

Matthes Elstermann
Albert Fleischmann
Christian Stary
Ronny Hengartner

Modellieren und Programmieren unter Verwendung der Satzstruktur natürlicher Sprachen

Eine Anleitung

OPEN ACCESS

 Springer Vieweg

Modellieren und Programmieren unter Verwendung der Satzstruktur natürlicher Sprachen

Matthes Elstermann · Albert Fleischmann ·
Christian Stary · Ronny Hengartner

Modellieren und Programmieren unter Verwendung der Satzstruktur natürlicher Sprachen

Eine Anleitung

Matthes Elstermann
Mannheim, Deutschland

Albert Fleischmann
Pfaffenhofen, Deutschland

Christian Stary
Institut für Wirtschaftsinformatik
Johannes Kepler Universität Linz
Linz, Österreich

Ronny Hengartner
Flims Dorf, Schweiz



ISBN 978-3-658-47103-3 ISBN 978-3-658-47104-0 (eBook)

<https://doi.org/10.1007/978-3-658-47104-0>

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <https://portal.dnb.de> abrufbar.

JKU Linz, Institut für Wirtschaftsinformatik-Communications Engineering

Dieses Werk wurde gefördert durch JKU Linz, Institut für Wirtschaftsinformatik-Communications Engineering.

Publiziert mit Unterstützung des Publikationsfonds der Johannes Kepler Universität Linz und des Landes Oberösterreich.

© Der/die Herausgeber bzw. der/die Autor(en) 2025. Dieses Buch ist eine Open-Access-Publikation.

Open Access Dieses Buch wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor*in(nen) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Buch enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des/der betreffenden Rechteinhaber*in einzuholen.

Die Wiedergabe von allgemein beschreibenden Bezeichnungen, Marken, Unternehmensnamen etc. in diesem Werk bedeutet nicht, dass diese frei durch jede Person benutzt werden dürfen. Die Berechtigung zur Benutzung unterliegt, auch ohne gesonderten Hinweis hierzu, den Regeln des Markenrechts. Die Rechte des/der jeweiligen Zeicheninhaber*in sind zu beachten.

Der Verlag, die Autor*innen und die Herausgeber*innen gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag noch die Autor*innen oder die Herausgeber*innen übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Planung/Lektorat: Petra Steinmueller

Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer Fachmedien Wiesbaden GmbH und ist ein Teil von Springer Nature.

Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Wenn Sie dieses Produkt entsorgen, geben Sie das Papier bitte zum Recycling.

Inhaltsverzeichnis

1	Einleitung	1
2	Anwendungsbereiche des subjektorientierten Denkmodells	5
3	Grundkonzepte des subjektorientierten Modellierens	7
3.1	Grundstruktur von Sätzen in natürlichen Sprachen	7
3.2	Grundidee der Subjektorientierung	9
3.3	Kommunikation	9
3.3.1	Kommunikationsstruktur	9
3.3.2	Austausch von Nachrichten	10
3.4	Subjekte Intern	13
3.4.1	Subjektverhalten	13
3.4.2	Subjekt und Objekte	16
4	Strukturierung komplexer subjektorientierter Modelle	19
4.1	Strukturierung der Kommunikationsstruktur	19
4.1.1	Externe Subjekte	19
4.1.2	Netzwerke von Teilsystemen	21
4.1.3	Multiprozesse und Multisubjekte	24
4.2	Strukturierung des Subjektverhaltens	26
4.2.1	Makros/Unterverhalten	26
4.2.2	Wahlfreiheit	27
4.2.3	Ausnahmebehandlung	30
4.2.4	Verhaltenserweiterung	30
4.2.5	Verhaltensschalen	32
5	Konventionen für PASS	35
5.1	Benennung von Modellelementen	35
5.2	Layouting	36
5.2.1	Layouting SBDs	36
5.2.2	Layout-Prinzipien für SIDs	39

6	Subjektorientierte Prozesse realisieren	43
6.1	Modell, Instanz, Kontext und Implementierung	43
6.2	Realisierung von Subjekten	46
6.2.1	Realisierung des Input Pools	46
6.2.2	Realisierung des Subjektverhaltens	47
6.2.3	Realisierung von Objekten	48
6.3	Realisierung der Kommunikation	49
6.4	Umsetzungsplattformen	50
7	Management der Erstellung subjektorientierter Systeme	53
7.1	Projektmanagement Modellbildung	53
7.2	Projektmanagement Modellimplementierung	54
8	Können aus Geschichten Prozesse abgeleitet werden?	57
8.1	Von der natürlichsprachlichen Beschreibung zum ausführbaren Programm	57
8.2	Von Texten zu kontext-sensitivem Prozesswissen	58
8.2.1	Schritt 1: Identifikation in sich schlüssiger Geschichtenteile ...	59
8.2.2	Schritt 2: Kontextualisierung der Textteile	60
8.2.3	Schritt 3: Ableitung von Prozessmodellen	61
9	Soziotechnische Umsetzungsempfehlungen	65
9.1	Organisation der Implementierung	65
9.2	Detaillierung für die Implementierung	66
9.3	Schulung und Testen	68
9.4	Betrieb und Optimierung der Prozesse	69
10	Schlußgedanken	71
10.1	Subjektorientiertes Denkmodell	71
10.2	Weiterführende Aspekte der Subjektorientierung	72
	Literatur	75
	Stichwortverzeichnis	77

Abbildungsverzeichnis

Abb. 3.1	Subjekt InteraktionsDiagram (SID) Pannenservice	11
Abb. 3.2	Input pool Spezifikation des Subjekts <i>Support Requestor</i>	12
Abb. 3.3	Subjektverhaltensdiagramm (SVD) oder Subject Behavior Diagram (SBD) Support Requestor.	14
Abb. 3.4	Subject Behaviour Diagram (SBD) Call Center	15
Abb. 3.5	Datenobjekt Pannemeldung.	16
Abb. 4.1	Prozesse über externe (Interface) Subjekte verknüpfen	20
Abb. 4.2	Prozesse über externe (Interface-)Subjekte verknüpfen	20
Abb. 4.3	Komplexes Prozesssystem.	21
Abb. 4.4	Teilprozessmodell für die unmittelbare Pannenhilfe.	22
Abb. 4.5	Untermmodell zu Abb. 4.4 mit Detaillierung für die Teilprozesse im Incident Management	23
Abb. 4.6	Subjektebene eines hierarchischen Prozessnetzwerks	23
Abb. 4.7	Skizzierte Übersicht der Prozessmodellhierarchie	24
Abb. 4.8	Subject Interaction Diagram (SID) Call Center mit Towing Service als Multisubjekt	25
Abb. 4.9	SBD des Call Centers in Interaktion mit dem Multi-Subjekt Towing Service aus Abb. 4.8	26
Abb. 4.10	Makro Behaviour Diagram (MBD) Handle Support Request.	27
Abb. 4.11	Makro Behaviour Diagram (MBD) Handling Towing Car - Mit Kommunikation zu Multisubjekt	28
Abb. 4.12	Subject Behaviour Diagram (SBD) des Call Center Subjektes mit Makro aufrufen	28
Abb. 4.13	Alternativklausel (Choice Segment) zur überlappten Ausführung von zwei Makros	29
Abb. 4.14	Pannenhilfeprozess mit Nachrichten für die Möglichkeit die Pannenhilfe abbrechen.	31
Abb. 4.15	Guard Behavior für das Call Center Subjekt @@par (Base Behavior links im Hintergrund)	32

Abb. 4.16	Erweiterung (Extension) des Guard Behavior aus Abb. 4.15	33
Abb. 4.17	Skizze des Schichtkonzepts der Verhalten aus Abb. 4.12, 4.15 und 4.16 übereinander	33
Abb. 5.1	Explizite Benennung von Subjekten als aktive Einheiten und Nachrichten als Objekte	36
Abb. 5.2	Gruppierung und Layout im SBD	37
Abb. 5.3	Gemeinsames Führen von Transitionen in 5.2	38
Abb. 5.4	Beispiel für ein verbesserungswürdiges SBD ohne Beachtung von Konventionen – gleicher Inhalt wie Abb. 5.2	39
Abb. 5.5	Ursprüngliches SID eines realen Anwendungsfalls	40
Abb. 5.6	Überarbeitetes SID – gleicher Inhalt wie Abb. 5.5	40
Abb. 6.1	Konzept von Aufgabenträger (Subject Carrier) und Subjektinstanzen jeweils mit Input Pool	45
Abb. 6.2	Input Pool zur Aufnahme unterschiedlich realisierter Nachrichten	47
Abb. 8.1	Modellbildung: Von der natürlichen Sprache zum Programm	58
Abb. 8.2	Rhetorical Structure Theory (RST) Tree für erste Szene des Verpackungsprozesses auf Basis einer Storytelling-Ontologie	60
Abb. 8.3	Oberer Teil des RST Tree für die Textbeschreibung des Verpackungsprozesses von Industrie 4.0.	61
Abb. 8.4	Unterer Teil des RSTree für die Textbeschreibung des Verpackungsprozesses von Industrie 4.0.	62
Abb. 8.5	Subjektspezifikationen (SID) der beteiligten Komponenten im Verpackungsanwendungsfall mit Interaktionssequenzen unter Verwendung der Compunity Suite (compunity.eu)	62
Abb. 8.6	Verhaltensspezifikation eines Robotersystems, das beim Verpacken einen Sensor auswählt, unter Verwendung der Compunity Suite	63
Abb. 10.1	Mögliche Entwicklungsströmungen für die Subjektorientierung	72



Dies ist eine Anleitung zur Anwendung des subjektorientierten Denk- und Implementierungsmodells zur Spezifikation und Entwicklung von soziotechnischen Systemen. Systeme sind ein strukturiertes Ganzes aus verschiedenen interagierenden oder voneinander abhängigen Komponenten. Die Komponenten wirken zusammen um ein Ziel zu erreichen oder einen gewünschten Zweck zu erfüllen. Dazu wirken die beteiligten Komponenten zusammen.

Die hier betrachteten Systeme umfassen technische (Hardware und Software) als auch soziale Komponenten und ihre Interaktionen. Solche soziotechnischen Systeme betonen die Wechselwirkungen zwischen Menschen und Technologie, um Aufgaben zu erfüllen und Ziele zu erreichen. Beispiele für soziotechnische Systeme sind Geschäfts- und Produktionsprozesse, bei denen Mensch und Technik eng miteinander verknüpft sind.

Bevor solche soziotechnische Systeme realisiert werden, werden die Anforderungen an diese Systeme definiert, deren wesentliche Komponenten identifiziert und festgelegt wie diese zusammenwirken. Eine Komponente kann eine andere benutzen in dem sie Funktionen der benutzten Komponente aufruft und dann das jeweilige Ergebnis zurückerhält. Die aufrufende Komponente wartet so lange bis die aufgerufene Funktion abgeschlossen wurde und das Ergebnis an die aufrufende Komponente zurückgegeben wurde. Diese Beziehung wird als Benutzrelation bezeichnet. Durch Benutzrelationen werden Komponenten eng miteinander verknüpft. Ein Beispiel aus der IT zur Beschreibung solcher Systeme, die durch Benutzrelationen gekennzeichnet sind, sind Objektorientierte Methoden wie z. B. UML.

Eine zweite Möglichkeit der Interaktion ist die Kommunikation zwischen Komponenten d. h. eine Komponente sendet eine Nachricht an eine andere Komponente. Mit einer Nachricht kann die Ausführung einer Aufgabenfolge angestoßen werden. Die angestoßene Komponente kann, muss aber nicht, Nachrichten an die Komponente senden, die die Ausführungsfolge mit dem Senden einer Nachricht angestoßen hat. Die anstoßende Komponente

kann nach dem Senden einer Nachricht ihre Arbeit fortsetzen und bei Bedarf irgendwann auf eine Ergebnismnachricht warten. Komponenten die Nachrichten austauschen sind loser gekoppelt als Systeme deren Komponenten nur durch Benutzrelationen interagieren können. Dies kann insbesondere bei verteilten Systemen von Vorteil sein, und zwar, wenn Autonomie ein wesentliches Gestaltungsprinzip darstellt. Ein typisches Beispiel sind cyber-physische Systeme, wie wir sie im Bereich der digitalisierten Mobilität und Gesundheit vorfinden.

Eine Möglichkeit zu beschreiben in denen sowohl die Benutzrelation als auch die Kommunikationsrelation verwendet werden kann ist der hier vorgestellte subjektorientierte Ansatz. Dieser Ansatz basiert auf der allgemeinen Struktur von natürlich sprachlichen Sätzen mit Subjekt, Prädikat und Objekt. Diese subjektorientierte Modellierungssprache ist leicht erlernbar, da sie sich an die Struktur der natürlichen Sprachen anlehnt und zum anderen ist sie formal, so dass für die Softwareaspekte eines Systems, der Programmcode für die Realisierung direkt aus der Spezifikation abgeleitet werden kann (Low Code). Es wird also zum einem der Anwender und zum anderem der Entwickler unterstützt.

In dieser Abhandlung werden die entsprechenden Modellierungselemente und ihre Verwendung (Kap. 3, 4) sowie die Techniken mit denen diese Modelle implementiert werden können (Kap. 6) vorgestellt. Außerdem wird beschrieben wie eine Managementstruktur für die Erstellung komplexer subjektorientierter Systeme, angelehnt an die Systemstruktur, aussehen kann (Kap. 7). Den Abschluss bilden Überlegungen zur Generierung von Modellen aus natürlichsprachigen Texten (Kap. 8) sowie zur Implementierungspraxis (Kap. 9).

Diese Anleitung enthält keine Referenzen zu den beschriebenen Methoden und Vorgehensweisen. Die entsprechenden Begründungen und Hintergründe können der umfangreichen Literatur entnommen werden. Eine entsprechende Sammlung findet sich auf der Internetseite des Instituts für innovatives Prozessmanagement (siehe Link <https://i2pm.net>). Eine vollständige Beschreibung der Subjektorientierung enthält das Buch ‚Subjektorientiertes Prozessmanagement‘¹. Hinweise auf weitere Literatur die auf spezifische Aspekte eingeht findet man unter <https://i2pm.net/category/publications/>.

Zur Darstellung der Beispiele wird MS-Visio mit dem entsprechenden Satz an Formen, die von Matthes Elstermann entwickelt wurden verwendet (siehe <https://subjective-jimdofree.com/visio-modelling/>). Ansonsten sind für die Beschreibung von subjektorientierten Modellen durch die wenigen Symbole Bleistift und Papier ausreichend. Es muss auch nicht zwingend die in diesem Buch verwendete Symbolik verwendet werden. Subjektorientierte Modelle können z. B. auch durch eine Teilmenge der Sprache BPMN spezifiziert werden.

Diese Anleitung fokussiert ausschließlich auf das praktische Tun und enthält keine Begründungen für irgend welche Modellierungsaspekte der Subjektorientierung. Es geht

¹ Albert Fleischmann, Werner Schmidt, Christian Stary, Stefan Obermeier, Egon Börger, Subjektorientiertes Prozessmanagement, Hanser Verlag, 2011 bzw. in Englisch ‚Subject-oriented Business Process Management‘ by Albert Fleischmann, Werner Schmidt, Christian Stary, Stefan Obermeier, Egon Börger, Subject-oriented Business Process Management, Springer Verlag, 2012 oder als kostenloses Download unter <https://link.springer.com/book/10.1007/978-3-642-32392-8>.

darum wie ein System schnell und mit möglichst geringem Aufwand entwickelt werden kann. Ein vollständiges Beispiel wie ein subjektorientiertes Modell erstellt und automatisch in einen IT-basierten Workflow umgesetzt wird findet sich in dem Buch „S-BPM illustrated“².

Dieses Buch ist gedacht für diejenigen die an der Modellierung und Realisierung von soziotechnischen Systemen interessiert sind insbesondere Geschäftsprozessberater und Implementierer von Geschäftsprozessen. Spezielle Vorkenntnisse sind nicht schädlich aber auch nicht notwendig.

² Albert Fleischmann, Stefan Raß, Robert Singer, S-BPM Illustrated: A Storybook about Business Process Modeling and Execution, Springer Berlin Heidelberg, 2013.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.





Anwendungsbereiche des subjektorientierten Denkmodells

2

Subjektorientierte Konzepte sind dann sinnvoll, wenn Systeme betrachtet werden, in denen mehrere Handelnde, also aktive Entitäten, vorkommen die miteinander kommunizieren und Objekte benutzen. Wobei es zunächst egal ist, ob diese Handelnden Menschen, Maschinen, Computerprogramme oder eine Kombination aus all dem sind. Diese Realisierungsaspekte werden erst später bei der Implementierung der Modelle betrachtet (siehe Kap. 6). Die Subjektorientierung ist eine geeignete Methode für die Entwicklung von soziotechnischen Systemen in denen mehrere Menschen, Softwarekomponenten, Aktoren und Sensoren zusammenwirken, um eine vorgegebene Leistung zu erbringen. Diese Aspekte eines Systems werden durch eine entsprechende Software miteinander verknüpft. Das Subjektorientierte Denkmodell erlaubt es bereits existierende Teilsysteme oder Komponenten miteinander zu verknüpfen bzw. durch neue Systemkomponenten zu ergänzen.

Die Subjektorientierung bildet somit ein Rahmenwerk für den Entwurf und die Implementierung komplexer Systeme wie z. B. organisationsübergreifender Geschäftsprozesse, Produktionsprozesse, Maschinensteuerungen. In ein subjektorientiertes Modell können auch mehrere KI-Anwendungen eingebunden werden, die Beiträge für die Gesamtaufgabe eines komplexen Systems liefern, wie z. B. das automatisierte Fahren.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.





Grundkonzepte des subjektorientierten Modellierens

3

Dieses Kapitel beschreibt die Grundkonzepte der Subjektorientierung. Mit wenigen Konstrukten können prinzipiell alle Systeme beschrieben werden. Die zusätzlichen Konstrukte die in Kap. 4 vorgestellt werden, haben lediglich den Zweck die Beschreibung komplexer Systeme zu erleichtern und zu strukturieren.

3.1 Grundstruktur von Sätzen in natürlichen Sprachen

Sachverhalte werden in der Regel in natürlicher Sprache beschrieben. Beschreibungen von Sachverhalten erfolgen durch Sätze. Sätze in allen Sprachen bestehen aus den Komponenten Subjekt, Prädikat und Objekt, eine Aussage die sich durch eine Anfrage bei ChatGPT am 21. Mai 2024 durch die Frage

„Was ist die Grundstruktur von Sätzen in natürlichen Sprachen?“ bestätigen läßt:

Antwort von ChatGPT (kursiv):

Die Grundstruktur von Sätzen in natürlichen Sprachen variiert je nach Sprache, aber sie basiert typischerweise auf einigen grundlegenden syntaktischen Mustern. Ein gängiges Modell, um die Grundstruktur von Sätzen zu beschreiben, ist die Einteilung in Satzglieder wie Subjekt, Prädikat und Objekt. Hier sind die wesentlichen Bestandteile und ihre typischen Anordnungen:

- *Subjekt (S): Das Subjekt ist der Satzteil, der beschreibt, wer oder was die Handlung ausführt. Es ist normalerweise ein Nomen oder ein Pronomen.*

- *Prädikat (P):* Das Prädikat ist der Satzteil, der die Handlung oder den Zustand beschreibt. Es besteht aus einem Verb und eventuell weiteren Elementen wie Objekten, Adverbialen, etc.
- *Objekt (O):* Das Objekt ist der Satzteil, der die Handlung empfängt. Es gibt verschiedene Arten von Objekten, wie direktes Objekt (Akkusativobjekt) und indirektes Objekt (Dativobjekt).

Je nach Sprache können die Anordnungen dieser Satzteile unterschiedlich sein. Die häufigsten Wortstellungsmodelle sind:

SVO (Subjekt-Verb-Objekt): Diese Struktur ist in Sprachen wie Englisch und Spanisch üblich.

Beispiel (Englisch): „The cat (S) eats (V) the mouse (O).“

Beispiel (Spanisch): „Él gato (S) come (V) el ratón (O).“

SOV (Subjekt-Objekt-Verb): Diese Struktur findet sich in Sprachen wie Deutsch (in Nebensätzen) und Japanisch.

Beispiel (Deutsch, Nebensatz): „dass die Katze (S) die Maus (O) frisst (V).“

VSO (Verb-Subjekt-Objekt): Diese Struktur ist in Sprachen wie Arabisch und Walisisch zu finden.

Beispiel (Walisisch): „Bwyta (V) mae (S)’r gath (O).“

Es gibt auch andere, weniger verbreitete Wortstellungsmodelle wie VOS, OVS und OSV.

Zusätzlich zur Wortstellung spielen auch weitere syntaktische und morphologische Eigenschaften eine Rolle, wie:

Kasusmarkierung: In Sprachen mit ausgeprägtem Kasussystem (z. B. Latein, Deutsch) kann die Funktion eines Wortes im Satz durch Endungen oder Präpositionen angezeigt werden.

Kongruenz: Die Übereinstimmung von Subjekt und Prädikat in Person und Zahl, wie im Deutschen oder Spanischen.

Wortstellung im Satz: Besonders in Sprachen mit flexiblerer Wortstellung kann die Position von Wörtern durch Betonung und Informationsstruktur beeinflusst werden.

Insgesamt bietet die Satzstruktur einen Rahmen für die Anordnung von Wörtern und die Übermittlung von Bedeutungen, wobei jede Sprache ihre eigenen Regeln und Besonderheiten aufweist.

Natürliche Sprachen werden auch verwendet um technische Systeme die realisiert werden sollen, zu spezifizieren. Durch eine stärkere Formalisierung der erlaubten Satzstrukturen wird die Realisierung durch Menschen oder verschiedene Technologien erleichtert. Somit wird durch die Verwendung der Standardstruktur von natürlichen Sprachen sowohl die

Beschreibung der Anforderungen an ein zu realisierendes System erleichtert. Durch einen gewissen Grad an Formalisierung wird die Erstellung einer IT-basierten Lösung vorbereitet.

