

Fehlervermeidung und Prozessmonitoring
in komplexen und dynamischen
Großschadenslagen

DISSERTATION

zur Erlangung des akademischen Grades
doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat der Fakultät für Mathematik
und Informatik der Friedrich-Schiller-Universität Jena

von Dipl.-Inf. UWE KRÜGER
geboren am 29. Mai 1972 in Eisenberg



FRIEDRICH-SCHILLER-UNIVERSITÄT JENA
FAKULTÄT FÜR MATHEMATIK UND INFORMATIK
INSTITUT FÜR INFORMATIK

Gutachter:

1. Prof. Dr. CLEMENS BECKSTEIN, Praktische Informatik (Künstliche Intelligenz),
Institut für Informatik, Friedrich-Schiller-Universität Jena
2. Prof. Dr. STEFAN STROHSCHNEIDER, Forschungsstelle interkulturelle und komplexe
Arbeitswelten (FinkA), Fachgebiet Interkulturelle Wirtschaftskommunikation (IWK),
Friedrich-Schiller-Universität Jena

Tag der öffentlichen Verteidigung: 26. August 2014

Kurzfassung

In dieser Arbeit wird der Frage nachgegangen, inwieweit eine Computerunterstützung in komplexen, kritischen Arbeitsbereichen zu einer Harmonisierung der Handlungsabläufe und zur Verhinderung von Fehlern (z. B. Gedächtnisfehlern) beitragen kann. Als Beispiel eines komplexen und dynamischen Arbeitsbereiches wird das Agieren von *Behörden und Organisationen mit Sicherheitsaufgaben* (BOS) in Großschadenslagen betrachtet. Motiviert durch den großen Erfolg von Checklisten in der Luftfahrt und in der Intensivmedizin, wird in dieser Arbeit eine systematische Übertragung des Checklisten-Prinzips auf den deutschen BOS-Bereich vorgeschlagen. Hierfür wird ein Rahmenwerk eines intelligenten elektronischen Checklisten-Assistenzsystems erarbeitet, das auch für alternative, soziotechnische Arbeitssysteme mit dem BOS-Bereich ähnlichen Merkmalen hilfreich zur Fehlervermeidung eingesetzt werden kann.

Inhaltsverzeichnis

Kurzfassung	i
1 Einleitung	1
1.1 Motivation	1
1.2 Aufteilung der Arbeit	3
2 Akteure und Handlungsrouinen in Großschadenslagen	5
2.1 Behörden und Organisationen mit Sicherheitsaufgaben	5
2.2 Großschadensereignisse und deren Charakteristiken	8
2.3 Einsatzplanung und Einsatzvorbereitung	11
2.4 Handlungsrouinen und IT-Unterstützung	13
3 Stand der Forschung zur IT-Unterstützung für SOPs	17
3.1 Forschungsperspektiven zur SOP-Unterstützung	17
3.2 Business Process und Workflow Management	18
3.3 Ansätze der automatische Handlungsplanung	21
3.4 Fazit und Motivation des Checklisten-Ansatzes	27
4 Checklisten – eine Human Factors Perspektive	29
4.1 Strategien der Fehlervermeidung & Handlungsassistenz	29
4.2 Checklisten als kognitives Hilfsmittel	31
4.2.1 Was sind Checklisten? – eine Arbeitsdefinition	32
4.2.2 Funktionen, Einsatzbereiche und Zielgruppen	33
4.2.3 Items und Anwendungsmethoden	36
4.2.4 Repräsentationsformen von Checklisten	38
4.2.5 Nachteile und Akzeptanzproblematik	39
4.3 Unterscheidung der Begriffe Checkliste und SOP	40

5	Intelligente elektronische Checklisten-Assistenz	43
5.1	Motivation & Anforderungen einer elektronischen Checklisten-Assistenz . . .	43
5.2	Grundlegende Konzepte und Designentscheidungen	48
5.2.1	Rollen und Rollenübernahme in einem Einsatz	48
5.2.2	Charakteristische Merkmale von Checklisten und SOPs	51
5.2.3	Komplexität handhaben durch Hierarchien	57
5.2.4	HTN-Planung und Theorien der Handlungsregulation	64
5.2.5	Checklisten-Auswahlprozess und -Zuordnung	69
5.2.6	Standardisiertes Vokabular	71
6	Technische Realisierung	75
6.1	Komponenten der CASIE-Architektur	75
6.1.1	CL-Client, Benutzerschnittstelle und Endgeräte	77
6.1.2	CL-Manager	79
6.1.3	CL-Repository & -Logbook	80
6.1.4	Knowledge-Base (System) / BOS-Ontologie	81
6.2	Dynamik / reaktives Systemverhalten	88
6.2.1	Zustandsübergänge einer Checkliste und deren Items	88
6.2.2	(Intelligentes) reaktives Systemverhalten	90
6.3	Konfiguration/Wartung	94
6.3.1	Entwicklung einer BOS-Ontologie	95
6.3.2	Entwicklung von ICLs	103
7	Vorteile der CASIE-Anwendung in der Praxis	107
7.1	Beitrag zur Fehlervermeidung	107
7.2	Prozessmonitoring	108
7.3	Zusätzliche Vorteile und Ausblick	109
8	Zusammenfassung	117
	Literaturverzeichnis und Photonachweise	122
	Abbildungsverzeichnis	135

Kapitel 1

Einleitung

„Die Menschen stolpern nicht über Berge,
sondern über Maulwurfshügel.“ (Konfuzius)

1.1 Motivation

Wir Menschen sind zu großen geistigen Leistungen fähig, jedoch nicht unfehlbar in dem, was wir tun. So stoßen wir zum Beispiel beim Versuch, alle Schritte einer komplexen und ungewohnten Handlungssequenz korrekt auszuführen, in den Situationen an unsere Grenzen, in denen wir zusätzlich hohem physischen und psychischen Stress ausgesetzt sind. Denkblockaden und Vergesslichkeit sind die Folge. Dem steht eine Zunahme an Komplexität in vielen Arbeitsbereichen gegenüber, wodurch sich das Risiko menschlicher Fehler erhöht [Rea94]. Vor allem in dynamischen Ad-hoc-Arbeitsumgebungen, in denen Teams unterschiedlicher Profession an der Abarbeitung komplexer und verteilter Aufgabenstellungen arbeiten, stellt eine fehlerfreie Abarbeitung und Koordinierung aller dabei gebotenen Handlungen eine besondere Herausforderung dar. Das Agieren von *Behörden und Organisationen mit Sicherheitsaufgaben* (BOS) in Großschadenslagen ist ein solch komplexer und zugleich dynamischer Arbeitsbereich. Er enthält eine Vielzahl kritischer Handlungsabläufe, bei denen menschliche Fehler enorme Auswirkungen haben können. Unter Komplexität eines solchen sog. soziotechnischen Arbeitssystems (vgl. [BHL08], S. 157 ff.) werden die u. a. von SIMON [Sim62] und DÖRNER [Dör75] charakterisierten Merkmale wie *Anzahl* der Aufgaben und Akteure, *Vielfalt* der möglichen Handlungsausprägungen und die *Vernetztheit* verstanden (vgl. MANSER [Man09] und SCHAUB [Sch06]). Die *Dynamik*, *Intransparenz* und *Unbestimmtheit* sind weitere charakteristische Merkmale eines solchen Arbeitsumfeldes.

In Großschadenslagen unterliegen Einsatzkräfte in besonderem Maße großem physischen und psychischen Stress. Es ist daher nicht zu erwarten, dass alle erlernten Arbeitsabläufe korrekt erinnert und im Ernstfall fehlerfrei durchgeführt werden [Rea90; BHL08]. Schätzungen gehen davon aus, dass gerade bei besonders großen Einsatzlagen, mit Dutzenden von Schwerverletzten, bis zu ein Drittel der Verletzten wegen Missmanagement am Einsatzort und in der Klinik unnötig versterben (vgl. [KHK+06]). Auch wenn große Verletztenzahlen bei nur äußerst selten eintretenden Ereignissen wie z. B. Terroranschlägen oder Naturkatastrophen zu verzeichnen sind, offenbaren die Auswertungen bereits kleinerer Einsätze und Übungen nicht selten eine Diskrepanz zwischen der A-priori-Vorstellung dessen, wie ein Schadensereignis zu managen ist („best practice“) und dem tatsächlichen Verlauf. Insbesondere wird das Management in Großschadenslagen zusätzlich dadurch erschwert, dass

eine große Anzahl von freiwilligen Einsatzkräften beteiligt sind, welche weniger routiniert in dem fehlerfreien Anwenden von großschadensspezifischen Handlungsabläufen sind.

Aus Eigeninitiative der BOS und aus Interesse von Forschungseinrichtungen und Ländern untersuchen diese in jüngster Zeit verstärkt den Einsatz von Computerunterstützung im Krisenmanagement und bei der Abarbeitung von Großschadenslagen. Ein Beispiel hierfür ist der Forschungsschwerpunkt „Schutz und Rettung von Menschenleben“ im Rahmen des Sicherheitsforschungsprogramms der Deutschen Bundesregierung, in dem eine Vielzahl von Projekten (siehe [BMBF]) die Einsatzmöglichkeiten von IT in Großschadenslagen untersucht. Während sich viele dieser Projekte auf eine robuste Kommunikation zwischen den Einsatzkräften, eine Telemetrieunterstützung, eine Bereitstellung eines allumfassenden Informationssystems oder auf eine Kartendarstellung des Einsatzortes fokussieren, bleibt eine praktikable Assistenz von Handlungsabläufen bisher noch weitgehend unberücksichtigt. Auch haben die auf internationaler Ebene in den letzten 20 Jahren vorgestellten Lösungsansätze zur IT-Unterstützung im Krisenmanagement keinen Niederschlag in die heutige (deutsche) BOS-Praxis gefunden. Ein Grund hierfür liegt darin, dass der BOS-Bereich (im Vergleich zur Industrie oder speziell dem Militär) aktuell erst im Begriff ist, die Möglichkeiten eines Computereinsatzes am Einsatzort auszuloten und vereinzelt zu testen.

Diese Arbeit geht der Frage nach, inwieweit eine Computerunterstützung in komplexen, kritischen Arbeitsbereichen zu einer Harmonisierung der Handlungsabläufe und zur Verhinderung von Fehlern (z. B. Gedächtnisfehlern) beitragen kann. Als Beispiel wird der Arbeitsbereich der deutschen BOS betrachtet, dessen besondere Merkmale und Anforderungen herausgearbeitet werden. Maßgeblich für die Wahl des hier vorgestellten Lösungsansatzes ist die Beobachtung, dass ein vermeintlich simpler und im BOS-Bereich momentan wenig beachteter Kontrollmechanismus dazu dienen kann, menschliches Handeln fehlerfreier und konsistenter bezüglich Standardhandlungsabläufen werden zu lassen. Die Rede ist von *Checklisten* (CL), deren Anwendung mittlerweile eine Erfolgsgeschichte in vielen Hochsicherheitsbereichen wie etwa in der Luftfahrt [DW90] und der Intensivmedizin [Gaw10] verzeichnen kann. So entwickelte z. B. die Luftfahrt bereits Ende der 1960er Jahre eine gesunde Ehrfurcht gegenüber ihren immer komplexer werdenden Flugzeugen und den dazugehörigen Arbeitsprozessen und entschied, Checklisten als kognitives Hilfsmittel dem menschlichen Geist bei der täglichen Arbeit an die Seite zu stellen. Heutzutage sind sie ein fest etablierter Bestandteil aller sicherheitskritischen Prozeduren im Cockpit (siehe z. B. [DW90; DW93]). Warum also nicht von diesem positiven Beispiel lernen?

Basierend auf diesen Erfahrungen des Checklisten-Einsatzes wird in dieser Arbeit eine systematische Übertragung des Checklisten-Prinzips auf den deutschen BOS-Bereich vorgeschlagen. Hierzu wird nach einer Analyse der Praxisbedingungen und einer systematischen Betrachtung des Checklisten-Prinzips selbst ein Rahmenwerk für ein *intelligentes elektronisches Checklisten-Assistenzsystem* erarbeitet. Die Arbeit adressiert noch keine Implementierung und Evaluation unter Praxisbedingungen. Vielmehr gibt sie einen Anstoß für einen zukünftigen Transfer des Checklisten-Prinzips in den BOS-Bereich. Sie soll interessierten Anwendern als Grundlage für kommende Implementierungen dienen. Die Arbeit diskutiert darüber hinaus die zu erwartenden Vorteile einer computerverarbeitbaren Lagerepräsentation mittels einer formalen Wissensbasis. Das hierfür erarbeitete Rahmenwerk ist prinzipiell domänenunabhängig und auf alternative soziotechnische Arbeitssysteme mit hohen Sicherheitsanforderungen und ähnlichen Merkmalen wie die im BOS-Bereich übertragbar.

