN 2.0)
	processType: ProcessType = none { none | executable | non-executable | public }	Attribute type in the attribute type group **BPMN 2.0 attributes** of model type BPMN process diagram (BPMN 2.0): * Process type (AT_BPMN_PROCESS_TYPE) Attribute values: * Undefined (= none), * Executable process (AVT_BPMN_EXECUTABLE), * Non-executable process (AVT_BPMN_NON_EXECUTABLE) * Public process (AVT_BPMN_PUBLIC)
	auditing: Auditing [0..1]	Currently not implemented.
	monitoring: Monitoring [0..1]	Currently not implemented.
	laneSets: LaneSet [0..*]	Object type **Lane** (OT_BPMN_LANE) Symbol: Lane (ST_BPMN_LANE_1)

Class	BPMN attribute name	Implementation in ARIS
	IsClosed: boolean = false	Attribute type **Is closed** (AT_BPMN_IS_CLOSED) in attribute type group **BPMN 2.0 attributes** of the BPMN process diagram
	supports: Process [0..*]	Currently not implemented.
	properties: Property [0..*]	Currently not implemented.
	definitionalCollaborationRef: Collaboration [0..1]	The BPMN collaboration diagram (BPMN 2.0) that contains the process

A process is a particular construct: On the one hand it is a model. On the other hand a process can be visualized within a pool in a collaboration. But a pool is not identical with a process, and vice versa. A pool represents a participant in a collaboration (see chapter Collaboration (Seite 263)). A pool may contain the process the participant uses in a specific collaboration.

The core elements for modeling a BPMN process are those constructs which can be connected to each other by sequence flow. They are called flow nodes. The corresponding ARIS object types and their symbols provided in the **Symbols** bar are listed in the table below.

BPMN element	ARIS object type	ARIS symbol	API name
Event	Event (OT_EVT)	Start event Intermediate event End event	ST_BPMN_START_EVENT ST_BPMN_INTERMEDIATE_EVENT ST_BPMN_END_EVENT
Activity	Function (OT_FUNC)	Task Subprocess Call activity	ST_BPMN_TASK ST_BPMN_SUBPROCESS ST_BPMN_CALL_ACTIVITY
Gateway	Rule (OT_RULE)	Gateway	ST_BPMN_RULE_1

These constructs are described in detail in the separate chapters below.

10.4.1 Activities

The BPMN activity is represented by the ARIS object type **Function**.

BPMN 2.0 differentiates three basic types of activities: task (atomic activity), subprocess (non-atomic activity) and call activity. The symbols depicting these activity types are provided in the ARIS **Symbols** bar.

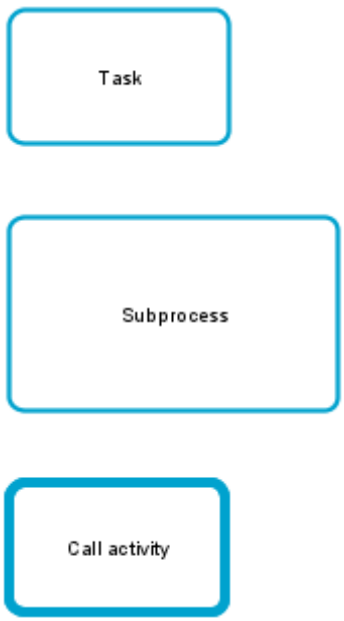


Abbildung 159: Symbols representing activities in the Symbols bar

When the modeler places an activity symbol, the software sets the corresponding value of the ARIS attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE). This activity type controls the correct behavior of the symbol. For example: A subprocess may have **embedded** flow elements, a task must not; a call activity may reference another task or process, tasks and subprocesses must not.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Activity	inherits from FlowElement	Object type: Function (OT_FUNC) Attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE) in the attribute type group **BPMN 2.0 attributes** of object type **Function** Attribute values: * Task (AVT_BPMN_TASK) * Subprocess (AVT_BPMN_SUBPROCESS) * Call activity (AVT_BPMN_CALL_ACTIVITY)
	Compensation activity:: boolean = false	Attribute type **Compensation activity**: (AT_BPMN_COMPENSATION_ACTIVITY.TRM=Compensation activity) in the attribute type group **BPMN 2.0 attributes** of object type **Function**
	loopCharacteristics: LoopCharacteristics [0..1]	see below: Loop characteristics
	resources: ActivityResource [0..*]	Currently not implemented.
	default: SequenceFlow [0..1]	Attribute type **Sequence flow condition** (AT_BPMN_SEQ_FLOW_CONDITION) in the attribute type group **BPMN 2.0 attributes** of the following conenction types: * Activity creates ..., * Activity is predecessor of ..., * Activity leads to ... The attribute value must be set to **Default sequence flow**.
	ioSpecification: InputOutputSpecification [0..1]	Currently not implemented.
	properties: Property [0..*]	Currently not implemented.
	boundaryEventRefs: BoundaryEvent [0..*]	Connection type: Function can trigger event CT_BPMN_CAN_TRIGGER

Class	BPMN attribute name	Implementation in ARIS
	dataInputAssociations: DataInputAssociation [0..*]	Currently not implemented.
	dataOutputAssociations: DataOutputAssociation [0..*]	Currently not implemented.
	startQuantity: integer = 1	Currently not implemented.
Class	**completionQuantity**: integer = 1	Currently not implemented.

10.4.1.1 Resource assignment

Resource assignments are not included in the current version of the BPMN 2.0 implementation. They will be dealt with in detail when implementing the execution design level in ARIS.

See: Business Process Model and Notation (BPMN), version 2.0.

10.4.1.2 Performer

Resource assignments are not included in the current version of the BPMN 2.0 implementation. They will be dealt with in detail when implementing the execution design level in ARIS.

See: Business Process Model and Notation (BPMN), version 2.0.

10.4.1.3 Activity type: Task

See: Business Process Model and Notation (BPMN).

BPMN 2.0 distinguishes eight task types which are represented by different symbols (see). Only the Abstract task is available in the **Symbols** bar. The symbols of the remaining seven special task types are not available in the **Symbols** bar, they are handled by the program.

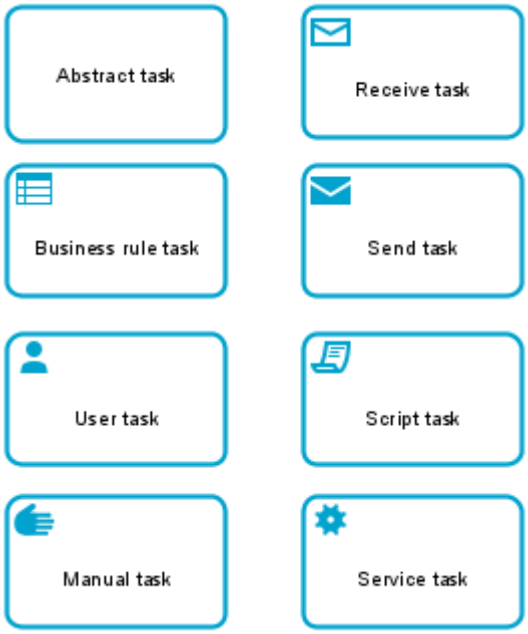


Abbildung 160: Task symbols

When the modeler selects a specific task symbol the software sets the corresponding value of the ARIS attribute type **Task type**. This attribute type is read-only. It provides the following values: Abstract task, Business rule task, Manual task, Script task, Send task, Service task, Receive task, and User task.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Task	inherits from Activity	The value of the attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE) is set to **Task** in the attribute type group **BPMN 2.0 attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Task (ST_BPMN_TASK) or a special task symbol (see below)

Class	BPMN attribute name	Implementation in ARIS
Service task	inherits from Activity	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Service task** in the attribute type group **BPMN 2.0 attributes/Task attributes** of object type **Function** Object type: Function (OT_FUNC) Symbol: Service task (ST_BPMN_SERVICE_TASK)
	implementation: Implementation = Web Service {Web Service | Other | Unspecified}	Currently not implemented.
	operationRef: Operation [0..1]	Currently not implemented.
Send task	inherits from Activity	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Send task** in the attribute type group **BPMN 2.0 attributes/Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Send task (ST_SEND_TASK)
	messageRef: Message [0..1]	Connection type in the BPMN collaboration diagram (BPMN 2.0) * Function sends message.
	operationRef: Operation [0..1]	Currently not implemented.
	implementation: Implementation = Web Service {Web Service| Other | Unspecified}	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
Receive task	inherits from Activity	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is s**et to Receive task in the attribute type group BPMN 2.0 attributes/Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Receive task (ST_RECEIVE_TASK)
	messageRef: Message [0..1	Connection type in the BPMN collaboration diagram (BPMN 2.0) * Message is received from function
	Instantiate: boolean = False	Currently not implemented.
	operationRef: Operation [0..1]	Currently not implemented.
	implementation: Implementation = Web Service {Web Service | Other | Unspecified}	Currently not implemented.
User task	inherits from Activity	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **User task** in the attribute type group **BPMN 2.0 attributes/Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: User task (ST_USER_TASK)
	Implementation: UserTaskImplementation = Other {HumanTaskWebService | WebService | Other | Unspecified}	Currently not implemented.
	renderings: Rendering [0..*]	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
Manual task	inherits from Activity	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Manual task** in the attribute type group **BPMN 2.0 attributes/Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Manual task (ST_MANUAL_TASK)
Business Rule Task	inherits from Activity	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Business rule task** in the attribute type group **BPMN 2.0 attributes/Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Business rule task (ST_BUSINESS_RULE_TASK)
	Implementation: BusinesRuleTaskImplementation = Other {BusinessRuleWebService | WebService | Other | Unspecified}	Currently not implemented.
Script task	inherits from Activity	The value of the attribute type **Task type** (AT_BPMN_TASK_TYPE) is set to **Script task** in the attribute type group **BPMN 2.0 attributes/Task attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: Script task (ST_SCRIPT_TASK)
	scriptLanguage: string [0..1]	Currently not implemented.
	script: string [0..1]	Currently not implemented.

10.4.1.4 Human interactions

User tasks and manual tasks are relevant for modeling human interactions. Their attributes and model associations can also be found in chapter Activity type: Task (Seite 218). They will be dealt with in detail when implementing the execution design level in ARIS.

See: Business Process Model and Notation (BPMN), version 2.0.

10.4.1.5 Activity type: Subprocess

See: Business Process Model and Notation (BPMN), version 2.0.

BPMN 2.0 knows four types of subprocesses:

- Subprocess (standard)
  (A ([standard] subprocess corresponds to the embedded subprocess in BPMN 1.x.)
- Event subprocess
- Transaction, and
- Ad hoc subprocess

Each type of a subprocess can be displayed as

- Subprocess (collapsed) or
- Subprocess (expanded)

Collapsed subprocesses have a special marker displayed at the bottom of the corresponding subprocess symbol:

10.4.1.6 Subprocess type: Subprocess

A standard subprocess shares the same shape as a task. In the collapsed form, the subprocess object uses the **+**-marker to distinguish it from a task. Expanded subprocesses have no marker, they reveal their **embedded** objects.

The symbol representing the expanded subprocess is available in the **Symbols** bar, the symbol representing the collapsed subprocess is handled by the software.

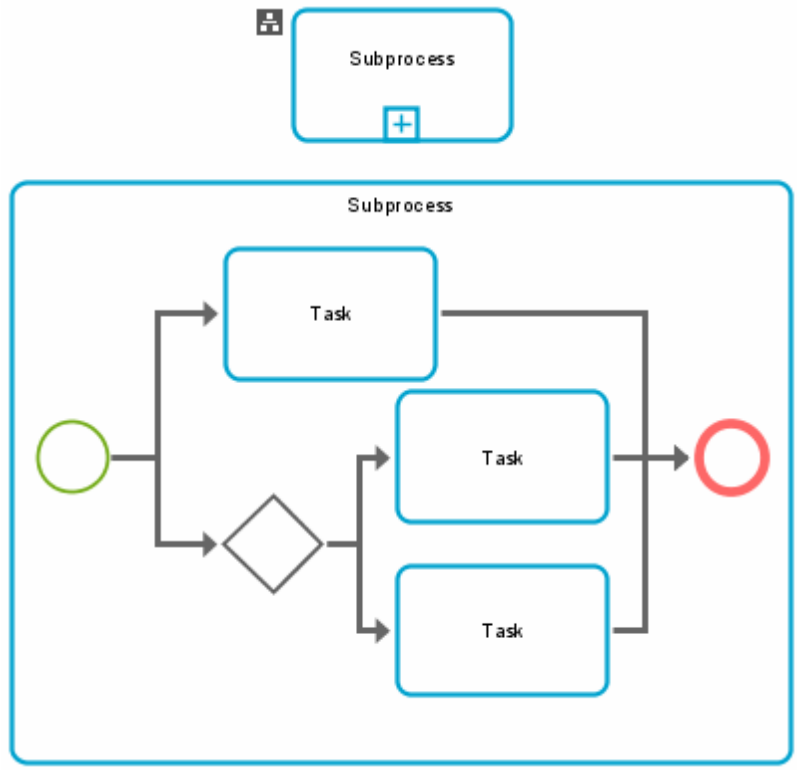


Abbildung 161: Symbols of a standard subprocess

The attributes and model associations of a subprocess and their mapping to ARIS constructs are listed in the table below.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Subprocess	inherits from Activity inherits from FlowElementsContainer	The value of the attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE) is set to **Subprocess** in the attribute type group **BPMN 2.0 attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbols: * Subprocess (ST_BPMN_SUB_PROCESS) * Subprocess collapsed (ST_BPMN_SUB_PROCESS_COLLAPSED) * or a special subprocess symbol (see below)
	triggeredByEvent: boolean = false	Attribute type **Event subprocess** (AT_BPMN_EVENT_SUB_PROCESS) in the attribute type group **BPMN 2.0 attributes/Subprocess attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbols: * Event subprocess (ST_BPMN_EVENT_SUBPROCESS) * Event subprocess (collapsed) (ST_BPMN_EVENT_SUBPROCESS_COLLAPSED) The symbols are rendered by the program.

10.4.1.7 Subprocess type: Event subprocess

An event subprocess is a specialized subprocess that is used within a process or a subprocess. Unlike a standard subprocess which uses the flow of its parent process as a trigger, an event subprocess is not part of the normal flow of its parent process, there is no incoming and outgoing sequence flow. An event subprocess has a start event with a trigger. Each time the start event is triggered while the parent process is active, then the event subprocess will start.

The symbols of an event subprocess are shown below. If the event subprocess is collapsed, its start event is used as a marker in the upper left corner of the symbol. The software will render this marker.

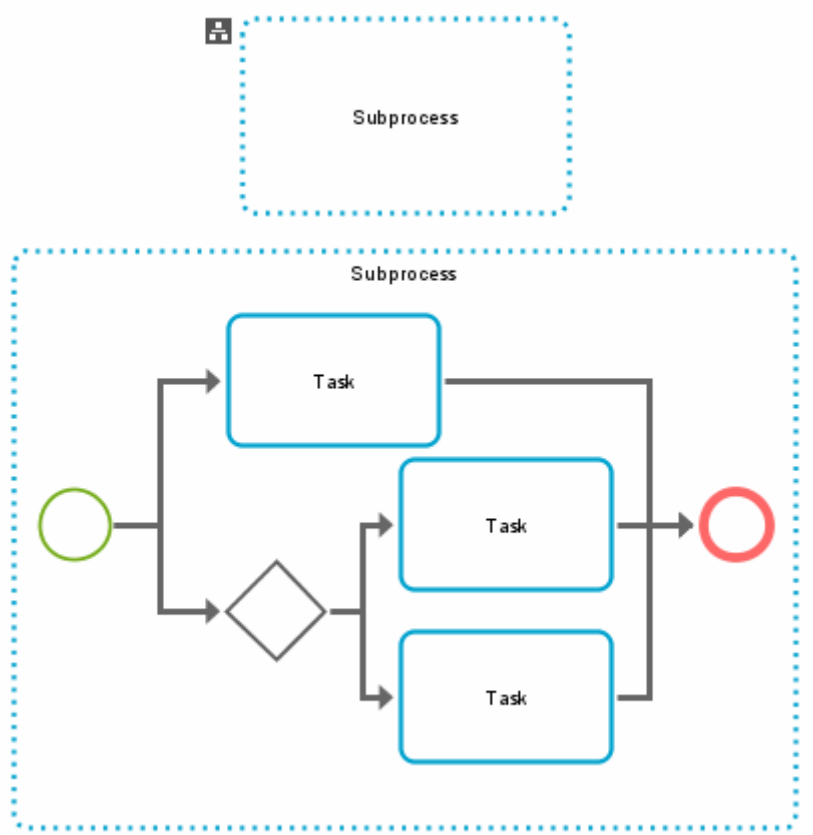


Abbildung 162: Symbole des Event Subprocess

There is a Boolean ARIS attribute type **Event subprocess** representing the BPMN attribute **triggeredByEvent** (see table under Subprocess type: Subprocess (Seite 224)). This attribute type is read-only and used by the software.