3.2 Grundidee der Subjektorientierung

Die Grundidee des subjektorientierten Denkmodells ist, dass aktive Entitäten, als Subjekte bezeichnet, passive Objekte benutzen und miteinander durch den Austausch von Nachrichten interagieren. Subjekte sind aktiv und es ist noch nicht festgelegt, ob sie bei der Realisierung von Menschen, Maschinen, Computerprogramme oder Kombinationen aus all dem repräsentiert werden. Subjekte sind aktiv, in dem sie mit anderen Subjekten kommunizieren oder interne Aktionen auf Objekten ausführen. Analog den Subjekten sind Nachrichten ebenfalls abstrakt. Sie haben einen Namen und eine Nutzlast die sie vom Sendesubjekt zum Empfangssubjekt transportieren. Somit kann eine Nachricht ein Brief sein oder auch ein Gegenstand der transportiert wird. Diese Aspekte der Realisierung werden zum Zeitpunkt der Modellbildung noch nicht betrachtet sondern werden erst in der Implementierung festgelegt.

Die Objekte auf denen ein Subjekt Operationen ausführt, sind Teil eines Subjekts und werden zunächst ebenfalls abstrakt definiert. Objekte können Daten, Papierformulare oder auch Werkstücke sein die bearbeitet werden. Wie ein Objekt beschaffen ist und durch welche Mittel eine Aktion ausgeführt wird, wird also während der Modellbildung noch nicht betrachtet sondern erst im Rahmen der Implementierung festgelegt. Ein System wird beschrieben durch die beteiligten Subjekte und die Nachrichten, die sie austauschen (Kommunikationsstruktur). Ein zweiter Teil der Beschreibung definiert in welcher Reihenfolge die Subjekte Nachrichten senden und empfangen und welche internen Aktionen auf ihren Objekten (Subjektverhalten) ausführen. In den folgenden Abschnitten werden die beiden Aspekte Kommunikationsstruktur und Subjektverhalten näher beschrieben.

3.3 Kommunikation

In der subjektorientierten Betrachtung tauschen Subjekte Nachrichten aus und benutzen Objekte.

3.3.1 Kommunikationsstruktur

In diesem Modellierungsschritt ist es bedeutungslos, wie eine Nachricht oder deren Versand implementiert wird. Eine Nachricht kann als Nutzlast Informationen, einen physikalischen Gegenstand oder Kombinationen daraus transportieren und wird irgendwie vom Sender zum

Empfänger transportiert. Die entsprechende Realisierung wird zum Implementierungszeitpunkt definiert.

Die Namen von Subjekten und Nachrichten geben einen ersten Eindruck von den Aufgaben eines Subjekts, bzw. von der Bedeutung einer Nachricht. Die Namen der Objekte die als Nutzlast von Nachrichten transportiert werden sollen ebenfalls einen Hinweis auf deren Verwendung geben.

Ein subjektorientiertes Modell ist also zunächst eine abstrakte Beschreibung eines Sachverhalts. Subjekte und die ausgetauschten Nachrichten werden bei der Beschreibung existierender Systeme unabhängig von ihrer momentanen Realisierung, bzw. bei neu zu entwickelnden Systemen oder Systemteilen unabhängig von der späteren Realisierung beschrieben.

Die Kommunikationsstruktur mit den Subjekten und den zwischen ihnen ausgetauschten Nachrichten wird als Modell grafisch in einem sogenannten Subjekt Interaktions Diagramm (SID) dargestellt. Das Abb. 3.1 zeigt die Kommunikationsstruktur eines Systems zur Bearbeitung von Fahrzeugpannen. Dieses Beispiel ist ein stark vereinfachter Auszug aus einem realen Prozess.

Der Start des Systems erfolgt durch das Subjekt *Support Requestor*. Das Startsubjekt wird durch ein Dreieck in der rechten oberen Ecke gekennzeichnet. Das Subjekt *Support Requestor* sendet die Nachricht *Support Request* zum Subjekt *Call Center*. Dieses Subjekt kommuniziert mit anderen Subjekten um die entsprechende Unterstützung an das Subjekt *Support Requestor* zu liefern. Das Subjekt *Support Requestor* erhält entweder die Nachricht *Request rejected*, wenn die Unterstützung ungerechtfertigt angefordert wurde (kein Servicevertrag), oder die Nachricht *Announcement towing truck* wenn ein Abschleppwagen losgeschickt wurde. Die Nachricht *Towing Truck Confirmation* vom Subjekt *Towing Service* zum Subjekt *Support Requestor* beschreibt das Kommen des Abschleppwagens.

3.3.2 Austausch von Nachrichten

Jedes Subjekt verfügt über einen sogenannten Input Pool. Ein Subjekt das eine Nachricht sendet legt diese Nachricht im Input Pool des Empfängers ab. Damit gilt eine Nachricht als gesendet. Kommt das empfangende Subjekt in einen Zustand in dem es eine bestimmte Nachricht benötigt, wird überprüft ob sich diese Nachricht im Input Pool befindet, und zur Weiterverarbeitung entnommen (siehe Abschn. 3.4).

Das Ablegen einer Nachricht im Input Pool kann an Bedingungen des Input Pool geknüpft sein, die auch Auswirkungen auf das sendende Subjekt haben.

- Ein Input Pool hat eine bestimmte Größe. Diese gibt an wie viele Nachrichten maximal in dem Input Pool enthalten sein dürfen. Ist diese Kapazität allgemein oder für bestimmte Nachrichten erschöpft, dann können folgende Auswirkungen auf das sendende Subjekt definiert werden:

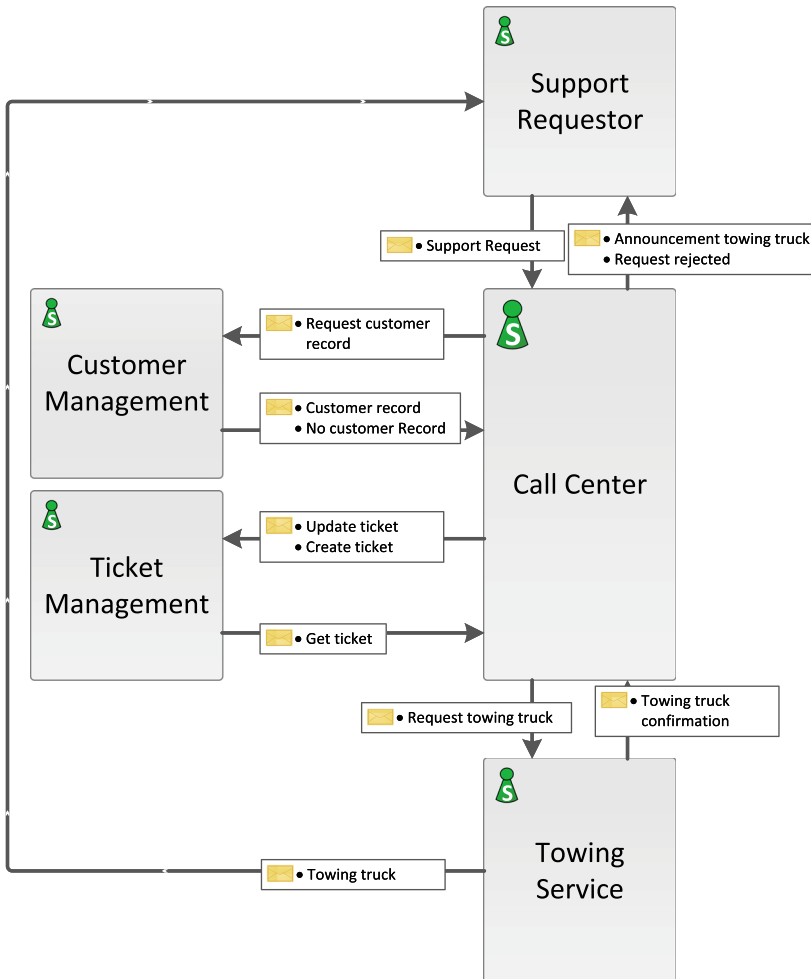


Abb. 3.1 Subjekt InteraktionsDiagram (SID) Pannenservice

- Das sendende Subjekt wird solange blockiert bis das Empfangssubjekt eine Nachricht aus dem Input Pool zur Weiterverarbeitung entnimmt und damit ein Platz frei wird zur Ablage der neu eingetroffenen Nachricht.
- Durch die neu Nachricht wird die älteste oder jüngste Nachricht im Input Pool überschrieben und das Sendesubjekt wird nicht blockiert.
- Bei einer Input-Pool-Größe von Null müssen die Nachrichten direkt übergeben werden, d. h. das sendende Subjekt wird solange blockiert bis das empfangende Subjekt die entsprechende Nachricht erwartet (siehe Abschn. 3.4) und abnimmt.

- Neben der maximalen Größe ist auf dem Input Pool noch eine Struktur definiert. Dazu gibt es folgende Varianten:
 - Es ist festgelegt welche maximale Anzahl von Nachrichten eines bestimmten Namens von einem bestimmten Subjekt in dem Input Pool enthalten sein dürfen. Ist diese maximale Größe erreicht und trifft eine weitere solche Nachricht ein, kann ein Verhalten analog, wie beim Erreichen der Maximalgröße festgelegt werden.
 - Ist für die maximale Anzahl Null angegeben muss eine eintreffende Nachricht direkt übergeben werden, wie bei der Input-Pool-Größe Null.
 - Es wird die maximale Anzahl von erlaubten Nachrichten im Input Pool definiert unabhängig von welchem Subjekt sie gesendet wurde. Beim Erreichen dieser Maximalzahl kann ein Verhalten, wie beim Erreichen der Maximalgröße festgelegt werden. Bei der maximalen Anzahl Null müssen diese Nachrichten direkt übergeben werden.
 - Es wird eine maximale Anzahl von Nachrichten von einem Subjekt festgelegt unabhängig vom Nachrichtennamen. Beim Erreichen dieser Maximalzahl kann ein Verhalten, wie beim Erreichen der Maximalgröße festgelegt werden. Bei der maximalen Anzahl Null müssen Nachrichten von diesen Subjekten direkt übergeben werden.

InputPoolConstraintEditor

Note: Non-Defined Responses are assumed to happen instantaneous (0 days)

Correspondence Subject:

Subjects

- Call Center
- Towing service

Incoming Messages

- Towing truck

Constraint Limit: Handling Strategy: Blocking ▾

Existing Input Pool Constraints				
Type	Subject	Message	Constraint Value	Handling Strategy
Sender/Message	Call Center	Announcement towing truck	1	Blocking
Sender/Message	Call Center	Request rejected	1	Blocking
Sender/Message	Towing service	Towing truck	0	Blocking

Constraint Limit: Handling Strategy: Blocking ▾

Abb. 3.2 Input pool Spezifikation des Subjekts *Support Requestor*

Das Abb. 3.2 zeigt wie mit dem verwendeten Werkzeug der Input Pool des Subjekts *Support Requestor* spezifiziert wird. Der Input Pool des Subjekts *Support Requestor* hat maximal drei Plätze. Die Nachrichten *Announcement towing truck* und *Request rejected* dürfen nur einmal im Input Pool liegen. Die Nachricht *Towing truck confirmation* vom Towing Service wird direkt an den Empfänger übergeben. Der Abschleppwagen wird direkt erwartet.

3.4 Subjekte Intern

In einem Subjekt sind das Verhalten und die benötigten Objekte gekapselt.

3.4.1 Subjektverhalten

Im Subjektverhalten wird beschrieben, in welcher Reihenfolge Nachrichten gesendet oder empfangen werden, bzw. welche Operationen wann auf welchen Objekten ausgeführt werden. Abb. 3.3 zeigt das Verhaltensmodell des Subjekts *Support Requestor* als sogenanntes Subjektverhaltensdiagramm (SVD).

Das Subjekt *Support request* führt zunächst die interne Operation *Create Support Request* aus. Dies bedeutet dass die Informationen für die Anforderung der Unterstützung in einem entsprechenden Datenobjekt (siehe Abschn. 3.4.2) bereitgestellt werden. Wie diese Informationen in dem Objekt bereitgestellt werden z. B. als Notizen auf Papier wird in der Implementierungsphase (siehe Kap. 6) festgelegt. Im logischen Modell wird nur festgelegt dass dies getan werden muss. Sind die für die benötigte Unterstützung benötigten Daten zusammengestellt, werden diese als Nutzlast mit der Nachricht *Support Request* an das Subjekt *Call Center* gesendet. Die Realisierung dieses Sendevorgangs wird in der Implementierungsphase festgelegt z. B. per App oder Telefon.

Danach wartet das Subjekt *Support Requestor* auf eine Antwort vom Subjekt *Call Center*. Vom Subjekt *Support Requestor* kann entweder die Nachricht *Request rejected* oder die Nachricht *Announcement towing truck* eintreffen. Trifft die Nachricht *Request rejected* ein, so wird in den Zustand *Ende* übergegangen und das Verhalten beendet (Punkt in der rechten oberen Ecke des Zustandssymbol). Die Nachricht *Request rejected* können Informationen als Nutzlast enthalten die den Grund für die Zurückweisung beschreiben.

Wird die Nachricht *Announcement towing truck* erhalten, so wird auf die Nachricht *Towing truck* vom Subjekt *Towing service* gewartet. Der Empfang dieser Nachricht wird durch das Eintreffen des Abschleppwagens implementiert. Die interne Funktion *Do towing to repair shop* bedeutet, dass das kaputte Fahrzeug zur Werkstatt geschleppt wird. Nach dem Übergang *Repairshop arrived* wird eine Endzustand erreicht. Das Subjekt hat seinen Arbeit erledigt.

Das Abb. 3.4 zeigt das Verhalten des Subjekts *Call Center* als Gegenüber des Subjekts *Support Requestor*. Im Startzustand wird die Nachricht *Support Request* erwartet. Trifft diese ein, d. h. ist diese im Input Pool enthalten, wird sie entnommen und die mitgesendeten Daten

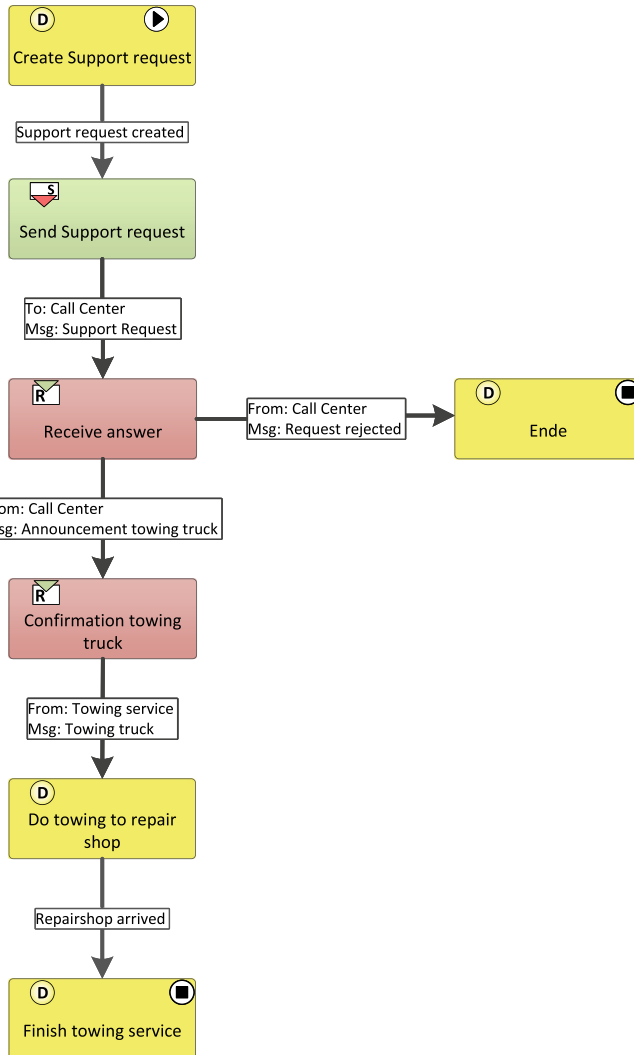
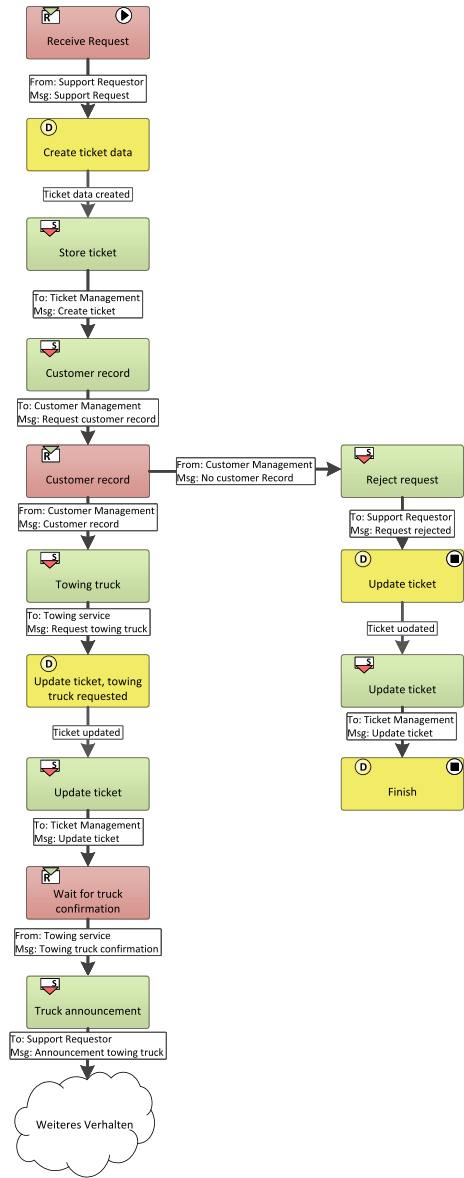


Abb. 3.3 Subjektverhaltensdiagramm (SVD) oder Subject Behavior Diagram (SBD) Support Requestor

(Nutzlast) intern abgespeichert. Diese Daten werden zur Erstellung eines Servicetickets durch die interne Operation *Create Ticket data* verwendet. Danach werden diese Ticketdaten als Nutzlast in der Nachricht *Create ticket* an das Subjekt *Ticket Management* gesendet. Darauf folgend werden mit der Nachricht *Request customer record* an das Subjekt *Customer Management* die Kundendaten angefordert. Kommt vom Subjekt *Customer Management* die Nachricht *No customer Record* zurück so wird an das Subjekt *Support Requestor* die

Abb. 3.4 Subject Behavior Diagram (SBD) Call Center



Nachricht *Support Rejected* gesendet, da der Anforderer des Service keine Kunde ist. Die Ablehnung der Anforderung wird im Ticket vermerkt d.h. das Ticket wird aktualisiert und an das Ticket Management gesendet.

Ist der Anforderer ein Kunde wird der Einsatz eines Abschleppwagens organisiert.

Die beiden Verhaltensdiagramme sind nur Ausschnitte aus dem gesamten Verhalten der Subjekte *Support Requestor* bzw. *Call Center*. Aus Platzgründen wird nicht das gesamte Verhalten dargestellt. Um das komplexe Verhalten zu strukturieren gibt es verschiedene Beschreibungskonstrukte, die in Kap. 4 beschrieben werden.

Wer zuständig ist für die Ausführung der Verhaltensbeschreibung, wird erst in der Implementierungsphase festgelegt. Bei der Modellerstellung wird davon ausgegangen, dass dies in einer passenden Form geschieht.

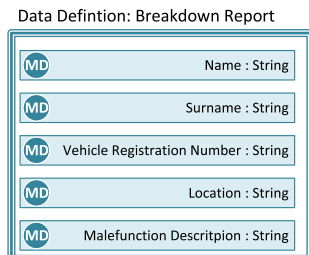
3.4.2 Subjekt und Objekte

Ein Subjekt bearbeitet Objekte durch interne Operationen, versendet Objekte als Nutzlast von Nachrichten bzw. erhält Objekte als Nutzlast von Nachrichten. Die Nutzlast von empfangenen Nachrichten kann für die spätere Verwendung intern abgelegt werden. Die Daten für die Nutzlast von Nachrichten, die gesendet werden, werden aus internen Objekten erstellt.

Analog den Nachrichten haben Objekte ebenfalls einen Namen. Es wird zwischen Datenobjekten und physikalischen Objekten unterschieden. Bei physikalischen Objekten können die physikalischen Eigenschaften wie Größe, Gewicht usw. angegeben werden. Die Struktur und der Inhalt von Datenobjekten können mit den gängigen Beschreibungsmethoden wie z. B. UML definiert werden. Abb. 3.5 zeigt in einer einfachen Darstellung die Inhalte des Datenobjekts Pannemeldung, das als Nutzlast mit der Nachricht *Support Request* gesendet wird.

Die Nachricht *Towing truck* hat als Nutzlast das physikalische Objekt Abschleppwagen und muss in der Modellbildung nicht weiter detailliert werden. Dies geschieht in der Implementierungsphase.

Abb. 3.5 Datenobjekt
Pannemeldung



Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.





Strukturierung komplexer subjektorientierter Modelle

4

In Kap. 3 wurden die Grundkonzepte der subjektorientierten Modellbildung eingeführt. Diese Konstrukte sind prinzipiell ausreichend um alle Prozesse zu beschreiben (Turing-Mächtigkeit). Die in diesem Kapitel eingeführten Konstrukte dienen dazu, komplexe Prozesse und Verhaltensweisen zu strukturieren und damit einfacher nachvollziehbar zu machen.

Zunächst werden Konstrukte eingeführt um komplexe Verhaltensdiagramme hierarchisch zu strukturieren. Im Folgeabschnitt werden Konstrukte detailliert, die bestimmte Verhaltensmuster transparenter und kompakter beschreiben lassen.

4.1 Strukturierung der Kommunikationsstruktur

4.1.1 Externe Subjekte

Mehrere Prozesse eines Prozesssystems können über sogenannte externe Subjekte verknüpft werden. Abb. 4.1 zeigt den Pannenhilfeprozess, allerdings soll dieser nun mit einem Serviceprozess verknüpft werden. Dieser Serviceprozess organisiert eine Werkstatt in die das Pannenfahrzeug geschleppt werden soll und bereitet die Reparatur vor. Um diesen Prozess anzustoßen sendet das Subjekt *Call Center* die Nachricht *Organize Repair* an das externe Subjekt *Service Center*. Dieses Subjekt ist ein sogenanntes externes Subjekt, da es nicht zum Pannenhilfeprozess gehört sondern zum Serviceprozess. Es ist nur für das Subjekt *Call Center* sichtbar und kann deshalb Adressat einer Nachricht von dort sein.