Die Arbeit gibt einen Lösungsvorschlag zur Fragestellung, wie in der komplexen und dynamischen Arbeitsumgebung einer Großschadenslage ein Beitrag zur Fehlervermeidung geleistet werden kann. Die Arbeit geht dabei von folgender Hauptthese aus:

These 1

Eine IT-unterstützte Umsetzung des Checklisten-Prinzips ist im BOS-Bereich besonders geeignet, um einen signifikanten Beitrag zur **Fehlervermeidung** und besseren Handhabung von Komplexität im Management einer Großschadenslage zu leisten.

Zusätzlich wird gezeigt, wie ein Checklisten-Assistenzsystem einen Beitrag zum Aufbau eines einheitlichen Lagebildes leisten kann:

These 2

Die Anwendung vernetzter computerintegrierter Checklisten kann für ein **Prozessmonitoring** genutzt werden, welches zusätzlich dem **Aufbau eines Lagebildes** dient und somit zu einem einheitlichen Lageverständnis (Situation Awareness) führt.

Darüber hinaus weist das vorgestellte Rahmenwerk eine Vielzahl von Zusatzeffekten auf:

These 3

Eine elektronische Checklisten-Assistenz kann zu einer **qualitativen Verbesserung der gesamten Einsatzabwicklung** beitragen, die weit über die positiven Effekte der Fehlervermeidung und des Prozessmonitorings hinausreicht.

Der Grundansatz dieser Arbeit basiert auf Erkenntnissen und Arbeiten aus dem BMBF Forschungsprojekt SPEEDUP¹ (siehe [KGK+10; WYM+11; WKK11; KWB12]), im Zuge dessen bereits Fragen und Lösungsansätze einer möglichen IT-Assistenz im BOS-Bereich bearbeitet wurden. Darüber hinaus wurden die Wahl und die Umsetzung des Lösungsansatzes durch Erkenntnisse der *Human-Factors*-Forschung beeinflusst, die sich unter anderem mit Phänomenen menschlicher Handlungen in komplexen Arbeitswelten beschäftigt. Nicht zuletzt flossen frühere berufliche Erfahrungen² des Autors mit in die Problemanalyse und die Konzeption eines vor allem praktikablen Lösungsansatzes ein.

1.2 Aufteilung der Arbeit

Das **Kapitel 2** erläutert den Arbeitsbereich der deutschen BOS. Hierfür werden zu Beginn die wichtigsten Organisationen vorgestellt und deren typische Einsatzvorbereitung und -koordination skizziert. Es wird auf besondere Einsatzlagen und deren Spezifika eingegangen sowie die Grundlagen der Führungsorganisationen, die Einsatzdynamik und die Charakteristiken der Einsatzphasen aufgezeigt. Darüber hinaus werden die in der Praxis üblichen vorbereitenden Maßnahmen sowie Möglichkeiten der Erfassung und Standardisierung von Handlungsrouninen betrachtet. Es wird der Frage nachgegangen, welchen Stellenwert das Auftreten möglicher Fehler in der Organisationskultur der BOS hat und welche

¹ Titel: „Untersuchung von mobilen und selbstorganisierenden Kommunikations- und Datenplattformen sowie Organisations- und Handlungsstrategien für komplexe Großlagen“. (Mai '09 - Juli '12)

² Der Autor war 6 Jahre im aktiven Rettungsdienst und als Disponent einer Rettungsleitstelle tätig.

Strategien zur Fehlervermeidung heute in der Praxis angewandt werden. Abschließend wird auf den derzeitigen Einsatz von IT im Einsatzmanagement eingegangen.

In **Kapitel 3** werden bisherige Forschungsarbeiten zur Thematik IT-Einsatz zur Prozessfassung und -unterstützung in Krisenszenarien betrachtet. Hierbei werden zwei Herangehensweisen vorgestellt: Arbeiten aus dem Umfeld der Geschäftsprozessmodellierung und Arbeiten aus dem Themenumfeld der automatischen Handlungsplanung der Künstlichen Intelligenz. Zusätzlich werden für die Problemstellung relevante Ergebnisse des Forschungsbereiches Human Factors studiert, welche sich unter anderem mit der Thematik Fehlervermeidung befassen.

Das **Kapitel 4** gibt einen Überblick über wichtige Aspekte des Checklisten-Prinzips und geht auf die Anwendungsprinzipien in kritischen Arbeitsbereichen ein. Hierfür werden die positiven Erfahrungen aus Hochsicherheitsbereichen wie Luftfahrt und Medizin analysiert und Grundprinzipien hinsichtlich Funktionen und Anwendungen herausgearbeitet. Weiterhin wird geklärt, welche Grundbestandteile Checklisten aufweisen und *wer* sie *wann* und *wie* einsetzt. Das Kapitel zeigt darüber hinaus, welchen Problemen Checklisten entgegenwirken sollen und welche bekannten Nachteile sie haben.

In **Kapitel 5** wird ein Rahmenwerk eines intelligenten elektronischen Checklisten-Assistenzsystems vorgestellt, das jeder BOS (und Organisationen mit ähnlichem Arbeitsumfeld) eine Hilfestellung bei der Sicherstellung fehlerfreier Anwendung von Standardhandlungsroutinen ermöglicht. Hierzu wird zu Beginn die Lösungsidee motiviert und daraus abgeleitet die drei Thesen der Arbeit aufgestellt. Weiterhin werden die Anforderungen an ein elektronisches Checklisten-Assistenzsystem benannt und dessen grundlegende Konzepte und Designentscheidungen vorgestellt.

Im **Kapitel 6** wird eine zu den erarbeiteten Anforderungen passende Architektur vorgestellt und die Dynamik des Assistenzsystems erläutert. Neben der Vorstellung der Architekturkomponenten wird im Besonderen die Wissensbasis als die ausschlaggebende Komponente für ein intelligentes Systemverhalten behandelt. Auf die Fragen der richtigen Konfiguration eines solchen Systems geht der Schluss des Kapitels ein.

Das **Kapitel 7** geht auf die Pragmatik und die sich zusätzlich für die Praxis ergebenden Anwendungsmöglichkeiten ein. Es wird eine Reihe positiver Konsequenzen der Anwendung eines auf der CASIE-Architektur basierenden Assistenzsystems vorgestellt, die neben der gewünschten Fehlervermeidung und dem Prozessmonitoring einen qualitativen Sprung im Management einer Großschadenslage darstellt.

Abschließend zieht **Kapitel 8** ein Resümee der Arbeit und fasst das Erreichte zusammen. Darüber hinaus gibt das Kapitel einen Ausblick auf zukünftige Aufgabenstellungen und nennt noch ungelöste Problemfelder.

Kapitel 2

Akteure und Handlungsrouninen in Großschadenslagen

Dieses Kapitel skizziert den komplexen und dynamischen Arbeitsbereich (ausgewählter) deutscher *Behörden und Organisationen mit Sicherheitsaufgaben* (BOS). Es werden hierzu Charakteristiken besonderer Einsatzlagen und die sich daraus für die Einsatzkräfte ergebenden Schwierigkeiten vorgestellt. Weiterhin wird kurz auf den organisationsinternen Umgang mit Fehlern eingegangen und der heutige Stand der IT-Unterstützung mit dem Blick auf die Standardhandlungsroutinen diskutiert.

Eine detaillierte Einführung in die Thematik kann und soll dieser Abschnitt nicht bieten. Für weitere Informationen sei der interessierte Leser auf die entsprechende Fachliteratur verwiesen. Als Einstieg empfiehlt sich z. B. die SEGmente Reihe des Stumpf + Kossendey Verlags (siehe z. B. [PM01]) und die Abhandlung von LIPP [Lip09].

2.1 Behörden und Organisationen mit Sicherheitsaufgaben

Bevor auf die Charakteristiken von Großschadenslagen eingegangen wird, sollen die beteiligten Akteure mit ihren unterschiedlichen Aufgabenbereichen vorgestellt werden. In Deutschland zählen die Feuerwehr, der Rettungsdienst und die Polizei mit zu den wichtigsten BOS. Zusammen mit weiteren Organisationen decken sie in der Regel unterschiedliche Aufgabenbereiche ab.

Feuerwehr

Im Rahmen der nichtpolizeilichen Gefahrenabwehr übernimmt die Feuerwehr Aufgaben aus den Bereichen *Retten, Löschen, Bergen* und *Schützen*, die durch Gesetze der jeweiligen Bundesländer und Kommunen geregelt sind. Laut dem Bundesministerium des Inneren (BMI) hat die Feuerwehr die Aufgabe, innerhalb des deutschen Notfallvorsorgesystems bei Bränden, Unfällen, Überschwemmungen und ähnlichen Ereignissen Hilfe zu leisten.

Obwohl jede größere Stadt typischerweise eine sog. Berufsfeuerwehr vorweist, deren Einsatzkräfte meist Beamte der jeweiligen Kommune sind, bilden die freiwilligen Feuerwehren zahlenmäßig das Rückgrat des deutschen Feuerwehrsystems.

In Deutschland spielen die Feuerwehr-Dienstvorschriften (FwDV) eine große Rolle bei der Regelung der Aufgabenbereiche und der Durchführung von Einsätzen. Je nach Tätigkeitsfeld existieren eine Reihe unterschiedlicher Dienstvorschriften. Als Beispiel sei eine der wichtigsten genannt:

- **FwDV 100:** *Führung und Leitung im Einsatz.* Zweck der Feuerwehr-Dienstvorschriften ist es, „die erforderliche Einheitlichkeit im Feuerwehrdienst herbeizuführen und auch zukünftig sicherzustellen. Sie gelten für den Einsatz und für die Ausbildung“ (siehe [FwDV100]).

Die Dienstvorschriften sind jedoch als grobe Richtlinie zu verstehen und geben lediglich einen Rahmen für eine konkrete Ausgestaltung vor. Sie werden den Bundesländern zur Einführung empfohlen und werden aufgrund der unterschiedlichen Rechtslagen länderspezifisch umgesetzt. Die aktuell gültigen Dienstvorschriften der einzelnen Bundesländer unterscheiden sich daher mehr oder weniger stark.

Die Struktur der Einsatzleitung auf der operativ-taktischen Führungsebene (Führungstab, Technische Einsatzleitung, Örtliche Einsatzleitung und Gemeinsame Einsatzleitung vor Ort) ist von der Gefahrenlage, dem Schadensereignis und den zu führenden Einheiten abhängig. Wer konkret die Einsatzleitung übernimmt, ist in den Feuerwehr- bzw. den Katastrophenschutzgesetzen der Länder geregelt [FwDV100].

Ist die Feuerwehr bei einer Schadenslage beteiligt, so benennt der oberste Verwaltungsbeamte einen **Einsatzleiter** für die Leitung des kompletten Schadensereignisses. Er ist typischerweise ein erfahrener und speziell geschulter Feuerwehrbeamter. Zur Bewältigung der vielfältigen Aufgaben wird der Einsatzleiter durch einen zu etablierenden Führungstab unterstützt. Die Einsatzleitung koordiniert die Tätigkeiten aller beteiligten Stellen und stellt die Schnittstelle zu den nachgeschalteten Führungsstäben und der zuständigen Rettungsleitstelle dar. Sie etabliert sich generell am Einsatzort z. B. in einem Einsatzleitwagen (ELW) [PM01].

Rettungsdienst

Der Aufgabenbereich des Rettungsdienstes liegt in der Sicherstellung einer notfallmedizinischen Grundversorgung am Notfallort, dem Herstellen der Transportfähigkeit und dem unter medizinisch-fachlicher Betreuung durchzuführenden Transport in ein geeignetes Krankenhaus zur weiteren Versorgung. Der Rettungsdienst wird je nach Region vorrangig von Hilfsorganisationen wie z. B. dem *Deutschen Roten Kreuz* (DRK), dem *Arbeiter-Samariter-Bund* (ASB), der *Johanniter-Unfall-Hilfe* (JUH) und dem *Malteser Hilfsdienst* übernommen. Durch eine sog. Bedarfsplanung ist festgelegt, wie viele Rettungswagen (RTW), Notarzteinsetzfahrzeuge (NEF) und Krankentransportwagen (KTW) in der jeweiligen Region vorzuhalten sind, und wie dementsprechend die zeitlich-personelle Besetzung (Dienstplan) der Rettungsmittel zu erfolgen hat. Typischerweise beteiligt sich in Städten, die eine Berufsfeuerwehr unterhalten, auch die Feuerwehr mit eigenen RTW und NEF an dieser Aufgabe. Hinzu kommen in einigen Regionen private Rettungsdienstunternehmen.

Die gesetzlichen Grundlagen der Aufgaben des Rettungsdienstes sind in den Rettungsdienstgesetzen bzw. Katastrophenschutzgesetzen der verantwortlichen Kommunen und Ländern geregelt. Als Beispiel eines von 16 Gesetzen sei das entsprechende Gesetz von Thüringen genannt:

- **ThürRettG:** *Thüringer Rettungsdienstgesetz* (2008), dessen Zweck die Regelung einer „Sicherstellung einer bedarfsgerechten medizinischen Versorgung der Bevölkerung mit Leistungen des Rettungsdienstes“ ist (siehe [ThürRettG]).