10.4.1.8 Subprocess type: Transaction

A transaction subprocess, denoted with a double-lined boundary, is a specialized type of subprocess. In a transaction subprocess all activities must either complete successfully or the subprocess must be rolled back to its original consistent state. A transaction subprocess has a special behavior: It is associated with a transaction protocol that has to verify that all activities have been successfully completed. The symbols are not available in the **Symbols** bar, they are handled by the software. The program also sets the value of the ARIS attribute type **Subprocess type** to **Transaction**.

Abbildung 163: Symbol for a collapsed transaction

A transaction inherits from **Activity**. The attributes and model associations of a transaction subprocess and their mapping to ARIS constructs are shown in the table below.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Transaction	inherits from Activity	The value of the attribute type **Subprocess type** (AT_BPMN_SUBPROCESS_TYPE) is set to **Transaction** in the attribute type group **BPMN 2.0 attributes/Subprocess attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbols: * Transaction (ST_BPMN_TRANSACTION) * Transaction (collapsed) (ST_BPMN_TRANSACTION_COLLAPSED_1) The symbols are rendered by the software.
	protocol: string [0..1]	Currently not implemented.
	method: TransactionMethod = compensate { compensate | store | image }	Currently not implemented.

10.4.1.9 Subprocess type: Ad hoc subprocess

A transaction subprocess, denoted with a double-lined boundary, is a specialized type of subprocess. In a transaction subprocess all activities must either complete successfully or the subprocess must An Ad hoc subprocess, denoted with a tilde marker, is a specialized type of subprocess. It contains a set of activities that could be performed. Sequence flow between activities is optional in an Ad hoc subprocess. What activities are performed as well as the sequence and the number of performances is determined by the performers of the activities. During execution of the (parent) process, any one or more of the activities may be active.

The ARIS Method provides the tilde marker as mini-symbol for the value **Ad hoc subprocess** of the attribute **Subprocess type**. The program will render the symbols for the Ad hoc subprocess.

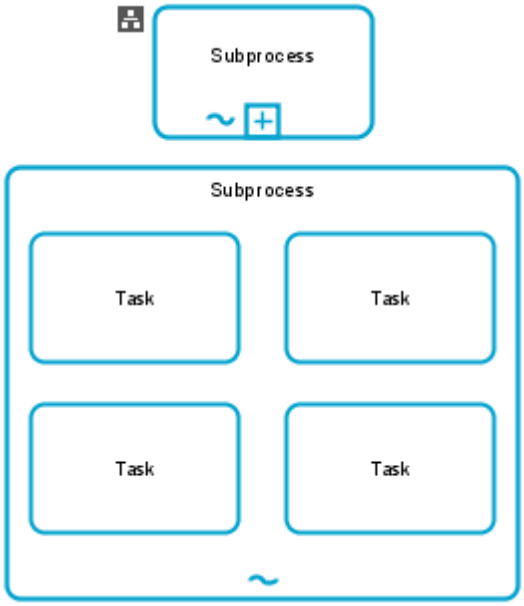


Abbildung 164: Symbol for a collapsed and expanded Ad hoc subprocess

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Ad hoc subprocess	inherits from Activity	The value of the attribute type **Subprocess type** (AT_BPMN_SUB_PROCESS_TYPE) is set to **Ad hoc subprocess** in the attribute type group **BPMN 2.0 attributes/Subprocess attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbols: The mini-symbol **tilde** is rendered by the program.
	completionCondition: Expression	Attribute type **Ad hoc completion condition** (AT_BPMN_COMPLETION_CONDI) in the attribute type group **BPMN 2.0 attributes/Subprocess attributes/Ad hoc subprocess attributes** of object type **Function**.
	ordering: AdHocOrdering = parallel { parallel | sequential }	Currently not implemented.
	cancelRemainingInstances : Boolean = True	Currently not implemented.

10.4.1.10 Subprocess type: Call Activity

A call activity represents the invocation of either a reusable global task or a process. The call activity represents the calling element, and the global task or process represents the called element.

The symbol **Call activity** is available in the **Symbols** bar. If the modeler places this symbol, the value of the attribute **Activity type** is set to **Call activity** and the software provides a dialog where the modeler selects the task or the process being called. Depending on this selection, the value of the attribute type **Called element** is set to **Global task** or **Global process**. The program renders the symbol for the call activity. It corresponds to the symbol for the called task or process, but it is drawn with a thick border.

If a task is selected the program automatically creates a connection (call activity invokes task) on definition level. If a process is selected, the related process diagram is assigned to the call activity.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
CallActivity	inherits from Activity	The value of the attribute type **Activity type** (AT_BPMN_ACTIVITY_TYPE) is set to **Call activity** in the attribute type group **BPMN 2.0 attributes** of object type **Function**. Object type: Function (OT_FUNC) Symbol: The symbol depends on the activity being called. The program will render the symbol.

Class	BPMN attribute name	Implementation in ARIS
	calledElement: CallableElement [0..1]	**For tasks**: The value of the attribute type **Called element** (AT_BPMN_CALLED_ELEMENT) is set to **Global task**. The program creates the connection type **Function invokes [CT_INVOKES] function** on definition level. **For processes**: The value of the attribute type **Called element** (AT_BPMN_CALLED_ELEMENT) is set to **Global process**. The BPMN process diagram of the called process is assigned to the Call activity. In both cases the software provides an appropriate dialog.

10.4.1.11 Global task

The global task is described in chapter Callable Elements (Seite 197).

A global task has no specific attributes and model associations, it inherits from callable elements.

10.4.1.12 Loop characteristics

BPMN 2.0 provides two alternatives to model repeating activities (both tasks and subprocesses):

- Loop activity (= standard loop)
- Multi-instance activity

10.4.1.13 Loop characteristics representations

An activity can be specified to repeat based on a condition. That is called standard loop activity in BPMN. A standard loop is equivalent to the **do while** and **do until** structure in programming. The number of iterations is unknown.

A multi-instance activity is another type of repeating activity useful for performing actions on a list of items. A multi-instance activity is equivalent with a **for each** structure in programming. The number of iterations is known when the activity starts. It is the number of items in the list. Iterations of a multi-instance activity can be performed concurrently or sequentially.

The marker for a standard loop is a circular arrow at the bottom center of the activity symbol.

Abbildung 165: Symbols of Standard loop activities

The markers for multi-instance activities are three bars at the bottom center of the task or subprocess symbol.

Vertical bars are used to represent concurrent/parallel performances:

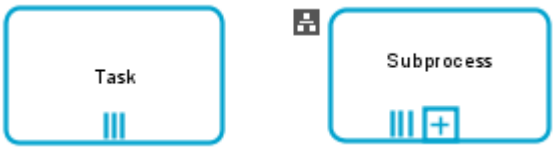


Abbildung 166: Symbols of BPMN multi-instance (parallel) activities

Horizontal bars are used to represent sequential performances:

Abbildung 167: Symbols for activities of the BPMN multi-instance (parallel)

Loop characteristics has no specific attributes, it inherits the attributes and associations of base element. The attribute type **Loop type** is used in ARIS to specify whether the loop is a standard loop, a multi-instance parallel loop, or a multi-instance sequential loop. The attribute values are visualized by mini-symbols.

10.4.1.14 Standard and multi-instance loop characteristics and complex behavior definition

The attributes and model associations of standard activities, multi-instance loop activities, and complex behavior definition are summarized in the table below.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
LoopCharacteristics	inherits from BaseElement	Attribute type group **Loop characteristics** (AT_BPMN_LOOP_CHARACTERISTICS) in attribute type group **BPMN 2.0 attributes** of object type **Function** (OT_FUNC).

Class	BPMN attribute name	Implementation in ARIS
StandardLoopCharacteristics	inherits from BaseElement	The value of the attribute type **Loop type** (AT_BPMN_LOOP_TYPE_2) is set to **Standard loop** (AVT_BPMN_STANDARD_LOOP) in the attribute type group **BPMN 2.0 attributes/Loop characteristics** of object type **Function**.
	testBefore: boolean = False	Attribute type **Test before** (AT_BPMN_LOOP_TEST_TIME) in the attribute type group **BPMN 2.0 attributes/Loop characteristics/Standard loop attributes** of object type **Function**.
	loopMaximum: Expression [0..1]	Attribute type **Loop maximum** (AT_BPMN_MAX_LOOP) in the attribute type group **BPMN 2.0 attributes/Loop characteristics/Standard loop attributes** of object type **Function**.
	loopCondition: Expression [0..1]	Attribute type **Loop condition** (AT_BPMN_LOOP_CONDITION) in the attribute type group **BPMN 2.0 attributes/Loop characteristics/Standard loop attributes** of object type **Function**.
MultiInstanceLoop Characteristics	inherits from BaseElement	The value of the attribute type **Loop type** (AT_BPMN_LOOP_TYPE_2) is set to **Multi-instance sequential loop** (AVT_BPMN_MULTI_INSTANCE_SEQUENTIAL_LOOP) or **Multi-instance parallel loop** (AVT_BPMN_MULTI_INSTANCE_PARALLEL_LOOP) in the attribute type group **BPMN 2.0 attributes/Loop characteristics** of object type **Function**.
	isSequential: boolean = False	isSequential = true corresponds to: Loop type = Multi-instance sequential loop isSequential = false corresponds to: Loop type = Multi-instance parallel loop

Class	BPMN attribute name	Implementation in ARIS
	loopCardinality: Expression [0..1]	Attribute type **Loop cardinality** (AT_BPMN_LOOP_CARDINALITY) in the attribute type group **BPMN 2.0 attributes/Loop characteristics/Multi-instance loop attributes** of object type **Function**.
	loopDataInput: DataInput [0..1]	Currently not implemented.
	loopDataOutput: DataOutput [0..1]	Currently not implemented.
	inputDataItem: Property [0..1]	Currently not implemented.
	outputDataItem: Property [0..1]	Currently not implemented.
	completionCondition: Expression [0..1]	Currently not implemented.
	behavior: MultiInstanceBehavior = all { none | one | all | complex }	Currently not implemented.
	complexBehaviorDefinition: ComplexBehaviorDefinition [0..*]	Currently not implemented.
	oneBehaviorEventRef: EventDefinition [0..1]	Currently not implemented.
	noneBehaviorEventRef: EventDefinition [0..1]	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
ComplexBehaviorDefinition	inherits from BaseElement	Currently not implemented.
	condition: Formal Expression	
	event: ImplicitThrowEvent	

10.4.2 Items and Data

As mentioned above, the current implementation of BPMN 2.0 in ARIS focuses on the business process level. Therefore, only data objects and data stores are provided - as input or output of activities. Detailed data modeling aspects (for example data structures, data states, data associations) are omitted.

See: Business Process Model and Notation (BPMN), version 2.0.

10.4.2.1 Data object

In BPMN 1.x data were considered as an artifact; in BPMN 2.0, data objects were upgraded to objects in the BPMN semantic model.

On the business level, where the data structures and mappings are not included, data objects are represented in ARIS by six symbols of the object type **Cluster/Data model**:

- Data object
- Data collection
- Data input
- Data input collection
- Data output
- Data output collection

When you place a Data object the object symbols **Data object**, **Data input**, and **Data output** are provided. You can select the object symbols **Data collection**, **Data input collection**, and **Data output collection** using the **Object appearance** page of the **Object properties** dialog.

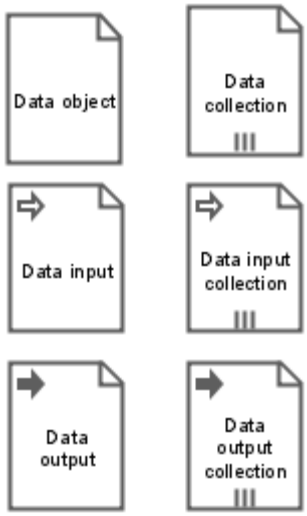


Abbildung 168: Symbols of data objects

Only the **Data object** symbol is available in the **Symbols** bar.

Data objects can represent the input or output of activities by using the following connection types:

- Cluster/Data model **is input for** function
- Function **has as output** Cluster/Data model

The data input symbols must not be the target of a **has as output** connection, and the data output symbols must not be the source of an **is input for** connection. The software ensures this.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
ItemAware Element	inherits from BaseElement	
	itemSubjectRef: ItemDefinition [0..1]	Currently not implemented.
	dataState: DataState [0..1]	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
Data object	inherits from FlowElement & ItemAwareElement	Object type: Cluster/Data model (OT_CLST) Six symbols: * Data object (ST_BPMN_DATA_OBJECT) * Data collection (ST_BPMN_DATA_COLLECTION) * Data input (ST_BPMN_DATA_INPUT) * Data input collection (ST_BPMN_DATA_INPUT_COLLECTION) * Data output (ST_BPMN_DATA_OUTPUT) * Data output collection (ST_BPMN_DATA_OUTPUT_COLLECTION)
	isCollection: Boolean = False	Represented by special symbols of the object type **Cluster/Data model** (OT_CLST)

10.4.2.2 Data store

Unlike data objects, which live only as long as the process instance is running, a Data store represents information that persists beyond the lifetime of a particular process. On the business level, a Data store is represented by the symbol **Data store** of the ARIS object type **Information carrier**. This symbol is available in the **Symbols** bar.

On the business level, where the data structures and mappings are not included, data objects are represented in ARIS by six symbols of the object type **Cluster/Data model**:

- Information carrier **provides input for** function

- Function **creates output to** information carrier.

Abbildung 169: Symbol for a data store

A Data store inherits from FlowElement and ItemAwareElement.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
DataStore	inherits from FlowElement and ItemAwareElement	Object type: Information carrier (OT_INFO_CARR) Symbol: Data store (ST_BPMN_DATA_STORE)
	name: string	Attribute type **Name** (AT_NAME) of object type **Information carrier** (OT_INFO_CARR)
	capacity: Integer [0..1]	Currently not implemented.
	isUnlimited: Boolean = False	Currently not implemented.
DataStoreReference	inherits from FlowElement and ItemAwareElement	
	dataStoreRef: DataStore	Occurrence copies of the (referenced) data store.

10.4.3 Events

BPMN events are represented in ARIS by the object type **Event**. Altogether there are sixty-three symbols available in BPMN 2.0. The main event types are:

- Start event
- Intermediate event
- End event

Only these three events are provided in the **Symbols** bar in ARIS (see type = None). The remaining symbols are provided as symbols in the ARIS Method.

BPMN events: (See: Business Process Modeling Notation (BPMN), version 2.0, page 233 and the following).

Types	**Start**			**Intermediate**				**End**
	Top level	Event Sub-process Inter-rupting	Event Sub-process Non-Inter-rupting	Catching	Boundary Inter-rupting	Boundary Non-Inter-rupting	Throwing	
None								
Message								
Timer								
Error								
Escalation								
Cancel								
Compensation								
Conditional								

Types	Start			Intermediate				End
Link								
Signal								
Terminate								
Multiple								
Parallel multiple								

10.4.3.1 Catch events and throw events

Events can be:

- catch events (all start and a number of intermediate events)
- throw events (all end events a number of intermediate)

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Event		Object type: Event (OT_EVT) Symbols: sixty-three different symbols (see below)
Catch Event	inherits from FlowElement	Object type: Event (OT_EVT) Symbols: different start or intermediate event symbols
	eventDefinitionRefs: EventDefinition [0..*]	Occurrence copy of the corresponding throw event.
	eventDefinitions: EventDefinition [0..*]	Attribute type **Event definition** (AT_BPMN_EVENT_DEFINITION) in the attribute type group **BPMN 2.0 attributes** of object type **Event** (OT_EVT). The values of this attribute type are: None, Message, Timer, Error, Escalation, Cancel, Compensation, Conditional, Link, Signal, Multiple, Parallel multiple (as special case of Multiple). Each event definition has a specific marker inside the event symbol.
	dataOutputAssociations: DataOutputAssociation [0..*]	Currently not implemented.
	dataOutput: dataOutput [0..*]	Connection type in the BPMN process diagram (BPMN 2.0) and the BPMN collaboration diagram (BPMN 2.0): Event (symbol: only catch events) has as output Cluster/data model

Class	BPMN attribute name	Implementation in ARIS
	outputSet: OutputSet [0..1]	Currently not implemented.
Throw event	inherits from FlowElement	Object type: Event (OT_EVT) Symbols: different intermediate or end event symbols
	eventDefinitionRefs: EventDefinition [0..*]	Occurrence copy of the corresponding catch event.
	eventDefinitions: EventDefinition [0..*]	Attribute type **Event definition** (AT_BPMN_EVENT_DEFINITION) in the attribute type group **BPMN 2.0 attributes** of object type **Event** (OT_EVT). The values of this attribute type are: None, Message, Error, Escalation, Cancel, Compensation, Link, Signal, Terminate, Multiple. Each event definition has a specific marker inside the event symbol.
	dataInputAssociations: DataInputAssociation [0..*]	Currently not implemented.
	dataInput: DataInput [0..*]	Connection type in the BPMN process diagram (BPMN 2.0) and the BPMN collaboration diagram (BPMN 2.0): Cluster/data model is input for event (symbol: only throw events)
	inputSet: InputSet [0..1]	Currently not implemented.
Implicit Throw Event	inherits from ThrowEvent	Currently not implemented.