Das Abb. 4.2 zeigt die Kommunikationsstruktur des Serviceprozesses. Dieser Prozess besteht aus nur einem Subjekt, das mit den Subjekten *Call Center* im Prozess *Pannenhilfe*

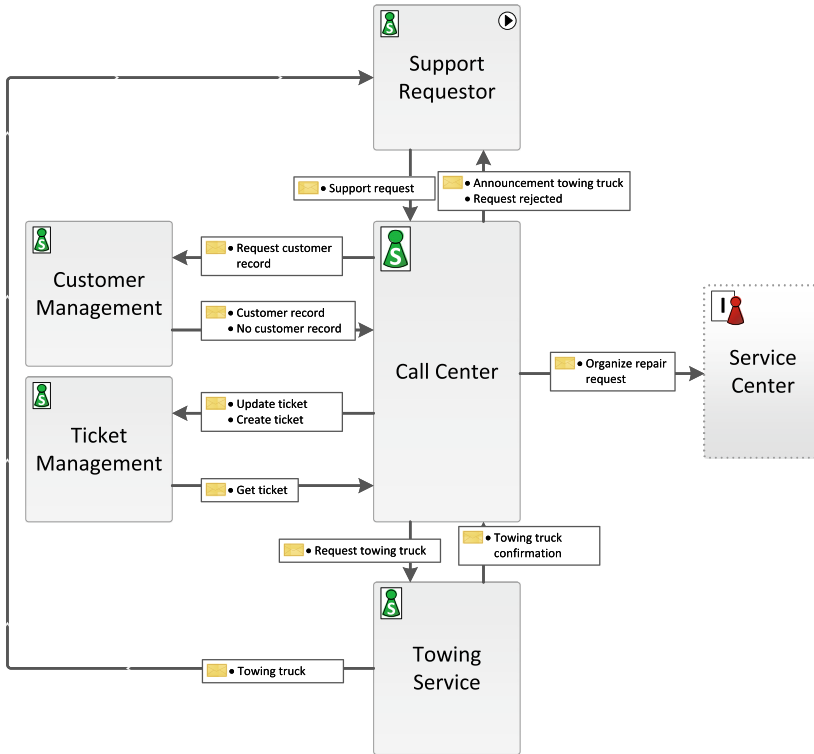


Abb. 4.1 Prozesse über externe (Interface) Subjekte verknüpfen

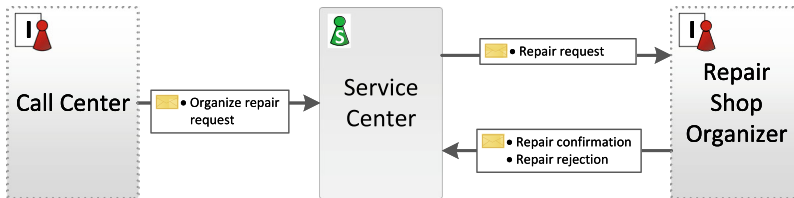


Abb. 4.2 Prozesse über externe (Interface-)Subjekte verknüpfen

und dem Subjekt *Repair Shop Organizer* im Prozess *Terminabsprache* kommuniziert. Beide Subjekte sind extern.

Durch diese Möglichkeit über externe Subjekte mehrere Prozesse miteinander zu verknüpfen, können komplexe Prozesse in Teilprozesse zerlegt werden. Externe Subjekte sind die Schnittstelle zu anderen Prozessen, die unter Umständen noch nicht Teil der Betrachtung sind. Damit kann auch die Modellierung und Implementierung von Prozessen zeitlich oder organisatorisch strukturiert werden.

4.1.2 Netzwerke von Teilsystemen

Das Konzept der externen Subjekte erlaubt zum einen die Strukturierung von Prozessen mittlerer Komplexität, so wie im letzten Abschn. 4.1.1 gezeigt. Dabei sind die einzelnen Teilprozesse verknüpft und liegen auf der gleichen Ebene.

Bei komplexen Systemen kann es aber zum anderen hilfreich sein, Hierarchien von Prozessnetzen zu bilden, wobei die Modelle auf unterschiedlichen (Abstraktions)Ebenen liegen. Abb. 4.3 zeigt ein Beispiel eines komplexen Prozesssystems. Das Prozesssystem umfasst die Teilprozesse um eine Panne vollständig abzuwickeln, einschließlich der Werkstatt, der Ersatzteilversorgung und der Schadensregulierung über die Versicherung.

Jedes Rechteck entspricht einem Prozesssystem. Die Linien zwischen Rechtecken zeigen dass zwischen den jeweiligen Teilsystemen Nachrichten ausgetauscht werden. Die Rechtecke sind mit dem Namen des Teilsystems beschriftet und die Verbindungslinien mit den Namen der Nachrichtenmenge, die zwischen den Teilsystemen ausgetauscht wird. Statt Mengen von Nachrichten zwischen Teilprozessen können die ausgetauschten Nachrichten auch unmittelbar angegeben werden. Aus dem Teilprozess *Towing Service* heraus wird die Nachricht *Broken Car* in den Prozess *Car Repair Workshop* hinein gesendet.

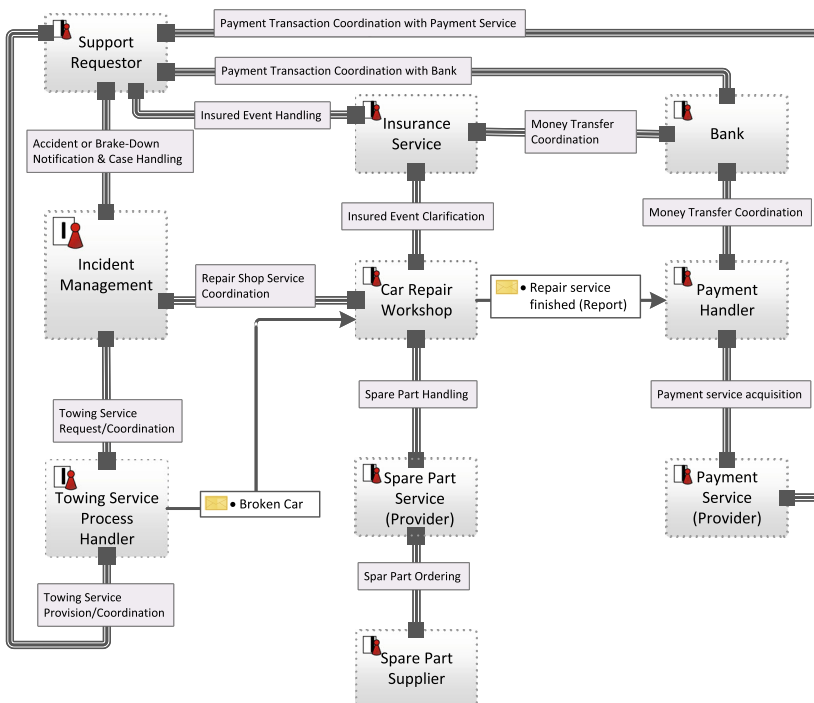


Abb. 4.3 Komplexes Prozesssystem

In einem ersten Detaillierungsschritt soll nur die unmittelbare Pannenhilfe betrachtet werden. Dies betrifft die Teilprozesse *Car User*, *Incident Management*, *Towing Service* und *Car Repair Workshop*. Abb. 4.4 zeigt den jeweiligen Ausschnitt aus dem Gesamtsystem.

Der Teilprozess *Incident Management* ist in zwei weitere Teilprozesse aufgeteilt. Abb. 4.5 zeigt das betrachtete Prozesssystem *Incident Management* mit den Teilprozessen *Support Request Process* und *Service Center Process*.

Damit erreicht die Prozesshierarchie die Subjektebene. Jeder Teilprozess hat externe Subjekte, Border Subjects oder auch Interface Subjects genannt, die von Subjekte in anderen Prozessen adressiert werden können, das heißt an diese können Nachrichten gesendet bzw. von ihnen Nachrichten erhalten werden. Abb. 4.6 zeigt die externen Subjekte der jeweiligen Teilprozesse, mit denen diese verknüpft sind.

Im letzten Schritt können nur noch die Subjekte betrachtet werden die für den betrachteten Prozess notwendig sind. Dies entspricht dann dem Prozess in Abb. 3.1.

Abb. 4.7 zeigt eine Übersicht über die Modelle, die in diesem Abschnitt erstellt wurden und wie sie eine Hierarchien von Prozessnetzen bilden. Die höheren Ebenen sind prinzipiell allgemeiner bzw. abstrakter, während die tieferen Ebene spezieller oder detaillierter sind.

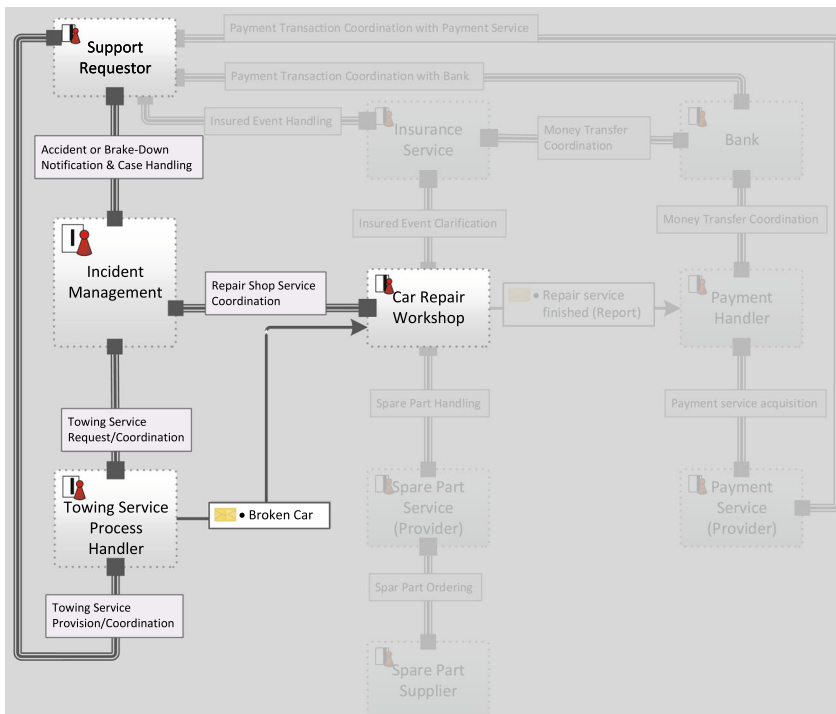


Abb. 4.4 Teilprozessmodell für die unmittelbare Pannenhilfe

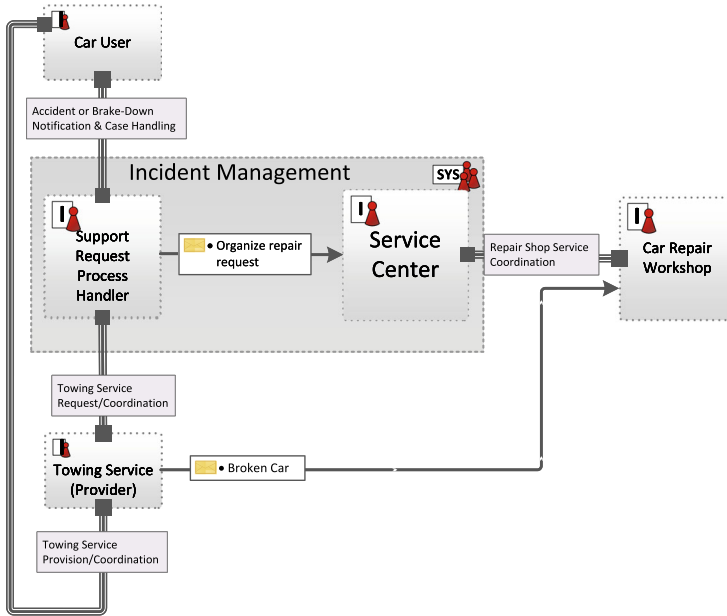


Abb. 4.5 Untermodell zu Abb. 4.4 mit Detaillierung für die Teilprozesse im Incident Management

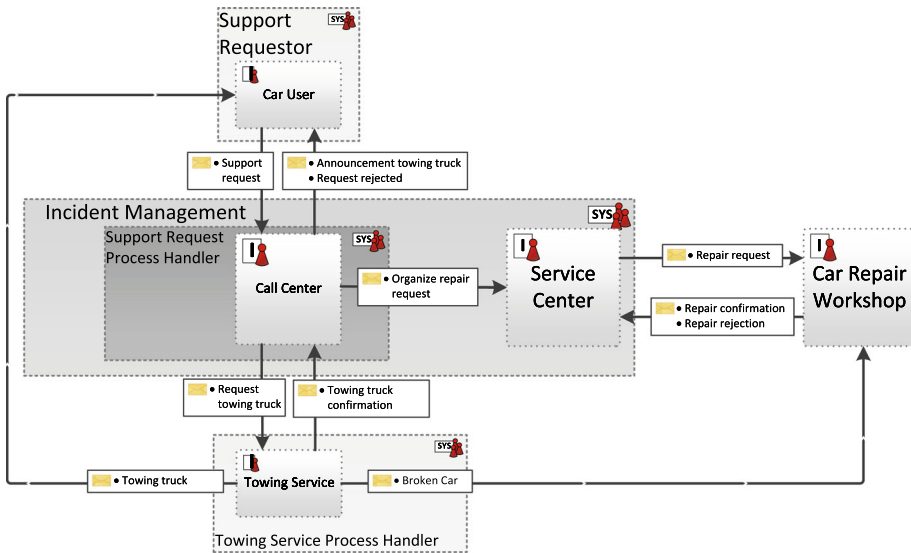


Abb. 4.6 Subjektebene eines hierarchischen Prozessnetzwerks

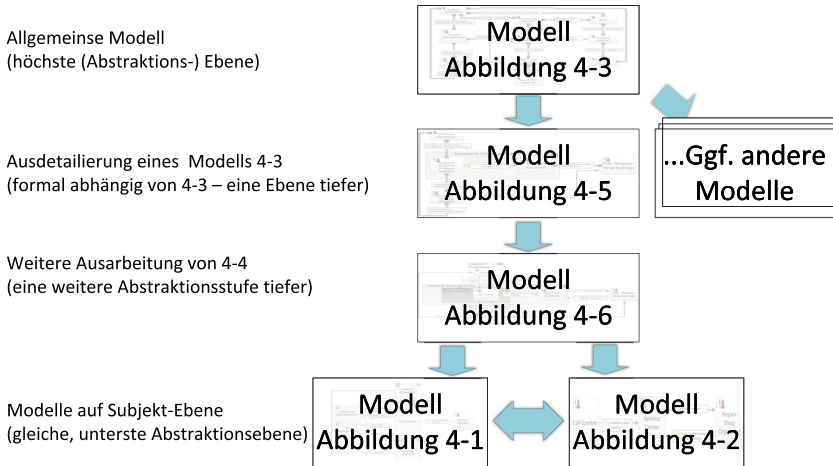


Abb. 4.7 Skizzierte Übersicht der Prozessmodellhierarchie

4.1.3 Multiprozesse und Multisubjekte

In einem Geschäftsprozess kann es mehrere identische Teilprozesse geben, die bestimmte gleichartige Aufgaben parallel und unabhängig voneinander ausführen. So ist es z. B. bei einem Beschaffungsvorgang oft üblich, Angebote von mehreren Anbietern einzuholen.

Ein Prozess oder Teilprozess wird bei der Ausführung folglich mehrfach gleichzeitig oder nacheinander durchlaufen. Eine Menge solcher gleichartiger, unabhängig voneinander ausgeführter Prozesse oder Teilprozesse bezeichnen wir als Multiprozess. Die konkrete Anzahl dieser unabhängigen Teilprozesse wird erst zur Laufzeit ermittelt.

Eine einfache Möglichkeit, dies in PASS umzusetzen, sind die sogenannten *Multisubjekte*. Dabei wird davon ausgegangen, dass das Verhalten (SBD) eines Multi-Subjektes den mehrfach auftretenden Multiprozess darstellt, also mehrfach existiert bzw. *instanziiert* werden kann.

Um unser Beispiel zu erweitern: Der Prozess *Pannenhilfe* soll um die Möglichkeit erweitert werden Angebote von verschiedenen Abschleppdiensten einzuholen um den geeigneten zu beauftragen. Abb. 4.8 zeigt das entsprechend geänderte SID. Das Subjekt *Towing Service* wird zum Multisubjekt. Das Subjekt *Call Center* fordert von verschiedenen *Instanzen* des Multi-Subjektes *Towing Service* ein Angebot an. Aus den verschiedenen Angeboten, die mit der Nachricht *Request towing truck offer* eintreffen wird das passende ausgewählt und eine Beauftragung mit der Nachricht *Towing truck order* versendet. Die einzelnen Subjektelemente eines Multisubjektes werden zur Laufzeit mit Indizes unterschieden. Diese werden bei der Benennung der Adressaten bzw. Absender mit verwendet.

Eine erweiterte Möglichkeit Multiprozesse zu beschreiben ist die Erweiterung von Interface Subjekten um das Multisubjektconcept. Statt einem einzelnen Subjektverhalten gibt es

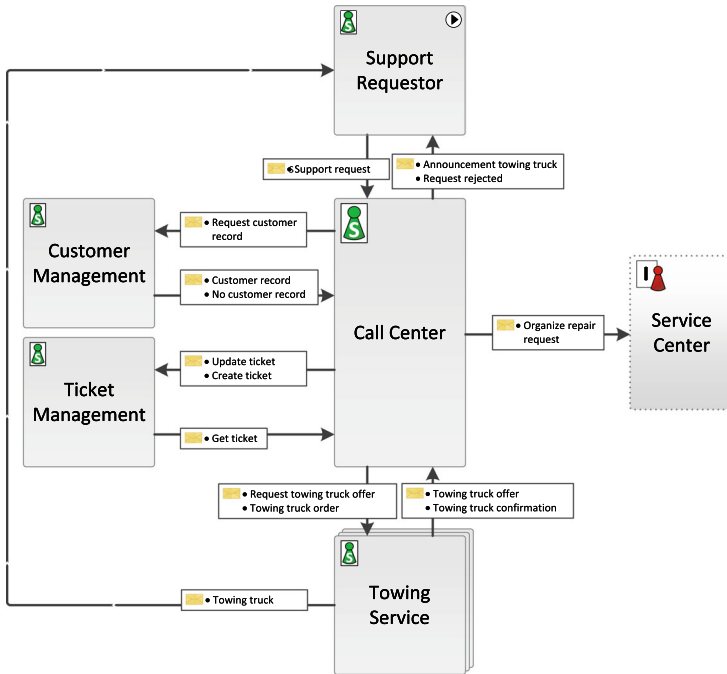


Abb. 4.8 Subject Interaction Diagram (SID) Call Center mit Towing Service als **Multisubjekt**

dann mehrere gleichartige Prozesse, die dann eben als Interface Subjekt ein Multisubjekt haben. Die Multiprozesse werden dann in gleicher Weise beschrieben wie normale Prozesse eines Prozessnetzwerks.

In beiden Fällen ist eine Ergänzung für das beauftragende Subjekt erforderlich, das mit einem Multi-Subjekt bzw. mit einem Subjekt aus einem Multiprozess kommuniziert. Es muss wissen, wie viele und welche Kopien es von einem Multiprozess erzeugt hat. Daher werden bei der Beschreibung seines Verhaltens die einzelnen Kopien wie Elemente eines Feldes indiziert, um die betreffende Prozesskopie bei Zustandsübergängen identifizieren zu können. Möchte etwa ein Subjekt mit einem Subjekt einer bestimmten Prozesskopie aus dem Multiprozessfeld kommunizieren, gibt es bei der entsprechenden Sende- bzw. Empfangsaktion den Index des jeweiligen Prozessexemplars an.

Abb. 4.9 zeigt, wie ein entsprechendes SBD bezüglich Kommunikation mit einem Multi-Subjekt aussehen kann. Dabei muss z. B. angegeben werden, ob mit dem Senden einer Nachricht eine bzw. wie viele neue Instanzen des Multisubjektes erzeugt werden sollen (*Send to new*), oder ob und zu wie vielen bekannten Instanzen (*Send to known - Min: 3 Max: **) Nachrichten geschickt werden sollen. Ebenso kann das Empfangen quantifiziert werden. Im Beispiel müssen mindestens 2 (*Min: 2*) „Towing truck offer“-Nachrichten empfangen werden um weiter machen zu können und diese dann zu vergleichen.

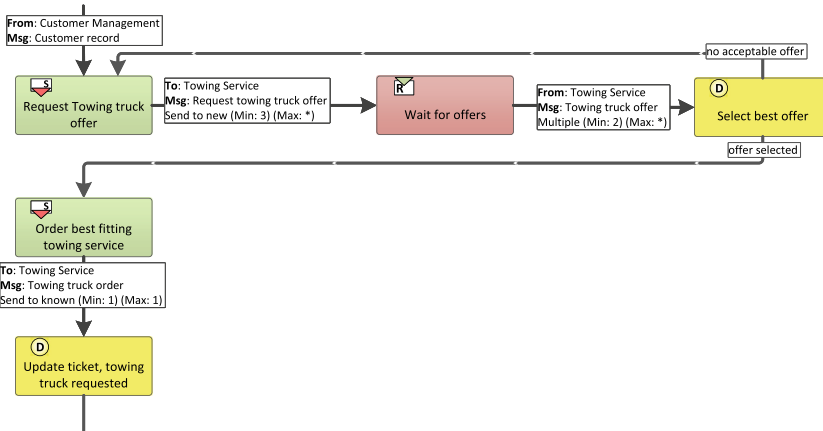


Abb. 4.9 SBD des Call Centers in Interaktion mit dem Multi-Subjekt Towing Service aus Abb. 4.8

4.2 Strukturierung des Subjektverhaltens

4.2.1 Makros/Unterverhalten

Es kommt häufig vor, dass ein bestimmtes Verhaltensmuster innerhalb eines Subjekts wiederholt auftritt. Das geschieht genau dann, wenn in verschiedenen Prozessabschnitten identische Aktionen erforderlich sind. Stehen nur die Grundkonstrukte zur Verfügung, muss das Subjektverhalten mehrfach beschrieben werden.