Bei einer größeren Einsatzlage sind im Rettungsdienst zwei verschiedene Einsatzleitertypen dafür verantwortlich, das Notarzt- bzw. Rettungsdienstpersonal zu führen. Stellvertretend für die Rettungsdienste sei folgend aus dem *Thüringer Rettungsdienstgesetz* [ThürRettG] zitiert. Dort heißt es unter § 17 *Rettungsdienstliche Versorgung in besonderen Fällen:* „Zur

Sicherstellung der rettungsdienstlichen Versorgung bei größeren Notfallereignissen unterhalb der Katastrophenschwelle mit mehreren Verletzten oder Erkrankten, bei denen die Tätigkeiten des eingesetzten Personals koordiniert werden müssen, hat der Aufgabenträger des bodengebundenen Rettungsdienstes eine rettungsdienstliche Einsatzleitung vor Ort einzurichten. Dieser gehören insbesondere ein Leitender Notarzt und ein Organisatorischer Leiter an.“ Die einzelnen Aufgabenbereiche werden ebenfalls im § 17 spezifiziert:

- **Leitender Notarzt (LNA):** *„Der Leitende Notarzt leitet den rettungsdienstlichen Einsatz, stimmt alle medizinischen Maßnahmen aufeinander ab und überwacht deren Durchführung. [...] Er ist gegenüber dem Personal des Rettungsdienstes, den eingesetzten Ärzten und den sonstigen zur rettungsdienstlichen Versorgung eingesetzten Kräften weisungsbefugt.“*
- **Organisatorischer Leiter Rettungsdienst (OrgL):** *„Der Organisatorische Leiter unterstützt den Leitenden Notarzt, indem er organisatorische Führungs- und Koordinationsaufgaben übernimmt. Er ist gegenüber dem Personal des Rettungsdienstes und den zur rettungsdienstlichen Versorgung eingesetzten Kräften weisungsbefugt.“*

Polizei

Die Aufgaben der Polizei ergeben sich aus Recht und Gesetz. Sie umfassen insbesondere die Gefahrenabwehr einschließlich Gefahrenvorsorge und die vorbeugende Bekämpfung sowie die Verfolgung von Straftaten und Ordnungswidrigkeiten [PDV100]. Speziell in einer Großschadenslage mit vielen Verletzten und Toten ist die Polizei zuständig für Amts- und Vollzugshilfe, die Betreuung und Identifizierung unbekannter hilfloser Personen, die Einleitung von Todesermittlungsverfahren sowie für die Beweissicherung, Ursachenforschung und Eigentumssicherung [Wei02].

Die Polizeidienstvorschriften (PDVs) regeln die Aufgaben und die Vorgehensweisen der Polizeikräfte. Beispiele hierfür sind:

- **PDV 100: Führung und Einsatz der Polizei (Ausgabe 1999).** In der Verordnung werden alle Facetten der Führung und des Einsatzes der Polizei skizziert [PDV100].
- **PDV/DV 800: Fernmeldeeinsatz (Ausgabe 1986).** Diese Dienstvorschrift befasst sich mit Planung, Aufbau und Betrieb des Fernmeldeverkehrs und hat ebenfalls im Katastrophenschutz sowie der Feuerwehr Bedeutung.

Die konkrete Ausgestaltung der Dienstvorschriften fällt auch hier bei den unterschiedlichen Polizeipräsidien unterschiedlich aus. Der Aufbau der Führungsorgane ist (ähnlich wie bei der Feuerwehr) abhängig von der Art der sog. *Aufbauordnung* und den jeweils zu führenden Organisationseinheiten.

Die Führungsstruktur der Polizei unterscheidet die *Allgemeine Aufbauordnung* (AAO) und die *Besondere Aufbauordnung* (BAO). Während die AAO im Rahmen des täglichen Dienstes etabliert ist, ändert sich die Struktur hin zu einer BAO, wenn es sich um einen größeren Einsatz handelt. Der Aufbau der BAO gliedert sich in zwei Phasen. Phase 1: Sofortmaßnahmen zur Stabilisierung der Lage, Vorbereitung der Phase 2 und Alarmierung aller zu Beginn benötigten Kräfte. Phase 2: Bewältigung des Einsatzes mit dem Führungsstab bzw. der Führungsgruppe und in Abstimmung mit allen beteiligten nicht-polizeilichen Kräften. In der Phase 1 übernimmt in der Regel der diensthabende Dienstgruppenleiter (DGL) der Leitstelle (Polizeipräsidium) die Funktion des sog. **Polizeiführers**. Der Polizeiführer trägt die Gesamtverantwortung und trifft die grundsätzlichen Entscheidungen.

Weiterhin werden in einer BAO verschiedene Einsatzabschnitte (EA) mit jeweiligen Unterabschnitten (UA) etabliert, für die es jeweils wiederum spezielle **Einsatzabschnittsleiter** bzw. **Unterabschnittsleiter** gibt. Falls möglich, sollte im Polizeiführungsstab unverzüglich ein Fachberater des Rettungsdienstes integriert werden.

Weitere Spezialorganisationen

Je nachdem welche Merkmale eine Großschadenslage aufweist, kommen zusätzliche Organisationen zum Einsatz. So unterstützen Einheiten des *Katastrophenschutzes* (KatSchutz) die Feuerwehr und den Rettungsdienst in Sachen Technik, Betreuung und Versorgung. Weiterhin existiert zur Überbrückung der zeitlichen Lücke zwischen dem Eintreffen des Katastrophenschutzes und der benötigten Unterstützung der regulären Einsatzkräfte das (teils überholte) Konzept der *Schnellen-Einsatz-Gruppe* (SEG). Die SEG übernimmt dabei ähnliche Aufgaben wie die Kräfte des Katastrophenschutzes, ist jedoch in der Regel schneller alarmiert und somit früher am Einsatzort verfügbar. Zusätzlich kommen bei Verschüttungen *Rettungshundestaffeln* zum Einsatz und bei Einsätzen der Wasserrettung die Kräfte der *Deutschen Lebens-Rettungs-Gesellschaft*. Nicht zuletzt sei noch das als Bundesbehörde organisierte *Technische Hilfswerk* (THW) genannt, welches bundesweit bei Bedarf technische Hilfestellung in Großschadenslagen anbietet. Der Großteil der Einsatzkräfte der eben genannten Organisationen setzt sich aus hochmotivierten und freiwilligen Einsatzkräften zusammen.

Hinweis: In der folgenden Arbeit wird sich bei den Beispielen auf die Feuerwehr und den Rettungsdienst beschränkt, da diese wesentliche Aufgabenbereiche einer Großschadenslage abdecken und eng in einer solchen zusammenarbeiten. Dies ermöglicht eine für den BOS-Bereich stellvertretende Betrachtung typischer Problemstellungen und interessanter Beispiele von Standardhandlungsroutinen.

2.2 Großschadensereignisse und deren Charakteristiken

Kleinere Einsatzlagen, wie z. B. ein Rettungseinsatz nach einem Verkehrsunfall mit einer oder zwei verletzten Personen, stellen für die Einsatzkräfte eine gewisse Routine dar. Die Art und der Umfang einer solchen Schadenslage gehört zum Tagesgeschäft, sodass die gebotenen Arbeitsabläufe routiniert und fehlerfrei ablaufen. Die Lage ist überschaubar und die eingesetzten Mittel und Kräfte sind ausreichend. Auch kommen im Alltagsgeschäft zumeist eingespielte Teams zum Einsatz, die einen hohen Grad an Professionalität aufweisen.

Hingegen werden die Einsatzkräfte in einem größeren Schadensereignis, wie z. B. einem *Massenanfall von Verletzten oder Erkrankten* (MANV), mit besonderen Herausforderungen konfrontiert. Ein MANV ist zu Beginn von der Diskrepanz zwischen der Anzahl der verletzten Personen und den zunächst zur Verfügung stehenden Einsatzkräften und -mitteln gekennzeichnet [PM01]. Da diese Situation vom täglichen Einsatzgeschehen abweicht, entstehen besondere Koordinationsaufgaben, die nur durch eine organisatorische und taktische Planung sicherzustellen sind [PMU01]. Es gilt, einer großen Anzahl verletzter Personen eine möglichst schnelle und angemessene Versorgung zukommen zu lassen, und das mit anfangs zu knappen Ressourcen. Weitsichtiges Handeln ist gerade zu Beginn eines der Erfolgskriterien der späteren Lagebewältigung. Anfängliche kleine Fehler können später zu gravierenden Auswirkungen führen.

In Großschadenslagen sind die Einsatzkräfte mit besonderen Widrigkeiten konfrontiert:

- Der zu Beginn herrschende Informationsmangel erschwert vor allem den Führungskräften die Entscheidung über das weitere Vorgehen.
- Viele Maßnahmen müssen fast zeitgleich eingeleitet und abgearbeitet werden.
- Auch kann sich die Einsatzlage jederzeit ändern, neue Anforderungen können hinzukommen und ältere plötzlich obsolet sein. Die Lageentwicklung ist schwer voraussagbar.
- Oftmals erschweren auch interorganisationale Zielkonflikte das Einsatzmanagement.
- Die Maßnahmen sind zumeist nicht umkehrbar und teils ist unklar, ob sie die angestrebten Effekte erzielen werden.
- Weiterhin muss die zum Teil ungewohnte Koordination und Kommunikation mit den anderen Organisationen gemeistert werden, und das ggf. bei einer geografisch verteilten Einsatzlage.
- Nicht zuletzt können die unterschiedlichen Erfahrungs- und Ausbildungsstände der Einsatzkräfte eine reibungslose Einsatzbewältigung behindern.

In einer Großschadenslage etablieren sich am Einsatzort die bereits angesprochenen besonderen Leitungsstrukturen, die im Wesentlichen die Führung und Koordination der Einsatzkräfte vor Ort übernehmen. Die Charakteristiken der Schadenslage bestimmen hierbei die beteiligten Hilfsorganisationen und deren Strukturen. Großschadensereignisse werden typischerweise anhand ihres Ausmaßes klassifiziert. Ein Beispiel hierfür ist die MANV-Stufeneinteilung der *Planungsplattform des Deutschen Städtetages*. Sie dient als eine grobe Richtlinie und ist im deutschen Katastrophenschutz geläufig [Lüd07] (vgl. Abb. 2.1).

Versorgungsstufen	I			II		III		IV
Ausmaß	Rettungsdienst			MANV 1	MANV 2	MANV 3	MANV 4	
	Grundbedarf	Spitzenbedarf	Sonderbedarf	<50 Betroffene	<=500 Betroffene	>500 Betroffene	zusätzlich zerstörte Infrastruktur	
politische Ebene	Zuständige Behörde im Tagesgeschäft			Bürgermeister, Landräte, OA	Bürgermeister, Landräte, OB; ggf. Landesreg.		Landesregierung; Bundesreg.	
administrativ-organisator. Ebene	Integrierte Leitstelle, Äztl. Leiter			Verwaltungsstab-behörde; Führungsstab	interdisziplinäre Krisenstäbe		interdisziplinäre Krisenstäbe	
operativ-taktische Ebene	Notarzt bzw. RA/RS			Einsatzleitung vor Ort; LNA & OrgL	Einsatzleitung vor Ort; LNA & OrgL		Einsatzleitung vor Ort; LNA & OrgL	

Abbildung 2.1: MANV-Stufen versus Regelbetrieb (aus Sicht des Rettungsdienstes).

Es bleibt zu bedenken, dass die konkrete Maßnahmenplanung in den unterschiedlichen Bundesländern, Städten und Kommunen nicht einheitlich umgesetzt wird. So existieren z. B. in Großstädten andere Versorgungsmöglichkeiten und tägliche Vorhaltungen von Rettungskräften und -Mitteln als in ländlichen Gegenden, wo schon eine wesentlich geringere

Anzahl von Verletzten die zur Verfügung stehenden Einsatzkapazitäten erschöpfen können. So ist auch obige Stufeneinteilung kein bundesweiter Konsens und kann daher regional höchst unterschiedlich ausfallen. Zum Beispiel verwendet die Feuerwehr in Düsseldorf drei MANV-Kategorien, wobei die dritte nicht einmal an obige MANV-1-Stufe heranreicht [SBQ+09].

Je nach MANV-Stufe treten auch unterschiedliche Akteure und Führungsebenen in Kraft. Abbildung 2.1 macht dies in Form einer Matrix deutlich (vgl. [FwDV100], S. 21). Die Abbildung stellt die verschiedenen Führungsebenen bei Großschadensereignissen dar, und zeigt die unterschiedlichen Verantwortlichkeiten in den jeweiligen Stufen auf.

An dieser Stelle sei auf den Unterschied zwischen einer Großschadenslage und einer Katastrophe hingewiesen. Eine *Katastrophe* wird in [DIN13050] als ein „über das Großschadensereignis hinausgehendes Ereignis mit einer wesentlichen Zerstörung oder Schädigung der örtlichen Infrastruktur, das im Rahmen der medizinischen Versorgung mit den Mitteln und Einsatzstrukturen des Rettungsdienstes alleine nicht bewältigt werden kann“ definiert und ist bezüglich der Abbildung 2.1 somit einem MANV-4 zuzuordnen. Die Polizei spricht hierbei jedoch nicht von einem MANV, sondern von *Größere Gefahren und Schadenslagen, Katastrophen* (GGSK), was die MANV-Stufen 1 bis 4 umfasst.