10.4.3.2 Start event

The symbols of start events are depicted in chapter events (Seite 241). Only the start event **None** is available in the **Symbols** bar. When placing this start event, the modeler is guided by a special functionality of the program.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Start event	inherits from CatchEvent	Object type: Event (OT_EVT) Symbol: Start event (ST_BPMN_START_EVENT)
	isInterrupting: boolean	Interrupting start events are represented by specific event symbols.

10.4.3.3 Intermediate events

The symbols of intermediate events are depicted in chapter events (Seite 241). Only the intermediate event **None** is available in the **Symbols** bar. When placing this start event, the modeler is guided by a special functionality of the program. Intermediate events have no specific attributes and associations.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
IntermediateEvent		Object type: Event (OT_EVT) Symbol: Intermediate event (ST_BPMN_INTERMEDIATE_EVENT)

Some types of intermediate events can be attached to the boundary of activities, they are called **boundary events** (see column **Boundary Interrupting** and **Boundary Non-interrupting** in chapter events (Seite 241).

Boundary events are always catch events. Their attributes and model associations are shown in the table below.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Boundary events	inherits from CatchEvent	Boundary events are specific intermediate events.
	AttachedTo: Activity	Connection type in the BPMN process diagram (BPMN 2.0) and the BPMN collaboration diagram (BPMN 2.0): Function can trigger (CT_BPMN_CAN_TRIGGER) event (symbol: **intermediate event**)
	CancelActivity: boolean	(Non-)Interrupting events are represented by specific event symbols.

10.4.3.4 End event

The symbols of end events are depicted in chapter events (Seite 241). Only the none end event is available in the **Symbols** bar. When placing this start event, the modeler is guided by a special functionality of the program.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
EndEvent		Object type: Event (OT_EVT) Symbol: End event (ST_BPMN_END_EVENT)

10.4.3.5 Event definitions

BPMN 2.0 distinguishes the following event definitions: None, Message, Timer, error, Escalation, Cancel, Compensation, Conditional, Link, Signal, Terminate and Multiple (**Parallel multiple** is a special case **Multiple**). The different definitions are visualized by specific markers placed within the **None start**, **Intermediate** and **End event** symbol.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
EventDefinition	inherits from BaseElement	Attribute type **Event definition** (AT_BPMN_EVENT_DEFINITION) in the attribute type group **BPMN 2.0 attributes** of object type Event (OT_EVT) Attribute values: None, Message, Timer, Error, Escalation, Cancel, Compensation, Conditional, Link, Signal, Terminate, Multiple. This attribute is read-only and set automatically by the software.
CancelEventDefinition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Cancel intermediate event (ST_BPMN_CANCEL_INTERMEDIATE_EVENT) * Cancel end event (ST_BPMN_CANCEL_END_EVENT)

Class	BPMN attribute name	Implementation in ARIS
CompensationEvent Definition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Compensation start event (ST_BPMN_COMPENSATION_START * Compensation intermediate event (catch)(ST_BPMN_COMPENSATION_INTERMEDIATE_CATCH * Compensation intermediate event (throw) (ST_BPMN_COMPENSATION_INTERMEDIATE_THROW) * Compensation end event (ST_BPMN_COMPENSATION_END_EVENT)
	activityRef: Activity [0..1]	Attribute type **Wait for completion** (AT_BPMN_WAIT_FOR_COMPLETION) in the attribute type group **BPMN 2.0 attributes/Compensation event attributes** of object type **Event** (OT_EVT).
ConditionalEvent Definition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Conditional start event (ST_BPMN_RULE_START_EVENT) * Conditional start event (non-interrupting) (ST_BPMN_CONDITIONAL_START_NI) * Conditional intermediate event (ST_BPMN_RULE_INTERMEDIATE_EVENT) * Conditional intermediate event (non-interrupting) (ST_BPMN_CONDITIONAL_INTERMEDIATE_NI)

Class	BPMN attribute name	Implementation in ARIS
	condition: Expression	Attribute type **Condition** (AT_BPMN_RULE_EXPRESSION) in the attribute type group **BPMN 2.0 attributes/Conditional event attributes** of object type **Event** (OT_EVT).
ErrorEventDefinition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Error start event (ST_BPMN_ERROR_START) * Error intermediate event (ST_BPMN_ERROR_INTERMEDIATE_EVENT * Error end event (ST_BPMN_ERROR_END_EVENT)
	errorCode: string	Currently not implemented.
	error: Error [0..1]	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
EscalationEvent Definition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Escalation start event (ST_BPMN_ESCALATION_START) * Escalation start event (non-interrupting) (ST_BPMN_ESCALATION_START_NI) * Escalation intermediate event (catch) (ST_BPMN_ESCALATION_INTERMEDIATE_CATCH) * Escalation intermediate event (non-interrupting) (ST_BPMN_ESCALATION_INTERMEDIATE_NI) * Escalation intermediate event_throw (ST_BPMN_ESCALATION_INTERMEDIATE_THROW) * Escalation end event (ST_BPMN_ESCALATION_END)
	escalationCode: string	Currently not implemented.
	escalationRef: Escalation [0..1]	Currently not implemented.
LinkEventDefinition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Link intermediate event (catch) (ST_BPMN_LINK_INTERMEDIATE_CATCH) * Link intermediate event (throw) (ST_BPMN_LINK_INTERMEDIATE_THROW) Catch and throw link events are referred to each other by occurrence copies.
	name: string	Attribute type **Name** (AT_NAME) of object type **Event** (OT_EVT)

Class	BPMN attribute name	Implementation in ARIS
MessageEvent Definition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Message start event (ST_BPMN_MESSAGE_START_EVENT) * Message start event (non-interrupting) (ST_BPMN_MESSAGE_START_NI) * Message intermediate event (catch) (ST_BPMN_MESSAGE_INTERMEDIATE_CATCH) * Message intermediate event (non-interrupting) (ST_BPMN_MESSAGE_INTERMEDIATE_NI) * Message intermediate event (throw) (ST_BPMN_MESSAGE_INTERMEDIATE_THROW) * Message end event (ST_BPMN_MESSAGE_END_EVENT)
	MessageRef: Message [0..1]	Currently not implemented.
	operationRef: Operation [0..1]	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
Multiple event		Object type: Event (OT_EVT) Symbols: * Multiple start event (ST_BPMN_MULTIPLE_START_EVENT) * Multiple start event (non-interrupting) (ST_BPMN_MULTIPLE_START_NI) * Multiple intermediate event (catch) (ST_BPMN_MULTIPLE_INTERMEDIATE_CATCH) * Multiple intermediate event (non-interrupting)(ST_BPMN_MULTIPLE_INTERMEDIATE_NI) * Multiple intermediate event (throw) (ST_BPMN_MULTIPLE_INTERMEDIATE_THROW) * Multiple end event (ST_BPMN_MULTIPLE_END_EVENT)
None event		Object type: Event (OT_EVT) Symbols: * Start event (ST_BPMN_SE) * Intermediate event (ST_BPMN_IE) * End event (ST_BPMN_EE) These symbols are available in the Symbols bar.

Class	BPMN attribute name	Implementation in ARIS
Parallel multiple event		Object type: Event (OT_EVT) Symbols: * Parallel multiple start event (ST_BPMN_PARALLEL_MULTIPLE_START) * Parallel multiple start event (non-interrupting) (ST_BPMN_PARALLEL_MULTIPLE_START_NI) * Parallel multiple intermediate event (ST_BPMN_PARALLEL_MULTIPLE_INTERMEDIATE) * Parallel multiple intermediate event (non-interrupting) (ST_BPMN_PARALLEL_MULTIPLE_INTERMEDIATE_NI)
SignalEventDefinition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Signal start event (ST_BPMN_SIGNAL_START_EVENT) * Signal start event (non-interrupting) (ST_BPMN_SIGNAL_START_NI) * Signal intermediate event (catch) (ST_BPMN_SIGNAL_INTERMEDIATE_EVENT) * Signal intermediate event (non-interrupting) (ST_BPMN_SIGNAL_INTERMEDIATE_NI) * Signal intermediate event (throw) (ST_BPMN_SIGNAL_INTERMEDIATE_THROW) * Signal end event (ST_BPMN_SIGNAL_END_EVENT)
	signalRef: Signal	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
TerminateEvent Definition	inherits from BaseElement	Object type: Event (OT_EVT) Symbol: * Terminate end event (ST_BPMN_TERMINATE_END_EVENT)
TimerEventDefinition	inherits from BaseElement	Object type: Event (OT_EVT) Symbols: * Timer start event (ST_BPMN_TIMER_START_EVENT) * Timer start event (non-interrupting) (ST_BPMN_TIMER_START_NI) * Timer intermediate event (ST_BPMN_TIMER_INTERMEDIATE_EVENT) * Timer intermediate event (non-interrupting) (ST_BPMN_TIMER_INTERMEDIATE_NI)
	timeDate: Expression [0..1]	Attribute type **Time date** (AT_BPMN_TIMEDATE) in the attribute type group **BPMN 2.0 attributes/Timer event attributes** of object type **Event** (OT_EVT).
	timeCycle: Expression [0..1]	Attribute type **Time cycle** (AT_BPMN_TIMECYCLE) in the attribute type group **BPMN 2.0 attributes/Timer event attributes** of object type **Event** (OT_EVT)

10.4.4 Gateways

The ARIS object type **Rule** depicts BPMN gateways. Although BPMN 2.0 knows five different gateway types, only one symbol is available in the **Symbols** bar:

The remaining gateway symbols are handled by the program. The following figure depicts all (basic) gateway symbols.

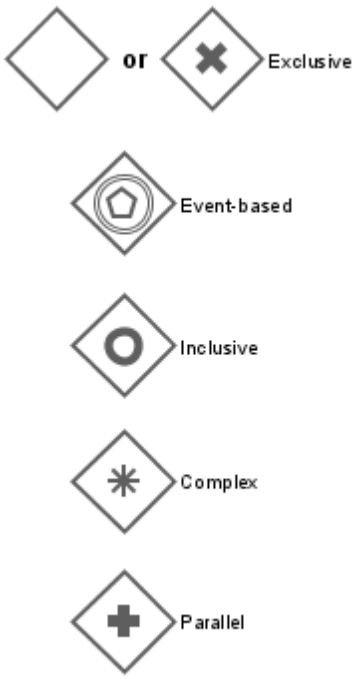
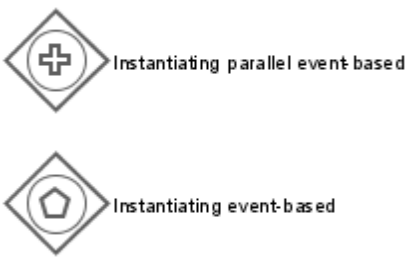


Abbildung 170: BPMN gateway types

For event-based gateways there are two additional symbols which are used to start a process:



All in all the ARIS Method will provide eight gateway symbols. Contrary to events, an ARIS attribute recording the gateway type is not required. It is up to the modeler to ensure that gateways are used in a semantically correct way. The modeler should not reuse gateways.

10.4.4.1 Exclusive gateway

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Exclusive gateway	inherits from Gateway	Object type: Rule (OT_RULE) Symbols: * Gateway (ST_BPMN_RULE_1) * Exclusive gateway (ST_BPMN_RULE_XOR_3) * Event-based gateway (ST_BPMN_RULE_XOR_4)
	default: SequenceFlow [0..1]	Attribute type **Sequence flow condition** (AT_BPMN_SEQ_FLOW_CONDITION) in the attribute type group **BPMN 2.0 attributes** of the following connection types: * Rule leads to (CT_LEADS_TO_2) event * Rule activates (CT_ACTIV_1) function * Rule links (CT_LNK_2) rule The attribute value must be set to **Default sequence flow**. The symbol (slash) is automatically set by the software.

10.4.4.2 Inclusive gateway

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Inclusive gateway	inherits from Gateway	Object type: Rule (OT_RULE) Symbol: * Inclusive gateway (ST_BPMN_RULE_OR_1)

Class	BPMN attribute name	Implementation in ARIS
	default: SequenceFlow [0..1]	See: exclusive gateway

10.4.4.3 Parallel gateway

Parallel gateways have no specific attributes.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Parallel gateway	inherits from Gateway	Object type: Rule (OT_RULE) Symbol: * Parallel gateway (ST_BPMN_RULE_AND_1)

10.4.4.4 Complex gateway

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Complex gateway	inherits from Gateway	Object type: Rule (OT_RULE) Symbol: * Complex gateway (ST_BPMN_RULE_COMPLEX_1)
	activationCondition: Expression [0..1]	Attribute type **Activation condition** (AT_ACTIVATION_CONDITION) in the attribute type group **BPMN 2.0 attributes/Complex gateway attributes** of object type Rule (OT_RULE)

10.4.4.5 Event-based gateways

All attributes are represented by specific gateway symbols.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Event-based gateway	inherits from Gateway	Object type: Rule (OT_RULE) Symbols: * Event-based gateway (ST_BPMN_RULE_XOR_4) * Instantiating event-based gateway (ST_BPMN_RULE_XOR_START) * Instantiating parallel event-based gateway (ST_BPMN_RULE_XOR_PARALLEL)
	instantiate: boolean = False	Represented by symbols. **True** if: * Instantiating event-based gateway (ST_BPMN_RULE_XOR_START) * Instantiating parallel event-based gateway (ST_BPMN_RULE_XOR_PARALLEL) **False** if: * Event-based gateway (ST_BPMN_RULE_XOR_4)
	eventGatewayType: EventGatewayType = Exclusive { Exclusive | Parallel }	Represented by symbols: **Exclusive** if: * Event-based gateway (ST_BPMN_RULE_XOR_4) * Instantiating event-based gateway (ST_BPMN_RULE_XOR_START) **Parallel** if: * Instantiating parallel event-based gateway (ST_BPMN_RULE_XOR_PARALLEL)

10.4.5 Lanes

A lane is a subdivision of a process or a pool. Lanes have no semantics in BPMN. BPMN 2.0 uses lanes as a way to categorizes Flow Elements. Most often lanes represent organizational elements, but in principle any categorization may be used for lanes. Lanes may contain nested sub-lanes. A lane set specifies the categorization represented by the lanes.

See: Business Process Model and Notation (BPMN), version 2.0.

Like a pool a lane is drawn as a rectangular box, its label is not boxed off.

Abbildung 171: Nested Lanes

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
LaneSet	inherits from BaseElement	Object type: Lane (OT_BPMN_LANE) Symbol: Lane (ST_BPMN_LANE_1)
	process: Process	The BPMN process model that contains the lane(s).
	lanes: Lane [0..*]	Object type: Lane (OT_BPMN_LANE) Symbol: Lane (ST_BPMN_LANE_1) The source objects in the connection type: Lane belongs to (CT_BELONGS_TO_1) lane
	parentLane: Lane [0..1]	The target object in the connection type: Lane belongs to (CT_BELONGS_TO_1) lane CT: Lane belongs to lane
Lane	inherits from BaseElement	Object type: Lane (OT_BPMN_LANE) Symbol: Lane (ST_BPMN_LANE_1)

Class	BPMN attribute name	Implementation in ARIS
	name: string	Attribute type **Name** (AT_NAME) of object type Lane (OT_BPMN_LANE)
	partitionElement: BaseElement [0..1]	Currently not implemented.
	partitionElementRef: BaseElement [0..1]	Currently not implemented.
	childLaneSet: LaneSet [0..1]	The source objects in the connection type: Lane belongs to (CT_BELONGS_TO_1) lane
	flowElementRefs: FlowElement [0..*]	The source objects in the following **belongs to** (CT_BELONGS_TO_1) connection types: * Function belongs to lane * Event belongs to lane * Rule belongs to lane * Cluster/data model belongs to lane * Information carrier belongs to lane

10.5 Collaboration

See: Business Process Model and Notation (BPMN), version 2.0.

A collaboration shows message exchanges between participants. A collaboration contains at least two pools representing the participants. A pool may include a process (white box) or may be shown as a black box with all details hidden. The message exchanges between the participants are represented by message flows that connect two pools (or the objects within the pools). Only one pool may be represented without a boundary.

The model type **BPMN collaboration diagram (BPMN 2.0)** has been introduced to model collaborations.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Collaboration	inherits from BaseElement and InteractionSpecification	Model type: BPMN collaboration diagram (BPMN 2.0) (MT_BPMN_COLLABORATION_DIAGRAM)
	name: string	Attribute type **Name** of the BPMN collaboration diagram (BPMN 2.0)
	choreographyRef: Choreography [0..1]	Currently not implemented.
	conversationAssociations: ConversationAssociation [0..*]	The relationships to conversations are represented by occurrence copies of participants (OT_BPMN_POOL; ST_BPMN_POOL_1), occurrence copies of message flow connections and the assignment of a BPMN collaboration model (BPMN 2.0) to the object type **Conversation** (OT_BPMN_CONVERSATION).
	conversations: Conversation [0..*]	BPMN collaboration diagram (BPMN 2.0) assigned to the object type **Conversation** (OT_BPMN_CONVERSATION)
	artifacts: Artifact [0..*]	Artifacts (Seite 191)

Class	BPMN attribute name	Implementation in ARIS
	participantAssociations : ParticipantAssociations [0..*]	The relationships to participants are represented by occurrence copies of participants (OT_BPMN_POOL; ST_BPMN_POOL_1)
	messageFlowAssociations: Message flow association [0..*]	The relationships to message flows are represented by occurrence copies of message flow connections (and the involved participants).
	IsClosed: boolean = false	Attribute type **Is closed** in the attribute type group **BPMN 2.0 attributes** of model type the BPMN Collaboration diagram (BPMN 2.0).