Alternativ kann ein solches Verhalten als sogenanntes *Verhaltensmakro* definiert werden. Ein solches Makro kann dann beliebig oft an unterschiedlichen Stellen eines Subjektverhaltens eingebunden werden. Damit lassen sich verschiedene Verhaltensvarianten zusammenfassen und das Gesamtverhalten strukturieren. bzw. ein Grundverhalten vereinfachen. Man kann ein solches Makro auch als eine Art Unterverhalten oder Unterprozess verstehen.

Abb. 4.10 zeigt ein Makro-Verhalten namens *Handle Support Request*, dass die Bearbeitung einer Pannenhilfeforderung kapselt. Dies ist ein Teilverhalten des Call Center Subjektes aus Abb. 3.4. Analog zeigt Abb. 4.11 ein Makro-Verhalten für die Organisation des Abschleppdienstes durch ein Makro mit Namen *Handling Towing Car*. Beide zeichnen sich dadurch aus, dass der letzte Zustand die Anweisung ist zu einem ursprünglichen Verhalten zurück zu kehren.

Nach Erreichen dieses Endzustands eines Makros (Zustand mit Quadrat in rechter oberen Ecke) wird das Makro also verlassen und in das übergeordnete Verhalten zurückgekehrt („Return to original state“).

In unserem Fall ist dies das Verhalten für das Subjekt *Call Center*. In der Verhaltensbeschreibung für das Subjekt *Call Center* kann also nun das Verhalten für die Annahme der Unterstützungsanforderung sehr vereinfacht werden in dem man einfach zwei Do-States

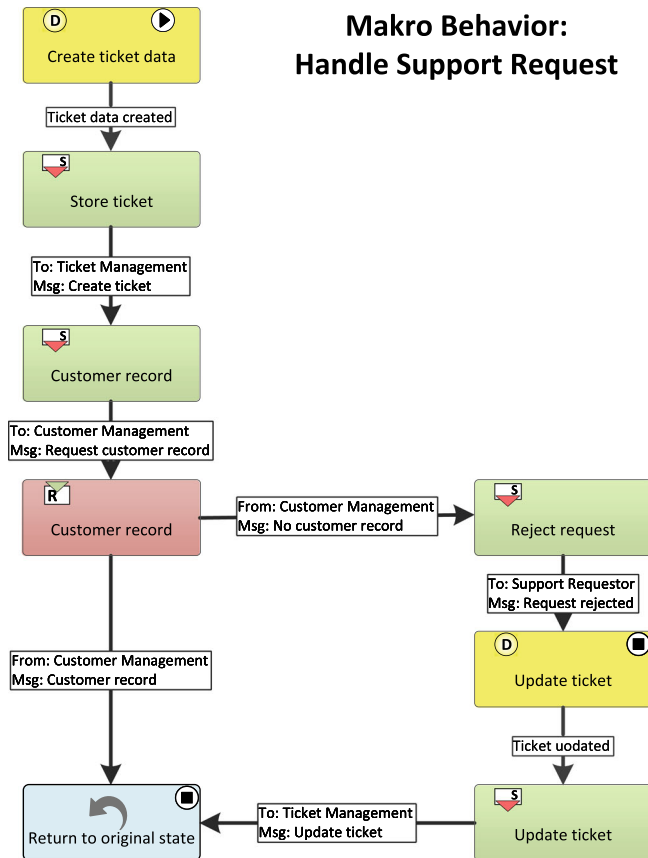


Abb. 4.10 Makro Behaviour Diagram (MBD) Handle Support Request

einfügt und dort jeweils das Makro *Handle Support Request* und die Organisation des Abschleppwagens durch den Aufruf des Makro *Handling Towing Car* ersetzt. Abb. 4.12 zeigt das resultierende Verhalten des Subjekts *Call Center*.

4.2.2 Wahlfreiheit

Bisher wurde das Verhalten von Subjekten immer als eine eindeutige Abfolge von internen Funktionen und Sende- und Empfangsaktivitäten betrachtet. In vielen Fällen ist aber die Reihenfolge der internen Abarbeitung nicht von Bedeutung und es ist wichtig zu zeigen, dass diese Reihenfolge auch nicht festgelegt ist.

Bestimmte Aktionsfolgen können überlappend ausgeführt werden. Wir sprechen hier von Wahlfreiheit. Dies bedeutet, dass der Modellierer keine strikte Reihenfolge der Aktivitäten

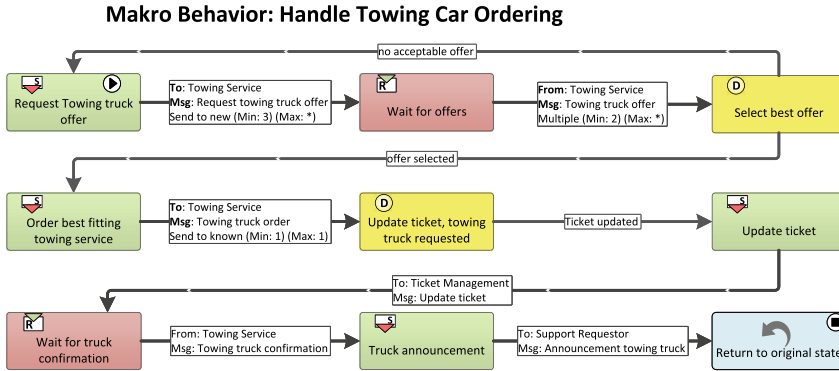
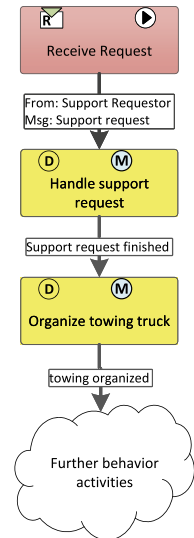


Abb. 4.11 Makro Behaviour Diagram (MBD) Handling Towing Car - Mit Kommunikation zu Multisubjekt

Abb. 4.12 Subject Behaviour Diagram (SBD) des Call Center Subjektes mit Makro aufrufen



vorgibt. Vielmehr kann ein Subjekt(träger) sein Verhalten zur Laufzeit in einem bestimmten Rahmen selbst organisieren. Die Wahlfreiheit im Verhalten wird über Alternativklauseln (Choice Segments) beschrieben, welche mehrere parallele Wege (Choice Segment Paths) skizzieren.

Zusätzlich zu ihrer Existenz können Optionen eingesetzt werden, die bedeuten, dass ein Alternativweg zwingend begonnen werden muss oder dass eine Alternative, einmal angefangen, vollständig durchlaufen werden muss. Dies führt zu folgenden Varianten:

- Anfang gesetzt und Ende gesetzt: Alternative muss bis zum Ende durchlaufen werden.
- Anfang gesetzt und Ende offen: Alternative muss begonnen, aber nicht beendet werden.

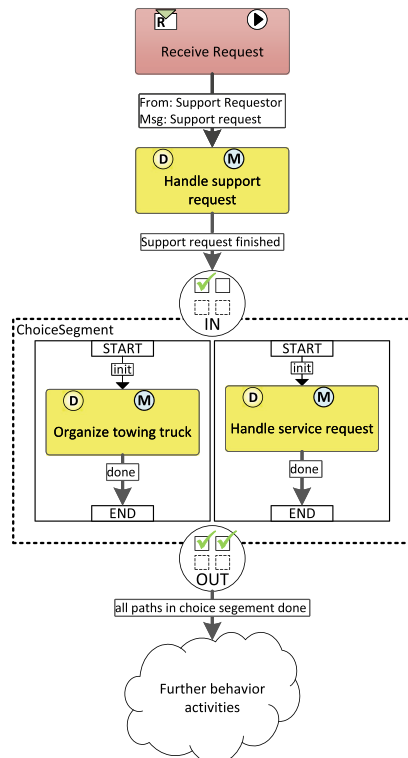
- Anfang offen und Ende gesetzt: Alternative kann durchlaufen, muss dann aber beendet werden.
- Anfang offen und Ende offen: Alternative kann durchlaufen, muss auch nicht beendet werden.

Die Ausführung einer Alternativklausel gilt als abgeschlossen, sobald alle Alternativsequenzen, die begonnen wurden und abgeschlossen sein müssen, vollständig bis zum Endeoperator der Alternativklausel durchlaufen worden sind. Zwischen den Alternativpfaden einer Alternativklausel darf es keine Übergänge geben. Eine Alternativsequenz beginnt in ihrem Startpunkt und endet vollständig in ihrem Endpunkt.

Abb. 4.13 zeigt ein Beispiel für die Modellierung von Alternativklauseln.

Die Makros *Handling Organize Towing* und *Handling Service Request* bzw. die Do-States in denen sie aufgerufen werden können überlappt ausgeführt werden d. h. die Aktionen für die Organisation des Abschleppwagens läuft überlappt mit der Organisation des Werkstatttermins. Beide Makros müssen ausgeführt und abgeschlossen werden.

Abb. 4.13 Alternativklausel (Choice Segment) zur überlappten Ausführung von zwei Makros



4.2.3 Ausnahmebehandlung

Eine Ausnahmebehandlung (auch „*Message Guard*“ oder Message Control, Nachrichtenüberwachung, Nachrichtenbeobachter) ist eine Verhaltensbeschreibung für ein Subjekt, die dann relevant wird, wenn eine bestimmte Ausnahmesituation während des Subjektverhaltens eintritt. In die Ausnahmebehandlung bzw. in das entsprechende „*Guard Behavior*“ wird gesprungen, wenn eine entsprechende Nachricht eintrifft. Der Übergang zur Ausnahmebehandlung hat dann höchste Priorität und wird erzwungen.

Ausnahmebehandlungen sind dadurch gekennzeichnet, dass sie in einem Prozess in vielen Verhaltenszuständen von Subjekten vorkommen können. Das Eintreffen bestimmter Nachrichten, beispielsweise zum Abbruch des Prozesses, hat immer das gleiche Bearbeitungsmuster zur Folge, welches für jeden Zustand, in dem es relevant ist, modelliert werden müsste. Ausnahmebehandlungen verursachen somit einen hohen Zeichenaufwand und unübersichtliche Prozessabläufe, da von jedem betroffenen Zustand ein entsprechender Übergang ausgehen muss. Um dies zu verhindern, wird ein Konzept eingeführt, dessen Methode dem Exception Handling in Programmiersprachen bzw. dem Interrupt Handling bei Betriebssystemen ähnelt.

Abb. 4.14 zeigt das SID des Pannenhilfeprozesses mit der Möglichkeit die Pannenhilfe abzubrechen. Dies wird durch die Nachricht *Cancel support request* veranlasst.

Die Nachricht *Cancel support request* kann zu einem beliebigen Zeitpunkt eintreffen d.h. es muß die Möglichkeit bestehen, dass diese Nachricht in dem jeweiligen Zustand angenommen werden kann. Dies explizit zu modellieren mit Hilfe der Grundkonzepte würde zu unübersichtlichen Verhaltensdiagrammen führen.

Daher wird das Verhalten für die Ausnahmebehandlung separat beschrieben, was die Modellierung vereinfacht.

Abb. 4.15 zeigt wie dies kompakt und übersichtlich beschrieben werden kann.

Wird ein Endzustand der Ausnahmebehandlung erreicht, kann entweder, wie bei einem Makro, in das unterbrochenen Verhalten an den Zustand zurückgekehrt werden von dem aus die Ausnahmebehandlung eingeleitet wurde. Alternativ kann auch festgelegt werden, dass zu einem festgelegten Zustand im ursprünglichen im Verhaltensdiagramm gesprungen wird. Oder, wie in unserem Beispiel, kann der Prozess der Pannenhilfe beendet werden, so dass der Endzustand des Ausnahmebehandlungsverhaltens auch der Endzustand des gesamten Subjektverhaltens des Subjekt *Call Center* ist.

4.2.4 Verhaltenserweiterung

Normalerweise sollten Modelle vor ihrer Implementierung (siehe nächstes Kap. 6) vollständig und aussagekräftig beschrieben worden sein. Jedoch ist keine Beschreibung von Anfang an vollständig, insbesondere bei komplexen Prozessen. Beispielsweise werden bei der Ausnahmebehandlung laufende Aktionen unterbrochen. Dies kann zu Inkonsistenzen bei

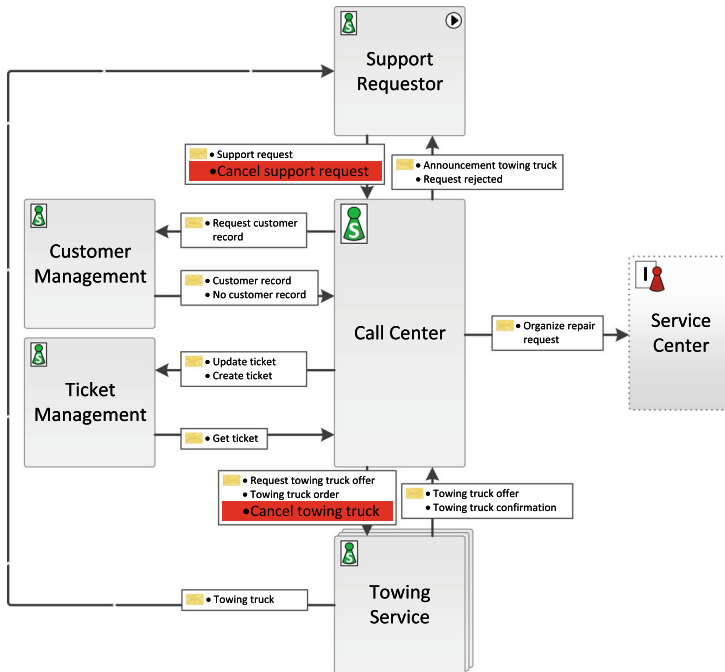


Abb. 4.14 Pannenhilfeprozess mit Nachrichten für die Möglichkeit die Pannenhilfe abzubrechen

der Bearbeitung von Geschäftsobjekten führen. So kann beispielsweise die Aktualisierung des Anforderungstickets im Zustand *Ticket update* durch das Eintreffen der Stornonachricht unterbrochen werden, und das Ticket nur teilweise ausgefüllt werden. Solche Konsequenzen werden wegen der Dringlichkeit von Stornonachrichten in Kauf genommen.

Wird diese Situation erst nach der Implementierung erkannt, kann es sein, dass der Modellierer das ursprüngliche Modell eines Subjektverhaltens nicht mehr ändern kann oder will, aber trotzdem, für z. B. auch einen Sonderfall eine Ergänzung beschreiben möchte. In solchen Fällen kann ein Modellierer durch eine ähnliche Vorgehensweise das Verhalten von Subjekten erweitern. Dazu wird eine Notation verwendet, die sehr ähnlich zur Ausnahmebehandlung ist. Statt mit ‚Ausnahme‘ wird das entsprechende Diagramm eben mit ‚Erweiterung‘ beschriftet. Verhaltenserweiterungen ergänzen ein Subjektverhalten mit Verhaltenssequenzen. In diesen Erweiterungen wird mit Referenzen (State Reference) auf bereits existierende Zustände im ursprünglichen Verhalten angegeben, wo das erweiterte Verhalten neue Verzweigungsmöglichkeiten einfügt oder auch bestehende *überschreiben* kann und wohin diese jeweils führen.

Abb. 4.16 zeigt eine einfache Erweiterung des Guard-Behaviors aus Abb. 4.15. Dabei wird die Prüfung, ob der Abschleppdienst schon bestellt wurde, ergänzt um die Option, dass man dabei feststellt, dass das Ticket noch nicht einmal wirklich erstellt worden ist und dem

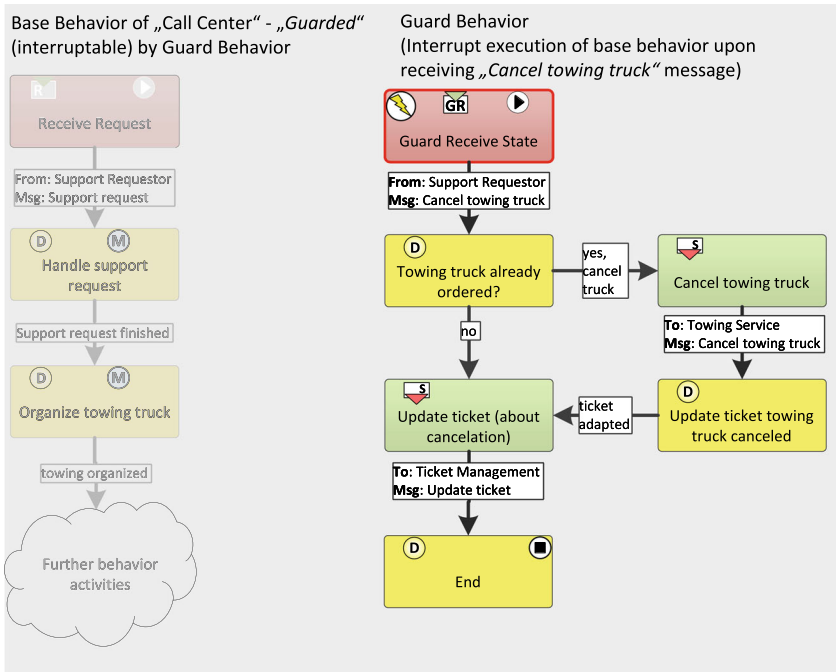


Abb. 4.15 Guard Behavior für das Call Center Subjekt (Base Behavior links im Hintergrund)

entsprechend kein Update möglich ist oder der Truck gecancelt werden muss. Viel mehr kann man nach dem Löschen des Ticketentwurfs direkt zum Ende des Verhaltens gehen, das wiederum durch eine Referenz auf das ursprüngliche Verhalten angezeigt wird. Das erweiterte Verhalten besteht also selbst nur aus den beiden Referenzen und einem Do-State.

4.2.5 Verhaltensschalen

Im anfangs erwähnten Modellierungstool werden Makros, Ausnahmebehandlungen und Verhaltenserweiterungen als sogenannte Schichten (Layers) angelegt. Dies bedeutet, es existiert ein Verständnis, dass diese Verhalten übereinander liegen. Dabei haben „höhere“ Verhalten eine höhere Ausführungspriorität. So ist dies z. B. der Fall für das Guard-Behavior aus Abb. 4.15. Dort wird auch angedeutet, dass das Base Behavior quasi „darunter“ liegt. Analog verhält es sich mit der Erweiterung aus Abb. 4.16 in deren Hintergrund das erweiterte Verhalten aus Abb. 4.15 zu sehen ist.

Abb. 4.16 zeigt eine Übersicht, wie das Zusammenspiel beider Arten von Subjektverhalten verstanden werden kann (Abb. 4.17).

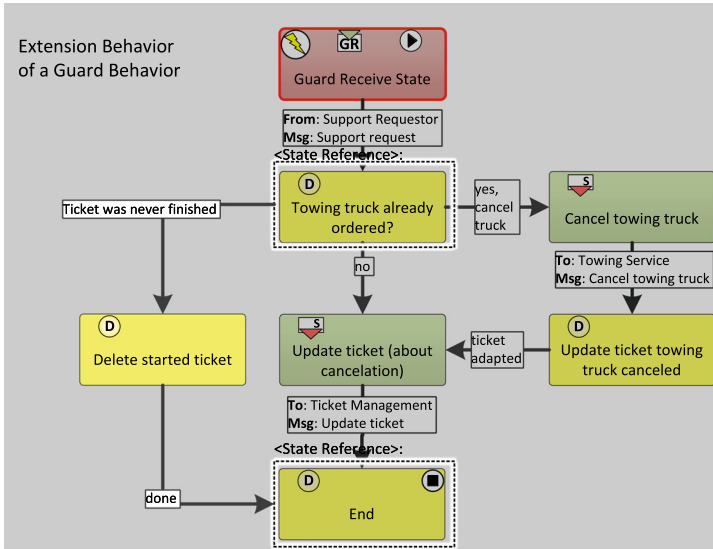
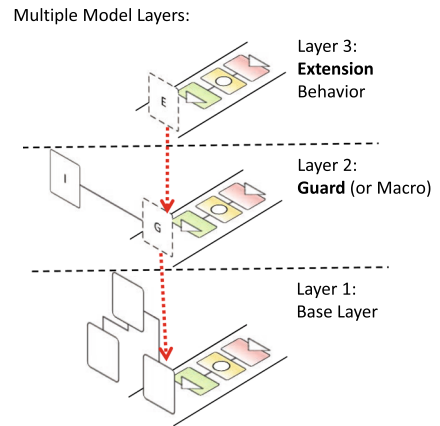


Abb. 4.16 Erweiterung (Extension) des Guard Behavior aus Abb. 4.15

Abb. 4.17 Skizze des Schichtkonzepts der Verhalten aus Abb. 4.12, 4.15 und 4.16 übereinander



Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.





Vor allem in Kap. 3 wurden die Grundkonzepte der subjekt-orientierten Modellierung mit der Modellierungssprache PASS – dem Parallel Activity Specification Schema – vorgestellt. Diese Sprache gehorcht dem diesem Buch zugrunde liegenden Gedankenmodell, dem Paradigma der Subjekt-Orientierung.

Um die Konzepte und das Paradigma effektiv einzusetzen und einfach anwenden zu können, bedarf es sowohl etwas Übung bei der Modellierung als auch Akzeptanz durch die Personen, die sich die Modelle ansehen und diese nutzen (werden). Hilfreich dabei sind *gute* Modelle, die inhaltlich wie visuell ansprechend sind. Dieses Kapitel enthält dafür einige praktische Tipps und Konventionen. So gelingen Modelle, die verständlich und leicht lesbar sind.