Für die weitere Arbeit soll sich ausschließlich auf die operativ-taktische Ebene konzentriert werden (vgl. Abbildung 2.1). Zusätzlich soll nicht mehr von MANV-Stufen die Rede sein, sondern allgemein der Begriff *Großschadenslage* für eine ganze Klasse von Schadensereignissen verwendet werden.

Einsatzdynamik und Einsatzphasen

Wie oben herausgestellt, liegt es in der Natur eines Großschadensereignisses, dass dessen Merkmale und dessen Entwicklungen schwer vorhersagbar sind. Dennoch gibt es invariante Charakteristiken, die sich in sog. Abarbeitungs-Phasen widerspiegeln. Wie lange jede einzelne Phase andauert und wie ausgeprägt sie ist, variiert von Schadensfall zu Schadensfall. Vor allem auf Seiten der polizeilichen Gefahrenabwehr hat sich hierfür das Verständnis einer groben Drei-Phasen-Einteilung etabliert, die prinzipiell auch auf die Bereiche der nicht-polizeilichen Gefahrenabwehr übertragbar sind (vgl. z. B. [BBK10; BBD+05]):

1. Die *Initial*-Phase, auch Chaos- oder Anlaufphase genannt, bezeichnet die Situation in den ersten Minuten eines Schadensereignisses. Sie ist aufgrund des in dieser Phase noch bestehenden lückenhaften Lagebildes (selbst auf Seiten der professionellen Einsatzkräfte) von einem unkoordinierten Handeln und einem gewissen augenscheinlichen „Durcheinander“ geprägt.
2. In der zweiten Phase, der *Konsolidierungs*-Phase (auch Führungs- oder Abarbeitungsphase genannt), werden alle Einsatzbereiche gemäß der Einsatzplanung und der Führungslehre strukturiert und deren Abläufe gemäß den Standardprozeduren harmonisiert. Hierzu gehört der Aufbau besonderer Organisationsstrukturen der Einsatzleitung und die Einrichtung funktionsbezogener Einsatzabschnitte mit klarer Festlegung der Verantwortungsbereiche.
3. Die *Rückführungs*-Phase, oder auch Demobilisierungs-Phase, bezeichnet den Zeitraum zwischen den abgeschlossenen Arbeiten am Einsatzort und der Wiederherstellung der Einsatzfähigkeit aller Kräfte und Technik. Hierzu zählen die Patientenübergabe in den Kliniken und die Rückführung aller Einsatzmittel und -kräfte, die Überprüfung und Archivierung der Einsatzdokumentation (Patientenregistrierung, Einsatztagebuch) und eine nachgelagerte Einsatznachbesprechung (engl. *debriefing*).

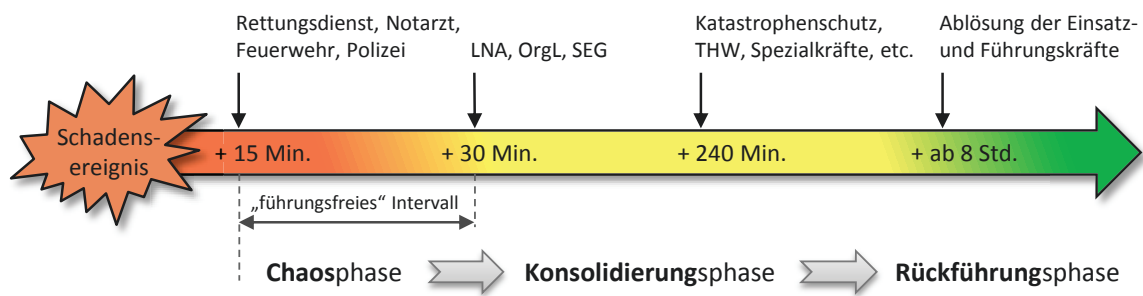


Abbildung 2.2: Reihenfolge des Eintreffens bestimmter Einsatzkräfte und Einsatzphasen.

Eine alternative Perspektive auf die Einsatzphasen einer Großschadenslage stellt die Betrachtung der zeitlichen „Eintreffreihenfolge“ aller beteiligten Einsatzkräfte dar. In Anlehnung an HACKSTEIN [Hac06] skizziert die Abbildung 2.2 (von links nach rechts) die typischen zeitlichen Dimensionen und die Reihenfolge der einzurichtenden Führungsstrukturen. Ebenfalls zu sehen ist das typischerweise zeitlich verzögerte Eintreffen nachgelagerter Spezialkräfte anderer Aufgaben- und Zuständigkeitsbereiche.

Die Abbildung macht weiterhin die Problematik des unterschiedlichen Erfahrungs- und Trainingslevels der Einsatzkräfte deutlich. So werden die erst-eintreffenden Einsatzkräfte in der Regel keine besondere Ausbildung bzw. Erfahrung im Management einer Großschadenslage besitzen. Sie müssen jedoch (kommissarisch) bereits zu Beginn Teile der Einsatzleitung übernehmen, wobei die Gefahr, strategische Fehler zu begehen, besonders groß ist. Die entsprechend geschulten Experten, wie z. B. der LNA und der OrgL, werden zumeist nach-alarmiert und treffen daher erst verzögert an der Einsatzstelle ein. Für sie stellt sich beim Eintreffen die Problematik, das bereits Geschehene und die eingeleiteten Maßnahmen zu erfassen und die Einsatzleitung in ihren Bereichen zu übernehmen.

2.3 Einsatzplanung und Einsatzvorbereitung

Komplett vorgefertigte Einsatzpläne, die angeben, *wann wer was* im Ernstfall zu tun hat, sind aufgrund der Vielfalt, der Unsicherheit und der Dynamik der zu erwartenden Lageentwicklungen unmöglich anzugeben. In der Praxis beschränkt sich daher die Planung auf eine *abstrakte* Einsatzvorplanung mit einer Bereitstellung grob spezifizierter Handlungsanweisungen und Dokumente, die für ein möglichst breites Spektrum an vorstellbaren Schadenslagen anwendbar sind. Im Sprachgebrauch der BOS haben die Begriffe *Plan*, *Planung* oder *Einsatzplanung* ein breites Bedeutungsspektrum. So wird unter Planung hauptsächlich eine Menge von vorbereitender Maßnahmen sowie die konkrete kognitive Leistung eines Einsatzleiters verstanden, der in einem Einsatz aufgrund der ihm vorliegenden Fakten das weitere Vorgehen plant. Beide Facetten der Planung werden im Folgenden skizziert.

Planung als vorbereitende Maßnahme

Als vorbereitende Maßnahme ist es üblich, spezielle „Pläne“ und Informationshilfen zu erarbeiten, von denen einige wichtige Vertreter im Folgenden vorgestellt werden.

Alarmierungspläne: Den (Integrierten) Leitstellen obliegt unter anderem die Aufgabe der Alarmauslösung für Rettungsdienst, Feuerwehr, SEG und Katastrophenschutz. Die hierfür zur Unterstützung erstellten Alarmierungspläne (auch *Alarm- und Ausrückordnungen* (AAO) genannt) sind oft sehr einfach gehaltene Listen, auf denen die zu alar-

mierenden Einheiten/Organisationen/Personen plus der jeweiligen Telefonnummer (bzw. der Möglichkeit, einen Alarmempfänger auszulösen,) aufgelistet sind. Alarmierungspläne sind eine der wichtigsten Voraussetzungen, um zu Beginn einer Schadenslage schnellstmöglich alle benötigten Einsatzkräfte gemäß dem erwarteten Schadensumfang zu alarmieren. Darüber hinaus kommen erweiterte Alarmierungspläne bei der Nachalarmierung von Sekundär- und Spezialeinsatzkräften zum Einsatz. Diese Listen wurden im Vorfeld auf Basis von Erfahrungen der Vergangenheit erstellt. Die Alarmierungspläne sind entweder auf Papier in sog. Alarmordnern hinterlegt oder aber auch zunehmend in elektronischer Form über die jeweilige Leitstellensoftware zugänglich. So existieren mittlerweile Leitstellensysteme, die den Disponenten vorher abgelegte Einsatzkategorien zur Auswahl geben und die je nach Schadenskategorie und Einsatzstichwort Standard-Alarmierungspläne bereitstellen.

Einsatzstichworte: Unterschiedliche Notfälle erfordern unterschiedliche Einsatzkräfte und somit Alarmierungspläne. Jeder zuvor „eingepflanzten“ Schadenssituation kann anhand von sog. Einsatzstichworten ein entsprechender Alarmierungsplan zugeordnet werden. Einsatzstichworte sind eine Möglichkeit einer abstrakten Beschreibung von Einsatzsituationen. Vorrangig wird dieses Konzept in Integrierten Rettungsleitstellen (IRL) dafür genutzt, um je nach Informationen des eingehenden Notrufs Rückschlüsse auf die benötigten Kräfte und Alarmstufen zu ziehen. Zusätzlich zu den Fakten der Notrufmeldung gehen noch allgemeine Informationen wie z. B. über die Tageszeit und die Wetterlage mit in die Bewertung ein. Die Listen von Einsatzstichworten unterscheiden sich von Region zu Region stark. Eine Standardisierung von Einsatzstichworten ist in Deutschland nicht anzutreffen und auch nicht angebracht. Vielmehr spiegeln die unterschiedlichen Einsatzklassifikationen die jeweiligen regionalen Besonderheiten und die jeweilige Technik- und Personalsituation der unterschiedlichen Kommunen wider.

Rahmenpläne für spezielle Szenarien: Alle denkbaren Szenarien im Vorfeld zu planen, ist ein unrealistisches Unterfangen und wird in der Praxis auch nicht getan. Dies bedeutet jedoch nicht, dass nicht trotzdem versucht wird, grobe Rahmenpläne für Szenarien auf Basis möglicherweise gefährdender regionaler Besonderheiten (wie Atomkraftwerke, Hochhäuser, regelmäßige Großveranstaltungen, Flugbetrieb, Überschwemmungsgefahren) zu erarbeiten. Ganz im Gegenteil, die Kommunen sind sogar im Rahmen der Sicherstellung der zivilen Sicherheit unter anderem dazu verpflichtet, Rahmenpläne für sicherheitskritische regionale Anlagen zu erarbeiten und vorzuhalten. Diese Art von Plänen sind zumeist detailliert ausgearbeitete Dokumente, die neben dem innerbetrieblichen Krisenmanagement auch die Zusammenarbeit mit den örtlichen BOS zur Gefahrenabwehr und zum Personenschutz betrifft. Im Falle einer Schadenslage erhalten die Einsatzkräfte der Feuerwehr durch die Rahmenpläne z. B. eine erste Orientierung über die zu erwartenden Gefahren, die Anfahrt zum Objekt, wichtige Kontaktpersonen und nicht zuletzt zumeist einen Grundriss des Gebäudes bzw. des Geländes.

Checklisten-Varianten: Vereinzelt kommen zur Überwachung der korrekten Ausführung von Prozessen papierbasierte Checklisten (PCLs) zum Einsatz, deren Aufbau und Abarbeitungskonzepte sich aber von Region zur Region und von BOS zu BOS stark unterscheiden sowie kein einheitliches Grundkonzept erkennen lassen (siehe [WKK11]). Vielmehr ist in der Praxis eine Vermischung zwischen Checklisten-Aspekten, Informationsblättern, Alarmierungsplänen und Kartenmaterialien zu beobachten. Obwohl (elektronische) Checklisten in den Rettungsleitstellen teils verbreitet sind, kommen sie (selbst in Papierform) für

die Einsatzkräfte und Einsatzleitung vor Ort selten zum Einsatz. In Kapitel 4 wird sich der Thematik Checklisten ausführlich gewidmet.

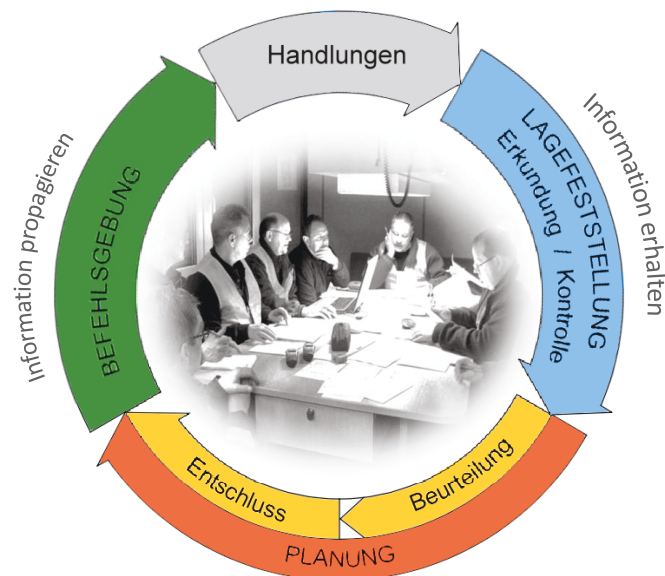


Abbildung 2.3: Klassisches Modell des Führungsvorgangs (nach [FwDV100]).