The object types and connection types of the BPMN collaboration diagram are detailed in the following chapters.

10.5.1 Pool and participant

Pools and participants play a central role in collaborations. They are described in detail in chapter Participant (Seite 204).

10.5.2 Object types and connection types reused from a process

As a pool may show a process (white box) all object types and connection types that are allowed in the BPMN process diagram (BPMN 2.0) are also available in the BPMN collaboration diagram (BPMN 2.0).

The object types and connection types taken over from the BPMN process diagram (BPMN 2.0) are described in detail in chapter Process (Seite 213).

The connection type **belongs to** is used to embed the object types of a visible process into a pool.

Source object type	Connection type	Target object type
Event (OT_EVT)	belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Function (OT_FUNC)	belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Rule (OT_RULE)	belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Lane (OT_BPMN_LANE)	belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Cluster/data model (OT_CLST)	belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)
Information carrier (OT_INFO_CARR)	belongs to (CT_BELONGS_TO_1)	Participant (OT_BPMN_POOL)

10.5.3 Message flow

The message flow between different participants is represented by an ARIS connection type of the same name. It connects two pools or the objects within a pool. The attributes and model associations of message flow are described in chapter Message flow (Seite 202).

To show the messages being exchanged in message flows the ARIS object type **message** represented by a message symbol is used. The message flow connection type is replaced by

two connection types: **sends** and **is received from**. This work around is required due to the fact that it is not possible in ARIS to assign object types to connection types.

The program will display the **sends** and **is received from** connection types like a normal message flow.

10.6 Conversation

See: Business Process Modeling Notation (BPMN), version 2.0, page 124 and the following.

The BPMN conversation diagram has been introduced with BPMN 2.0 to provide a big picture of the interactions (in terms of related message exchanges) between collaborating participants.

The BPMN conversation diagram is similar to the BPMN collaboration diagram, but its pools are not allowed to contain a process and a choreography is not allowed between the pools.

The BPMN conversation diagram differentiates three basic elements.

- Conversation nodes (Communication, Sub-conversation)
- Participants (Pools)
- Conversation links (message flow, participates in)

They are described in the next chapters.

10.6.1 Conversation container

The attributes and model associations of a conversation and conversation container are summarized in the table below.

MAPPING THE ATTRIBUTES AND MODEL ASSOCIATIONS TO ARIS

Class	BPMN attribute name	Implementation in ARIS
Conversation	inherits from CallableElement, InteractionSpecification, ConversationContainer	
	correlationKeys: CorrelationKey [0..*]	Currently not implemented.

Class	BPMN attribute name	Implementation in ARIS
	messageFlowRefs: MessageFlow [0..*]	Occurrence copies of message flows (and the involved participants)
Conversation container	inherits from BaseElement	Model type: BPMN conversation diagram (BPMN 2.0) MT_BPMN_CONVERSATION_DIAGRAM
	conversationNodes: ConversationNode [0..*]	see below
	artifacts: Artifact [0..*]	see chapter Artifacts (Seite 191)

10.6.2 Conversation nodes

BPMN 2.0 distinguishes three sub-types of conversation nodes.

- Communication
- Sub-conversation
- Call conversation
  (The concept of call conversation is not clear, thus, Call conversations are ignored in the current implementation of BPMN 2.0 in ARIS. However, in ARIS Call conversations can be particularly distinguished. See description below.)

Conversation nodes are represented in ARIS by the object type **Conversation**.

A **communication** is an atomic conversation element in a BPMN conversation diagram, it represents a set of message flow grouped together based on a single correlation key. A communication will involve at least two participants.

The symbol for a communication is available in the **Symbols** bar of the BPMN conversation diagram (BPMN 2.0).

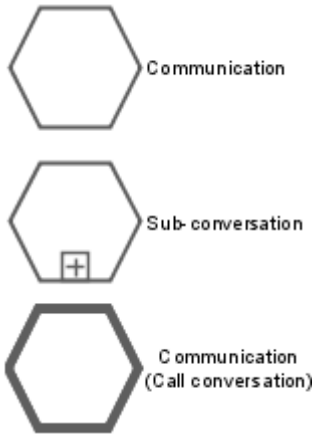


Abbildung 172: Symbols of Conversation nodes

A **Sub conversation** is a conversation which consists of lower-level conversations which are modeled in a separate BPMN conversation diagram assigned to the sub-conversation. A sub-conversation shares the participants of its parent conversation.

The ARIS method provides a **Sub conversation** symbol, it also shown in the **Symbols** bar.

A **Call conversation** identifies a place in a conversation where a global conversation or a global communication is used. A **global communication** is a reusable atomic communication definition that can be called from within any conversation by a Call conversation.

The concepts of Call conversations and global communication are very vague. Thus, the ARIS method does not provide specific symbols. But there is the Boolean ARIS attribute type **Call conversation** which allows the modeler to flag Call conversations. If the value is **true**, the **Call conversation** symbol is rendered by the software automatically.

10.6.3 Participant

Participants are represented by the ARIS object type **Participant**. The **Pool** symbol is available in the **Symbols** bar. If the ARIS attribute type **Multi-instance participant** is set to **true** the program will render the symbol: three vertical lines are displayed at the bottom of the pool symbol.

Participants/pools are described in detail in chapter Participant (Seite 204).

10.6.4 Artifacts

According to the metamodel artifacts are allowed in a conversation diagram. However, the relevance of groupings in a conversation diagram is not evident. For that reason only text annotations are implemented in the current version of the BPMN conversation diagram.

The symbol **Text annotation** is available in the **Symbols** bar. Artifacts and their usage are described in detail in chapter Artifacts (Seite 191).

10.6.5 Conversation link

A conversation link is used to link participants with conversation nodes. A conversation node has at least two participants.

There is an inconsistency in the specification: Sometimes the name **Communication links** is used, sometimes the name **Conversation link**.

N-ary (n > 2) conversations are allowed.

In ARIS the connection type **participates in** (Participant **participates in** Conversation) has been introduced. The passive name of the connection type is **has conversation link to** (Conversation **has conversation link to** Participant). Specific attributes are not required.

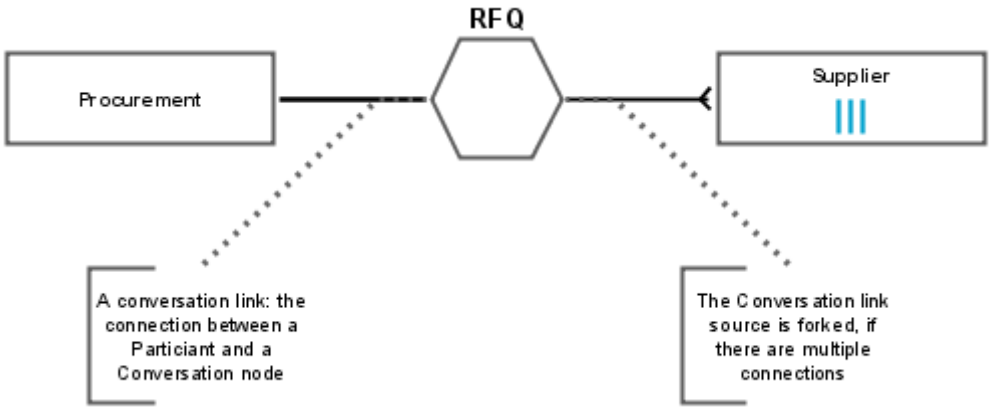


Abbildung 173: Conversation link with Participant multiplicity

The fork shown at the source of a conversation link must be manually set by the modeler using the property dialog of the relevant connection type.

10.6.6 Message flow in a conversation

According to the specification, it is allowed in the BPMN conversation diagram to model message exchanges between participants using message flows. (See: Business Process Model and Notation (BPMN), version 2.0.)

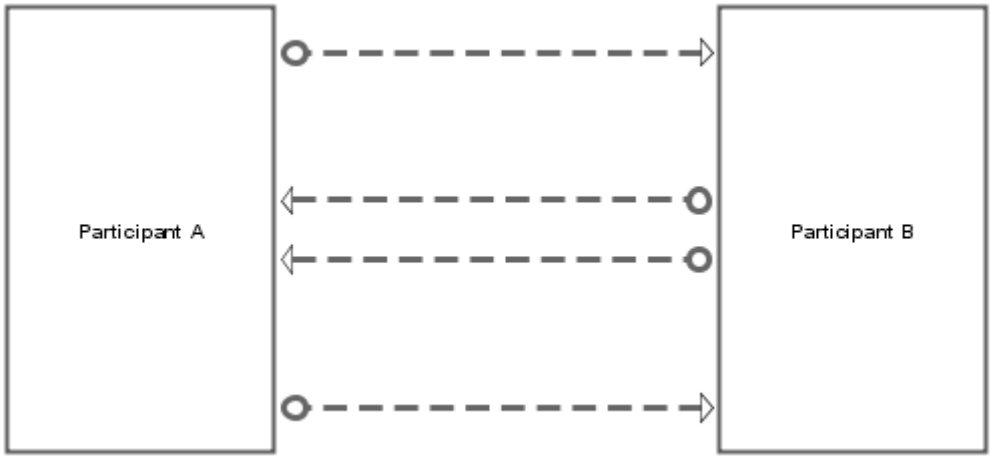


Abbildung 174: Message flow between Participants in a BPMN conversation diagram (BPMN 2.0)

Thus, the ARIS connection type **message flow** (Participant **message flow** Participant) is available in the BPMN conversation diagram (BPMN 2.0).

10.6.7 Model assignments

The object type **Conversation** has the following assignments:

- BPMN conversation diagram
- BPMN collaboration diagram.

Only one model of each type can be assigned to a conversation object.

10.7 Enterprise BPMN collaboration diagram

The model type **Enterprise BPMN Collaboration Diagram** is based on the BPMN Collaboration Diagram (BPMN 2.0).

It extents the **BPMN Collaboration Diagram (BPMN 2.0)** model type by ARIS constructs that are already available in the EPC, but that are out of scope in the BPMN specification. Thus, for example, the following object types can be (re-)used as lanes:

- Application system type
- Role
- Organizational unit
- Position
- Group

The **supports** connection is used to nest tasks in a lane object of the object type **Applications system type**. The **carries out** connection is used to nest tasks in the lane objects of the **Organizational** object types.

For all other nestings known from the BPMN specification the **belongs to** connection is used.

Similar to the EPC, an assignment relationship of the connection type **is process-oriented superior** is available between a function assigned to an Enterprise BPMN collaboration diagram and the tasks occurring in the assigned model.

11 Customer Experience Management (CXM)

Die Digitalisierung führt dazu, dass Verbraucher immer mehr Zeit online verbringen und Kundenprozesse immer häufiger auch digitale Interaktionen beinhalten. Resultierend daraus entwickelt sich das Customer Experience Management (CXM) wohl zu einem der wichtigsten Treiber für Innovation und Kundenloyalität.

CXM verfolgt das Ziel, das Konsumverhalten des Kunden positiv zu beeinflussen und jedem Kunden zu jederzeit über die richtigen Kanäle die Informationen zu liefern und anzubieten, die seinen Bedürfnissen entsprechen.

ARIS ermöglicht es, CXM-Projekte auf zwei unterschiedlichen Herangehensweisen zu realisieren. Zum einen besteht die Möglichkeit, ein CXM-Projekt mit dem Top-Down-Ansatz zu realisieren, d.h., das CXM-Projekt startet mit der Definition der Customer journey landscape und dem anschließenden Erstellen der Customer journey maps. Zum anderen besteht auch die Möglichkeit, die internen Prozesse um Customer touchpoints zu erweitern und diese bspw. in Customer touchpoint-Zuordnungsdiagrammen genauer zu spezifizieren und somit das CXM-Projekt nach Bottom-up-Ansatz zu gestalten.

11.1 Customer journey landscape

Das Modell **Customer journey landscape** ermöglicht die Beschreibung des Customer lifecycle mit den einzelnen Customer lifecylce stages und den Customer journeys. Dabei kann man das Modell entweder dafür nutzen, ausschließlich die Customer lifecycle stages zu modellieren und diese mit Customer journeys zu hinterlegen, oder man bildet sowohl die Customer lifecycle stages als auch die dazugehörigen Customer journeys in einem Modell ab.

Abbildung 175: Customer journey landscape

Geht man vom Top-Down-Ansatz aus, würde man mit der Customer journey landscape starten, welche eine klare Übersicht über alle Customer lifecylce stages und deren zugehörigen Customer journeys gibt.

Des Weiteren bietet der Objekttyp **Customer journey** die Möglichkeit, den Customer journey owner, den Geschäftsfaktor und den Business driver impact on transformation sowie die Overall Customer Experience mit Hilfe von Attributen abzubilden.

Durch die Aktivierung der zugehörigen CXM-Vorlagen ist es möglich, abhängig von den jeweilig definierten Attributwerten per Ampeldarstellung die jeweiligen Maßnahmen zu visualisieren, sodass der jeweils Verantwortliche direkt sehen kann, an welchen der Customer journeys welche Maßnahmen notwendig sind.

11.2 Customer journey map

Das Modell **Customer journey map** ist ein Spaltendiagramm und symbolisiert eine der Customer journeys aus dem Modell **Customer journey landscape**. Die Customer journey map ermöglicht es, die "Reise" des Kunden, die er mit der Organisation unternimmt und die seine Interaktionen mit dem Unternehmen charakterisieren, durch die Darstellung der Objekte **Customer journey step** und **Customer touchpoint** zu beschreiben. Dabei gestattet das Modell die detaillierte Beschreibung der Customer touchpoints u.a. mit den entsprechenden KPIs, organisatorischen Verantwortlichkeiten, Maßnahmen und Risiken.

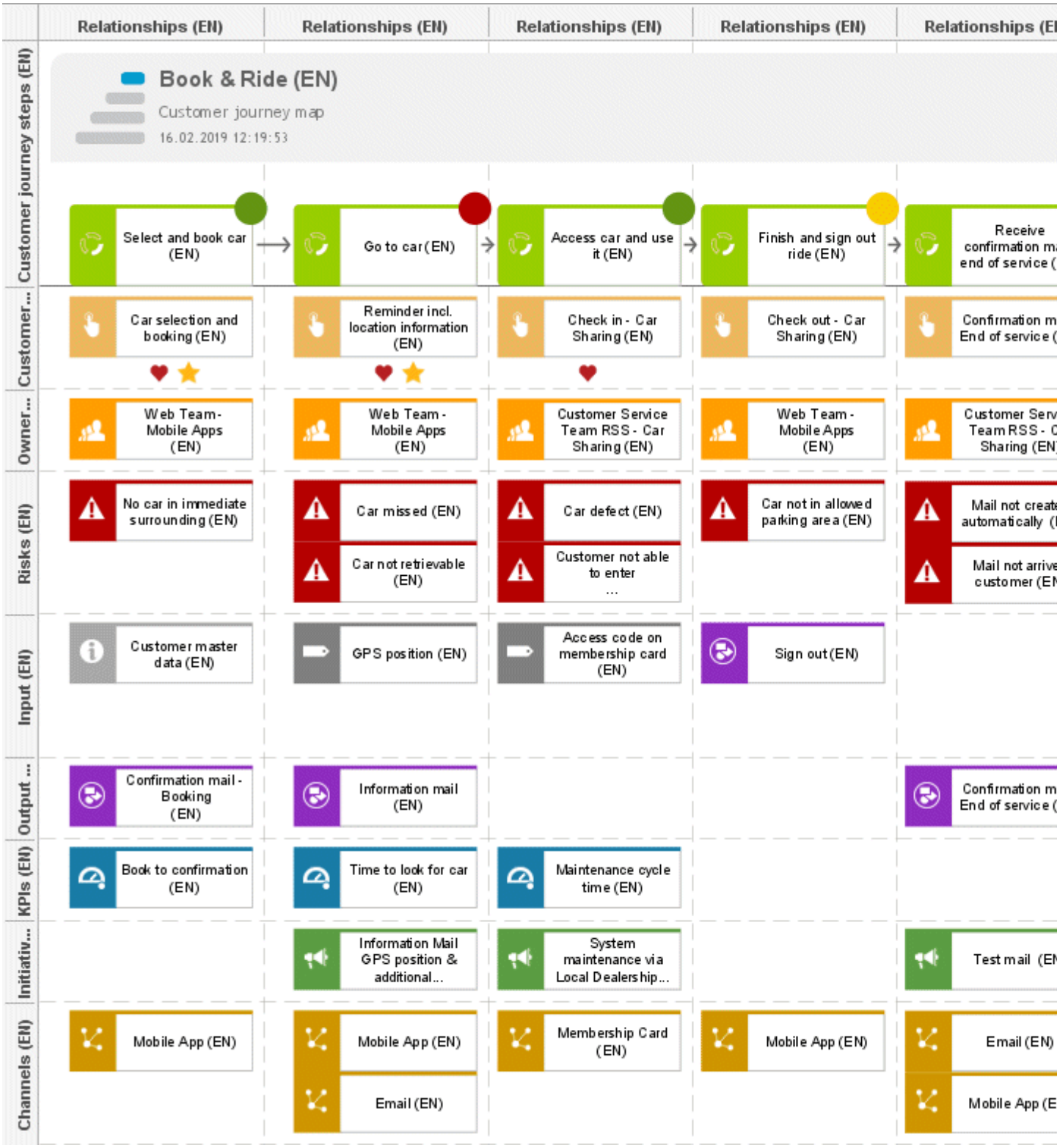


Abbildung 176: Customer journey map

Das zentrale Objekt des Modelltyps ist jedoch der **Customer touchpoint**, der mit einer Vielzahl unterschiedlicher Attribute beschrieben werden kann, zu diesen gehören u.a.:

- die Kundenziele
- die Kundenerwartungen und

- das Kundengefühl

Des Weiteren kann für den Customer touchpoint angegeben werden, ob es sich um einen Moment of truth, Pain point oder eine Best practice handelt. Diese Information wird aufgrund ihrer Wichtigkeit mit speziellen Icons im Modell angezeigt.