5.1 Benennung von Modellelementen

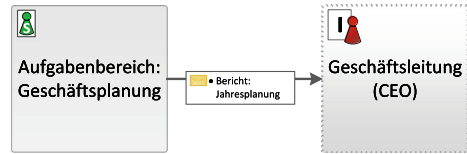
Wie in Kap. 3 beschrieben, ist die Grundidee subjekt-orientierten Modellierens, zwischen den drei Grundelementen der natürlichen Sprache zu unterscheiden: Subjekte, Objekte und Tätigkeiten bzw. in PASS Subjekte, Nachrichten und States.

Dementsprechend sollte bei der Modellierung darauf geachtet werden, dass die Benennung der einzelnen Elemente auch ihre Natur widerspiegelt. Dann können auch die Betrachter ohne Erfahrung die Idee eines Modells leichter verstehen und Verwechslungen mit anderen Sachverhalten sind kaum möglich.

Praktisch leiten sich daraus folgende Regeln ab:

Benennung von Subjekten: Subjekte sollten so benannt sein, dass ihre Natur klar zu erkennen ist. Wenn es also z. B. im SID eines Prozesses eine Abteilung gibt, die ‚Geschäftsplanung‘

Abb. 5.1 Explizite Benennung von Subjekten als aktive Einheiten und Nachrichten als Objekte



heißt, dann kann es nützlich sein ‚Aufgabenbereich:‘ vor den Begriff zu schreiben, damit klar wird, dass hier nicht die *Tätigkeit* der Geschäftsplanung selbst gemeint ist, sondern eben eine *aktive Einheit*. Es ist manchmal auch praktisch die prozessspezifischen Rollennamen um unternehmensspezifische Bezeichnungen zu ergänzen, die klar machen um wen es sich handelt. Beipielsweise kann der Aufgabenbereich der „Geschäftsleitung“ durch den oder die CEO übernommen werden, was alleine aber zu allgemein oder nur inoffiziell wäre. Hierbei muss aber genau abgewogen werden, ob Begriffe die eigentlich zum Themenbereich *Einbettung in die Organisationsstruktur* gehören, also Sachverhalte wie CEO oder Abteilung, bereits in einem allgemeinen Modell stehen sollten.

Benennung von Nachrichten: Gute Namen für Nachrichten machen klar, dass es sich um *passive Objekte* handelt und geben Auskunft über ihren Inhalt. Ein sprechender Nachrichtentname entspricht in etwa einer guten Betreffzeile einer E-Mail, die dem Empfänger klar macht was er erwarten kann. Wenn z.B. die Geschäftsplanungsabteilung die Ergebnisse ihrer Jahresplanung verschickt, dann ist das Ergebnis logischerweise ein *Bericht*, der dann auch verschickt wird und nicht die Tätigkeit der Jahresplanung selbst. Eine solche Ergänzung kann sehr praktisch sein. Siehe auch Abb. 5.1.

Benennung von Tätigkeiten: Im Verhalten stellen die Do-, Send- und Receive-States Tätigkeiten dar. Diese können natürlich Bezeichnungen mit substantivierten Verben tragen, wie z. B. *Jahresplanung* oder *Geschäftsplanung*. Um ihre Natur klarer hervorzuheben und auch die Idee der einfachen Sprache fortzuführen, ist die Empfehlung hier eindeutig zum aktiven Beschreiben wie z. B. *Jahresplanung durchführen*. Dabei kann man sich entscheiden ob man wie zuvor den Infinitiv verwendet oder das Ganze imperativ bzw. als Anleitung formuliert z. B. *Führe Jahresplanung durch*. Unabhängig von individuellen Präferenzen empfiehlt es sich, den Beschreibungsstil der States innerhalb eines Modells einheitlich zu halten.

5.2 Layouting

5.2.1 Layouting SBDs

Gruppierungen

Eine der ersten und wichtigsten Empfehlungen ist die Idee, Zustände in funktionale Blöcke zu gruppieren und zu beschreiben. Im verwendeten Modellierungswerkzeug erfolgt dies mit

dem Mittel der „State Group“. Obwohl dies keine Auswirkungen auf die Funktionalität des Modells hat, ist es eine große Hilfe für das Verständnis und das schnellere Erfassen von Ideen im SBD.

Oft gehören eine Kombination aus Empfangs-, Aktivitäts- und Sendezuständen zu einem einzelnen, zusammenhängenden, konzeptionellen Vorgang. Durch die Unterlegung mit farbigen Boxen wird ihre Verbindung sofort sichtbar. Darüber hinaus kann bei Folgeschritten in einem anderen SBD ihre Verbindung durch die Verwendung derselben Farbe dargestellt werden. Somit kann ein Aspekt der Kritik an PASS – die mangelnde, einfach zu verstehende Darstellung der zeitlichen Abfolge von Prozessschritten über Subjektgrenzen hinweg – durch ein schnelleres Verständnis der Zusammengehörigkeit reduziert werden.

Die folgende Abbildung zeigt das Prinzip anhand eines SBD.

Als praktische Anmerkung sei hier erwähnt: Es wird empfohlen, nur gedämpfte Farben zu verwenden. Zu knallige Farben wären für den Betrachter eher eine Belastung als eine Hilfe.

Es gibt auch ein paar weitere Ideen für diese Konventionen: in Abb. 5.2 wurde grüne Farbe für die „Ziellinie“, den Projektabschluss, verwendet, während rote Farben die wichtigen, zentralen Knotenpunkte anzeigen. Gruppen mit ähnlichem Zweck sollten auch innerhalb des gleichen Farbbereichs gehalten werden, wie z. B. die Berichterstattung an den Chef.

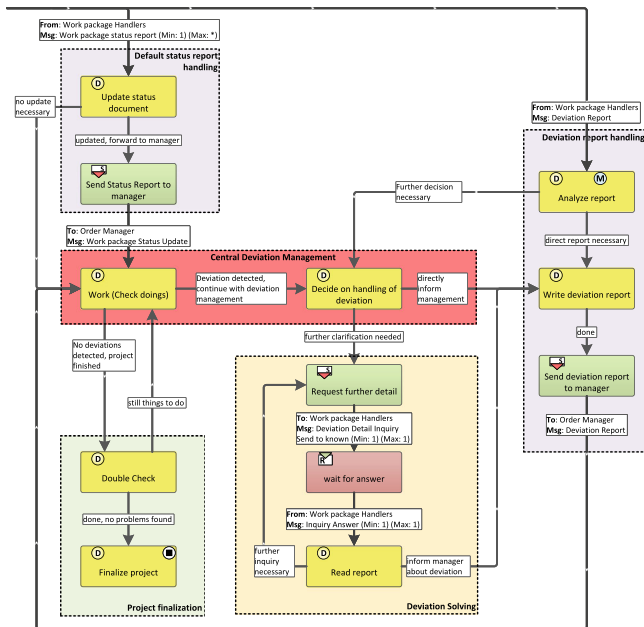


Abb. 5.2 Gruppierung und Layout im SBD

Vertikales oder horizontales Layout für zeitliche Abfolgen

Der folgende Vorschlag für einen Standard betrifft die Anordnung von Prozessschritten bzw. States bei der Modellierung eines SBD und wurde in den meisten SBDs in diesem Buch genutzt. Er sollte einfach umzusetzen sein:

Zeitlich nacheinander folgende Zustände (oder ihre Gruppen) kommen immer in eine vertikale Abfolge, während Verzweigungen für alternative Entscheidungen horizontal angeordnet werden.

Diese Wahl ist willkürlich und kann auch andersherum definiert werden. Wichtig ist, dass, soweit möglich, die jeweilige Dimension jeweils nur für einen Zweck genutzt wird, wobei kleine Ausnahmen möglich sind.

Ein Argument für einen horizontalen Zeitfluss wäre, dass für kleinere Prozesse der Modellgraph z. B. besser auf eine PowerPoint-Folie passt, und diese Ausrichtung der klassischen Leserichtung von links nach rechts entspricht.

Das vertikale Layout wird jedoch von den Autoren hier präferiert mit dem Hintergrund, dass in den meisten Prozessen zeitliche Abfolgen häufiger sind als alternative Entscheidungen und ein Scrollen über mehrere Bildschirmseiten für den temporalen Fluss wahrscheinlicher ist. Diese vertikale Scrollen ist in der Regel leichter, z. B. auf PCs mit Mäusen die dafür ein Mausrad haben. Und auch die Darstellung und das Scrollen auf mobilen Endgeräten, wie Smartphones, ist oft besser mit einem vertikalen Layout.

Vermeidung unnötiger Kreuzungen und gemeinsames Zurückführen

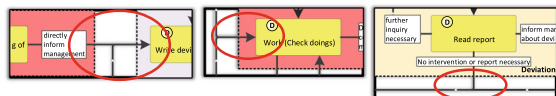
Wie schon bei der Gruppierung von States, ist eine ansprechende Anordnung von Zuständen und umso mehr auch die Anordnung der Transitionen bzw. Pfeile zwischen ihnen immer eine gute Idee. Eine gute Pfeilführung kann ein Diagramm sehr aufgeräumt und sauber erscheinen lassen. Andersherum kann eine unsaubere Führung das Diagramm leicht sehr unaufgeräumt und unübersichtlich werden lassen, was wiederum zu Ablehnung des ganzen Modells nur aus Darstellungsgründen führen kann.

Ein einfach umzusetzender Vorschlag für ein gut lesbares Layout ist daher, unnötige Kreuzungen von Transitionen so weit möglich zu vermeiden:

Um dies zu erreichen, hilft zu verstehen, dass Transitionen keine Leiterbahnen auf einer Elektronikplatine sind, die sich nicht berühren dürfen. Ganz im Gegenteil wird empfohlen, Pfade, die auf denselben Zielzustand hinleiten, gemeinsam übereinander zu führen.

Abb. 5.3 zeigt die Ausschnitte aus Abb. 5.2, in denen dies gut zu sehen ist.

Abb. 5.3 Gemeinsames Führen von Transitionen in 5.2



Dabei ist nur wichtig, dass das Zusammenführen erst nach den Beschriftungen der Transitionen erfolgt, da diese eindeutig sein müssen, um sehen zu können, welche Ausgangsbedingungen für einen Zustand gelten müssen, um zum Zielzustand zu gelangen.

Ein verbesserungswürdiges Beispiel

Zum Vergleich zeigt Abb. 5.4 (etwas übertrieben), wie das SBD aus Abb. 5.2 aussehen könnte, wenn die entsprechenden Konventionen nicht berücksichtigt werden. Es obliegt dem Leser selbst zu vergleichen und zu beurteilen wie wichtig oder nützlich die einzelnen Konventionen sind und ob sich der Aufwand für eine entsprechende Aufbereitung lohnt.

5.2.2 Layout-Prinzipien für SIDs

Natürlich gibt es auch für SIDs ähnliche Konventionen, um ihr optisches Ansprechen entsprechend verbessern zu können.

Abb. 5.5 zeigt ein SID aus einem realen Anwendungsfall, das zwar inhaltlich passend ist, bei dem aber keine Zeit bestanden hatte, es noch einmal sorgfältig hinsichtlich seiner optischen Qualitäten zu überarbeiten.

Abb. 5.6 zeigt inhaltlich das gleiche SID, das aber mit ein wenig Aufwand erheblich verbessert wurde.

Dabei wurden folgende Ideen umgesetzt:

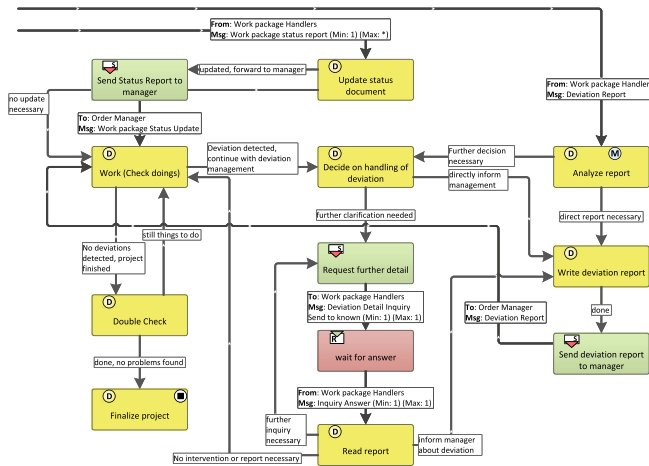


Abb. 5.4 Beispiel für ein verbesserungswürdiges SBD ohne Beachtung von Konventionen – gleicher Inhalt wie Abb. 5.2

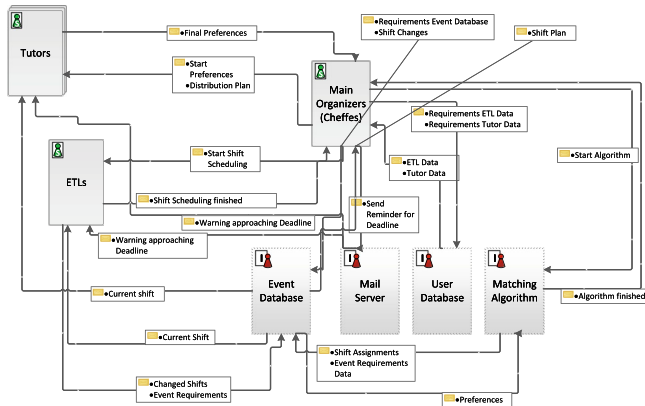


Abb. 5.5 Ursprüngliches SID eines realen Anwendungsfalls

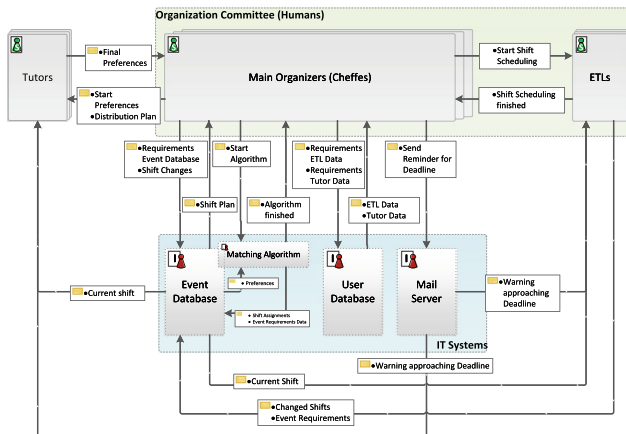


Abb. 5.6 Überarbeitetes SID – gleicher Inhalt wie Abb. 5.5

- **Gruppierung von Subjekten:** Eine zusätzliche, farbliche Unterlegung und Benennung hilft die Zusammengehörigkeit, hier die vom Organisationskomitee und den IT Systemen, auf einen Blick zu erkennen. Ebenso wie die Gruppierung von States in SBD hat dies in normalem PASS keine Bedeutung, verbessert aber die Verständlichkeit
- **Gerade und direkte Verbindungen:** Je mehr Ecken und Kurven Nachrichtenpfeile aufweisen, umso unaufgeräumter kann ein SID erscheinen. Lange, gerade Pfeile, die klare Linien schaffen, verbessern den optischen Eindruck. Direkte Verbindungen sind ideal. Damit dies umgesetzt werden kann, ist es manchmal notwendig *zusätzliche Verbindungspunkte* an Subjekten anzubringen, insbesondere wenn diese in ihrer Form verändert wurden.

Wenn Subjekte nebeneinander liegen, sollte darauf geachtet werden, dass sie die gleichen

Abmessungen auf zumindest den ihnen zugewandten Kanten haben, damit die Verbinder auch ohne zusätzliche Verbindungspunkte direkt zueinander passen.

- **Vermeidung von Überschneidungen:** Dort, wo möglich, wurden die Pfeile so gelegt, dass sie einander nicht überschneiden. Dies ist nicht immer möglich. Wo dies machbar ist, sollte es aber versucht werden. Hier kann auch daran erinnert werden, dass, wie bei den Transitionen in den SBDs, Pfeile ggf. übereinander gelegt werden können, solange klar bleibt, von welchem Subjekt welche Nachricht gesendet wird.
- **Anordnung:** Damit die Nachrichtenverbinder gerade und überschneidungsfrei gelegt werden können, sollte auf die Platzierung von Subjekten geachtet werden, so dass, wo möglich, direkt kommunizierende Subjekte nebeneinander liegen.
Eine direkte Berührung zweier Subjekte hat keine formale Bedeutung, zeigt aber auf, dass diese eng zusammen hängen, auch wenn sie im Modell als zwei getrennte, aktive Einheiten betrachtet werden.
- **Größe und Form:** Die meisten Modellierungsumgebungen geben eine Standardgröße für Subjekte vor. Wenn diese aber verändert werden kann, dann kann das genutzt werden um die Bedeutung von bestimmten Subjekten zu erhöhen oder zu verringern. Dies wurde beispielsweise mit den Subjekten der Hauptorganisatoren (Main Organizers) und des Matching-Algorithmus gemacht, um ihre Bedeutung hervorzuheben bzw. für den Algorithmus diese etwas zu reduzieren. Außerdem können größere Subjekte leichter durch gerade Verbinder mit mehreren Subjekten verbunden werden.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.





Die vorangegangenen Kapitel haben die Grundidee des subjekt-orientierten Denkens beschrieben und erläutert, wie dem entsprechend Prozessmodelle subjekt-orientiert erstellt werden. Die Modellierung ist ein wichtiger Schritt des Business Process Managements (BPM). Nach der initialen Erstellung eines Modells kann dieses genutzt werden, um, idealerweise unter Einbeziehung der Prozessbeteiligten, Analysen und Verbesserungen des Prozesses zu entwickeln, die dann wiederum als Modell beschrieben werden – siehe auch die Begriffe IST- und SOLL Prozesse. Aber was geschieht danach mit dem Modell eines entwickelten Prozesses? Es sollte umgesetzt oder *realisiert* werden. Damit also die Modelle nicht in irgendeinem Schrank oder auf einer Festplatte verstauben, sollte dafür gesorgt werden, dass die modellierten Umstände so ausgeführt werden wie vorgesehen.

6.1 Modell, Instanz, Kontext und Implementierung

Prozessinstanzen

Prozessmodelle sind Beschreibungen von Prozessabläufen. Sobald ein beschriebener Prozess gestartet wird, sprechen wir von der *Ausführung des Prozessmodells*. Die Durchführung eines Prozesses gemäß eines Prozessmodells bezeichnen wir eine *Prozessinstanz*. Diese wird beim Starten des Prozesses erzeugt. Entsprechend können anhand eines Prozessmodells viele Prozessinstanzen erzeugt werden, die parallel zu einander ablaufen können. Trifft für einen beschriebenen Bestellprozess das Startereignis „Bestellung von Huber“ ein, so wird zur Bearbeitung dieses Ereignisses eine Prozessinstanz des Bestellprozessmodells für genau diese Bestellung *angelegt* und gemäß der Modellbeschreibung ausgeführt. Kommt auch vor dem Ende der Bearbeitung der Bestellung von Huber eine zweite Bestellung von

Müller herein, dann wird dafür eine zweite Instanz erzeugt und beide unabhängig voneinander abgearbeitet, solange die Bearbeitungsressourcen, wie z. B. eine Verpackungsmaschine, nicht von der jeweils anderen Instanz blockiert sind. Eine Prozessinstanz wird wieder aufgelöst wenn ein Endzustand erreicht wurde. Ein Prozessmodell ist also eine Schablone für das Kreieren von Prozessinstanzen zur Abarbeitung der zugehörigen Geschäftsereignisse.

Prozessinstanzen, Subjektinstanzen und Aufgabenträger

Durch das Konzept der Subjekte in der Subjektorientierung werden bei der Instanziierung von entsprechenden Prozessmodellen *innerhalb* der Prozessinstanz für jedes Subjekt *Subjektinstanzen* gebildet. Wurden im Modell nur einzelne Subjekte beschrieben, so gibt es in eine Prozessinstanz jeweils nur eine Subjektinstanz pro Subjekt. Wurden jedoch einzelne Subjekte als Multisubjekte modelliert (siehe Abschn. 4.1.3), dann kann es innerhalb einer Prozessinstanz mehrere Instanzen des gleichen Subjekts geben. Ein Prozessinstanz gilt dann als beendet bzw. terminiert, wenn alle Subjektinstanzen in einem *Endzustand* sind.

Prozess- und Subjekt-Instanzen sind dabei immer abstrakte Ideen. Damit tatsächlich etwas in der Realität geschieht muss es reale *Aufgabenträger* (Subjekt-Carrier) geben, die tatsächlich Aktivitäten setzen. Bei der Erzeugung einer Prozessinstanz oder in deren späterem Verlauf muss also jedem beschriebenen Subjekt ein Aufgabenträger zugeordnet werden. Der Aufgabenträger kann dabei ein Mensch, ein physikalisches Gerät, ein Computerprogramm oder eine Kombination aus diesen Möglichkeiten sein. Wie und in welcher Form genau diese Zuordnung erfolgt hängt vom *Kontext* ab in dem das Modell realisiert bzw. *eingebettet* wird (siehe nächster Abschnitt).

Wichtig ist dabei, dass Subjekte jeweils nur prozessspezifische Rollen oder Aufgabengebiete sind und nicht allgemein die Handlungen von z. B. realen Menschen beschreiben. Das bedeutet im Gegenzug, dass eine reale Person z. B. die Aufgaben von unterschiedlichen Subjekt-Instanzen in unterschiedlichen Prozessinstanzen übernehmen kann, wie es in Abb. 6.1 skizziert ist. Auch können Gruppen von realen Akteuren für die Ausführung eines einzelnen Subjektes verantwortlich sein oder sich dabei abwechseln bzw. ausgetauscht werden.

Die Art der Aufgabenträger, mit der Subjekte oder Objekte realisiert werden, hat Auswirkungen auf die Instanzbildung. Sollen für ein Prozessmodell zu einem Zeitpunkt mehrere Instanzen existieren, so müssen für die Realisierung auch instanzierungsfähige Aufgabenträger eingesetzt werden. Instanzierungsfähige Aufgabenträger sind in der Lage ihre Aufgaben in einer Prozessinstanz auszuführen, ohne dass dadurch andere Prozessinstanzen, in denen sie auch beteiligt sind, beeinflusst werden.