Planung als kognitiver Prozess im Führungsvorgang

Die Abbildung 2.3 skizziert das gelehrte klassische Modell dieses Führungsvorganges. Ein erfolgreicher Führungsvorgang ist unter der Prämisse der Sicherstellung des Einsatzenerfolges nur durch ein geordnetes Handeln aller Beteiligten zu erreichen [FwDV100]. Zu Beginn steht die unbekannte Lage, die von den erst-eintreffenden Kräften erkundet werden muss. Die Planung besteht aus der Beurteilung der Fakten und einem konkreten Handlungsentschluss. Der Entschluss zum Handeln beinhaltet die durchzuführenden Maßnahmen, die hier als ein Plan angesehen werden, der *klar*, *einfach* und *ausführbar* sein soll [FwDV100].

Die in der Praxis ablaufenden komplexen Denk- und Handlungsabläufe, können mit diesem recht vereinfachten Bild allerdings nur schwer vermittelt werden. In der Literatur (vgl. z. B. [PMU01]) wird ein detaillierteres Führungsmodell vorgestellt, das den Führungsvorgang besser widerspiegelt. Abbildung 2.4 (siehe folgende Seite) skizziert diesen erweiterten Führungsvorgang (siehe auch FwDV100, S. 24 ff.). Es wäre jedoch falsch, anzunehmen, dass der in Abbildung 2.4 gezeigte Führungsvorgang ähnlich dem Vorgang eines starren Algorithmus oder eines Workflows in der Praxis exakt befolgt wird. Vielmehr dient das Schema zur Schulung der Führungskräfte und wird in der Praxis mehr oder weniger flexibel angewandt. Die dynamische Lage und die Vielzahl der erst zum Schadenszeitpunkt bekannten Randbedingungen sind Gründe dafür, die Einsatzführung so flexibel wie möglich zu gestalten.

2.4 Handlungsroutinen und IT-Unterstützung

Trotz regelmäßiger Übungen stellt sich für die Führungskräfte jede Großschadenslage als ein besonderes Ereignis dar, bei dessen Bewältigung es zusätzlich darauf ankommt, die wesentlichen Standardprozeduren je nach (BOS-spezifischem) Training, lokalen Vereinbarungen oder übergeordneten Regelwerken, soweit es die Situation erlaubt, korrekt zu befolgen. Die

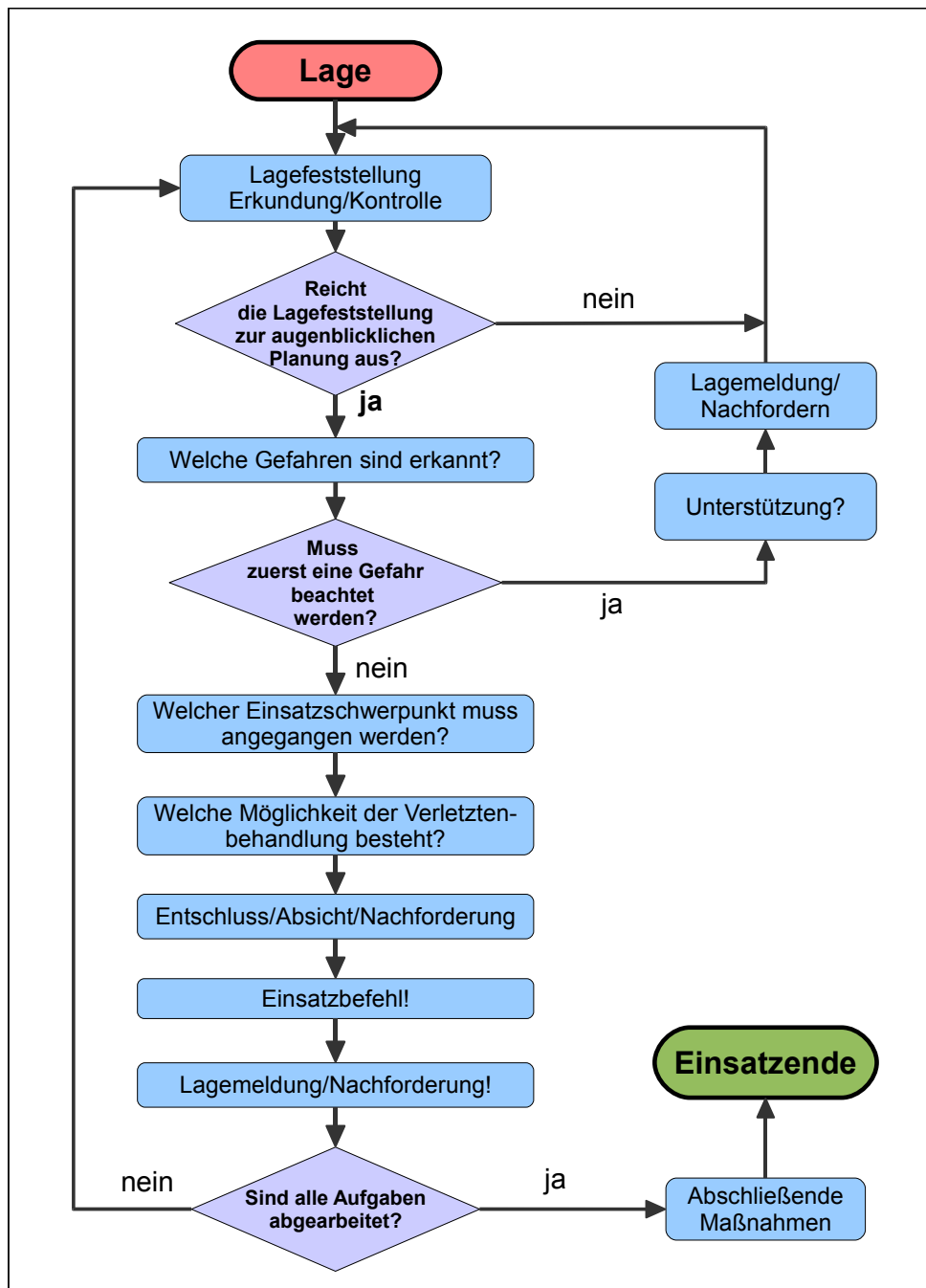


Abbildung 2.4: Gelehrter Führungsvorgang aus Sicht der Feuerwehr (Quelle: [CP07]).

Ausgestaltung der Arbeitsabläufe im BOS-Bereich unterliegt einer Vielzahl von Regelwerken, Lehrmeinungen, Erfahrungen, Dienstvorschriften und Gesetzen. Die jeweilige Interpretation und Ausgestaltung der Regelwerke obliegt größtenteils der Eigenverantwortung der BOS und ist den jeweiligen regionalen Besonderheiten angepasst [CG05]. Als Assistenz zur Einsatzzeit sind diese Regelwerke aufgrund ihres Inhalts und ihrer Form nicht geeignet und auch nicht vorgesehen.

Ohne entsprechende Standards ist es schwierig, eine gleichbleibend hohe Qualität der Maßnahmen sicherzustellen, insbesondere während langer, komplexer oder ungewohnter Einsatzsituationen. Die Anwendung von Standardhandlungsroutinen (im Folgenden auch

Standard Operation Procedures, kurz *SOPs*, genannt) stellt eine Möglichkeit dar, immer wiederkehrende Routineaufgaben möglichst fehlerfrei und unter Vermeidung unnötiger Reibungsverluste durchzuführen. SOPs können die Zusammenarbeit zwischen den Einsatzkräften und den unterschiedlichen Organisationen deutlich befördern und somit die Qualität einer gesamten Lagebewältigung erhöhen. SOPs machen umfangreiche Leitlinien operabel (vgl. [Ell11]) und müssen den jeweiligen regionalen und organisationspezifischen Eigenheiten angepasst werden.

Die Standard-Einsatz-Regeln (SER) aus dem deutschen Feuerwehrbereich sind Beispiele für SOPs. Sie stellen den Versuch einer Übertragung der in den USA weit verbreiteten SOPs dar (siehe hierzu die Bemühungen von CIMOLINO¹ und GRAEGER [CG05]). Sie sollten unter Beteiligung möglichst aller Führungsebenen erstellt und allen Mitarbeitern bekannt gemacht werden. Laut CIMOLINO ermöglicht die schriftliche Fixierung solcher Routinen eine Einsatzvorbereitung und kann somit die Grundlage einer gesamten groben Einsatz- bzw. Trainingsplanung sein. Eine komplette Vorplanung aller nur denkbaren Einsatzszenarien ist jedoch schlichtweg unmöglich.

Allerdings sind in der BOS-Praxis niedergeschriebene SOPs (oder auch operationalisierte Einsatzpläne) in der Regel nicht anzutreffen. Von *den* für die BOS gültigen Standardhandlungsroutinen kann daher nicht die Rede sein. Dies deckt sich mit der Feststellung von ROSE et al.: „*Most communities have very vague emergency plans. Procedures seem to be locked in the head of experienced staff, but are not traceable by electronic or paper means*“ [RPA08]. So weiß z. B. in der Regel jeder erfahrene Feuerwehreinsatzleiter, was er grob in welcher Situation zu tun hat und welches die in seinem Feuerwehr-Ausrückebereich vereinbarten und akzeptierten Standardprozeduren sind. SOPs spiegeln hierbei eine momentane „best-practice“ des jeweiligen Arbeitsgebietes wider. SOPs dürfen jedoch nicht als „Workflows“ verstanden werden, sondern eher als eine grobe Arbeitsbeschreibung für spezielle Aufgaben.

Nichtsdestotrotz besteht in der Praxis sehr wohl ein großes Interesse, gemäß der „best practice“ zu handeln und Fehler durch Vergessen weitgehend zu vermeiden. Hierzu wird bis heute jedoch ausschließlich auf Training und den Aufbau von Erfahrung auf Seiten der Einsatzkräfte gesetzt.

Neben den professionellen Einsatzkräften kommen in Großschadenlagen viele freiwillige Einsatzkräfte zum Einsatz. Obwohl von diesen Einsatzkräften ebenfalls ein hohes Maß an Professionalität abverlangt wird, kann dies in der Praxis schwerlich erreicht werden. Doch vor Fehlern im Ablauf von Handlungsroutinen sind weder die Experten noch die wenig erfahrenen Einsatzkräfte gefeit.

Die Art und Weise, wie eine Organisation mit eigenen Fehlern umgeht, ist ihre „Fehlerkultur“. Gemeint ist im weitesten Sinne der Umgang mit Fehlern, die in einer Organisation bei der Arbeit auftreten. Von der Polizei, der Feuerwehr und dem Rettungsdienst wird ein hohes Maß an Professionalität erwartet. Sie agieren zumeist im öffentlichen Raum und stehen daher in besonderem Maße im Fokus der Aufmerksamkeit von Medien und Bevölkerung. Dies macht den Umgang und das offizielle Eingestehen von Fehlern problematisch; von „Fehlern“ wird überhaupt nur äußerst ungern gesprochen. In der Human-Factors-Forschung, die sich unter anderem mit dem bewussten Umgang mit menschlichen Fehlern beschäftigt, wird daher statt von Organisationen mit einer Fehlerkultur auch von *High Reliability Organization* (HRO) gesprochen [HPK+12]. Gemeint ist die Widerstandsfähigkeit der Arbeitsabläufe einer Organisation gegenüber menschlichen Fehlern und äußeren Einflüssen.

¹ Der Autor hat einige Broschüren zum Thema *Standard-Einsatz-Regeln* verfasst und stellt unter www.standardeinsatzregel.org (abgerufen am 20.12.2013) weitere Informationen zum Thema bereit.

IT-Unterstützung für SOPs

Im BOS-Bereich findet sich eine umfassende IT-Unterstützung auf Seiten der Rettungsleitstellen sowie der Krisenstäbe von Kommunen und Ländern. So ist in heutigen modernen Rettungsleitstellen der Computereinsatz nicht mehr wegzudenken. Leitstellendisponenten nutzen professionelle Softwarelösungen, um z. B. eine automatische Alarmierung durchzuführen, alle Notfalleinsätze und die dazu zugewiesenen Einsatzfahrzeuge zu disponieren oder ein digitales Einsatztagebuch zu führen – um nur einige Beispiele zu nennen.

Auf Seiten der Krisenstäbe kommen teils bundeslandspezifische Informationsplattformen zum Einsatz, die speziell auf den Informations- und den Nachrichtenverteilungsbedarf der Krisenstäbe zugeschnitten sind. So z. B. in Hessen die browsergestützte Befehlsstellensoftware ILIAS-HE, welche zur Unterstützung der Stabsarbeit bei der Polizei, der Feuerwehr und dem Katastrophenschutz/Landeskrisenstab dient. ILIAS-HE wird in der hessischen Polizei landesweit seit 2004 eingesetzt [Hei09]. Bei der Bayerischen Polizei und im Thüringer Landeskrisenzentrum ist hingegen das webbasierte Einsatz-Protokoll-System (EPSweb) flächendeckend im Einsatz. Weitere Beispiele sind das Stabsorganisationssystem *Stabos* in Nordrheinwestfalen (vgl. [NKW08]) und das deutsche Notfallvorsorge-Informationssystem (deNIS II, siehe www.denis.bund.de (abgerufen am 20.12.2013)).