Abbildung 177: CXM-Symbole

Der Moment of Truth (kurz **MoT**) zeigt an, dass es sich bei dem Touchpoint um einen sehr wichtigen und entscheidenden Touchpoint für das Unternehmen und den Prozess handelt. Ein MoT kann darüber entscheiden, ob eine Beziehung zwischen Kunde und Unternehmen vertieft oder abgebrochen wird. Das Identifizieren von MoTs sollte daher eine hohe Priorität haben, da diese einen direkten Einfluss auf das Geschäft haben.

Um die Modellierung der Customer journey map so einfach wie möglich zu gestalten, bietet ARIS das Platzieren der Objekte per Drag & Drop an. Auch das Ziehen von Beziehungskanten ist bei diesem Modelltyp bis auf die Objekte des Typs **Customer journey step** nicht notwendig, da diese automatisch in Beziehung gesetzt werden. Alle Objekte in einer Spalte gehören zum Objekt **Customer touchpoint**. Dieser wiederrum steht in Beziehung zum Objekt **Customer journey step**.

Sollte es auf Grund mehrere Kanäle (Multichannel) dazu kommen, dass ein Customer journey step mehr als einen Customer touchpoint besitzt, besteht die Möglichkeit, die einzelnen Touchpoints mit dem Customer touchpoint-Zuordnungsdiagramm zu hinterlegen und detaillierter zu beschreiben. Wenn Sie die Touchpoints nicht hinterlegen, werden alle Objekte unter dem Customer touchpoint für jeden Touchpoint dieses Schrittes verwendet. Sie stellen damit nur eine allgemeine Spezifikation dar.

11.3 Customer touchpoint-Zuordnungsdiagramm

Das Modell **Customer touchpoint-Zuordnungsdiagramm** dient dazu, einen Customer touchpoint detaillierter mit den entsprechenden KPIs, Organisationseinheiten, Maßnahmen, Risiken etc. zu beschreiben.

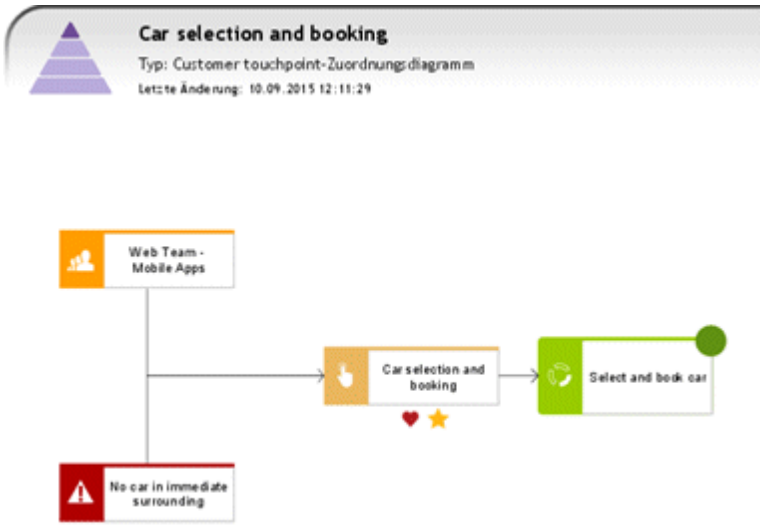


Abbildung 178: Customer touchpoint-Zuordnungsdiagramm

Neben der Hinterlegung des Customer touchpoint-Objekts in der Customer journey map bietet sich dieses Modell vor allem dann an, wenn Unternehmen bereits ARIS nutzen und die Customer touchpoints ausschließlich in ihren internen Prozessen und nicht zusätzlich in Customer journey map-Modellen abbilden möchten. In diesem Modell stehen zur Detaillierung des Customer touchpoints die gleichen Objekttypen zur Verfügung wie im Modell **Customer journey map**.

11.4 Customer touchpoint map

Das Modell **Customer touchpoint map** listet alle Customer touchpoints auf und kann z. B. als Startpunkt eines Customer Experience-Projekts genutzt werden, um die Interaktionspunkte mit dem Kunden zu identifizieren. Die Customer touchpoints werden anhand eines Kriteriums gruppiert, das für die Analyse wichtig sein kann, z. B. nach Organisationseinheit, Channel oder Risiko.

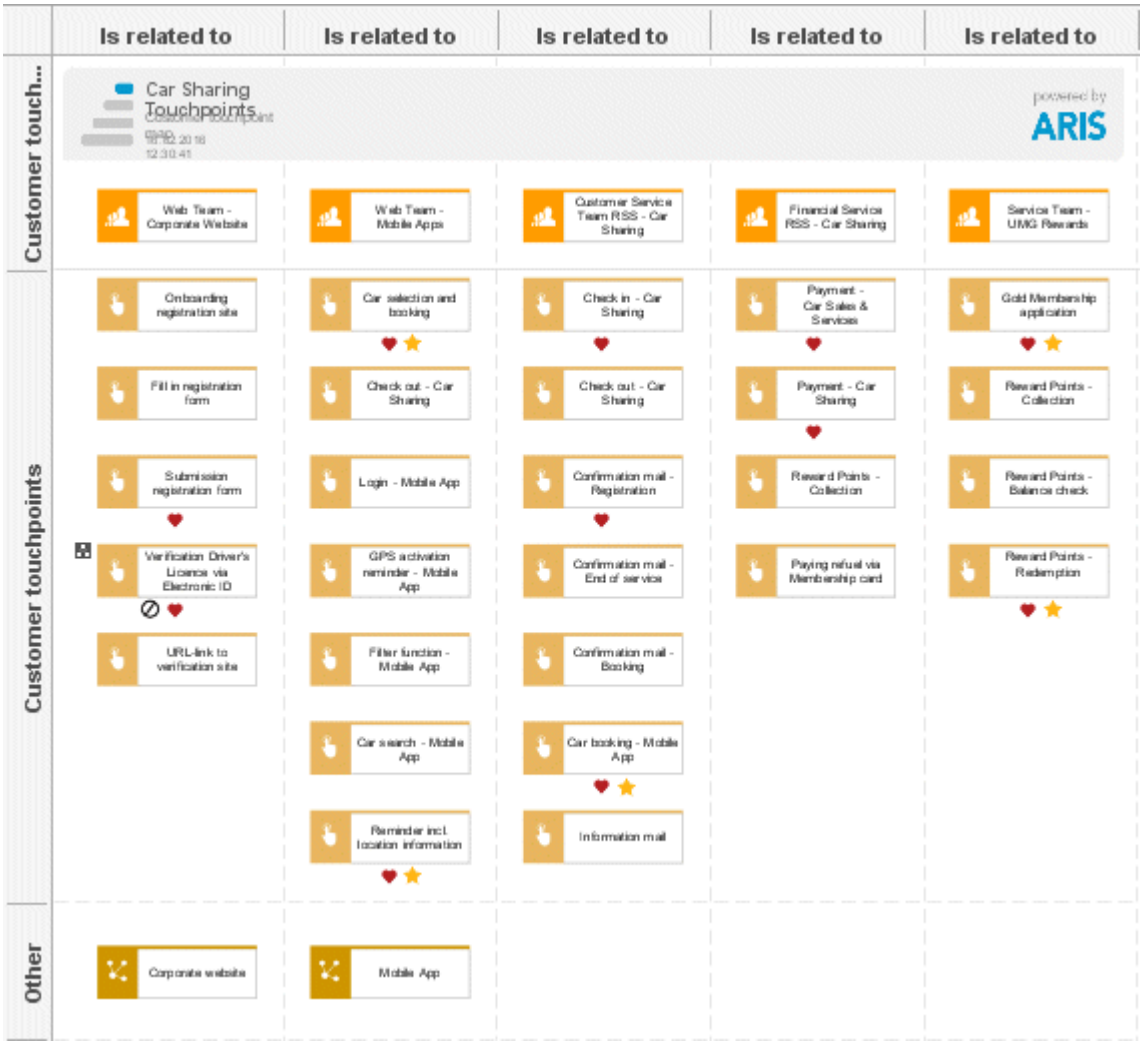


Abbildung 179: Customer touchpoint map

Die Customer touchpoint map kann jedoch auch dafür genutzt werden, um in einem vorhandenen Customer Experience-Projekt eine Übersicht über alle Touchpoints zu bekommen und diese zu administrieren. Sie ermöglicht es, auf einen Blick zu erkennen, bei welchen Customer touchpoints es sich um einen Moment of truth, Pain point und/oder Best practice handelt.

11.5 Verknüpfung von CXM und BPM

ARIS ermöglicht es, sowohl die internen als auch die externen Prozesse und Perspektiven abzubilden und diese mit Hilfe des Objekts **Customer touchpoint** miteinander in Verbindung zu setzen. Hat das Unternehmen bspw. einen Customer touchpoint in einer Customer journey map definiert, kann es diesen in einem internen Prozess ausprägen und wiederverwenden, z. B. in einer EPK, die für die Customer Journey wichtig ist. Durch diese Möglichkeit werden die beiden Modelle miteinander in Verbindung gesetzt und man kann z. B. durch Analysen schnell einen Überblick darüber bekommen, welche internen Prozesse durch Änderungen am Touchpoint beeinflusst werden.

Abbildung 180: Verknüpfung von CXM und BPM

11.5.1 Analysemöglichkeiten

Neben der Möglichkeit, CXM-Daten in den entsprechenden Modelltypen in ARIS zu visualisieren, können diese Daten per Report oder Abfragen ausgewertet werden.

11.5.1.1 Report

Der Report **Kundenzufriedenheit analysieren** visualisiert in einer Infografik, wie Kunden die Interaktion mit dem Unternehmen während einer Customer journey erleben und soll dabei helfen, die Zufriedenheit und die Probleme des Kunden aufzudecken.

Folgende Informationen werden ausgewertet und dargestellt:

- **Customer journey steps** mit **Customer touchpoints**
- **Moment of truth** und **Pain points** mit Beschreibung
- **Best practice**
- **Bedeutung für den Kunden** (Importance) & das **Kundengefühl** (Satisfaction)
- Prozentualer **Anteil von Pain points** in der **Customer journey map**
- Anzahl betroffene **interne Prozesse**

- Prozentualer **Anteil zufriedener Kunden**

Abbildung 181: CXM-Infografik

11.5.1.2 Abfragen

Mit Hilfe von Abfragen können Sie mit nur wenigen Mausklicks komplexe Zusammenhänge innerhalb einer ARIS-Datenbank visualisieren. Die Visualisierung der Daten und Zusammenhänge wird dabei grafisch vorgenommen, kann jedoch zusätzlich auch tabellarisch stattfinden.

11.5.1.3 Gesamtübersicht einer oder mehrerer Customer journeys erzeugen

Die Abfrage **Gesamtübersicht einer oder mehrerer Customer journeys erzeugen** wird per Objekttyp **Customer journey** gestartet.

Die Abfrage sammelt alle hinterlegten Customer journey maps der markierten Customer Journey-Objekte ein und liefert alle verfügbaren Informationen über die Customer journey steps und den zugehörigen Customer touchpoints. Die Information, die zu einem Customer touchpoint gehören, beschreiben den Customer touchpoint im Detail, z. B. die mit ihm verbundenen Risiken.

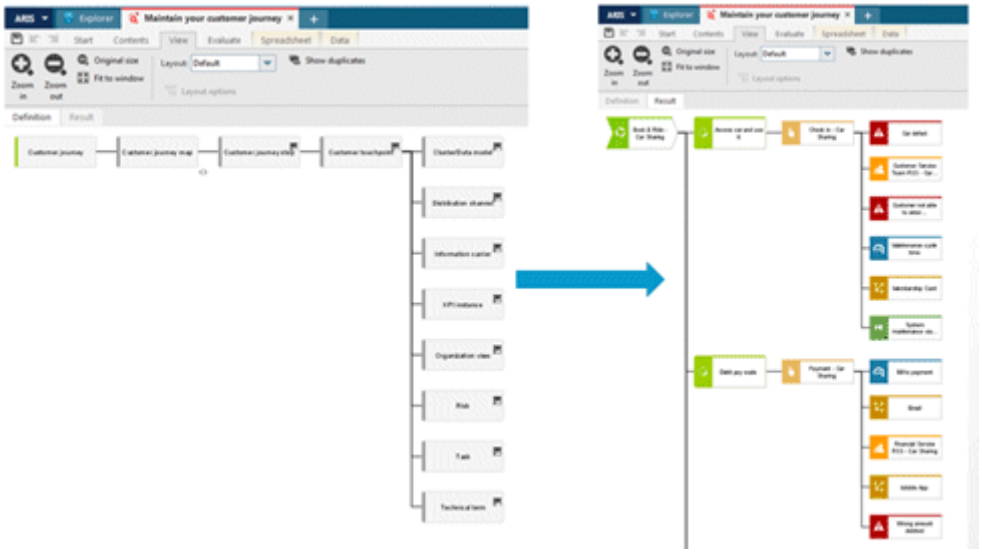


Abbildung 182: Customer journey-Übersicht

Neben der graphischen Darstellung liefert diese Abfrage auch eine Tabelle zur kompletten Übersicht über die Customer journey, welche unter anderem zur Datenpflege genutzt werden kann.

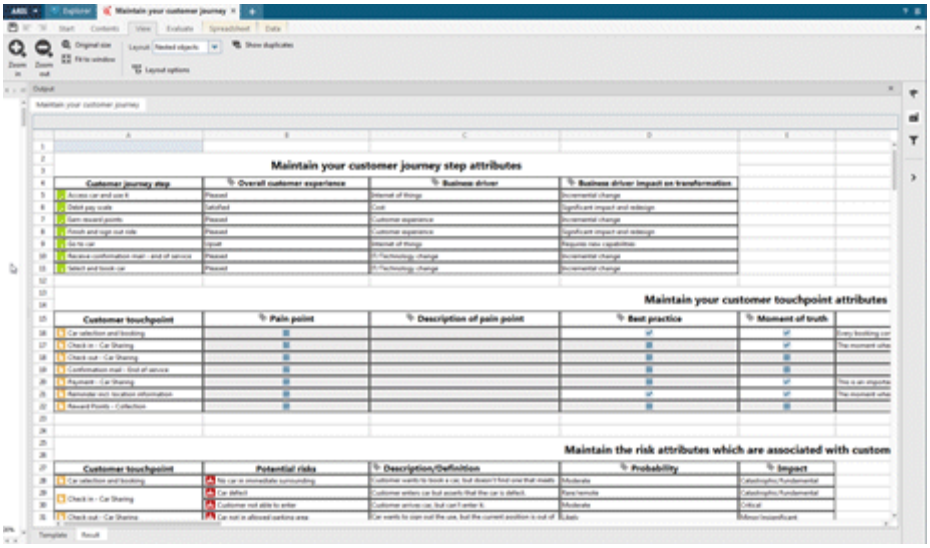


Abbildung 183: Gesamtübersicht: Customer journey-Tabelle

Die Tabelle enthält Informationen über die einzelnen Customer journey steps, z. B. die Gesamtkundenerfahrung, sowie alle wichtigen Details zu den Customer touchpoints der markierten Customer journey, z. B. bei welchen Touchpoints es sich um einen Pain point für den Kunden handelt und bei welchen nicht.

Die Tabelle bietet nicht nur die Möglichkeit, sich eine Übersicht zu verschaffen, sondern auch die Möglichkeit, Attribute zu pflegen.

11.5.1.4 Zum Risiko gehörende Customer touchpoints

Diese Abfrage wird per Objekttyp **Risiko** gestartet. Sie gibt alle zu einem Risiko gehörenden Customer touchpoints aus, um so eine schnelle Übersicht zu erhalten.