Umgekehrt sollten sich die Aktivitäten eines Aufgabenträgers in anderen Prozessinstanzen nicht auf Aktionen in der betrachteten Instanz auswirken. Ein Beispiel für instanzierungsfähige Aufgabenträger sind Menschen. Menschen können mehrere Prozessinstanzen bearbeiten. Zu einer bestimmten Zeit bedienen sie die Prozessinstanz 1 und dann wechseln sie zu einer anderen Instanz 2. Da ein Aufgabenträger, insbesondere ein Mensch, in Regel nicht wirklich parallel arbeiten kann, werden parallel ablaufende Prozessinstanzen,

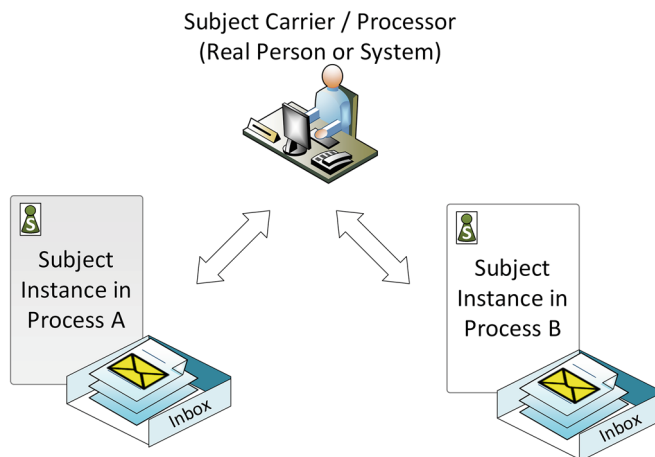


Abb. 6.1 Konzept von Aufgabenträger (Subject Carrier) und Subjektinstanzen jeweils mit Input Pool

die dem gleichen Aufgabenträger zugeordnet sind, von diesem in entsprechenden Zeitscheiben abwechselnd bedient.

Einbettungskontext

Die Möglichkeit zur Erzeugung von Prozessinstanzen bzw. die Art wie diese tatsächlich realisiert werden hängt vom *Kontext* bzw. der *Umgebung* ab, in dem das Prozessmodell für die Ausführung der Instanzen *eingebettet* werden soll.

Einbetten in einen Kontext bedeutet, ein Modell an die *Elemente* eines Kontextes bzw. eine Umgebung zu geben, damit ab dem Zeitpunkt der Einbettung Prozessinstanzen auf Basis des Modells starten und ausführen können – also, dass diese das Modell dadurch realisieren können.

Durch den Kontext wird dabei definiert durch welche Aufgabenträger der in einem Subjekt beschriebene Ablauf ausgeführt wird. Der Aufgabenträger kann dabei, wie schon erwähnt, ein Mensch, ein physikalisches Gerät, eine Computerprogramm oder eine Kombination aus diesen Möglichkeiten sein.

Ähnlich ist es bei den benutzten Objekten. Die in einem Modell beschriebenen Objekte (Nachrichten) können bei der Realisierung IT-Objekte wie z. B. Datensätze innerhalb von Informationssystemen, aber auch physische Gegenstände wie z. B. Papierformulare mit den jeweiligen darauf definierten Operationen sein. Die jeweiligen Objekt- und Subjektrealisierung müssen dabei zusammenpassen. Ein Softwareprogramm kann nicht direkt eine Kurbel drehen d. h. wenn zur Realisierung eines Subjekts oder Objekts Programme verwendet werden, so müssen zwischen ihnen Übergänge zur Verfügung stehen. Werden für die Realisierung eines Prozesses Personen und IT-Systeme eingesetzt, wird die Schnittstelle zwischen diesen beiden Implementierungstechniken in der Regel durch Bildschirmmasken (GUIs) der Software realisiert, d. h. ein Anwender gibt die erforderlichen Informationen in

das IT-System ein. Steuert ein IT-System eine Maschine, so werden die von der Maschine benötigten Informationen durch entsprechende elektronische Schnittstellen vom IT-System an die Maschine übermittelt.

Ein Prozessmodell kann in beliebig viele unterschiedliche Kontexte eingebettet werden. So kann ein Modell für einen Verkaufsprozess z. B. in die Niederlassung Europa oder in der nordamerikanischen Niederlassung integriert werden. Dies bedeutet, dass das gleiche Modell verwendet wird, allerdings zur Ausführung andere Personen und auch unterschiedlich IT-Systeme verwendet werden. Es gibt dann laufende Prozessinstanzen des Modelles in Europa und andere in Nordamerika.

Die Interpretation, dass es sich beim Kontext, wie hier angegeben, um existierende Organisationseinheiten an verschiedenen Orten der Welt handelt, ist nur eine Möglichkeit. Der Begriff kann jedoch auch etwas spezieller aufgefasst werden, wenn zur Realisierung ein zentrales IT-System verwendet wird. Das Prozessmodell wird dabei digital als eine Art Programm in ein sogenanntes Workflowsystem bzw. einer Workflow-Engine Software geladen. Die Prozessinstanzen existieren dann innerhalb der Software und werden von dieser bzw. den Nutzern der Software digital ausgeführt. In diesem Fall bezieht sich der Einbettungskontext auf die technische Umgebung, in der ein digitales Modell umgesetzt wird.

In der Realität ist es aber auch der Fall, dass beides gleichzeitig gemeint sein kann, da entsprechende IT-Systeme von Organisationen eingesetzt werden, wodurch sich sowohl ein digitaler Software- als auch ein organisationsbedingter Kontext ergibt.

Der Kontext in den ein Modell eingebettet wird, und die notwendige Instanzfähigkeit bilden den Rahmen für die Implementierung der Subjekte und der zugehörigen Kommunikation.

6.2 Realisierung von Subjekten

Das Ausführungskonzept von PASS versteht sich so, dass eine Subjektinstanz immer aus einem Verhalten, den verwendeten Objekten und einem Input Pool besteht. Jedes dieser Teile kann durch Menschen, physischen Komponenten, IT-Systeme oder Kombinationen daraus realisiert werden.

6.2.1 Realisierung des Input Pools

Der Input Pool ist der „Briefkasten“ in dem „Nachrichten“ an eine Subjektinstanz gespeichert werden. Der Input Pool selbst ist dabei nicht direkt modelliert, aber es gibt die *Input Pool Restrictions* wie sie in Abschn. 3.3 beschrieben wurden.

Abhängig von der Realisierung der Nachrichten können in einem Input Pool elektronische Nachrichten, Papierdokumente, oder andere physische Objekte eingelagert werden. Ein Input Pool kann konzeptionell also sowohl physische und elektronische Objekte

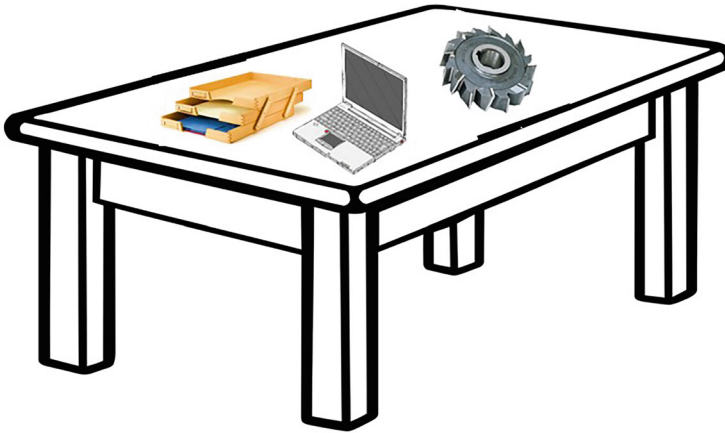


Abb. 6.2 Input Pool zur Aufnahme unterschiedlich realisierter Nachrichten

aufnehmen. Er besteht damit z. B. aus einem elektronischen Maileingang für die elektronischen Nachrichten und einer Kiste, in die die physischen „Nachrichten“ eingelagert werden. Erhält z. B. eine Subjektinstanz eine im Modell beschriebene Nachricht „Ersatzteil“, so kann damit gemeint sein, dass das Empfängersubjekt dieses Ersatzteil physisch in die Kiste gelegt bekommt und gleichzeitig die Begleitinformationen elektronisch in einer Mailbox als E-Mail landen. Durch einen Aufkleber auf dem Ersatzteil kann der Bezug zu den Begleitinformationen hergestellt werden.

Abb. 6.2 zeigt das Konzept eines Input Pools, der Nachrichten aufnehmen kann, die unterschiedlich implementiert sind. Der Input Pool wurde dabei durch einen Tisch realisiert, auf dem in einer Papierablage Papierformulare abgelegt werden können, elektronische Nachrichten, die über einen PC empfangen werden, und auf dem erhaltene physikalische Objekte abgelegt werden können.

6.2.2 Realisierung des Subjektverhaltens

Das Verhalten von Subjekten, d. h. die Reihenfolge, in der die einzelnen Operationen und Kommunikationsaktionen ausgeführt werden und die in den Verhaltensdiagrammen des Prozessmodells beschrieben werden, kann durch Personen, physische Maschinen oder IT-Systeme, oder aus entsprechenden Kombinationen umgesetzt werden. Dies hängt, wie zuvor in Abschn. 6.1 beschrieben, vom Aufgabenträger ab, der dem Subjekt für die Instanzierung in einem Kontext zugewiesen wurde.

Eine Person erhält eine Beschreibung des Verhaltens und führt gemäß dieser Beschreibung die entsprechenden Aktionen aus. Eine solche Beschreibung ist Bestandteil der Arbeitsbeschreibung des entsprechenden Mitarbeiters. Die Ablaufbeschreibung beginnt z. B. damit,

dass der Mitarbeiter aufgefordert wird, im Input Pool nachzusehen, ob ein Verpackungsauftrag eingetroffen ist. Dieses „Nachsehen“ hängt davon ab, wie der Empfang eines Versandauftrags und sein Input Pool realisiert sind. Ist der Empfang des Auftrags als e-Mail realisiert, so sieht der Mitarbeiter in seiner Mailbox nach und öffnet den vorhandenen Versandauftrag digital. Danach soll die Aktion „Versandpapiere vorbereiten“ ausgeführt werden. Dazu wird er in seiner Prozessbeschreibung aufgefordert das Versandsystem aufzurufen. Dieses wiederum fordert ihn auf, ein elektronisches Formular auszufüllen, welches ausgedruckt wird. Aus seiner Prozessbeschreibung entnimmt er, dass dieses Formular an dem Ersatzteil befestigt wird. Dieses Ersatzteil hat er vorher durch ein Transportband erhalten. Gemäß der Prozessbeschreibung versendet er das Ersatzteil zur Weiterbearbeitung an eines seiner Nachbarsubjekte. Das Versenden dieses physischen Objekts wird ebenso durch ein Transportband realisiert.

Bei einer digitalen Umsetzung kann das Subjekt auch durch ein entsprechendes Workflowsystem gesteuert werden. Das Workflowsystem stößt ein Anwendungsprogramm an und fordert einen Benutzer auf, eine Bildschirmmaske auszufüllen. Diese Informationen werden dem Anwendungsprogramm übergeben, das ein Formular ausdruckt. Danach fordert das Workflowsystem den Benutzer auf, das Formular an ein physisches Objekt zu kleben und den Abschluss dieser Aktion dem Workflowsystem zu bestätigen. Dies führt zur Aufforderung seitens des Workflowsystem an den Benutzer, das physische Objekt auf ein Laufband zu legen. Auch diese Aktion muss dann dem Workflowsystem bestätigt werden. Anstelle des Abarbeitens der Papierbeschreibung des Prozesses wird der Mitarbeiter durch ein Workflowsystem durch den Prozess navigiert.

Die Steuerung eines Subjekts kann auch durch ein elektromechanisches Gerät erfolgen. Dabei fordert die Maschine den Bediener auf, bestimmte Knöpfe, Hebel usw. in einer bestimmten Reihenfolge zu betätigen.

6.2.3 Realisierung von Objekten

Objekte, vor allem die Nachrichten in den Modellen, können als Datenobjekte in Software oder als physische Gegenstände realisiert sein. Auf Softwareobjekten, die innerhalb von IT-Systemen existieren können entsprechende Datenmanipulationen ausgeführt werden, während physisch realisierte Geräte entsprechende Operationen auf physischen Objekten ausführen.

Dabei ist z. B. ein Objekt Drehmaschine notwendig um verschiedene Operationen auf einem Werkstück auszuführen, während die Reihenfolge der Bedienschritte durch eine Person ausgeführt wird. Die Reihenfolge der Bedienschritte sind Teil des Subjektverhaltens. In diesem Fall implizieren sie aber die Existenz der Drehmaschine, obwohl diese nicht explizit als Nachricht oder Subjekt beschrieben wurde.

Physische Objekte können mit elektronischen Schnittstellen versehen sein, um bestimmte Aktionen des Objekts auszuführen, z. B. der Knopf der Drehmaschine, den ein Subjekt drücken kann.

Bei Datenobjekten in IT-Systemen hingegen muss es entsprechende Eingabemasken oder GUIs geben, damit ein Mensch entsprechende Veränderungen der Daten vornehmen kann.

6.3 Realisierung der Kommunikation

Folgen wir der Philosophie, findet frei nach Luhmann, Kommunikation immer statt, und zwar zwischen Subjekten, sowie zwischen Objekten und zwischen Subjekten und Objekten. In einem PASS Prozessmodell wird Kommunikation allerdings nur zwischen aktiven Subjekten beschrieben. In dieser vereinfachten Betrachtung können Subjekte mit Objekten zwar *interagieren* aber nicht *kommunizieren*, da dort kein aktiver Empfangsvorgang stattfinden kann.

In jedem Fall kann die Kommunikation aber sowohl elektronisch oder physisch oder in einer Mischform erfolgen, auch in Abhängigkeit der Realisierung der Objekte (Nachrichten). Um die Realisierung der jeweiligen Kommunikation zu vereinfachen, werden häufig bereits verfügbare Plattformen eingesetzt.

Kommunikation zwischen Subjekten bedeutet, dass die sendende Subjektinstanz die Nachricht in den Input Pool des Empfängers ablegt. Eine Nachricht hat einen Namen und eine Nutzlast. Mit dem Namen wird in der Regel die Nutzlast bezeichnet.

Die Nutzlast sind oft Daten, und kann sowohl aus physischen Objekten, elektronischen Objekten oder Mischformen bestehen. Beide können Daten enthalten oder einfach durch ihre Existenz Dinge ausdrücken. Ein ausgefülltes Papierformular ist ein physisches Objekt, das Daten enthält. Analog kann mit einer e-Mail ein elektronisches Dokument mit entsprechenden Daten gesendet werden. Ein einfaches Werkstück, das in den Eingang einer Bearbeitungsstation gelegt wird, enthält dagegen keine Nutzlast und nur sein Vorhandensein bzw. sein Empfang impliziert, dass es an dieser Station danach weiter bearbeitet werden soll.

Abhängig von der Realisierung der sendenden Subjektinstanz und der Realisierung des Input Pools der empfangenden Subjektinstanz können entsprechende Realisierungsanpassungen notwendig sein. Sind beide durch Software realisiert, so werden Nachrichten in der Regel auch digital übertragen. Für die Übertragung der elektronischen Nachricht müssen die beiden Subjekte eine Übereinkunft über die verwendeten Übertragungsprotokolle treffen und im Besitz entsprechender Geräte zum Austausch sein, z. B. im Fall von E-Mail. Allerdings können Nachrichten zwischen elektronisch realisierten Subjekten auch physisch übertragen werden. Dazu wird z. B. die Nachricht beim sendenden Subjekt ausgedruckt, von einem Menschen in einen Briefumschlag gesteckt, per Post versendet, und beim Empfänger wird der Papierinhalt wieder in eine elektronische Form übertragen und dann in den Posteingang des Empfängers gelegt werden – siehe auch Realisierung des Input Pools (siehe auch Abschn. 6.2).

Die Beispiele zeigen wie abhängig von der Realisierung der sendenden und empfangenden Subjekte entsprechende Implementierungsadaptionen zwischen Subjekt- und Kommunikationsrealisierungen vorgenommen werden.

6.4 Umsetzungsplattformen

Um die Realisierung eines neuen Prozessmodells zu vereinfachen und vor allem um Änderungen am Prozess bzw. die Veränderung des damit verbundenen Prozessmodells schneller und kostengünstiger ausführen zu können, werden in der Regel „*Plattformen*“ verwendet, die für bestimmte wiederkehrende Aufgaben Bausteine zur Verfügung stellen. Solche Bausteine können die Implementierung von Subjektablaufen, die Kommunikation zwischen Subjekten und den Zugriff von Subjekten auf Objekte vereinfachen.

Mit Plattform meinen wir hier in der Regel IT bzw. spezieller Workflowsysteme, in die Prozessmodelle digital hinein geladen werden. Die Plattform erzeugt dann bei Bedarf die Instanzen des Prozesses bzw. die Subjektinstanzen, stellt den realen Aufgabenträgern Schnittstellen zur Interaktion zur Verfügung und realisiert auch intern die Kommunikation zwischen den Subjekt-Instanzen. Dafür wird aus den Ablaufbeschreibungen eines Prozessmodells automatisch entsprechender Programmcode erzeugt. In diesen Code können dann auch über entsprechende Programmierschnittstellen die benutzten Objekte als digitale Daten erstellt werden. Menschliche Benutzer können dann durch Bildschirmmasken integriert werden, die ebenfalls automatisch durch die Codes erzeugt werden.

Ähnlich wie menschliche Benutzer durch Bildschirmmasken in den Subjektablauf integriert werden, können auch Sensoren oder Aktuatoren durch entsprechende Softwarebausteine an ein solches Workflowsystem angebunden und damit in den Ablauf eingebaut. Ebenso können Aufrufe von Funktionen aus existierenden Softwaresystemen wie z. B. SAP Systemen aus dem Code der Implementierung des Subjektverhaltens heraus aufgerufen werden.

In der weiterführenden Literatur (siehe Anhang) sind mehrere Implementierungen solcher Unterstützungssysteme beschrieben. Da sich diese kontinuierlich weiter entwickeln, wird hier nicht weiter auf Details eingegangen.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.





Management der Erstellung subjektorientierter Systeme

7

Ein subjektorientiertes Modell besteht aus Subjekten, Prädikaten und Objekte die zusammen einen Satz bilden. Wobei die Subjekte als zentrale Elemente entweder miteinander über Nachrichten kommunizieren oder Objekte durch den Aufruf der jeweiligen Operationen benutzen. Diese beiden Relationen bilden die Struktur eines subjektorientierten Modells, wobei die Kommunikationsbeziehungen zwischen Subjekten den Aufrufbeziehungen zwischen Subjekt und Objekt übergeordnet sind. Das durch die Kommunikationsbeziehungen aufgebaute Netzwerk von Subjekten bildet den Ausgangspunkt für die Strukturierung von Realisierungsprojekten. Dieses Netzwerk ist der Ausgangspunkt für die Erstellung einer Work Break Down Structure, welche dann Schritt für Schritt in einen Projektplan mündet. In den folgenden Abschnitten wird beschrieben, wie die einzelnen Schritte für die Erstellung eines Projektmanagementplans für ein subjektorientiertes System aussehen.

7.1 Projektmanagement Modellbildung

Mit Hilfe eines *Projektstrukturplans* werden die in einem Projekt anfallenden Aufgaben in eine geeignete Struktur gebracht. Um dies zu erreichen werden die relevanten Aufgaben möglichst vollständig erfasst. Dabei werden die Aufgaben funktionsorientiert, objektorientiert oder zeitorientiert gegliedert.

Bei der Funktionsorientierung steht die Art der Tätigkeit im Vordergrund, während bei der Objektorientierung der Schwerpunkt auf dem Produkt selbst liegt. Das Produkt wird in entsprechende Teilkomponenten gegliedert. Durch eine zeitorientierte Gliederung werden dann Aufgaben aus der Funktions- bzw. Objektorientierung in eine zeitliche Abfolge für die Realisierung gebracht und die notwendigen Ressourcen zugeordnet. Die funktions- bzw.

objektorientierten Aufgaben werden in der Regel zur besseren Übersichtlichkeit hierarchisch gegliedert.

Die Einführung eines Geschäftsprozesses teilt sich vornehmlich in zwei Teile auf. Im ersten Schritt wird das Modell erstellt, validiert und optimiert. In dieser ersten Phase geht es im wesentlichen um folgende Aufgaben:

- Identifikation der Prozessumgebung. Welche Nachrichten kommen aus der Umgebung und werden an die Umgebung übermittelt?
- Identifikation der Subjekte an der Schnittstelle zur Umgebung. Welche Nachrichten werden aus der Umgebung erhalten bzw. in diese Umgebung gesendet und von welchen Subjekten aus der Umgebung kommen diese Nachrichten bzw. an welche gehen sie?
- Detaillierung der Teilsysteme. Welche Subjekte eines Teilsystems sind von anderen Teilsystemen aus sichtbar und welche Nachrichten senden bzw. erhalten sie von sichtbaren Subjekten (Grenzsubjekten oder Schnittstellensubjekten) anderer Teilsysteme?
- Identifikation der Subjekte in einem Teilsystem und der ausgetauschten Nachrichten inklusive der Nutzlasten.
- Identifikation der in einem Subjekt verwendeten Objekte mit den benötigten Funktionen.
- Beschreibung der jeweiligen Subjektverhalten.

Je nach Umfang des Prozesses können die Teilsysteme der obersten Ebene einem Teilprojektleiter zugewiesen werden. Die Teilsysteme können je nach Umfang des Teilsystems in weitere kleinere Teilsysteme gegliedert werden, für die ebenfalls Teilprozessleiter zuständig sein können. Dieses Vorgehen kann bis zur Subjektebene fortgesetzt werden. Ein Teilprojektleiter kann dabei auch für mehrere Teilsysteme zuständig sein. Durch dieses Vorgehen werden die Projektstruktur ermittelt und die Verantwortlichkeiten für die Teilsysteme definiert. Nach der Strukturierung des Systems können die ausgetauschten Nachrichten identifiziert werden. Die Teilprojektleiter stimmen die Nachrichten ab, die zwischen den jeweiligen Teilsystemen ausgetauscht werden. Danach werden die Subjektverhalten und die benötigten Objekte mit ihren Funktionen beschrieben. Die wichtigste Aufgabe der jeweiligen Teilprojektleiter ist die Kommunikation zwischen den Teilprojektleitern der darunter liegenden Ebene zu organisieren.