Die obere Aufzählung ließe sich noch weiterführen. Eine IT-Unterstützung im operativen Bereich vor Ort hat sich bis heute nicht etablieren können. Vereinzelt finden sich jedoch Insellösungen, die oft in Eigeninitiative einzelner engagierter Entscheidungsträger entstanden sind. Diese IT-Lösungen sind auf die speziellen Belange eines bestimmten Einsatzbereiches und der jeweiligen BOS-Organisation zugeschnitten. So kommt z. B. in der mobilen Einsatzleitung des Rettungsdienstes (DRK) Lemgo (Kreis Lippe, Nordrhein-Westfalen) eine IT-Unterstützung zum Einsatz, die eine Verwaltung aller Einsatzfahrzeuge und aller Verletzten ermöglicht. Der Abtransport der Verletzten kann somit besser organisiert und dokumentiert werden.

Ein weiteres Beispiel stellt die im SPEEDUP-Projekt prototypisch umgesetzte Kommunikationsplattform dar, welche die Führungskräfte z. B. bei der Kommunikation und der Telemetrie der Patientendaten vor Ort entscheidend unterstützt. Nach Wissen des Autors wird der Computereinsatz, im Sinne einer Assistenz zur Vermeidung von Fehlern bei SOPs, von keinem der heutigen Praxissysteme adressiert. Es zeichnet sich jedoch ab, dass sich mit einem zunehmenden Einsatz von IT auch am Einsatzort den Rettungskräften die Möglichkeit erschließt, kritische Arbeitsschritte durch eine computergestützte Hilfestellung zu kontrollieren.

Die vorliegende Arbeit widmet sich im Folgenden der Fragestellung, wie sich eine geeignete Unterstützung von Handlungsroutinen im schwierigen Umfeld der BOS realisieren lässt. Zuvor werden jedoch im nächsten Kapitel verwandte Forschungsansätze betrachtet, welche sich unter anderem mit dem Einsatz des Computers im Umfeld der Prozesse eines Krisenmanagements beschäftigen.

Kapitel 3

Stand der Forschung zur IT-Unterstützung für SOPs

Dieses Kapitel gibt einen Überblick über bisherige Forschungsarbeiten zum Thema IT-Unterstützung für (Standard-)Prozesse im BOS-Bereich bzw. in vergleichbaren (internationalen) Einsatzbereichen.

3.1 Forschungsperspektiven zur SOP-Unterstützung

Nicht zuletzt motiviert durch die Erfahrungen des 11. Septembers und einer Zunahme an Risiken für Mensch und Infrastruktur durch Naturkatastrophen, rücken in letzter Zeit vermehrt die Bereiche „zivile Sicherheit“ und „Krisenmanagement“ in den Fokus des Forschungsinteresses. Als Beispiel sei hierfür die Auflage des Programms „Sicherheitsforschung – Forschung für die zivile Sicherheit“ der deutschen Bundesregierung genannt, in dessen Rahmen sich Projekte auch mit der Frage einer geeigneten Bereitstellung von IT-Assistenz im Bereich „Schutz und Rettung von Menschen“ beschäftigten. Mit dem Ziel, Forschung und Praxis zu vernetzen, hat sich international die seit 2003 jährlich organisierte *International Conference on Information Systems for Crisis Response and Management* (ISCRAM) etabliert – auf der bereits zwei Vorarbeiten zu dieser Arbeit (siehe [WYM+11; KWB12]) vorgestellt wurden.

Während sich viele Projekte mit der Bereitstellung einer allgemeinen und allumfassenden Kommunikationsplattform – sog. „Emergency Management Information System“ (EMIS) –, einer Triage-Unterstützung (*triage* „Sichtung“) oder einer Visualisierung durch Kartendarstellungen befassten, wurden konkrete Fragen nach einer IT-Assistenz bezüglich einer Sicherstellung reibungsfreier Prozesse nur in einigen wenigen Forschungsprojekten adressiert (siehe z. B. die Projekte MobisPro und I-LOV, weiter unten).



Abbildung 3.1: Zwei Ansätze: IT-Assistenz für manuelle vs. automatische Handlungsplanung.

Die Lösungsansätze dieser Projekte lassen sich zwei (größtenteils) disjunkten Forschungsrichtungen zuordnen (vgl. Abb. 3.1). Beide Ansätze ähneln sich in ihren Zielen, unterscheiden sich jedoch vor allem in ihren speziellen Problemräumen und Lösungsansätzen. Die Rede ist von Arbeiten aus dem Themenumfeld des (Business) Process Managements (BPM) und den Arbeiten, die sich der SOP-Unterstützung aus Richtung der automatischen Handlungsplanung nähern. Letztere ist ein Teilgebiet der Künstlichen-Intelligenz-Forschung. Während sich die eine Seite mit der Anwendung von Techniken und Softwaretools des BPM beschäftigt, liegt der Ansatz der Künstlichen Intelligenz (KI) in der Anwendung eines konkreten Planungsparadigmas für Probleme der Einsatzplanung. Ziel der KI-Handlungsplanung ist es, anhand einer geeigneten Modellierung der Domäne und entsprechend expliziten Zielvorgaben einen (Lösungs-)Plan zur Reaktion auf ein beliebiges Krisenszenario zu berechnen.

Im Folgenden werden Arbeiten beider Herangehensweisen projekt- bzw. forschungsgruppenorientiert betrachtet, wobei die Begriffe Krisenszenario und Krisenmanagement für eine Vielzahl von Schadensereignissen größeren Ausmaßes bzw. deren Abarbeitungsmanagement stehen.

3.2 Business Process und Workflow Management

In diesem Abschnitt werden Forschungsarbeiten vorgestellt, die sich mit der Idee der Übertragung von Techniken aus dem Business Process Management auf das Krisenmanagement befassen. Unter dem Begriff *Process* soll an dieser Stelle im weitesten Sinne ein Hergang, Fortgang bzw. Ablauf eines Vorganges verstanden werden. Im Kontext betrieblich-organisatorischer Abläufe wird in diesem Zusammenhang auch von Geschäftsprozessen (*Business Processes*) gesprochen [Kla01].

Eine Vielzahl von Forschungsansätzen geht davon aus, dass sich die ablaufenden Prozesse einer Großschadenslage mittels BPM-Techniken/-Tools erfassen und modellieren lassen (siehe z. B.: BÜRMANNS [Bür05]; ROSE et al. [RPA08]; PEINEL et al. [PR09]; LINDEMANN et al. [LPK09]; SOBOLL et al. [SBQ+09]). Die Prozessmodelle dienen in einem ersten Schritt der Erfassung (Niederschrift) von Prozessen. Auf Basis des Prozessmodells können Fehler im Ablauf erkannt und die Prozesse optimiert werden. Hierfür kommen verschiedene Modellierungstechniken und -sprachen wie z. B. *Ereignisgesteuerte-Prozessketten* (EPKs) (vgl. SCHEER [KNS92]) oder *Petri-Netze* [Pet62], BPML, BPMN, XPDL sowie UML zum Einsatz (für einen umfassenden Überblick siehe BARTONITZ [Bar05]). In einem weiterführenden zweiten Schritt werden die formalen Prozessmodelle genutzt, um deren praktische Ausführung mit Hilfe des Computers zu überwachen und teils zu steuern. Diese Aufgabe übernehmen spezielle Prozessausführungssysteme (sog. Workflow-Management-Systeme (WfMS), siehe [Jab95]). Im Folgenden soll ein repräsentativer Teil dieser Forschungsansätze vorgestellt werden, der die Charakteristiken dieser Lösungsansätze gut widerspiegelt.

Im Rahmen des europäischen Verbundprojektes ERMA (Electronic Risk Management Architecture; '06 bis '08) wurden Prozessmodellierungsansätze für die Einsatzplanung und Simulation von Großschadensereignissen erarbeitet. Hierzu motivieren PEINEL & ROSE in [RPA08; PR09] die Notwendigkeit einer formalen Modellierung der geplanten Vorgehensweisen als computerintegrierte Prozesse. Aus Sicht der Autoren ist diese Art von Modellierung zuerst einmal hilfreich für eine Bewertung der Prozesse selbst, und weniger für eine spätere automatische Ausführung bzw. Ausführungsüberwachung, „da die überwiegende Anzahl der Maßnahmen außerhalb von IT-Systemen ausgeführt werden“ [PR09]. Hierbei widmeten sich PEINEL et al. unter anderem auch der Fragestellung, wie bereits vor-

handene umfangreiche Regelwerke und Vorgehensbeschreibungen konkret am Beispiel des Einsatzkonzeptes der „Rheinischen Projektgruppe MANV Überörtlich (ÜMANV)“ (siehe [Sch07]) mittels SOPs operationalisiert werden können. PEINEL et al. diskutieren in diesem Zusammenhang auch die Möglichkeit einer Wiederverwertung von sich in der Praxis bewährten Prozessbausteinen beim Management zukünftiger Großschadenslagen. Weiterhin schlagen die Autoren ein allgemeines Metamodell vor, welches die Unterteilung der Prozesse in folgende Bestandteile (Sichten) vorsieht: *Strategy view* (strategische Einsatzziele), *Concept view* (wesentliche Maßnahmen in zeitlichem und sachlogischem Zusammenhang), *Flow view* (detaillierte Ablaufspezifikation), *Organizations view* (Einsatzorganisation, Rollen und Qualifikationen), *Objectives view* (strategische und taktische Einsatzziele), und die *Requisites view* (Anforderungen). Begonnen bei den strategischen Einsatzzielen folgt der Modellierungsprozess den jeweiligen Sichten des Metamodells. Die Sichten sollen als Hilfestellung für die gesamte Prozessmodellierung dienen.

Konkret wurde unter Nutzung eines EPK-zentrierten Modellierungswerkzeuges ein Prozessmodell des ÜMANV entwickelt, das eine erste Analyse der Vorgehensweisen ermöglichte (vgl. auch [Ars08]). Das erstellte Modell soll einen Überblick über alle Prozesse geben und somit bereits bei der Diskussion der Einsatzvorbereitung dienlich sein.

PEINEL et al. resümieren jedoch in [PR09], dass sich bis heute in der BOS-Praxis keine formale Prozessmodellierung etablieren konnte, und vertreten jüngst in [PRW12a] den Standpunkt, „*that business process management methods and tools cannot be directly applied to the emergency management domain*“. Als Grund hierfür sehen sie den Fakt, dass die Domänenexperten aus den BOS (wie z. B. Feuerwehren oder Rettungsdienste) nicht auch zugleich Experten auf dem Gebiet der Prozessmodellierung sind, aber nur sie über das benötigte fachliche Know-how verfügen, um die Prozesse der Praxis entsprechend modellieren zu können. PEINEL et al. vertreten daraufhin die Meinung, dass heutige Techniken und Tools des BPM zur Planung von Katastrophenschutzplänen ungeeignet sind, vor allem dann, wenn man eine selbständige Prozesserfassung durch die BOS-Experten im Sinn hat [PRW12a].

Jüngst griffen PEINEL et al. in [PRW12b] den Ansatz der Checklisten-Perspektive von WUCHOLT & KRÜGER [WKK11] auf, und erweiterten das von WOLLERT & MANNABETH [WM11] vorgeschlagene Plan-Autoring-Tool für Katastrophenschutzmanagement um die Perspektive der Modellierung von Checklisten statt um die zuvor angestrebten Einsatzpläne. Hierzu berücksichtigen die Autoren domänenspezifische Merkmale und geben in ihrem Checklisten-Meta-Modell, das in Form eines Entity-Relationship-Diagramms gegeben ist, Konzepte wie: Item, Category, Event, Organisations, Persons, Resources und Measure an. Zum Beispiel steht Measure für eine bestimmte Aufgabe, die durch einen Satz beschreibender Merkmale gegeben sein soll. Der zentrale Editor soll es den Einsatzkräften ermöglichen, neben den Items einer Checkliste weitere detaillierte Angaben über das Ziel, die Zeitdauer, den Ort oder die genutzten Ressourcen anzugeben. PEINEL et al. wollen somit das Erstellen sog. „smarter Checklisten“ ermöglichen, die die Organisation im Gegensatz zu papierbasierten Checklisten erleichtert. Der „smart“-Faktor soll dabei auf zwei Konzepten beruhen. Zum einen soll das System – wie in [WKK11] motiviert – das Vokabular dem Sprachgebrauch der jeweiligen Organisation des Benutzers anpassen können, und zweitens sollen Ressourcen-Konflikte zwischen der Zuordnung zweier Checklisten erkannt werden.