Abbildung 184: Customer touchpoints je Risiko (Abfrage)

11.5.1.5 Zur Verantwortlichkeit gehörende Customer touchpoints

Diese Abfrage wird per Objekttyp **Organisationseinheit** gestartet und gibt alle zu einem verantwortlichen Bereich gehörenden Customer touchpoints aus, um so eine schnelle Übersicht zu erhalten.

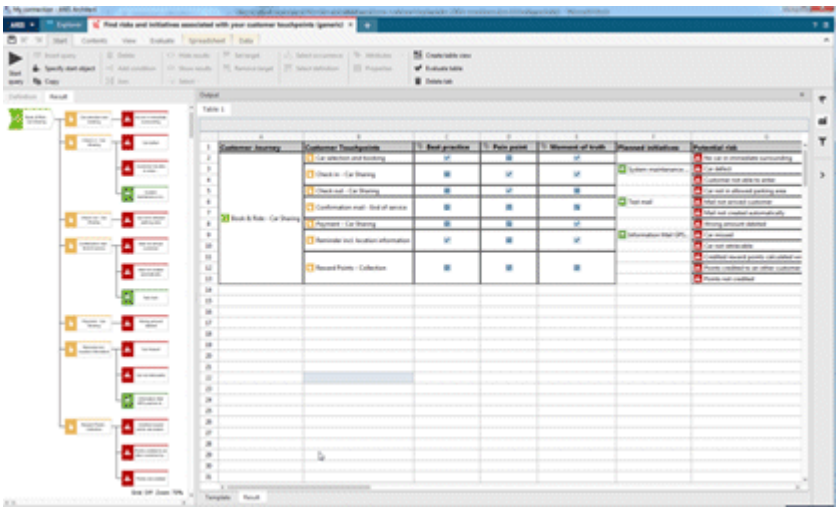


Abbildung 185: Risiken und Initiativen aller Customer touchpoints (Abfrage)

11.5.1.6 Zum Channel gehörende Customer touchpoints

Diese Abfrage wird per Objekttyp **Vertriebsweg** gestartet und gibt alle zu einem Vertriebsweg gehörenden Customer touchpoints aus, um so eine schnelle Übersicht zu erhalten.

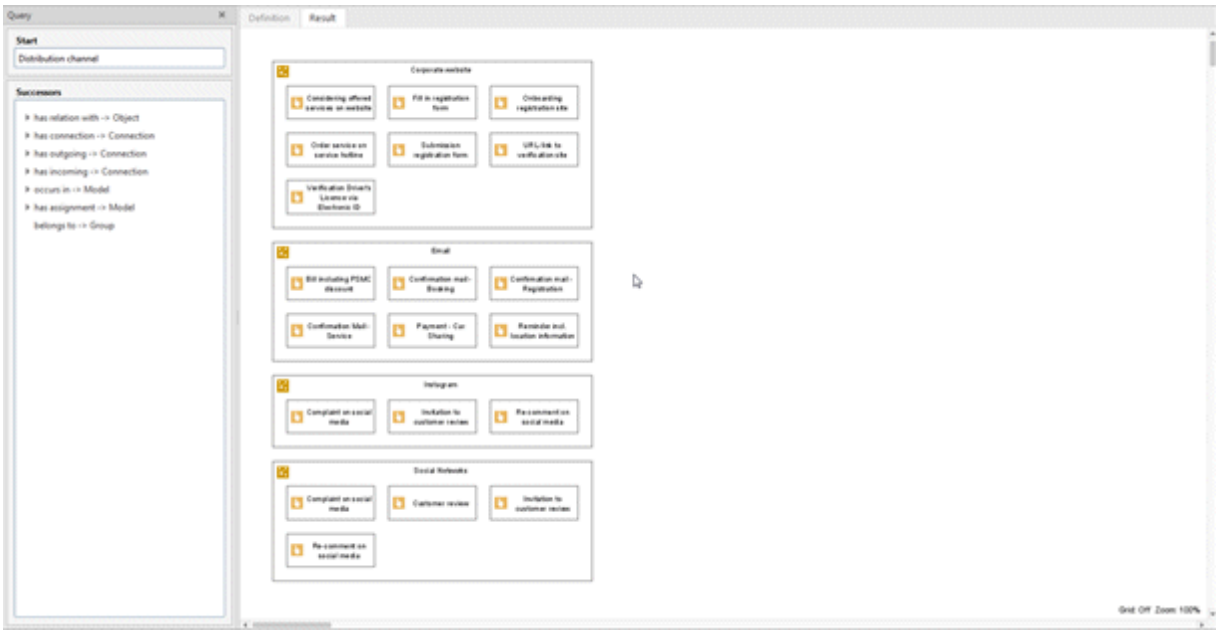


Abbildung 186: Zum Channel gehörende Customer touchpoints (Abfrage)

11.5.1.7 Risiken und strategische Initiativen aller Customer touchpoints

Die Abfrage **Risiken und strategische Initiativen aller Customer touchpoints** wird per Customer journey-Objekt gestartet und sammelt alle hinterlegten Customer journey maps der markierten Customer journey-Objekte und gibt die Customer touchpoints, ihre zugehörigen Risiken und geplante Maßnahmen aus.

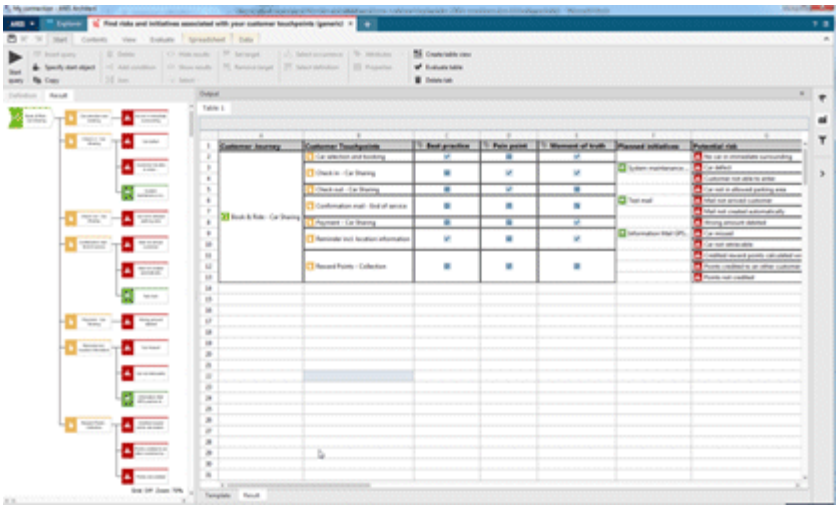


Abbildung 187: Risiken und Initiativen aller Customer touchpoints (Abfrage)

In der dazugehörigen Tabelle werden Informationen zu den Customer journey-Objekten und den dazugehörigen Customer touchpoints abgebildet. Weitere Informationen zu den Touchpoints sind die Werte zu den Attributen Pain point, Moment of truth und Best practice. Außerdem enthält die Tabelle die Risiken, die mit den einzelnen Touchpoints einhergehen, sowie die geplanten Maßnahmen.

11.5.1.8 Risiken und strategische Initiativen mangelhafter Customer touchpoints

Die Abfrage **Risiken und strategische Initiativen mangelhafter Customer touchpoints** wird per Customer journey-Objekt gestartet. Sie beinhaltet ähnliche Informationen wie die Abfrage **Risiken und strategische Initiativen aller Customer touchpoints**, jedoch mit dem Unterschied, dass nicht alle Customer touchpoints ausgewertet werden, sondern nur die, für die das Attribut **Pain point** als zutreffend gepflegt und somit der Touchpoint für den Kunden negativ ist. Der Fokus der Abfrage liegt hier auch auf den Risiken der einzelnen Touchpoints und liefert somit Informationen über die Risiken, z. B. die Eintrittswahrscheinlichkeit.

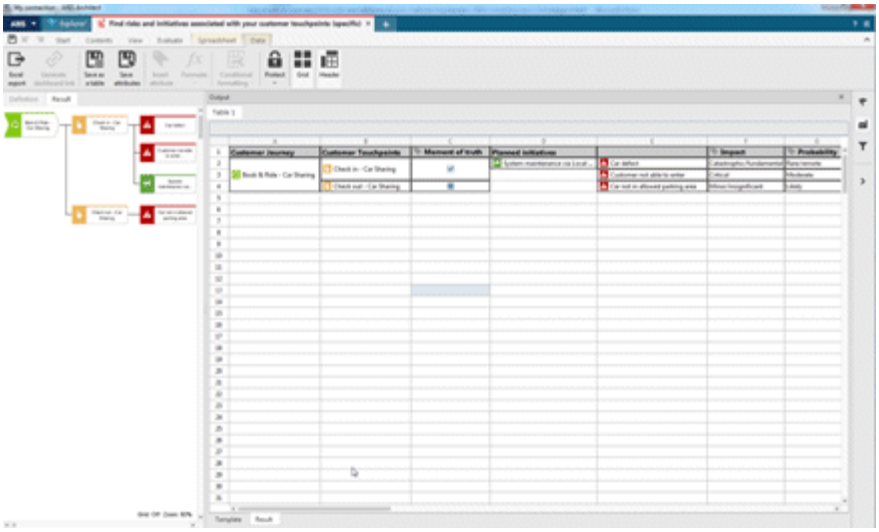


Abbildung 188: Risiken und Initiativen mangelhafter Customer touchpoints (Abfrage)

11.5.1.9 Mit Customer journeys verbundene Prozesse

Die Abfrage wird per Funktion eines Wertschöpfungskettendiagramms (WKD) gestartet.

Sie dient dazu Customer touchpoints in internen Prozessen zu finden und damit festzustellen, welche Customer journey map mit welchem internen Prozess in Verbindung steht.

Die Abfrage gibt zunächst alle EPKs aus, die den markierten Funktionen hinterlegt sind. Dann gibt Sie alle Funktionen der EPKs aus, die eine Beziehung zu Customer touchpoints haben.

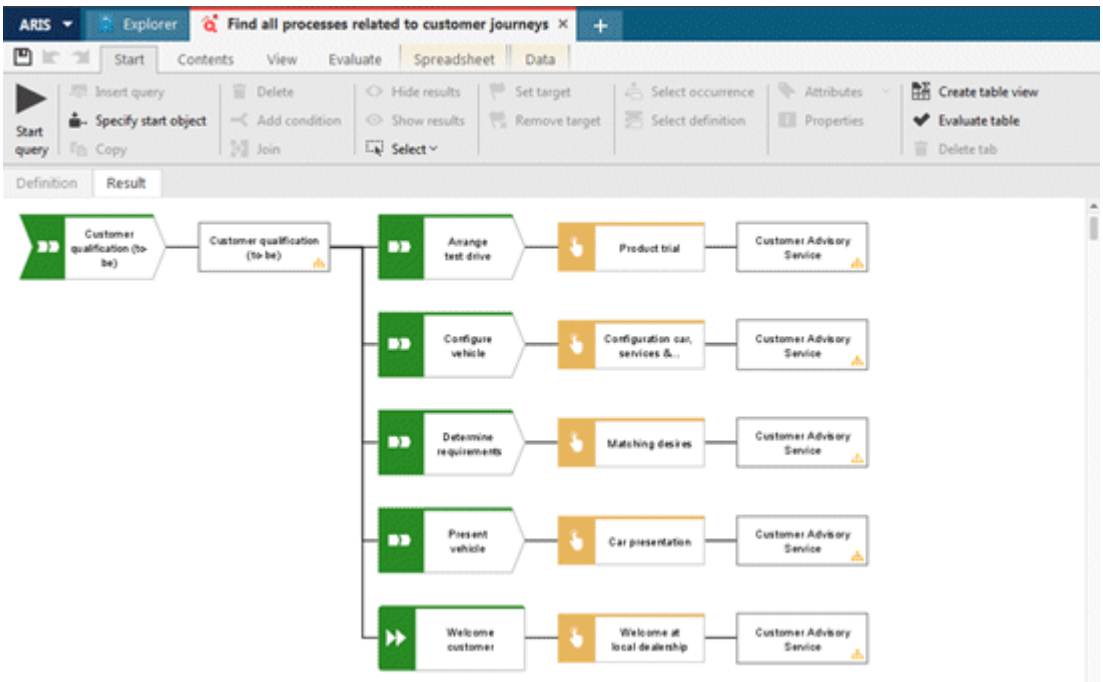


Abbildung 189: Mit der Funktion verbundene Customer journeys

Neben der grafischen Darstellung der Zusammenhänge enthält diese Abfrage eine Tabelle, die folgende Informationen enthält:

- Die relevante EPK.
- Den Prozess, der eine Beziehung zu einem Customer touchpoint hat.
- Den Customer touchpoint.
- Die Information darüber, ob der Customer touchpoint ein Pain point oder ein Moment of truth ist.
- Den Namen der Customer journey, in der der Customer touchpoint ausgeprägt ist.

12 Anwendungsfälle

Dieses Kapitel soll helfen, für bestimmte betriebswirtschaftliche Problemstellungen die richtige Unterstützung durch ARIS zu finden. Hierzu ist das Kapitel in Anwendungsszenarien (Unterkapitel) aufgeteilt.

Für jedes Anwendungsszenario ist kurz beschrieben, was unter dem Szenario zu verstehen ist und welche Tätigkeiten standardmäßig in dem Szenario durchgeführt werden. Anschließend sind Aufgaben beschrieben, die typischerweise in dem Szenario vorkommen. Zu jeder Aufgabe ist dargestellt, wie man ARIS bei der Lösung der Aufgabe einsetzen kann.

Eine Übersicht der beschriebenen Anwendungsfälle nebst zu verwendenden Modelltypen zeigt die nachfolgende Tabelle:

Szenario	Szenarioaufgaben	Modelltypen
Allgemeine Unternehmensdokumentation (Seite 292)	Dokumentation der Unternehmensziele Dokumentation der Unternehmenswertschöpfung Dokumentation der Organisationsstruktur Dokumentation der Unternehmensfunktionen Prozessdokumentation Process Warehousing	Zieldiagramm Wertschöpfungskettendiagramm Organigramm Funktionsbaum EPK
Datenbankmanagement/ Data Warehousing (Seite 293)	Datenstrukturierung/Datenbankentwurf Datenbank-/Zugriffsverwaltung	ERM IEF-Datenmodell Tabellendiagramm Klassendiagramm
PC-Hardware- und Netzwerkmanagement (Seite 294)	Ermittlung des informationstechnischen Infrastrukturbedarfs Dokumentation der informationstechnischen Infrastruktur Zugriffsrechte	Netztopologie Netzdiagramm
Prozesskostenrechnung (Seite 295)	Beschreibung der Prozess- und Organisationsstrukturen Kostenstellenanalyse Prozesskalkulation	EPK Organigramm

Szenario	Szenarioaufgaben	Modelltypen
Qualitätsmanagement (Seite 296)	Aufbau von QM-Dokumentationen Zertifizierungsvorgehen Zertifizierungsunterlagen	Produktbaum Produktauswahlmatrix EPK Wertschöpfungskettendiagramm Gliederungsmodell Organigramm
Reorganisationsmaßnahmen (Seite 297)	Projektdokumentation Reorganisationsdurch-führung	Wertschöpfungskettendiagramm EPK Organigramm Produktmodell Leistungsmodell Zieldiagramm
SAP R/3-Einführung (Seite 299)	Phase Analyse, Spezifikation (Project Preparation) Phase Design (Business Blueprint); 3 Aufgabenfälle vorhanden	EPK Organigramm
Software-Entwicklung und -einführung (Seite 300)	Projektdokumentation Festlegung von Anwendungssystemen und Modulen Beschreibung der DV-Prozesse Entwicklung der Systemoberfläche	Wertschöpfungskettendiagramm Organigramm EPK Use-Case-Diagramm Anwendungssystemtypdiagramm Programmablaufplan Maskendiagramm
Wissensmanagement (Seite 302)	Wissenslandkarte bzw. Yellow Page Wissenskategorisierung Wissensverarbeitung in Geschäftsprozessen	Wissenslandkarte Wissensstrukturdiagramm EPK Funktionszuordnungsdiagramm
Workflow-Management (Seite 303)	Prozess-Customizing von Workflow-Management-Systemen	EPK Funktionszuordnungsdiagramm Anwendungssystemdiagramm Anwendungssystemtypdiagram

Szenario	Szenarioaufgaben	Modelltypen
		m

12.1 Allgemeine Unternehmensdokumentation

Zu Schulungs-, Präsentations- und Bewertungszwecken jedweder Art können Unternehmenscharakteristika, z. B. Abläufe, Strukturen und Daten, in geeigneter Form dokumentiert werden. Die wichtigsten Aufgaben der Unternehmensdokumentation sind im Folgenden kurz beschrieben.

AUFGABE: DOKUMENTATION DER UNTERNEHMENSZIELE

ARIS-Unterstützung: Zur Hierarchisierung von Unternehmenszielen und den dazugehörigen Erfolgsfaktoren können Zieldiagramme verwendet werden.