7.2 Projektmanagement Modellimplementierung

In der Implementierungsphase gilt es folgende Aufgaben zu erledigen:

- Implementierung des Subjektverhaltens
- Implementierung der Objekte
- Implementierung der Kommunikation zwischen den Subjekten
- Implementierung des Zugriffs auf Funktionen von Objekten

Die Struktur des Prozessmodells kann dabei direkt als Ausgangspunkt für die Struktur des Projektmanagements dienen. Die Teilprojektmanager organisieren die notwendige Kommunikation zwischen den Mitarbeitern in ihren jeweiligen Teilprojekten, d. h. die Kommunikationsstruktur in dem zu realisierenden Prozesssystem entspricht der Kommunikationsstruktur zwischen den jeweiligen Realisierungsverantwortlichen.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.





Können aus Geschichten Prozesse abgeleitet werden?

8

In diesem Abschnitt gehen wir der Frage nach, wie aus natürlichsprachiger Information ablauffähige Prozessmodelle abgeleitet werden können. Dabei sind zwei Gesichtspunkte wesentlich: Zum einen, welche Elemente der natürlichen Sprache für ablauffähige Modelle elementaren Charakter besitzen. Dieser Frage gehen wir in Abschn. 8.1 nach. Zum anderen, in welcher Form relevanter Ablaufkontext, welcher aus Fließtext stammt, modelliert werden kann, um die Nachvollziehbarkeit des Modellierungsvorgangs zu erhalten. Dieser Frage gehen wir in Abschn. 8.2 nach.

8.1 Von der natürlichsprachlichen Beschreibung zum ausführbaren Programm

Probleme aus der Geschäftswelt werden anfangs zunächst in natürlicher Sprache beschrieben, einschließlich einer möglichen Lösung des beschriebenen Problems. Sind diese Probleme operativer Natur d. h. liegen sie in unzureichenden Geschäftsabläufen (Prozessen) muss die Prozesslogik umgearbeitet oder deren Implementierung angepasst werden. Heute sind Prozesse in der Regel digitalisiert d. h. sie werden durch Software implementiert, in deren Ablauf Menschen einbezogen sind (siehe Kap. 6). Zur Problemlösung ist somit von einer informellen Problembeschreibung zu einem durch digitale Systeme ausführbaren Programm zu kommen. Dieser Schritt beinhaltet umfangreiche Aufgaben, so dass heute eine informelle natürliche Beschreibung zunächst in ein Modell umgesetzt wird. Dieses bereits zumindest halbformale Modell wird dann in ein Computerprogramm umgesetzt. Je formaler dieses Modell ist, um so einfacher lässt sich daraus ein Programm ableiten (Low code).

Die Herausforderung im Rahmen der Modellierung besteht nun darin, mittels Modellierungskonstrukten nahe an der natürlichen Sprache schrittweise von einer natürlichsprachigen

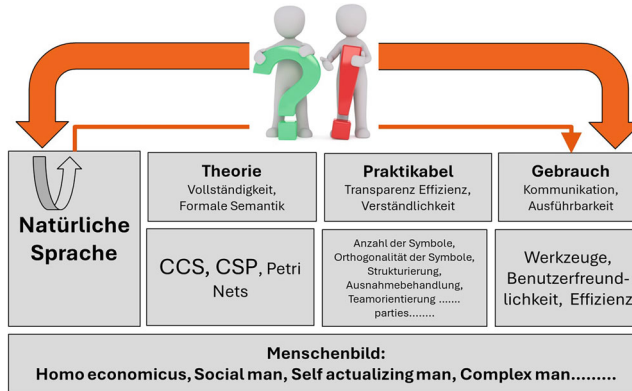


Abb. 8.1 Modellbildung: Von der natürlichen Sprache zum Programm

Darstellung von relevanten Sachverhalten zu einem aussagekräftigen Modell zu kommen, um den Aufwand niedrig zu halten. Zusätzlich sollte das Modell formal genug sein, um computergestützt ausführbar zu sein, und damit die entsprechende Umsetzung in ein durch digitale Systeme ausführbares Programm einfach gestaltbar sein.

Abb. 8.1 zeigt die Brücke, die von der natürlichen Sprache zum Computerprogramm geschlagen werden soll.

8.2 Von Texten zu kontext-sensitivem Prozesswissen

Anhand eines industriellen Verpackungsprozesses mit modernen Technologien aus dem Industrie4.0-Umfeld demonstrieren wir, wie Fließtexte organisiert und zur Ableitung von Geschäftsprozessen analysiert werden können, um schließlich prozessrelevantes Wissen in seinem Kontext darstellen zu können. Wir nutzen dazu die Erkenntnisse der Forschung aus dem Bereich der Rhetorik, im Speziellen der Rhetorical Structure Theory (RST).

Der zu Anschauungszwecken genutzte logistische Anwendungsfall betrifft das Verpacken eines medizinischen Gutes gemäß den spezifischen Anforderungen für dessen Handhabung und Lagerung.

Das in der Folge beschriebene Vorgehen zur Strukturierung des Texts and darauf aufbauend, der Ableitung von Prozesswissen, besteht aus 2 Schritten: Im ersten Schritt zerlegt der Leser den Text in schlüssige Teile, welche den Verpackungsprozess in seiner Ursprungstextform beschreiben. Im zweiten Schritt kommen rhetorisch begründete Strukturen zum Einsatz. Wir nutzen beispielhaft eine Ontologie, wie sie von Nakasone et al.¹ für Storytelling

¹ Arturo Nakasone, Mitsuru Ishizuka, M. (2006). SRST: A storytelling model using rhetorical relations. Proc. 3rd Int. Conf. on Technologies for Interactive Digital Storytelling and Entertainment, TIDSE 2006, Springer, Berlin, 127–138.

eingeführt wurde. Sie umrahmt den sogenannten RST Tree (RSTree). Das ist jene Struktur, die mit den im ersten Schritt bestimmten Textteilen mit Hilfe der Ontologie konstruiert wird. Die verwendeten Beziehungen der RST-Ontologie liefern den Kontext, der für die anschließend vorgestellte themenorientierte Spezifikation relevant ist, und bereichern so den Modellierungsprozess.

8.2.1 Schritt 1: Identifikation in sich schlüssiger Geschichtenteile

Im logistischen Anwendungsfall hat unser Leser folgende Textbereiche voneinander unterschieden:

1. Der Prozess des Verpackens medizinischer Güter ist von entscheidender Bedeutung, da er Teil gesellschaftsrelevanter Dienstleistungen (Gesundheitsversorgung) ist. Bei der Anwendung innovativer Konzepte aus der industriellen Transformation und der Nutzung cyber-physischer Intelligenz über das Internet ist aufgrund der „Systemrelevanz“ des Gesundheitssektors besondere Aufmerksamkeit erforderlich.
2. Für den Transport von medizinischen Gütern müssen bestimmte Bedingungen erfüllt sein. Wenn also ein medizinisches Gut, wie z. B. ein Impfstoff, von einem Arzneimittelhersteller zu einem Gesundheitsdienstleister transportiert werden muss, muss sein Transport entsprechend den Lagerungs- und Handhabungsanforderungen vorbereitet werden, die von den Besonderheiten dieses Gutes abhängen.
3. Die vom Arzneimittelhersteller bereitgestellten Informationen zu den Lagerungs- und Handhabungsanforderungen werden dem medizinischen Gut beigelegt. Die beigelegten Informationen erfordern eine Überprüfung der Lagerungs- und Handhabungsanforderungen, bevor der Verpackungsvorgang beginnen kann.
4. Sobald das medizinische Gut eingetroffen ist, bewegt sich das Robotertransportsystem der Verpackungseinheit, z. B. Spot®, von seinem Standort in der Verpackungseinheit zum Lieferort, um sich für die Verpackung vorzubereiten.
5. Das Robotersystem aktiviert ein Scangerät, um die Anforderungen zu erfassen, die mit dem medizinischen Gut verbunden sind, das entsprechend verpackt werden muss.
6. Sobald die gescannten Informationen analysiert wurden, bewegt sich das Robotertransportsystem zu einem Regal im Lagerbereich der Verpackungseinheit, wo die entsprechenden Sensoren und Kisten verfügbar sein sollten.
7. Nach der Verfügbarkeitsprüfung der relevanten Sensoren und einer Kiste, die zu den Sensoren und dem medizinischen Gut passt, nimmt ein Roboterarm diese als Nutzlast des Transportroboters auf.
8. Das Robotertransportsystem bewegt sich zu dem Ort, an dem das Gut ursprünglich abgeladen wurde. Er entlädt die Sensoren und die Kiste, bestückt die Kiste mit den Sensoren und schließt den Packvorgang ab, indem er das Gut in die Kiste legt und diese schließt.

Die gewählte Strukturierungsform des Textes stellt die durchzuführenden Aufgaben sowie deren Umsetzung mittels cyber-physischer Systemkomponenten in den Mittelpunkt der Betrachtung.

8.2.2 Schritt 2: Kontextualisierung der Textteile

In nun folgenden Schritt wird eine weitere Strukturierung des Textes vorgenommen. Die folgende Abb. 8.2 zeigt das Ergebnis der Anwendung der von Nakasone et al. vorgeschlagenen Metadaten und ihren Verbindungen aus der eingeführten Ontologie auf die Textbeschreibung des aus dem logistischen Anwendungsfall abgeleiteten Wissens über den Geschäftsprozess.

Die Anfangsszene ist die Ankunft des medizinischen Gutes. Das Konzept interagierender Netzwerkpartner, die die choreografische Packintelligenz erfassen, bezieht sich auf das Internet of Behavior(s). Das Transportsystem der Packeinheit wird von einem Roboter implementiert, der auch andere Aufgaben ausführen kann, wie z. B. 3D-Scannen. In der ersten Szene, mit dem Ereignis des Beginns des Packvorgangs, muss sich das Robotersystem zu dem Ort bewegen, an dem das zu verpackende medizinische Gut geliefert wurde. Neben der Lokalisierung des medizinischen Gutes müssen vor dem nächsten Schritt die Anforderungen für die Lagerung und Handhabung des Gutes erfasst werden.

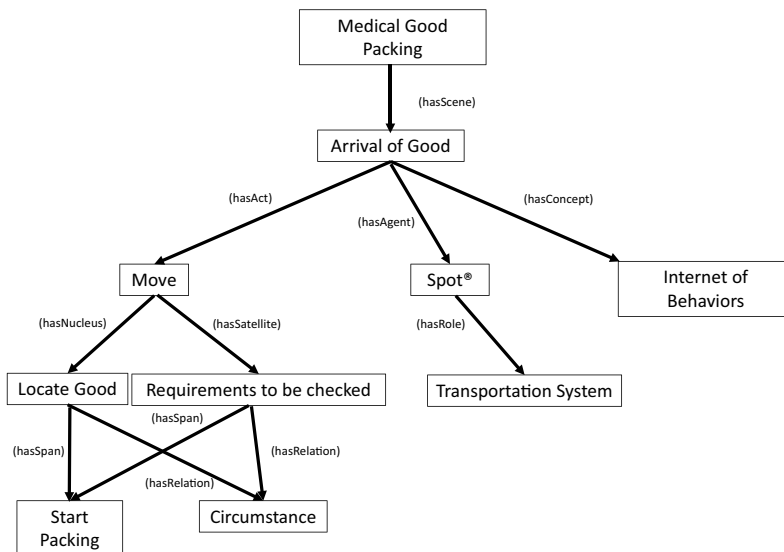


Abb. 8.2 Rhetorical Structure Theory (RST) Tree für erste Szene des Verpackungsprozesses auf Basis einer Storytelling-Ontologie

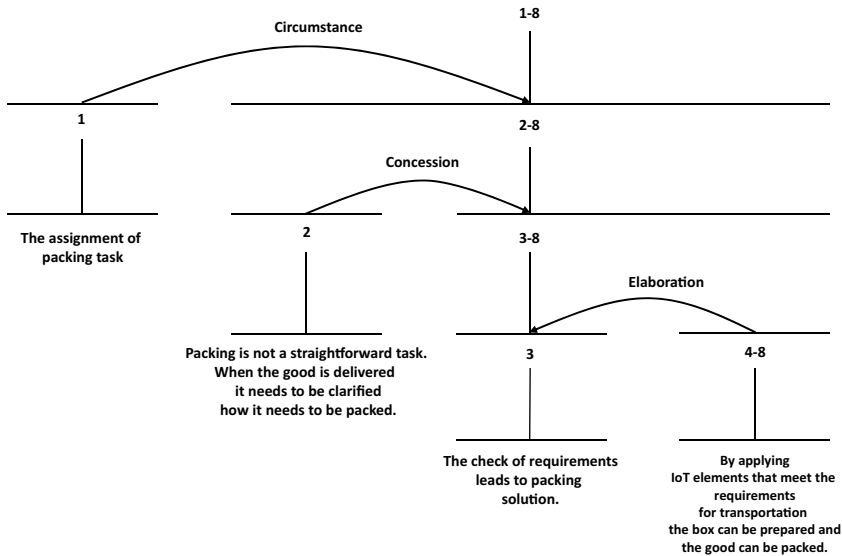


Abb. 8.3 Oberer Teil des RST Tree für die Textbeschreibung des Verpackungsprozesses von Industrie 4.0

Abb. 8.3 zeigt eine typische Anreicherung prozessbezogener Informationen. Jede Relation enthält auch einen kurzen Text, der die wichtigsten Elemente zusammenfasst. Neben der Umstandsrelation wird die Konzessionsrelation angewendet, um die durch die Ausarbeitungs- und Sequenzrelation beschriebenen Schritte besser zu verstehen (Abb. 8.4).

8.2.3 Schritt 3: Ableitung von Prozessmodellen

Wie in Abb. 8.5² gezeigt, ermöglicht der RSTree die Einordnung der im Prozess ausgeführten Kernfunktionsaktivitäten entlang der Sequenzrelation. Auf diese Weise kann zum einen die Begründung formalisierter Arbeitsabläufe bereitgestellt werden. Zum anderen kann die erforderliche Vorverarbeitung von Informationen, wie z. B. die Überprüfung der Voraussetzungen zur Durchführung des Verpackungsprozesses, erfasst werden. Beide Fähigkeiten erleichtern die Identifizierung von Kernkomponenten (Subjekten), die im Prozessmodell dargestellt werden sollen.

In Abb. 8.5 sind die Kernkomponenten des Packprozesses inklusive ihrer Interaktionen dargestellt. Das SID betrifft die Transportvorbereitung des Robotersystems. Der Nachrichtenfluss zwischen den Subjekten Raum, Roboter, Scanner, Arm-Unit und Transport-Box,

² Thomas E. Jost, Christian Stary, Richard Heiningner (2022). Geo-spatial context provision for digital twin generation. Applied Sciences, 12(21), 10988.

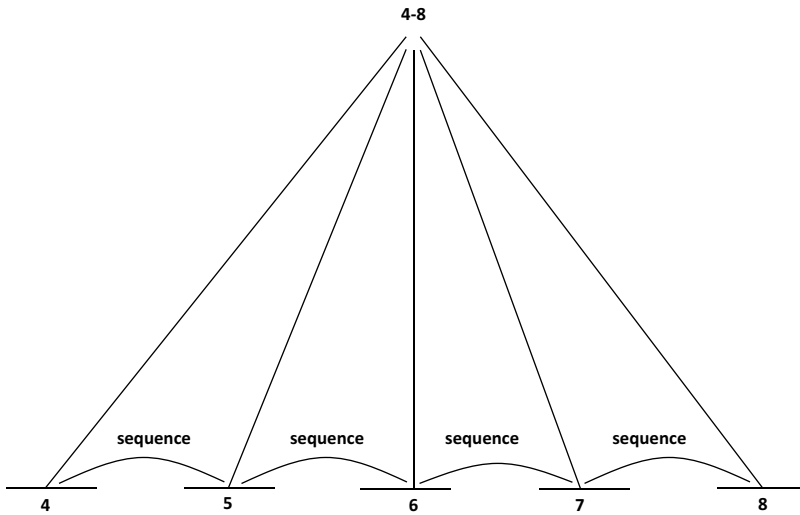


Abb. 8.4 Unterer Teil des RSTree für die Textbeschreibung des Verpackungsprozesses von Industrie 4.0

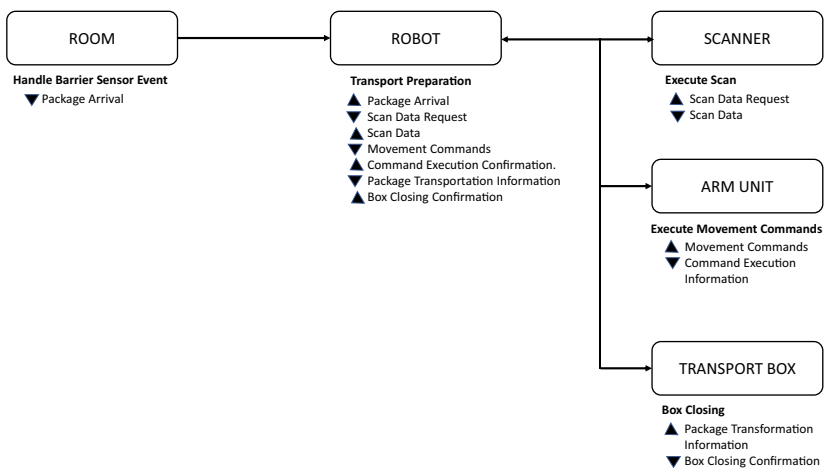


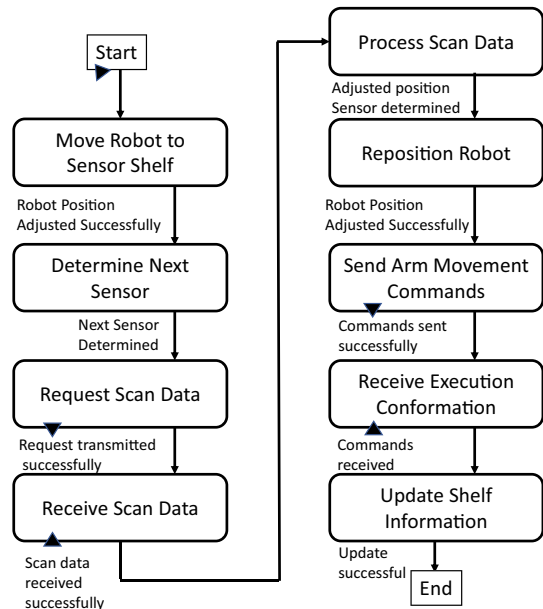
Abb. 8.5 Subjektspezifikationen (SID) der beteiligten Komponenten im Verpackungsanwendungsfall mit Interaktionssequenzen unter Verwendung der Compunity Suite (compunity.eu)

die Teil des Transportprozesses zum Packen sind, umfasst die Benachrichtigung der Packeinheit (Subjekt Raum), die das Verhalten des Transportroboters auslöst, und die Befehle des Roboters an seine Transportnutzlast (Subjekte Scanner, Arm, Transport-Box), um die in den „Blättern“ (Leave Node Entries) des RSTree angegebenen Aufgaben auszuführen.

Die Interaktion zwischen den für den Transport erforderlichen Gegenständen ist im Robotersystem zusammengefasst. Es erfasst die Bewegung des Roboters und die Fähigkeit, die Transportbox zu schließen, um Transportgüter zu sichern, als elementare Funktionen. Um das Verpacken an dem Ort durchzuführen, wo sich der Transportroboter bewegen kann, müssen die Interaktionsmuster den Empfang eines Pakets mit den medizinischen Warendaten bei der Ankunft, die Aktivierung des Scanners zum Identifizieren der zum Verpacken erforderlichen Sensoren und die Steuerung des Roboterarms zum Aufnehmen von Nutzlastartikeln und zum Abliefern dieser zum Verpacken umfassen.

Die folgende Abb. 8.6³ zeigt die Verhaltensspezifikation des Robotersystems für den Transport in der Verpackungseinheit beim Auswählen und Aufnehmen eines Sensors. Der Roboter beginnt, sich zum Lagerbereich der Verpackungseinheit zu bewegen. Beim Eintreffen am Sensorregal wählt er die Sensoren entsprechend der gescannten Informationen aus – die Sende- und Empfangszustände des SBD werden durch entsprechende Umschlagssymbole gekennzeichnet. Jeder der erforderlichen Sensoren wird von einem Roboterarm

Abb. 8.6 Verhaltensspezifikation eines Robotersystems, das beim Verpacken einen Sensor auswählt, unter Verwendung der Community Suite



³ Thomas E. Jost, Christian Stary, Richard Heining (2022). Geo-spatial context provision for digital twin generation. Applied Sciences, 12(21), 10988.

aufgenommen, um Teil der Nutzlast des Roboters zu werden. Der dargestellte Verhaltens- teil berücksichtigt nur den „Happy Path“, d. h. alle erforderlichen Sensoren sind verfügbar. Schließlich muss der Regalstatus nach dem Aufnehmen eines Sensors aktualisiert werden.

Subjektverhaltensspezifikationen dieser Art haben den Vorteil, dass sie nach der Vali- dierung automatisch ausgeführt werden können. Ein solcher verhaltenszentrierter Realitätscheck wird durch die Ausführung der Modelle aus jeder Rollenperspektive unterstützt, einschließlich der Interaktion mit anderen Rollen. In cyber-physischen und Mixed-Reality- Umgebungen können solche Modelle zur Steuerung von Systemen während des Entwurfs und der Laufzeit verwendet werden und als digitale Zwillinge dienen.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.





Nachdem die subjektorientierten Prozessmodelle erstellt wurden, sollen nun mit einer Plattform, also einem IT basierten Workflowsystem, die einzelnen Teilprozesse implementiert werden. Eine erfolgreiche Prozessimplementierung hängt nicht nur von der technischen Umsetzung ab, sondern wesentlich von den beteiligten Menschen. Da es sich, wie erwähnt, um ein soziotechnisches System handelt, ist dem entsprechend Rechnung zu tragen. Daraus sind folgende Empfehlungen für eine erfolgreiche Implementierung entstanden.