Wie und auf welcher Basis dies ermöglicht werden soll, bleibt leider unklar. Auch die Art der Ressourcen-Konflikte und das Konzept der Zielmerkmale bleibt noch vage beschrieben. Außer der Angabe eines Entity-Relationship-Diagramms für das Checklisten-Konzept wird jedoch keine genaue Systembeschreibung bzw. keine Architektur des Gesamtsystems angegeben. Zudem erscheint der Anspruch der Modellierung von Ressourcen aufgrund der

Komplexität und Dynamik realer Großschadensereignisse unrealistisch. Nichtsdestotrotz stellt der so entstandene Prototyp einen vielversprechenden Ansatz eines Autoring-Tools dar, mit dessen Hilfe Einsatzkräfte ihre eigenen Checklisten erstellen und editieren können. Für einen Einsatz direkt zur Einsatzzeit sind dieser Prototyp und das vorgestellte Konzept nach Meinung der Autoren jedoch (noch) nicht geeignet.

Nicht zuletzt entstanden im Rahmen des Sicherheitsforschungsprogramms der Bundesregierung, im Bereich „Computeranwendung und Integration in Konstruktion und Planung“ der Universität Paderborn, Arbeiten zum Thema Prozessmodellierung im BOS-Bereich. Die Arbeiten profitierten dort besonders von einer guten Zusammenarbeit mit Berufsfeuerwehren (wie z. B. der Feuerwehr Düsseldorf), wodurch viele Lösungsansätze direkt bei den Anwendern in der Praxis evaluiert werden konnten. So z. B. das Forschungsprojekt SHARE (siehe [KPC06]), dessen allgemeines Ziel es war, durch Bereitstellung einer digitalen Kommunikationsplattform die Effizienz und Effektivität von Feuerwehreinsätzen zu verbessern.

Als ein weiteres Forschungsprojekt verfolgte *Mobis-Pro* ('08 bis '12) eine Optimierung der gesamten Prozesskette vom vorbeugenden zum abwehrenden Brandschutz. LINDEMANN et al. [LPK09] untersuchten hierbei ebenfalls Einsatzmöglichkeiten von EPKs. Die Autoren stellten jedoch fest (siehe [LPK09]), dass es „wegen der theoretisch unendlichen Anzahl von Kombinationen und Variationen einzelner Ereignisse unrealistisch“ ist, alle denkbaren Informationen über die möglichen Prozesse den Einsatzkräften mit an die Hand zu geben bzw. sie im Vorfeld weitgehend vollständig zu erfassen.

Weiterhin schlugen BECKER et al. [BLK11] ein im Rahmen des „I-LOV“-Projektes ('08 bis '11) konzipiertes SOP-gestütztes Informationssystem für den Einsatz in der Einsatzführung des Technischen Hilfswerks (THW) vor. Ziel war es, SOPs der Arbeitsprozesse am Beispiel „Ortung und Bergung von Verschütteten“ zu entwickeln und somit den gesamten Einsatzablauf abzubilden. Die Autoren erkannten jedoch, dass reale Einsätze eine gewisse Flexibilität in ihren SOPs benötigen, und schlugen daher den Einsatz sogenannter „interaktiver SOPs“ vor. Ziel war es, die Vorteile von Checklisten, Flussdiagrammen und die herkömmliche, textbasierte Darstellung der SOP zu vereinen.

Im Kern wird in [BLK11] ein ähnliches System skizziert, wie es bereits in KRÜGER et al. [KKG+10] motiviert wurde. Der Ansatz von BECKER et al. [BLK11] ähnelt jedoch eher einem Versuch, eine Art THW-spezifisches Expertensystem (vgl. [Jac98]) aufzubauen. Über Ja-/Nein-Fragen soll der Nutzer im Einsatz durch die komplexe SOP-Struktur eines kompletten THW-Einsatzes geführt werden.

Das Konzept der interaktiven SOPs war zusätzlich Bestandteil in der Entwicklung des SOP-gestützten Informationssystems SOPHIE (Standard Operation Procedures Hierarchical Information Exchanger), das den Einsatzkräften neben den interaktiven SOPs noch eine Karten- und Kommunikationkomponente bereitstellen soll. Veröffentlichungen zu diesem Thema sind jedoch bis heute nicht zu finden. Auch ist das von BECKER et al. beschriebene Informationssystem recht vage beschrieben. So wird z. B. auf die Frage, wie auf sich ändernde Situationen in den dynamischen SOPs reagiert wird, nur andeutungsweise eingegangen. Ebenso bleibt offen, wie und auf Basis welcher Konzepte eine automatische Zusammenstellung der dynamischen SOP konkret erfolgt.

Allen oben genannten Ansätzen ist die prinzipielle Schwierigkeit der Prozess-Modellierung gemein. Wenn bereits die Modellierung der im Krisenmanagement angewandten Workflows problematisch ist, so verwundert es nicht, dass bis heute (nach Wissen des Autors) im BOS-Bereich auch keine WfMS praktische Anwendung finden. Die Diskussion über ein Für

und Wider von WfMS im Krisenmanagement wird bereits seit Jahren geführt, zumeist motiviert von Verfechtern einer bestimmten Modellierungsmethode/-Sprachen (z. B. EPKs, BPEL oder BPMN) bzw. von Herstellern der WfMS selbst.

Aktuell wird der Einsatz sog. adaptiver WfMS diskutiert (siehe z. B. [RW07; HBS13]), welche den Anforderungen einer Großschadenslage durch dynamische Anpassung an neue Gegebenheiten gerecht(er) werden sollen. Diese Systeme sollen eine ausreichende Flexibilität durch alternative und parallele Workflow-Instanzen bereitstellen. Ein konkretes Beispiel hierfür ist die von HOFMANN et al. vorgeschlagene Architektur eines *Disaster Response Workflow Management Systems* (DRWfMS) [HBS13]. Die Autoren schlagen ein System vor, dass je nach vorliegender Situation initial einen Satz von noch unspezifizierten Prozessbeschreibungen aus einem vorkonfigurierten Repository auswählt. Die richtige Auswahl soll anhand der vorliegenden Fakten und einer geeigneten Beschreibung der Prozesse erfolgen. Zusätzlich soll ein Abgleich mit ebenfalls zuvor hinterlegten Einsatzregeln sicherstellen, dass nur die zu einer bestimmten Einsatzlage passenden Prozesse ausgewählt werden. Die Autoren schlagen hierzu ein *disaster rule repository* (DRR) vor, in dem, neben den Regeln der Anwendbarkeit von Prozessfragmenten, auch die Relationen zwischen verschiedenen Einsatztypen und die Abhängigkeiten der Prozesse untereinander abgelegt sind. Als Kernkomponente des Architekturvorschlages analysiert ein sog. *disaster process analyser* die Informationslage, selektiert die notwendigen Prozesse aus dem DRR und fügt die Prozesse zu einem gesamten „Plan“ zusammen – die Autoren sprechen hierbei von *process tailoring*. Schlussendlich sollen Instanzen der zuvor ausgewählten Prozesse durch ein adaptives WfMS ausgeführt und überwacht werden. Neben HOFMANN et al. beschäftigten sich auch PEINEL et al. mit der Fragestellung einer geeigneten Ausführungsüberwachung von Prozessen im Krisenmanagement [PRW12b].

Alles in allem steht der Ansatz von HOFMANN et al. als ein Beispiel dafür, dass auf Seiten der BPM-Forschung erkannt wurde, dass Prozesse in einem Krisenszenario keine starren, unveränderlichen Prozesse sind. HOFMANN et al. lassen jedoch in ihrem Vorschlag eine Vielzahl von Fragen offen. So bleibt z. B. offen, wie ein DRR genau aufgebaut werden kann und wie es strukturiert ist, wie die Prozess-Templates gewonnen werden und wie vor allem das Verbinden der Fragmente zu einem Gesamtworkflow erfolgt. Gänzlich unausgesprochen bleibt die Frage, wie sich solch ein System in die Arbeitskultur der an einer Großschadenslage beteiligten Organisationen einbinden lässt und wie die automatische Prozessausführung des WfMS mit den physischen Handlungen der Einsatzkräfte abgeglichen werden kann.

3.3 Ansätze der automatische Handlungsplanung

Aus dem Umfeld der automatischen Handlungsplanung finden sich eine Reihe von Arbeiten zur Planungsassistenz in Großschadenslagen und im Krisenmanagement verschiedenster Art (Bekämpfung von Waldbränden [ACF+05b], Katastrophenschutzplanung [BS01], militärische Einsatzplanung [WD92; TDB+04; GHA+05]). Allen Arbeiten ist gemein, dass sie das Paradigma der *Hierarchischen-Task-Netzwerk-Planung* (HTN-Planung) (vgl. SACERDOTI [Sac73], TATE [Tat76] und GHALLAB et al. [GNT04]) anwenden, in dem das Konzept der HTN-*Methoden* als Möglichkeit der Abstraktion von SOPs verstanden wird. Daneben gibt es mit der *Modelreduktion* [Ten88] und der *Operator-Dekomposition* [YM94; YP+94] weitere KI-Planungsansätze, die sich die hierarchische Abstraktion zu Nutze machen [Fox97]. Die HTN-Planung ist hiervon die in der Praxis am erfolgreichsten eingesetzte automatische Planungsmethode [EHN94a; GNT04]. Sie unterscheidet sich jedoch grundlegend von der sog. *STRIPS-style-Planung* in dem **für was** und dem **wie** geplant wird [EHN94b]: „*HTN planners search for plans that accomplish task networks, which can in-*

clude things other than just attainment goals“ [Ero96]. Mit anderen Worten, das Ziel der HTN-Planung liegt nicht darin, einen Weg zur Erreichung einer Menge von Zielmerkmalen zu finden, sondern darin, eine (vordefinierte) Menge von Arbeitsschritten (Tasks) auszuführen (zu *performen*) [GNT04].

Im Folgenden sollen einige wichtige Arbeiten dieses Ansatzes vorgestellt werden. Dabei wird davon ausgegangen, dass der Leser mit den Grundlagen der HTN-Planung vertraut ist. (Das Kapitel 11 in [GNT04] gibt einen umfassenden Überblick zum Thema.)

HICAP – HTN-Planung für militärischer Evakuierung-SOPs

Das System HICAP (Hierarchical Interactive Case-based Architecture for Planning) soll stellvertretend für eine Reihe von Arbeiten aufgeführt werden, die in den 1990er Jahren den Einsatz von sog. *case-based* Planern im Kontext komplexer SOPs für Krisenszenarien untersuchten (vgl. hierzu z. B. auch MITCHELL [Mit97]; GERVASIO et al. [GIL98]).

MUNOZ-AVILA et al. stellen in [MAB+99] das HICAP System als eine Kombination zwischen einer case-based Planung und einer sog. *doctrine-guided task decomposition* vor. Es soll der Erstellung sog. militärischer *Noncombatant Evacuation Operations* (NEOs) dienen. Die Rede ist von der komplexen Aufgabe der Evakuierung einer gefährdeten Region unter der Prämisse, so viele Zivilisten wie möglich unbeschadet aus einer Gefahrensituation zu verbringen. Wie MUNOZ-AVILA et al. darlegen, gilt es hierbei, eine Vielzahl von Regelungen, Vorgaben, SOPs, Ressourcen und Abhängigkeiten zwischen den SOPs zu beachten. Wie die Vergangenheit zeigte, sind die Pläne zu NEOs oft fehlerbehaftet, sodass sie des Öfteren zu einem vermeidbaren Verlust an Menschenleben geführt haben. Ziel von HICAP war es daher, den Verantwortlichen eine Assistenz an die Hand zu geben, die es ihnen ermöglicht, grobe taktische Pläne zu operationalisieren, d. h. den abstrakten und primitiven Tasks konkrete Ressourcen zuzuweisen. Strategische Belange, die meist politische und übergeordnete Zielstellungen verfolgen, wurden in HICAP ausgeklammert.

Neben der klassischen Einteilung in *primitive* und *nicht-primitive* Tasks wurden zusätzlich die *nicht-primitiven* Tasks in „eindeutig-reduzierbar“ (uniquely decomposable) und „mehrdeutig-reduzierbar“ (decomposable by multiple methos) unterschieden. Während die *primitiven* Tasks – wie in der klassischen HTN-Planung auch – konkreten Aktionen entsprechen, stehen die eindeutig-reduzierbaren *nicht-primitiven* Tasks für die Art von Tasks, für die es genau eine entsprechende (durch Lehrmeinung, Gesetz oder Regel) Sub-Task-Reduzierung gibt. In diesem Fall enthalten die Methoden keine Vorbedingung, d. h. die Methode wird unabhängig von der Situation, gemäß einer zu erfüllenden Vorschrift, angewandt. Für die Lösung mehrdeutig-reduzierbarer Tasks existieren hingegen mehrere Methoden mit entsprechenden Vorbedingungen. MUNOZ-AVILA et al. sprechen hierbei von einem spezifischen Problemlöse-Kontext, der als ein *Case-based* Planungsproblem aufgefasst wird. Methoden für problemspezifische Tasks repräsentieren *Cases*, deren Anwendungsbedingung aus einer Menge von Frage-Antwort-Paaren besteht. Die Cases selbst sind auf Basis von SOPs oder aus früheren erfolgreich angewandten (Teil-)Tasknetzwerken erarbeitet worden. Abbildung 3.2 zeigt eine HICAP Nutzerschnittstelle, in der auf der linken Seite eine hierarchische Tasks-Liste einer konkreten NEO-Mission zu sehen ist. Die rechte Seite stellt die Organisationsstruktur der Evakuierungsteams dar.