AUFGABE: DOKUMENTATION DER UNTERNEHMENSWERTSCHÖPFUNG

Grundlage vieler Unternehmensentscheidungen ist die Identifikation der an der Wertschöpfung einer Unternehmung beteiligten Funktionen.

ARIS-Unterstützung: Die an der Wertschöpfung beteiligten Unternehmensfunktionen können durch das Wertschöpfungskettendiagramm dargestellt werden. Dieses Modell macht sowohl die Reihenfolge der aufeinander folgenden Funktionen als auch über- und untergeordnete Funktionen deutlich.

AUFGABE: DOKUMENTATION DER ORGANISATIONSSTRUKTUR

ARIS-Unterstützung: Die Struktur einer Unternehmung kann durch Organigramme dokumentiert werden, die Hierarchie und Beziehungen von Organisationseinheiten darstellen.

AUFGABE: DOKUMENTATION DER UNTERNEHMENSFUNKTIONEN

ARIS-Unterstützung: Durch einen Funktionsbaum können einzelne Unternehmensfunktionen übersichtartig dargestellt werden. Die Funktionen werden entweder objektorientiert, prozessorientiert oder verrichtungsorientiert gegliedert.

AUFGABE: PROZESSDOKUMENTATION

ARIS-Unterstützung: Falls Prozessmodelle für SAP®-Anwendungen, Simulationen, Workflows etc. benutzt werden sollen, wird die Modellierung mittels EPKs empfohlen.

AUFGABE: PROCESS WAREHOUSING

Unter einem Process Warehouse versteht man die systematische Erfassung, Speicherung und Pflege des Geschäftsprozesswissens in einem Repository.

ARIS-Unterstützung: EPKs, angereichert durch Dokumente, Bilder und Videos, sind für die Modellierung des Prozesswissens nützlich. Damit können die Modelle für anspruchsvolle Auswertungen, wie Simulation oder Prozesskostenrechnung, genutzt werden können.

12.2 Datenbankmanagement/ Data Warehousing

Das Speichern von Unternehmensdaten in Datenbanken reduziert eine redundante Datenhaltung und ermöglicht den programmunabhängigen Zugriff auf unternehmensweite Daten. Durch Data Warehousing soll die Qualität, die Integrität und die Konsistenz der zugrunde liegenden Daten sichergestellt werden. Der Begriff des Data Warehouse bezeichnet generell eine von den operationalen DV-Systemen isolierte Datenbank, die als unternehmensweite Datenbasis für alle Ausprägungen managementunterstützender Systeme dient. Er ist durch eine strikte Trennung von operationalen und entscheidungsunterstützenden Daten und Systemen gekennzeichnet. Der Fokus des Data Warehouse-Konzeptes liegt auf der effizienten Bereitstellung und Verarbeitung großer Datenmengen für die Durchführung von Auswertungen und Analysen in entscheidungsrelevanten Prozessen.

AUFGABE: DATENSTRUKTURIERUNG/DATENBANKENTWURF

Die Struktur von Datenbanken wird durch die zugrunde liegenden Datenmodelle vorgegeben.

ARIS-Unterstützung: Die verbreitetste Methode der Datenmodellierung ist die des Entity-Relationship-Modells (ERM), welches die Grundlage zur Implementierung einer relationalen Datenbank legt.

Tabellen und Felder eines Datenbanksystems werden durch das Tabellendiagramm beschrieben.

Objektorientierte Datenbanksysteme können mit Hilfe der Unified Modeling Language (UML) entworfen werden. In der UML werden mit dem Class Diagramm die statischen Datenbeziehungen dargestellt.

AUFGABE: DATENBANK-/ZUGRIFFSVERWALTUNG

Zuordnung von Anwendern und Systemverantwortlichen zu den Datenbanksystemen.

ARIS-Unterstützung: Durch das Zugriffsdiagramm können in Verbindung mit Relationen und Systemkomponenten Zugriffsrechte von Organisationseinheiten, Stellen und Personen auf das Datenbanksystem festgelegt werden.

12.3 PC-Hardware- und Netzwerkmanagement

Unter Netzwerkmanagement versteht man die Steuerung, Überwachung und Koordination aller (verteilten) Ressourcen (Datennetze, Computer, Daten und Anwendungen), die eine Kommunikation in einem Computernetz ermöglichen.

AUFGABE: ERMITTLUNG DES INFORMATIONSTECHNISCHEN INFRASTRUKTURBEDARFS

Aus einer bestehenden Aufbauorganisation soll abgeleitet werden, durch welche Kommunikations- und Informationssystemlandschaften diese effizient unterstützt werden kann.

ARIS-Unterstützung: Mit dem Modelltyp Netztopologie sind die Anforderungen an den strukturellen Aufbau von Informationssystemen abbildbar. Bei der Darstellung der Anwendungssysteme, Netztypen und Hardwarekomponenten handelt es sich nicht um einzeln identifizierbare Exemplare (z. B. PC mit Inventarisierungsnummer 3423), sondern um Typisierungen auf gleicher technologischer Basis.

AUFGABE: DOKUMENTATION DER INFORMATIONSTECHNISCHEN INFRASTRUKTUR

Aufgabe ist die Abbildung einer bestehenden oder zu installierenden IT-Infrastruktur mit konkreten Hardwarekomponenten, Netzen und Anwendungssystemen.

ARIS-Unterstützung: Eine IT-Infrastruktur lässt sich mit Hilfe des Netzdiagramms darstellen, als konkrete Umsetzung einer Netztopologie.

AUFGABE: ZUGRIFFSBERECHTIGUNGEN

Aufgabe ist aufzuzeigen, welche Applikationen und Benutzer in welcher Weise Zugriff auf welche Daten haben.

ARIS-Unterstützung: Mit dem Zugriffsdiagramm kann beschrieben werden, welche Applikationen bzw. Applikationsmodule in welcher Form (schreiben/lesen/verändern) Zugriff auf Datenbestände und Informationsträger haben und ob die Daten als Input oder Output agieren. Weiterhin kann abgebildet werden, welche Benutzerberechtigungen und Ansichten bestimmte Benutzer bzw. Benutzergruppen auf die Applikationen bzw. Applikationsmodule besitzen.

12.4 Prozesskostenrechnung

Die Kostenrechnung stellt durch die Erfassung und Zurechnung der bei der betrieblichen Leistungserstellung anfallenden Kosten eine Dispositionsgrundlage und ein Kontrollinstrument zur Verfügung. Durch die Änderung der Kostenstrukturen, insbesondere durch den Anstieg der Gemeinkosten, werden die traditionellen Kostenrechnungsmethoden durch die Prozesskostenrechnung ersetzt. Die Prozesskostenrechnung bestimmt die Kosten von Prozessen kostenstellenübergreifend. Budgetplanung, Kostentransparenz in den indirekten Leistungsbereichen, Preisfindung und Unterstützung bei Make-or-Buy-Entscheidungen sind die Hauptvorteile der Prozesskostenrechnung.

AUFGABE: BESCHREIBUNG DER PROZESS- UND ORGANISATIONSSTRUKTUREN

Festlegen der Prozesse, auf die die Prozesskostenrechnung angewendet werden soll, sowie Beschreibung der Kostenstellen.

ARIS-Unterstützung: Prozesse werden mit den Standardmodelltypen (zum Beispiel EPK) abgebildet. Wichtig für die Prozesskostenrechnung sind die Pflege der Zeitenattribute sowie eine Zuordnung von Organisationseinheiten.

Die Unternehmensorganisation wird in einem Organigramm beschrieben, wobei die Organisationseinheiten Kostenstellen (u. a. mit den Attributen **Kostensatz** und **Leistung**) entsprechen.

AUFGABE: PROZESSKALKULATION

ARIS-Unterstützung: Voraussetzung ist eine vollständige Kostenstellenanalyse mit einer Ermittlung der Prozesskostensätze. Für die Durchführung der Prozesskalkulation müssen keine weiteren Modelle gepflegt werden. Die Ergebnisse werden in einer Kalkulationstabelle ausgewiesen.

12.5 Qualitätsmanagement

Unter Qualitätsmanagement (QM) versteht man alle Tätigkeiten, die die Qualitätspolitik, Ziele und Verantwortungen einer Unternehmung festlegen. Mittel zur Verwirklichung dieser Tätigkeiten sind Qualitätsplanung, Qualitätslenkung (Prozessmanagement), Qualitätssicherung und Qualitätsverbesserung (Qualitätsförderung).

AUFGABE: AUFBAU VON QM-DOKUMENTATIONEN

Um die Qualität von Produkten und Prozessen innerhalb einer Unternehmung zu sichern, ist eine geeignete Dokumentation anzufertigen, die Produkte und Prozesse bewertbar, vergleichbar und damit verbesserbar macht.

ARIS-Unterstützung: Zur Dokumentation von Produkten eignen sich Produktbäume, mit deren Hilfe sich Produkte klassifizieren lassen. Diese Art der Darstellung wird zunehmend im Dienstleistungsbereich und dort speziell in der öffentlichen Verwaltung angewendet. Mit der Produktauswahlmatrix ist weiter darstellbar, welche Unternehmensfunktionen zur Erstellung welcher Produkte notwendig sind und welche Organisationseinheiten für die Produkterstellung verantwortlich sind.

Weiterhin geht es bei QM-Dokumentationen hauptsächlich um die Dokumentation von Abläufen, die mit Hilfe von EPKs festgehalten werden, durch Reports ausgewertet werden und auf Dokumente und Anwendungen im Unternehmen verweisen können.

AUFGABE: ZERTIFIZIERUNGSVORGEHEN

Unterstützung des Projektmanagements bei der Zertifizierung nach national und international anerkannten Normen, wie ISO oder VDA, durch Vorgehensmodelle.

ARIS-Unterstützung: Ein Vorgehensmodell zur Zertifizierung (z. B. das ARIS-Vorgehensmodell) kann mit Hilfe eines Wertschöpfungskettendiagramms dargestellt werden. Die einzelnen Schritte können durch Hinterlegung weiterer Prozessmodelle detaillierter beschrieben werden.

AUFGABE: ZERTIFIZIERUNGSUNTERLAGEN

Erstellung der notwendigen Qualitätsunterlagen für die Zertifizierung.

ARIS-Unterstützung: Durch das Gliederungsmodell werden einzelne Zertifizierungsnormen in ihre Komponenten untergliedert. Den einzelnen Elementen des Gliederungsmodells können zur Qualitätslenkung Unternehmensmodelle hinterlegt werden. Diese Modelle können beispielsweise Prozessmodelle in Form von EPKs, Organigramme oder auch Wertschöpfungskettendiagramme sein .

Mit ARIS Architect können Reporte generiert werden, die als QM-Handbuch zur Zertifizierung anerkannt sind.

12.6 Reorganisationsmaßnahmen

Im Rahmen von Reorganisationsmaßnahmen zur Kosten- oder Zeitreduktion, Verbesserung der Ergebnisqualität oder Verbesserung der Arbeitsqualität werden Unternehmensabläufe entweder verändert (Prozess-Redesign) oder komplett neu entworfen (Prozess-Reengineering).

AUFGABE: PROJEKTDOKUMENTATION

Dokumentation der Planung, des Vorgehens und der Ergebnisse der Reorganisationsmaßnahmen.

ARIS-Unterstützung: Die Hauptprojektphasen der Reorganisation können als Vorgehensmodell durch ein Wertschöpfungskettendiagramm beschrieben werden.

Die einzelnen Projektaktivitäten im Reorganisationsprojekt können in ihren Abläufen mit Hilfe von EPKs dokumentiert werden.

Die Organisation der am Projekt teilhabenden Personen und Einheiten kann durch Organigramme dargestellt werden.

AUFGABE: REORGANISATIONSDURCHFÜHRUNG

Im Rahmen einer Reorganisation folgen nach der Projektvorbereitung und der strategischen Planung die Analyse der Ist-Situation, die Entwicklung des Soll-Konzepts und schließlich die Implementierung der Lösungen.

ARIS-Unterstützung: Zur Dokumentation der strategischen Rahmenbedingungen werden Produkt- und Leistungsmodelle sowie Zieldiagramme eingesetzt, damit die wesentlichen Geschäftsfelder des Unternehmens mit ihren Produkten, Dienstleistungen und Kundengruppen erfasst und die kritischen Erfolgsfaktoren sowie die Zielhierarchie des Unternehmens abgebildet werden.

Während der Ist-Analyse wird mit Hilfe von Wertschöpfungskettendiagrammen zunächst ein Rahmen mit den wesentlichen Geschäftsprozessen erarbeitet. Diese Geschäftsprozesse werden danach detailliert auf der Grundlage von Interviews mit den Mitarbeitern als EPK erfasst.

Nach einer Schwachstellenanalyse unter Berücksichtigung von Durchlaufzeiten, Prozesskosten, organisatorischen Brüchen, System- und Medienbrüchen, Datenredundanzen u. a. werden alternative Soll-Abläufe definiert. Diese werden, wie zuvor in der Ist-Erfassung geschehen, mit Hilfe von EPKs modelliert.

Zur Unterstützung der Implementierung werden System-, Organisations- und Datenkomponenten am Ende der Soll-Konzeption konkreter dargestellt. Beispielsweise kann die Angabe **Textverarbeitung** als Anwendungssystemkonstrukt nun als Microsoft® Word spezifiziert werden.

Hinweis: Die Phase der Schwachstellenanalyse kann durch Auswertungen mit Hilfe von Simulation unterstützt werden.

12.7 SAP R/3-Einführung

Die Unterstützungspotenziale von ARIS bei der Einführung der Standardsoftware R/3 der SAP AG orientieren sich an dem Lebenszyklus des Einführungsansatzes ASAP. Neben ASAP werden jedoch auch Ansätze unterstützt, die mehr an der Geschäftsprozessoptimierung (im weitesten Sinne) orientiert sind. Teile der nachfolgend beschriebenen ARIS-Unterstützung sind nur mit Extension pack SAP® möglich.

AUFGABE: PHASE ANALYSE, SPEZIFIKATION (PROJECT PREPARATION)

Aufgabe ist die Identifikation des Abdeckungsgrades der unternehmensspezifischen Prozesse durch das SAP®-System sowie die rechtzeitige Identifikation von möglichen Schwachstellen.

ARIS-Unterstützung: Wird die Untersuchung nicht direkt durch das SAP R/3-Referenzmodell durchgeführt, so können die zu unterstützenden, „betriebswirtschaftlich optimalen“ Prozesse mit ARIS modelliert werden (siehe allgemeine Unternehmensdokumentation). Anschließend kann durch Reporte eine erste Schätzung des Abdeckungsgrades ermittelt werden, indem das Unternehmensmodell mit dem SAP R/3-Referenzmodell gemappt wird.

AUFGABE: PHASE DESIGN (BUSINESS BLUEPRINT)

Prozess- und/oder Funktionsdefizite wurden in dem SAP R/3-Referenzmodell identifiziert.

ARIS-Unterstützung: Mit den aus dem SAP R/3-Referenzmodell bestehenden Bausteinen können in ARIS neue Prozess- und Szenariovarianten abgeleitet werden. Zusätzlich können die Prozess- und Szenariobausteine um neue Funktionen, Ereignisse und Regeln erweitert werden (d.h. ggf. durch die Entwicklung neuer SAP®-ABAP-Funktionen).

AUFGABE: PHASE DESIGN (BUSINESS BLUEPRINT)

Entwurf von Schnittstellen zwischen SAP® R/3 und Non-SAP®-Anwendungen.

ARIS-Unterstützung: Die Dokumentation der auszutauschenden Attribute kann im ARIS-Repository in Prozessmodellen durch Daten- bzw. Objektzuordnungen zu Funktionen und Daten- bzw. Objektmodellen detailliert beschrieben werden. Diese Informationen können als Reporte ausgegeben werden und bilden die Grundlage für die Schnittstellenentwicklung.

AUFGABE: PHASE DESIGN (BUSINESS BLUEPRINT)

Entwicklung der organisatorischen Gestaltung der Geschäftsprozesse und des SAP®-Systems.

ARIS-Unterstützung: Mittels Organigrammen kann die soziale Unternehmensstruktur und die SAP®-Organisationsstruktur dargestellt und einander gegenübergestellt werden.

12.8 Software-Entwicklung und -einführung

AUFGABE: PROJEKTDOKUMENTATION

Dokumentation der Planung, des Vorgehens und der Ergebnisse der Software-Entwicklung und -Einführung.

ARIS-Unterstützung: Die Hauptprojektphasen des Projektes können als Vorgehensmodell durch ein Wertschöpfungskettendiagramm beschrieben werden.

Die einzelnen Projektaktivitäten während Entwicklung und Einführung können in ihren Abläufen mit Hilfe von EPKs dokumentiert werden.

Die Organisation der am Projekt teilhabenden Personen und Einheiten kann durch Organigramme dargestellt werden.

AUFGABE: FESTLEGUNG VON ANWENDUNGSSYSTEMEN UND MODULEN

Aufgabe ist es, bestehend auf den Systemanforderungen, den strukturellen Aufbau eines Informationssystems darzulegen.