9.1 Organisation der Implementierung

Für die Implementierung der Teilprozesse, basierend auf dem subjektorientierten Prozessmodell, sind die Verantwortlichkeiten für die Implementierung klar festzulegen. Hierzu kann, wie in Kap. 7 beschrieben, das Netzwerk der Kommunikationsbeziehungen als Basis für die Work Break Down Structure des Projektplans verwendet und die Projekt-, resp. Teilprojektleitung zugewiesen werden. Dabei wird auch der Teilprozessleiter erwähnt, welcher für die Implementierung und wenn möglich auch für den Betrieb des Teilprozesses verantwortlich sein sollte. Im Idealfall ist dieser Teilprozessleiter bereits in der Prozessmodellierung involviert. So kann sichergestellt werden, dass betriebliche Aspekte bereits bei der Modellierung und bei der Implementierung berücksichtigt werden. Bei organisationsübergreifenden Prozessen kann es zu Konflikten zwischen dem Linienvorgesetzten und dem Teilprozessleiter kommen, da teilweise unterschiedliche Ziele verfolgt werden, andere Erwartungen vorhanden sind, oder die Verantwortlichkeiten unklar sind. Mit der Subjektorientierung können diese möglichen Konflikte einfacher identifiziert und somit gelöst werden. Wenn der Teilprozessleiter wie erwähnt bereits in der Prozessmodellierung beteiligt war, kann damit auch verhindert werden, dass die modellierten Prozessmodelle bzw. deren Sinn

während der Implementierung verändert werden. Diese Gefahr besteht durchaus, wenn nicht mehr das ganze System im Fokus steht, sondern nur der einzelne Teilprozess. Damit die Verantwortlichkeit für den (die) zu implementierenden Teilprozess(e) und insbesondere auch der Schnittstellen zu anderen Verantwortlichkeiten auch grafisch dargestellt wird, kann auch das Teilprozessmodell aus Abb. 4.4 im Implementierungsprojekt herangezogen werden.

Für eine reibungslose Prozessimplementierung ist ein aktives Change Management der involvierten Personen zu organisieren, um diese Personen auf die Veränderungen vorzubereiten und zu begleiten. Dazu beitragen kann auch das bekannte Vorgehen «betroffene zu involvierten Personen» zu machen. Das bedeutet, dass bereits in der Prozessmodellierung und in der Implementierung beteiligte Personen des Teilprozesses involviert werden sollten. Dies ist nicht für sämtliche am Teilprozess Beteiligte möglich, jedoch für einzelne Schlüsselpersonen, die wiederum die Informationen in die Fachbereiche tragen können und so automatisch die Rolle eines «Prozess Ambassador» übernehmen. Personen, welche von der Prozessmodellierung bis in die Implementierung und den Betrieb involviert sind, können die Gedankengänge, welche bei der Prozessmodellierung besprochen wurden, jedoch nicht in einem Prozessmodell bis ins Detail festgehalten werden können, in die Fachbereiche tragen. Um dies weiter zu unterstützen, ist es sinnvoll, einzelne Beweggründe für den gewählten Prozess bereits bei der Modellierung als Kommentar zu führen. Diese Kommentare können in der Umsetzung und in der Schulung hilfreich sein.

Ein weiterer Aspekt der Prozessimplementierung ist die Vorgehensweise. Dabei wird im Projektmanagement meist unterschieden zwischen der klassischen und der agilen Herangehensweise. Dies kann direkten Einfluss auf die Prozessmodellierung und Implementierung haben. Bei einer klassischen Vorgehensweise werden Prozessmodellierung und nachfolgend die Prozessimplementierung umgesetzt. In der agilen Methodik sind die Prozess Modellierung und die Implementierung aufeinander abzustimmen, sodass die einzelnen Zyklen (Sprints) zueinander aufbauend definiert und direkt umgesetzt werden.

9.2 Detaillierung für die Implementierung

Der zu implementierende Teilprozess ist weiter zu detaillieren hinsichtlich der Ausprägung von Subjekten, Objekten und Nachrichten zur Abbildung in einem Workflowsystem. Das Vorgehen wurde bereits im Kap. 6 beschrieben. Wie für die organisatorische Abgrenzung von Verantwortlichkeiten kann das Teilprozessmodell von Abb. 4.4 auch für die technische Abgrenzung eines Workflowsystems zu anderen Teilprozessen verwendet werden. Die Sicht auf die unterschiedlichen IT-Systeme zeigt das Subjekt-Interaktions-Diagramm (SID) von Abb. 5.6, woraus die notwendigen Schnittstellen ersichtlich sind und in der Implementierung definiert und realisiert werden müssen. Das kann wie beschrieben in Form der Nachricht und deren Nutzlast physisch oder elektronisch erfolgen, oder aber auch als technische Schnittstelle zwischen zwei IT-Systemen. Das Senden und Empfangen von Nachrichten innerhalb eines Teilprozesses werden üblicherweise vom Workflowsystem übernommen. Deren

Benennungen und Nutzlast sind jedoch im Workflowsystem abzubilden. Dabei empfiehlt es sich die Benennungen der Modellelemente für Subjekte, Objekte und Tätigkeiten aus der Prozessmodellierung (Empfehlung gemäß Abschn. 5.1) auch in der Prozessimplementierung im Workflowsystem zu verwenden. Konkret sollten die Elemente im Workflowsystem und im Prozessmodell die gleichen Benennungen aufweisen, wodurch der Bezug der subjektorientierten Modellierung in der Bearbeitung der Prozessinstanzen genutzt werden kann. So wird die Sprache innerhalb des Unternehmens zu einzelnen Teilprozessen und deren Inhalten einheitlich. Dies ist wichtig für die Schulung und den Bezug zwischen dem realen Prozess, dem Prozessmodell und dem IT-System.

Die Detaillierung der einzelnen Modellelemente sieht aus wie folgt:

- **Subjekt**

Anhand der Benennung des Subjekts in der subjektorientierten Modellierung leitet sich die aktive Einheit gemäß Abschn. 5.1 ab. Dabei ist zu beachten, dass in einem Workflowsystem idealerweise keine Personen, sondern Rollen, also eine Gruppe von Anwendern, oder Organisationsbereiche als Aufgabenträger hinterlegt werden. Dadurch können Änderungen der Aufbauorganisationen dynamisch in das Workflowsystem überführt werden. Ein weiterer wichtiger Punkt in der Implementierung ist die Definition von Stellvertretungen, welche auch im IT-System abgebildet werden sollten, nur so kann ein reibungsloser Prozessverlauf sichergestellt werden.
- **Objekt**

Wie beschrieben können Objekte physisch oder als Datensatz vorhanden sein. In beiden Fällen sind die Beschaffenheit der Objekte zu definieren und welche Aktion ausgeführt wird. Das beinhaltet in einem Workflowsystem mit integriertem Objektbezug die Beschaffenheit des Objekts vor und nach der ausgeführten Aktion. Dies stellt die Grundlage für die gewünschte Automatisierung durch das IT-System dar, oder bildet die Basis für die Realisierung von notwendigen Schnittstellen. Auch für physische Objekte gilt es die Beschaffenheit zu definieren. Häufig werden die Zustände des physischen Objekts auch im Datensatz dokumentiert. Dabei gilt es sicherzustellen, dass der Datensatz mit dem physischen Objekt übereinstimmt. Das ist in einem integrierten System wie z. B. einem SAP-System einfach möglich, weil der Workflow und der zum physischen Objekt gehörende Datensatz in einem System abgebildet werden.
- **Kommunikation**

Im Falle eines Workflowsystems wird die Kommunikation durch das IT-System bewerkstelligt. Die Kommunikationsstruktur ist vom Prozessmodell zu übernehmen und daraus sind die genauen Nachrichten abzuleiten, d. h. welche Texte werden dem Anwender bereitgestellt, in welchen Sprachen sind die Texte verfügbar, welche Texte oder Textbausteine sind dynamisch und leiten sich gegebenenfalls vom Objekt ab? Diese Fragen müssen beantwortet und im Workflowsystem hinterlegt werden.

Spezielles Augenmerk bei der Implementierung eines Workflowsystems ist der Ausnahmebehandlung gemäß Abschn. 4.2.3 zu widmen. Solche Ausnahmebehandlungen wie z. B. der Abbruch, führen in den meisten Fällen zu Auswirkungen auf die Objekte, also auf den Datensatz oder das physische Objekt, das abhängig von der eingetretenen Ausnahmebehandlung weiter zu bearbeiten ist.

9.3 Schulung und Testen

Eine große Bedeutung bei der Implementierung ist der Schulung des Prozesses, dem Workflowsystems und dem Testen einzuräumen. Für die Schulung können die erstellten subjektorientierten Modelle direkt verwendet werden. Wie bereits beschrieben basieren diese auf der natürlichen Sprache und sind somit einfach verständlich. Wie in Abschn. 3.1 aufgeführt, werden Geschäftsprozesse oder Probleme in der natürlichen Sprache beschrieben und bilden damit eine gute Basis für die Schulungsdokumentation. Weiter gilt es die Prozessbeteiligten im Workflowsystem auszubilden. Es ist also nicht nur das «WIE» in Form einer Anwenderschulung am IT-System durchzuführen, sondern idealerweise wird integriert auch das «WARUM» vermittelt, also auch die Prozessziele und der Prozesszweck. Dabei können auch die Kommunikationspartner und deren Aufgaben innerhalb des Teilprozesses oder von vor-/nachgelagerten Teilprozessen vermittelt werden.

Bevor das Workflowsystem und somit der neue oder angepasste Prozess produktiv genutzt werden können, sind umfangreiche Tests notwendig. Hierbei sollte der Fokus nicht nur den technischen Test des Workflowsystems gelegt werden, sondern auch der Prozess und deren Anwendbarkeit sollten intensiv betrachtet werden. Es sollten also auch z. B. folgende Fragen erfolgreich beantwortet werden können:

- Werden die Prozessziele erreicht?
- Sind die Prozessbeteiligten vollständig und korrekt zugeordnet?
- Entspricht der modellierte Prozess dem im Workflowsystem implementierten Prozess?
- Sind die Schnittstellen zu Teilsystemen und Teilprozessen vollständig (Siehe auch Abschn. 4.2 Externe Subjekte)?
- Können die Ausnahmebehandlungen angewendet werden (Siehe auch Abschn. 4.2.3)?

Wie bereits bei der Schulung, sollten beim Testen nicht nur die technischen Aspekte im Fokus stehen, sondern auch der Prozess selbst. Dabei werden viele Fragen der Prozessbeteiligten gestellt, die es zu beantworten gilt und im Sinne des aktiven Change Managements die Anwender auf den neuen Prozess und deren Aufgaben vorbereitet.

9.4 Betrieb und Optimierung der Prozesse

Es wurde beschrieben, wie neue Prozesse oder auch Änderungen definiert und realisiert werden können. Um die Prozesse ständig zu verbessern und auf ständige Änderungen des Umfeldes reagieren zu können, ist ebenfalls ein Prozess für die Anpassung von Prozessen zu definieren. Dieser Prozess kann in einem Unternehmen für sämtliche Prozesse oder für Teile des Unternehmens identisch sein und ist häufig im Rahmen des kontinuierlichen Verbesserungsprozesses bereits vorhanden. Optimierungswünsche, neue Anforderungen usw. sollten systematisch erfasst und bewertet werden. Bei der eigentlichen Optimierung des Prozesses sind die gleichen Schritte wie bei einem neu definierten Prozess zu durchlaufen. So kann auch sichergestellt werden, dass das Prozessmodell angepasst wird, der Prozesszweck nicht grundlegend verändert wird und evtl. betroffene Teilprozesse involviert werden.

Im laufenden Betrieb von Prozessen werden häufig auch Kennzahlen zum Prozess erwartet, wie z. B. die Anzahl abgeschlossener Prozessinstanzen, durchschnittliche oder maximale Durchlaufzeiten der Prozessinstanzen, Anzahl der Ausnahmebehandlungen, prozessabhängige Kennzahlen wie z. B. Kosten, Umsatz, Anzahl eingeholter Angebote, usw. Für eine einheitliche Sicht auf diese Kennzahlen aller Prozessbeteiligten und -verantwortlichen, sind diese frühzeitig zu definieren, damit diese auch während der Ausführung der Prozesse verfügbar sind. Diese Kennzahlen können wiederum Anhaltspunkte auf Optimierungspotential des Prozesses geben und liegen deshalb ebenfalls in der Verantwortung des Teilprozessleiters.

Der Überwachung der laufenden Prozessinstanzen ist bereits bei der Prozessimplementierung Beachtung zu schenken. Dabei geht es nicht um die Überwachung der einzelnen Prozessbeteiligten, sondern darum, zu erkennen, sobald Prozessinstanzen nicht wie definiert ablaufen korrigierend eingreifen zu können. Hierfür bieten Workflowsysteme entsprechende Reports. Jedoch ist deren Nutzung in der Prozessimplementierung zu definieren, um die Vergleichbarkeit der Auswertungen zu ermöglichen. Verantwortlich für die Überwachung der laufenden Prozessinstanzen sollte wiederum der Teilprozessleiter sein.

Wenn während der Prozessimplementierung neue Erkenntnisse entdeckt werden, die Einflüsse auf das Prozessmodell haben, sind diese nicht leichtfertig zu implementieren, sondern müssen über einen bewussten Entscheid in das Prozessmodell eingearbeitet und danach in die Prozessimplementierung übernommen werden. Es ist wichtig, dass sämtliche Anpassungen bewusst entschieden werden und in Form einer Änderung in das Prozessmodell zu übernehmen und diese Änderungen somit auch zu dokumentieren sind. Dieser Änderungsprozess ist für die Prozessmodellierung und Prozessimplementierung zu definieren und ist zwingend einzuhalten.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.





Die Subjektorientierung schlägt nicht nur die Brücke zwischen natürlicher Sprache und ausführbaren Prozessmodellen, sie erlaubt vielmehr die gleichwertige Berücksichtigung von Personen, Organisationen und Technik als kommunizierende Einheiten, und somit eine ganzheitliche System(entwicklungs)sicht.

10.1 Subjektorientiertes Denkmodell

Subjektorientierung ist sowohl ein Denkmodell als auch ein Praxiswerkzeug. Das Denkmodell Subjektorientierung beruht auf drei Grundsätzen.

Grundsatz Nummer eins lautet, Organisation ist Kommunikation. Dies entspricht der Organisationstheorie des Soziologen Niklas Luhmann, der die kleinste Einheit der Organisation als Kommunikation bezeichnete. Die Kommunikation findet dabei zwischen zwei informationsverarbeitenden Prozessoren statt. Wohlgedacht, diese Aussage kommt von einem Soziologen der bereits 1998 verstorben ist und nicht von einem Informatiker. Grundsatz zwei lautet, Geschäftsprozesse werden in ganzen Sätzen beschrieben. Dies bedeutet es wird die Frage geklärt, *wer*, das Subjekt, führt *welche Aktion*, das Prädikat, *auf oder mit welchem Objekt* durch. Bei mehreren Subjekten wird darüber hinaus gefragt *welches Subjekt sendet welche Nachrichten* an welches andere Subjekt bzw. *empfängt welche Nachrichten* von welchem Subjekt.

Damit wird ein Prozessmodell nach dem gleichen Grundmuster wie in natürlichen Sprachen beschrieben. Bei den Prädikaten Senden und Empfangen gibt es mit der Nachricht ein direktes Objekt und mit dem Empfänger bzw. Sender ein indirektes Objekt. Der dritte Grundsatz ist dass die Methode durch entsprechende Formalismen aus der Informatik präzise beschrieben ist.

Die Praxistauglichkeit des subjektorientierten Ansatzes ergibt sich aus den geschilderten Grundsätzen. Wenn einer Person erklärt wird, wie sie zusammen mit anderen etwas organisieren soll, dann bringt der Verantwortliche ihr dies in der Form näher, was sie oder die Partner tun sollen und wie sie ihre Tätigkeiten durch entsprechenden Nachrichtenaustausch, sei es per Mail, Telefon oder direktes Miteinandersprechen, synchronisieren. Die Beschreibung der Prozesse entspricht damit der Grundstruktur der natürlichen Sprache und kommt mit Senden, Empfangen und Tun aus. Durch diese drei Satzmuster und der präzisen Definition der Semantik können Prozessbeschreibungen vollautomatisch in ein ausführbares Programm umgewandelt werden.

10.2 Weiterführende Aspekte der Subjektorientierung

In den vorausgegangenen Kapiteln wurde der Kern einer Sprache zur Transformation eines in natürlicher Sprache beschriebenen Vorgangs in ein Programm beschrieben. Dieser Subjektorientierte Modellierungsansatz unterstützt auch den Einsatz verschiedener Techniken, Technologien und Methoden, in unterschiedlichen Anwendungs- und Betriebswirtschaftsbereichen wie der Prozesskostenrechnung. Weitere Bereiche in denen das Subjektorientierte Denkmodell hineinwirkt finden sich in Abb. 10.1.

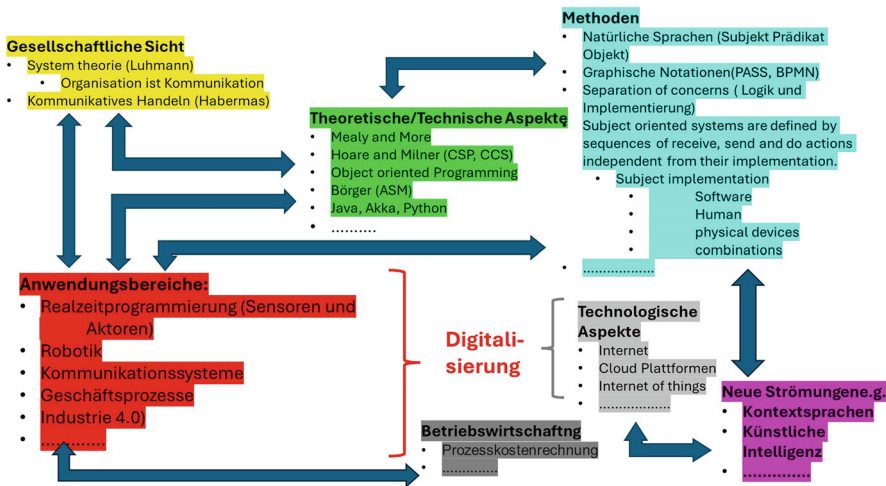


Abb. 10.1 Mögliche Entwicklungsströmungen für die Subjektorientierung

Ein wesentlicher neuer Aspekt ist die Betrachtung von Subjekten, die KI-Anwendungen enthalten. Die Ergebnisse dieser in einem Subjekt gekapselten KI-Anwendungen können über Nachrichten zwischen den jeweiligen Subjekten ausgetauscht werden, so dass ein Netzwerk von unterschiedlichen KI-Anwendungen entsteht, um auch hochkomplexe Probleme zu lösen¹.

¹ Dieses Literaturverzeichnis enthält weiterführende Literatur zum Denkmodell der Subjektorientierung.

Weitere Literatur zu speziellen Aspekten finden sich auf der Internetseite www.i2pm.net.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.



Literatur

1. ELSTERMANN, M. ; FLEISCHMANN, A. ; MOSER, C. ; OPPL, S. ; SCHMIDT, W. ; STARY, C. : *Ganzheitliche Digitalisierung von Prozessen: Perspektivenwechsel – Design Thinking – Werteleitete Interaktion*. Berlin, Heidelberg : Springer, 2023
2. FLEISCHMANN, A. ; RASS, S. ; SINGER, R. : *S-BPM Illustrated: A Storybook about Business Process Modeling and Execution*. Berlin, Heidelberg : Springer, 2013
3. FLEISCHMANN, A. ; SCHMIDT, W. ; STARY, C. : *S-BPM in the Wild – Practical Value Creation*. Berlin, Heidelberg : Springer, 2015
4. FLEISCHMANN, A. ; SCHMIDT, W. ; STARY, C. ; OBERMEIER, S. ; BÖRGER, E. : *Subject-Oriented Business Process Management*. Berlin, Heidelberg : Springer, 2012
5. FLEISCHMANN, A. ; SCHMIDT, W. ; STARY, C. ; OBERMEIER, S. ; BÖRGER, E. : *Subjektorientiertes Prozessmanagement: Mitarbeiter einbinden, Motivation und Prozessakzeptanz steigern*. München : Hanser Verlag, 2011

Stichwortverzeichnis

A

Aufgabenträger, 43
Ausnahmebehandlung, 30

B

Benennung, 35

C

Choice Segment, 27

D

Daten, 16

E

Einbettung, 45

G

Gestaltung, soziotechnische, 65
Guard, 30

H

Handlungsempfehlungen, 65

I

Implementierung, 43
Input Pool, 10, 13, 46
Restriction, 10, 13, 46

Instanz, 43

Interface Subjekt, 19, 21, 22
Interrupt, 30

K

Kommunikationsstruktur, 9
Kontext, 43, 45
Konvention, 35

L

Layout, 36, 39
Leitprinzipien, 65

M

Makro, 26
Message, 36
Multiprozess, 24
Multisubjekt, 24, 44

N

Nachricht, 36
Name, 35

O

Objekt, 16
Struktur, 16

P

Projektmanagement, 65

Prozesshierarchie, [24](#)

Prozessmodell, [43](#)

Prozessmodellhierarchie, [24](#)

Prozesssystem, [21](#)

R

Rhetorical Structure Theory, [57](#)

S

Storytelling, [57](#)

Subject, [35](#)

Subject Behavior Diagram (SBD), [13](#), [26](#), [36](#)

Subject Carrier, [43](#), [44](#)

Subject Interaction Diagram (SID), [9](#), [19](#), [21](#),
[35](#), [36](#), [39](#)

Subjekt, [24](#)

externes, [19](#), [21](#)

Instanz, [44](#)

U

Untermodell, [22](#)

Unterprozess, [26](#)

V

Verhalten, [13](#)

Verhaltensdiagramm, [13](#), [26](#)