ARIS-Unterstützung: Das Use-Case-Diagramm kann zur Identifikation der Anwendungsfälle des zu entwickelnden Softwaresystems dienen. Weiterhin werden dadurch die Systemanwender definiert und einzelnen Anwendungsfällen zugeordnet. Das Use-Case-Diagramm bildet häufig den Ausgangspunkt für die detaillierte Prozessmodellierung. Einzelnen Use-Cases können Prozessmodelle hinterlegt werden.

Mit dem Anwendungssystemtypdiagramm kann der hierarchische Aufbau von Anwendungssystemen auf Typebene durch Modultypen und DV-Funktionstypen beschrieben werden.

Für die spezielle Beschreibung der einzelnen Typen durch konkrete Ausprägungen kann das Anwendungssystemtypdiagramm verwendet werden.

AUFGABE: BESCHREIBUNG DER DV-PROZESSE

Aufgabe ist es, den zeitlich-logischen Ablauf innerhalb von Modulen und modulübergreifend zu beschreiben.

ARIS-Unterstützung: Mit dem Programmablaufdiagramm können DV-Prozesse modelliert werden.

AUFGABE: ENTWICKLUNG DER SYSTEMOBERFLÄCHE

Aufgabe ist die Entwicklung und Dokumentation der Benutzerschnittstelle.

ARIS-Unterstützung: Der geographische und funktionale Aufbau einer Bildschirmmaske (Fenster) kann mit einem Maskendiagramm beschrieben werden. Hieraus kann der Programmcode abgeleitet werden.

12.9 Wissensmanagement

Ausgangspunkt für die Gestaltung eines umfassenden Wissensmanagements ist die These, dass Wissen de facto zum dominanten Produktionsfaktor von Unternehmen geworden ist bzw. wird. Daraus ergibt sich die Notwendigkeit, Wissen als ebenso bewirtschaftbare Größe aufzufassen, wie die klassischen betriebswirtschaftlichen Produktionsfaktoren.

Folglich beschäftigt sich Wissensmanagement mit der Gewinnung, Darstellung und Verbreitung von Wissen. Wissensmanagement ist die Summe der Methoden, Maßnahmen und Systeme, die eine Organisation einsetzt, um Wissen zu entwickeln, es transparent zu machen und es zeit-, personen- und ortsunabhängig verfügbar zu machen. Ziel des Wissensmanagements ist es, das Wissen zu vergrößern und die Wissensbasis im Unternehmen optimal anzuwenden.

AUFGABE: WISSENSLANDKARTE BZW. YELLOW PAGE

Es soll abgebildet werden, welches Wissen wo im Unternehmen vorhanden ist.

ARIS-Unterstützung: Mit dem Modelltyp **Wissenslandkarte** lässt sich die organisatorische Verteilung unterschiedlicher Wissenskategorien abbilden. Es kann dargestellt werden, welche Organisationseinheit, Stelle oder welcher Mitarbeiter Know-how in bestimmten Wissenskategorien aufweist, und wie groß diese Kompetenz ist.

AUFGABE: WISSENSKATEGORISIERUNG

Aufgabe ist es, das Intellectual Capital einer Organisation analytisch zu klassifizieren, d.h. die verschiedenen Arten und Gruppen von Wissen darzustellen, um z. B. eine Wissensablagestruktur zu entwerfen.

ARIS-Unterstützung: Mit dem Wissensstrukturdiagramm lässt sich darstellen, wie sich die Wissensbasis einer Organisation in verschiedene Wissenskategorien aufteilt und wie sich diese in weitere Wissenskategorien und dokumentiertes Wissen auffächert. Für dokumentiertes Wissen kann zusätzlich abgebildet werden, auf welchen Informationsträgern dieses Wissen gespeichert ist.

AUFGABE: WISSENSVERARBEITUNG IN GESCHÄFTSPROZESSEN

Aufgabe ist es darzustellen, wo in den Geschäftsprozessen Wissen erzeugt, verändert und benötigt wird, um eine möglichst effiziente Nutzung der Ressource Wissen zu gewährleisten.

ARIS-Unterstützung: Innerhalb der Modelltypen **EPK**, **Vorgangskettendiagramm** und **Funktionszuordnungsdiagramm** können die Objekte **Wissenskategorie** und **Dokumentiertes Wissen** verwendet werden. Der Aufbau und die organisatorische Verteilung des Wissens kann durch das Wissensstrukturdiagramm und die Wissenslandkarte gesondert beschrieben werden.

12.10 Workflow-Management

Ein Workflow kann im weitesten Sinne als Geschäftsprozess interpretiert werden. Der Begriff Workflow steht für arbeitsteilige Prozesse, die zur Abwicklung von Geschäftsvorfällen initiiert werden. Dabei kann es sich sowohl um sehr einfache Geschäftsprozesse als auch um vielschichtige, organisationsübergreifende Prozesse handeln. Schwerpunkt der Betrachtung ist der dynamische Ablauf der Prozesse von Anfang bis Ende. Workflow-Management ist die Summe von Methoden, Maßnahmen und Systemen, die eingesetzt werden, um Workflows zu entwickeln, zu steuern und zu optimieren.

Ein Workflow-Management-System ist eine flexibel gestaltbare, nach einem organisatorischen Regelwerk arbeitende, aktiv einwirkende Software, die einen über mehrere Arbeitsplätze gehenden Vorgang steuert und bestehende technische Basiskomponenten einbindet. Mit Systemen der Vorgangssteuerung ist es möglich, komplexe Aufgabenzusammenhänge, an denen eine Vielzahl von Mitarbeitern und Stellen arbeiten, zu unterstützen.

AUFGABE: PROZESS-CUSTOMIZING VON WORKFLOW-MANAGEMENT-SYSTEMEN

ARIS unterstützt insbesondere die Überführung von allgemeinen Geschäftsprozessmodellen in Workflow-Modelle, mit denen dann verschiedene Workflow-Management-Systeme (halb-)automatisch konfiguriert werden können.

ARIS-Unterstützung: Mit der EPK wird analog zur Prozessmodellierung der Tätigkeitsfluss abgebildet. Hierbei ist auf eine streng methodenkonforme Modellierung zu achten! Zu jeder Funktion muss ein Funktionszuordnungsdiagramm erstellt werden, in dem der Funktion ein Bearbeiter sowie die Input- und Output-Daten zugeordnet werden - soweit dies nicht schon in der EPK abgebildet wurde.

Damit zur Run-Time die zu den Daten gehörigen Applikationen automatisch gestartet werden, muss in einem Anwendungssystemtypdiagramm bzw. Anwendungssystemtypdiagramm eine Zuordnung von Dateien zu Applikationen modelliert werden.

13 Literaturverzeichnis

13.1 Allgemeine Literaturliste

- Brombacher, R.; Bungert, W.: Praxis der Unternehmensmodellierung 1992

  Praxis der Unternehmensmodellierung, Seminar der IDS Prof. Scheer GmbH, Bad Soden/Taunus, 12./13. November 1992.
- Chen, P. P.: Entity-Relationship Model 1976

  The Entity-Relationship model: Toward a Unified View of Data, in: ACM Transactions on Database-Systems, Vol. 1 (1976), No. 1, S. 9 - 36.
- Hoffmann, W.; Kirsch, J.; Scheer, A.-W.: Modellierung mit Ereignisgesteuerten Prozeßketten 1993

  Modellierung mit Ereignisgesteuerten Prozeßketten (Methodenhandbuch, Stand: Dezember 1992), in: Scheer, A.-W. (Hrsg.), Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 101, Saarbrücken Januar 1993.
- Keller, G.; Hechler, H.-J.: Informationsmodell 1991

  Konzeption eines integrierten Informationsmodells für die Kostenrechnung des SAP®-Systems, in: Scheer, A.-W. (Hrsg.): 12. Saarbrücker Arbeitstagung Rechnungswesen und EDV: Kritische Erfolgsfaktoren in Rechnungswesen und Controlling, Heidelberg 1991, S. 67 - 106.
- Scheer, A.-W.: Architektur integrierter Informationssysteme 1992

  Architektur integrierter Informationssysteme - Grundlagen der Unternehmensmodellierung, 2. Aufl., Berlin et al. 1992.
- Scheer, A.-W.: EDV-orientierte Betriebswirtschaftslehre 1990

  EDV-orientierte Betriebswirtschaftslehre - Grundlagen für ein effizientes Informationsmanagement, 4. Aufl., Berlin et al. 1990.
- Scheer, A.-W.: Wirtschaftsinformatik 1994

  Wirtschaftsinformatik - Referenzmodelle für industrielle Geschäftsprozesse, 5. Aufl., Berlin et al. 1994.
- Schlageter, G.; Stucky, W.: Datenbanksysteme 1983

  Datenbanksysteme: Konzepte und Modelle, 2. Aufl. Stuttgart 1983.
- Seubert, M.: SAP®-Datenmodell 1991

  Entwicklungsstand und Konzeption des SAP®-Datenmodells, in: Scheer, A.-W. (Hrsg.): Datenbanken 1991 - Praxis relationaler Datenbanken: Vom Datenmodell zur Implementierung, Fachtagung vom 04.-05.06.1991 in Saarbrücken, Saarbrücken 1991, S. 87 - 109.
- Sinz, E. J.: Entity-Relationship-Modell 1990

Das Entity-Relationship-Modell (ERM) und seine Erweiterungen, in: HMD Theorie und Praxis der Wirtschaftsinformatik, 27 (1990), Heft 152, S. 17 - 29.

- Scheer, A.-W.: ARIS - Vom Geschäftsprozeß zum Anwendungssystem. 3. Aufl. Berlin et al. 1998
- Scheer, A.-W.: ARIS - Modellierungsmethoden, Metamodelle, Anwendungen. 3. Aufl. Berlin et al. 1998
- Scheer, A.-W., Jost, W.: ARIS in der Praxis 2002

  Gestaltung, Implementierung und Optimierung von Geschäftsprozessen, Berlin, Heidelberg, New York 2002.
- Scheer, A.-W., Abolhassan, F., Jost, W., Kirschmer, M.: Business Process Excellence 2002

  ARIS in Practice , Berlin, Heidelberg, New York 2002.

13.2 Themenbezogene Literaturliste

13.2.1 Unified Modeling Language in ARIS

13.2.1.1 UML-Spezifikation

UML-Spezifikation: https://www.uml.org.

13.2.1.2 Anwendung von UML

- Burkhardt, R.: UML Unified Modeling Language, Objektorientierte Modellierung für die Praxis, Bonn 1997.
- Fowler, M.; Scott, K.: UML Distilled - Applying the Standard Object Modeling Language, Reading et al. 1997.
- Oesterreich, B.: Objektorientierte Softwareentwicklung mit der UML, 3. Aufl., München-Wien 1997.

13.2.1.3 UML und Geschäftsprozessmodellierung

- Ambler, S. W.: What's Missing from the UML? Techniques that can help model effective business applications, Object Magazine 7 (1997)8
- Loos, P.; Allweyer, Th.: Process Orientation and Object-Orientation - An Approach for Integrating UML and Event-Driven Process Chains (EPC), Publication of the Institut für Wirtschaftsinformatik, Paper 144, Saarbrücken 1998.

13.2.2 Methoden für das Wissensmanagement

13.2.2.1 Wissensmanagement allgemein

- Probst, G.; Raub, S.; Romhardt, K.: Wissen managen. Wie Unternehmen ihre wertvollste Ressource optimal nutzen. Frankfurt Wiesbaden 1998.
- Bürgel, H. D. (Hrsg.): Wissensmanagement. Schritte zum intelligenten Unternehmen. Berlin et al. 1998.

13.2.2.2 ARIS-Einsatz für das Wissensmanagement

- Allweyer, Th.: 'Modellbasiertes Wissensmanagement' In: IM Information Management & Consulting 13 (1998) 1, S. 37-45.
- Allweyer, Th.: Wissensmanagement mit ARIS-Modellen. In: Scheer, A.-W.: Vom Geschäftsprozess zum Anwendungssystem. 3. Aufl. Berlin et al. 1998, S. 162-168.

13.2.3 Balanced Scorecard-Methode

Kaplan, Robert/Norton, David: The Balanced Scorecard-Measures that drive Performance, Harvard Business Review Januar/Februar 1992.

13.2.4 IT City Planning

- Schulman, Jeff: A New View of Architectures Needed for New Business Drivers, Gartner Briefing Presentations.
- Longépé, Christoph: Le projet d'urbanisation du système d'information, Dunod, Paris, 2001

13.2.5 Business Process Modeling

Business Process Modeling Notation, Working Draft (1.0); BPMI.org; August 25, 2003.

14 Rechtliche Informationen

14.1 Dokumentationsumfang

Die zur Verfügung gestellten Informationen beschreiben die Einstellungen und Funktionalitäten, die zum Zeitpunkt der Veröffentlichung gültig waren. Da Software und Dokumentation verschiedenen Fertigungszyklen unterliegen, kann die Beschreibung von Einstellungen und Funktionalitäten von den tatsächlichen Gegebenheiten abweichen. Informationen über solche Abweichungen finden Sie in den mitgelieferten Release Notes. Bitte lesen und berücksichtigen Sie diese Datei bei Installation, Einrichtung und Verwendung des Produkts.

Wenn Sie das System technisch und/oder fachlich ohne die von Software GmbH angebotenen Service-Leistungen installieren möchten, benötigen Sie umfangreiche Kenntnisse hinsichtlich des zu installierenden Systems, der Zielthematik sowie der Zielsysteme und ihren Abhängigkeiten untereinander. Aufgrund der Vielzahl von Plattformen und sich gegenseitig beeinflussender Hardware- und Softwarekonfigurationen können nur spezifische Installationen beschrieben werden. Es ist nicht möglich, sämtliche Einstellungen und Abhängigkeiten zu dokumentieren.

Beachten Sie bitte gerade bei der Kombination verschiedener Technologien die Hinweise der jeweiligen Hersteller, insbesondere auch aktuelle Verlautbarungen auf deren Internet-Seiten bezüglich Freigaben. Für die Installation und einwandfreie Funktion freigegebener Fremdsysteme können wir keine Gewähr übernehmen und leisten daher keinen Support. Richten Sie sich grundsätzlich nach den Angaben der Installationsanleitungen und Handbücher der jeweiligen Hersteller. Bei Problemen wenden Sie sich bitte an die jeweilige Herstellerfirma.

Falls Sie bei der Installation von Fremdsystemen Hilfe benötigen, wenden Sie sich an Ihre lokale Software GmbH-Vertriebsorganisation. Beachten Sie bitte, dass solche Hersteller- oder kundenspezifischen Anpassungen nicht dem Standard-Softwarepflege- und Wartungsvertrag der Software GmbH unterliegen und nur nach gesonderter Anfrage und Abstimmung erfolgen.

14.2 Support

Wenn Sie Fragen zu spezifischen Installationen haben, die sie nicht selbst durchführen können, wenden Sie sich bitte an Ihre örtliche Software GmbH-Vertriebsorganisation (https://www.softwareag.com/corporate/company/global/offices/default.html). Detaillierte Informationen und Support erhalten Sie über unsere Internetseiten.

Wenn Sie einen gültigen Supportvertrag haben, können Sie **Global Support ARIS** kontaktieren unter: **+800 ARISHELP**. Sollte diese Nummer von Ihrem Telefonanbieter nicht unterstützt werden, sehen Sie bitte in unserem Global Support Contact Directory nach.

Bei Fragen zur Produktdokumentation können Sie auch eine E-Mail an documentation@softwareag.com (mailto:documentation@softwareag.com) senden.

ARIS COMMUNITY

- Produkte, Updates und Fixes herunterladen
- Hier finden Sie Informationen, Fachartikel, Problemlösungen, Videos und Kommunikation mit anderen ARIS-Benutzern

Wenn Sie noch kein Konto haben, registrieren Sie sich bei der ARIS Community**.**

PRODUKTSCHULUNG

Hilfreiches Schulungsmaterial zu Ihrem Produkt finden Sie in unserem Learning Portal.

TECH COMMUNITY

Auf unserer Tech Community-Website können Sie mit Experten der Software GmbH zusammenarbeiten. Von hier aus können Sie z. B. Folgendes tun:

- Unsere riesige Wissensdatenbank durchsuchen.
- In unseren Diskussionsforen Fragen stellen und Antworten finden.
- Aktuelle News und Ankündigungen der Software GmbH erhalten.
- Unsere Communitys kennenlernen.
- In unseren öffentlichen Repositorys GitHub und Docker können Sie zusätzliche Software GmbH-Ressourcen entdecken.

PRODUKT-SUPPORT

Lizenzierte Kunden erhalten Support für Software GmbH-Produkte über unser Empower Portal (https://empower.softwareag.com/). Viele Services in diesem Portal setzen ein Konto voraus. Wenn Sie noch kein Konto haben, können Sie eines anfordern. Ein Konto bietet Ihnen z. B. folgende Möglichkeiten:

- Anfragen zu Produktmerkmalen hinzufügen
- Das Knowledge Center nach technischen Informationen und Tipps durchsuchen
- Frühwarnungen und kritische Alarmmeldungen erhalten
- Support-Vorfälle eröffnen und aktualisieren