In HICAP wurden eine HTN-Wissensbasis (bestehend aus kritischen Tasks) aufgebaut, indem eine Menge von Dienstvorschriften, Handbüchern und SOPs analysiert wurden. Detaillierte Beispiele finden sich hierfür jedoch nicht (mehr). HICAP konzentrierte sich vorwiegend auf eine Ressourcen-Zuweisung, sodass das HTN-Planungssystem hauptsächlich für Probleme eingesetzt wurde, die eher dem Bereich Scheduling als dem der Planung zuzu-

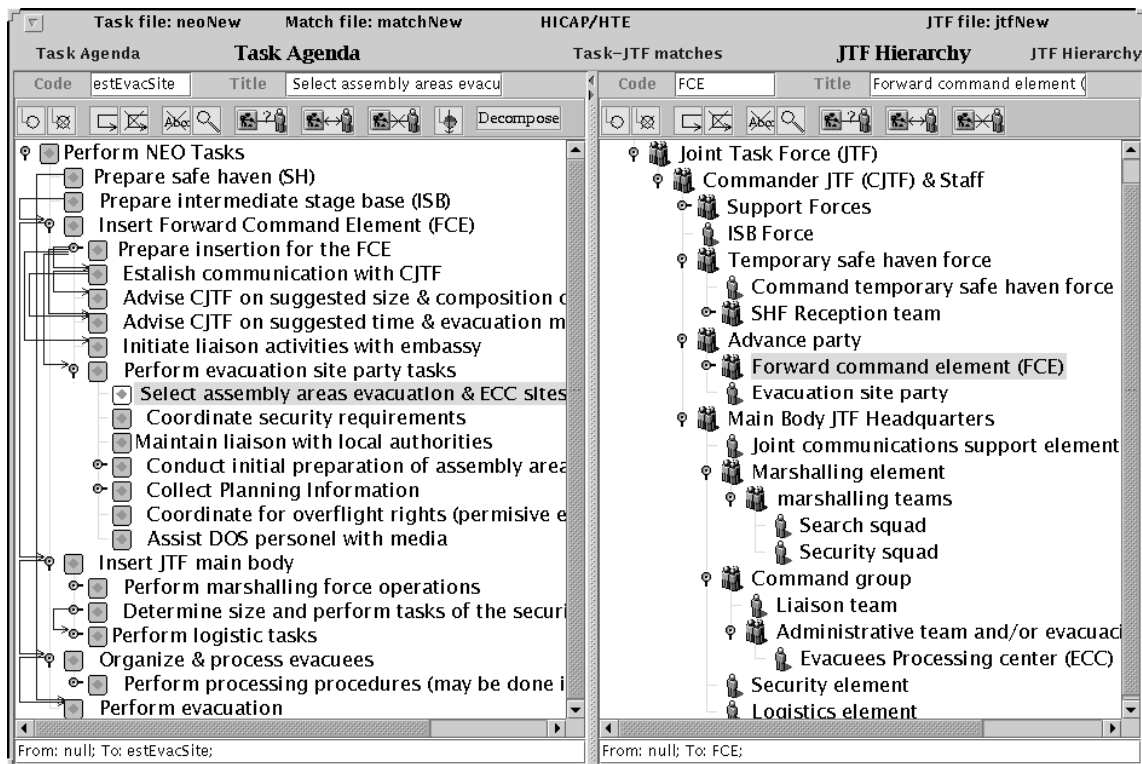


Abbildung 3.2: Anzeige der Task-Hierarchien im HICAP-Benutzerinterface. (Quelle: [MAB+99])

ordnen sind. In der Literatur wird Scheduling oft als ein Spezialfall der Handlungsplanung angesehen, der der eigentlichen Planung nachgelagert ist (vgl. [GNT04], Kap. 15).

CoSAR-TS – Coalition Search and Rescue Task Support

Das von TATE et al. [TDD96] entwickelte HTN-Planungssystem O-Plan soll menschliche Agenten in verschiedensten Arbeitsgebieten bei der Abarbeitung ihrer Tasks (Issues) unterstützen. Die Möglichkeiten der Anwendung von O-Plan und dessen Erweiterungen (siehe z. B. das *I-X*-System [Tat00]) wurden in den letzten 10 Jahren auch für die Domäne des Krisenmanagements untersucht. So auch im *Coalition Search and Rescue Task Support* (CoSAR-TS; '03 bis '06) Projekt¹. Im CoSAR-TS-Projekt wurde unter anderem die Arbeitsweise von intelligenten Softwareagenten im Zusammenhang mit einem KI-Planungssystem und dem Einsatz von Policies (im Sinne von Strategien, Taktiken oder Politiken) untersucht. Ziel des Projektes war die Integration des KI-Planers *I-X/I-Plan* mit den *KAoS Policies* in einer *Search & Rescue* (SAR) Service-Domäne. Bei *KAoS Policies* handelt es sich um ein spezielles Framework für Domänen und Policy Services (siehe [UBJ+03; UBJ+04]). Policies sind in CoSAR-TS in wohldefinierten, deklarativen Beschreibungen hinterlegt und regeln die Zugriffsmöglichkeiten von (elektronischen) Services verschiedenster Art.

Es liegt in der Natur von SAR-Operationen, dass eine sich rasch verändernde Umgebung eine gleichfalls schnelle Anpassung der zur Ausführung geplanten *policy-constrained* Services erfordert. Die Planungskomponente in CoSAR-TS soll dies durch eine *Semantic Web Services workflow composition* [TDB+04] unterstützen, bei der die Policies mit berücksichtigt werden.

¹ siehe <http://www.aiai.ed.ac.uk/project/cosar-ts/> (abgerufen am 20.12.2013)

Dem Benutzer stehen über ein sog. *Process Panel* Funktionalitäten eines KI-Planers bereit. Mit ihm können kontextsensitive Operationsfolgen zur Erfüllung eines festgelegten Zieles mit Blick auf die Performance und die einzuhaltenden Constraints geplant werden. Alle Services sind mit der Service-Modellierungs-Sprache OWL-S (vgl. [OWL-S]) beschrieben und mittels eines Konverters in eine geeignete Repräsentationsform für den KI-Planer umgewandelt.

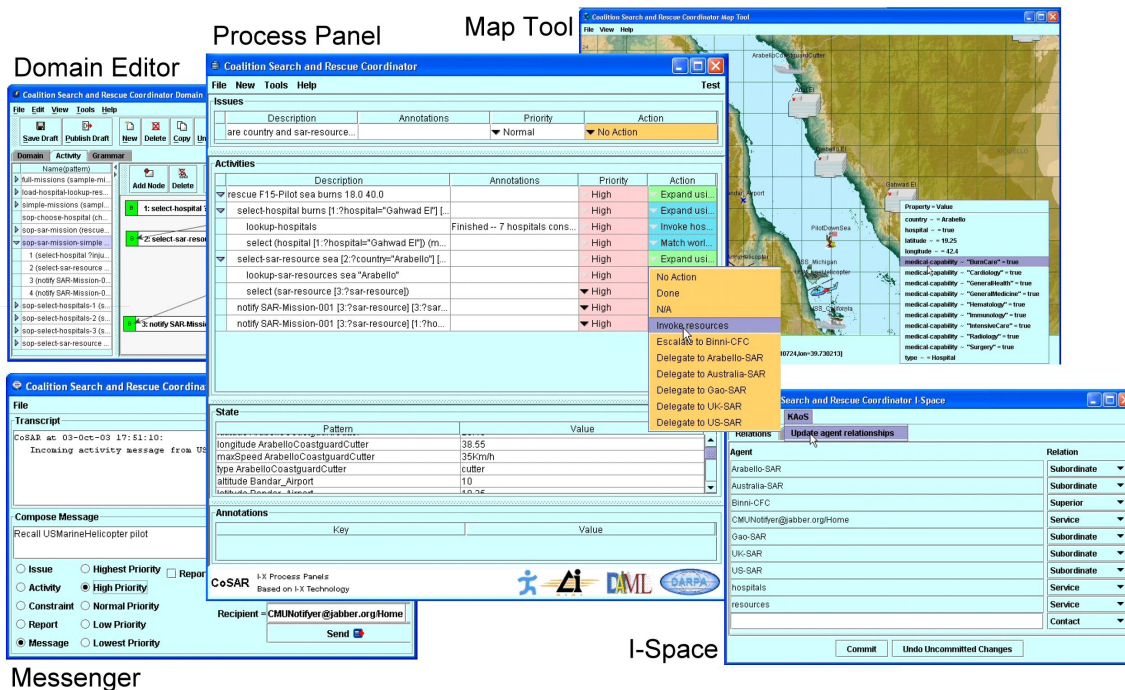


Abbildung 3.3: Einzelne Komponenten des CoSAR-TS GUI. (Quelle: [TDB+04])

In Abbildung 3.3 sind die Interface-Komponenten des CoSAR-TS Projektes zu sehen. So erlaubt ein *Domain Editor* das Erstellen, Verwalten und Bekanntmachen von HTN-Methoden, unter denen typische Vorgehensweisen zur Lösung komplexer und einfacher Problemstellungen verstanden werden. Über das *Process Panel* können die anfallenden Aufgaben je nach ihrer Priorität an die ausführenden Einheiten verteilt/ihnen zugewiesen werden. Gleichzeitig bietet das *Process Panel* einen Überblick über die aktuell laufenden Maßnahmen und zeigt die noch anstehenden unbearbeiteten Tasks an. Die Autoren sprechen in diesem Zusammenhang auch von der Bereitstellung von „*distributed and intelligent to-do lists*“ [WTP06]. Das *Map Tool* bietet eine geografische Lageübersicht aller sich im Einsatz befindlichen Einheiten.

Obwohl zu diesem Projekt einige Veröffentlichungen zugänglich sind, bleiben viele Details der Umsetzung unklar bzw. vage beschrieben. Auch ist außer einer Demonstration per Video keine frei verfügbare Applikation zum Test oder gar zur Weiterentwicklung verfügbar. Es bleibt zu vermuten, dass dies dem militärischen Anstrich des Projektes geschuldet ist. Auf den Webseiten des AIAI (Artificial Intelligence Applications Institute School of Informatics, University of Edinburgh) lassen sich jedoch Folgeprojekte finden. So greifen z. B. WICKLER und POTTER den Ansatz der HTN-Planung in [WP10] erneut auf und schlagen ein Wiki-basiertes System zur kollaborativen Entwicklung und Anwendungsassistentz von SOPs vor. Durch das erweiterte Wiki-System sollen die Domänenexperten auf einfache Art und Weise die formale Methodendefinition für das *I-X*-System erstellen können. Jede einzelne Wiki-Seite soll einer Task entsprechen und jede Wiki-Unterseite soll einer Sub-Task entsprechen.

SIADEx – HTN-Planung in der Waldbrandbekämpfung

Im Projekt SIADEx (Assisted Design of Forest Fire Fighting Plans by means of Artificial Intelligence Techniques) [AGP04; ACF+05b] wurde ein auf KI-Planungstechniken basierendes Assistenzsystem für das Generieren von Plänen zur Waldbrandbekämpfung entwickelt. Das System soll die manuelle Erarbeitung von Einsatzplänen durch eine HTN-Planungskomponente unterstützen. Die dynamische Einsatzdomäne von SIADEx beinhaltet Schwierigkeiten wie: plötzliche Ereignisse, eine sich schnell ändernde Umgebung und fehlende bzw. falsche Informationen über die Domäne. Darüber hinaus können Aktionen nicht den gewünschten Erfolg erzielen oder unerwartet nicht ausgeführt werden.

Die Architektur von SIADEx soll laut den Autoren all diesen Unwägbarkeiten Rechnung tragen [ACF+05b]. Das vorgeschlagene System soll zum einen als virtuelle Lernumgebung in Form einer Einsatzsimulation für taktische Waldbrandbekämpfung, zum Training von Einsatzleitern und zur Evaluation von Einsatzstrategien genutzt werden. Außerdem soll das System in realen Löscheinsätzen zur intelligenten Entscheidungsunterstützung – *intelligent decision support system* (IDSS) – eingesetzt werden. Besonderes Augenmerk wurde dabei auf ein *human-centred* Planungsframework gelegt, das eine einfache Interaktion von typischerweise KI-unbedarften Endanwendern mit dem Planungssystem ermöglicht. Mit dem Paper *Bringing Users and Planning Technology Together* [FCG+06] gewannen die Autoren diesbezüglich auf der ICAPS (International Conference on Automated Planning and Scheduling) 2006 den „Best Application Award“.

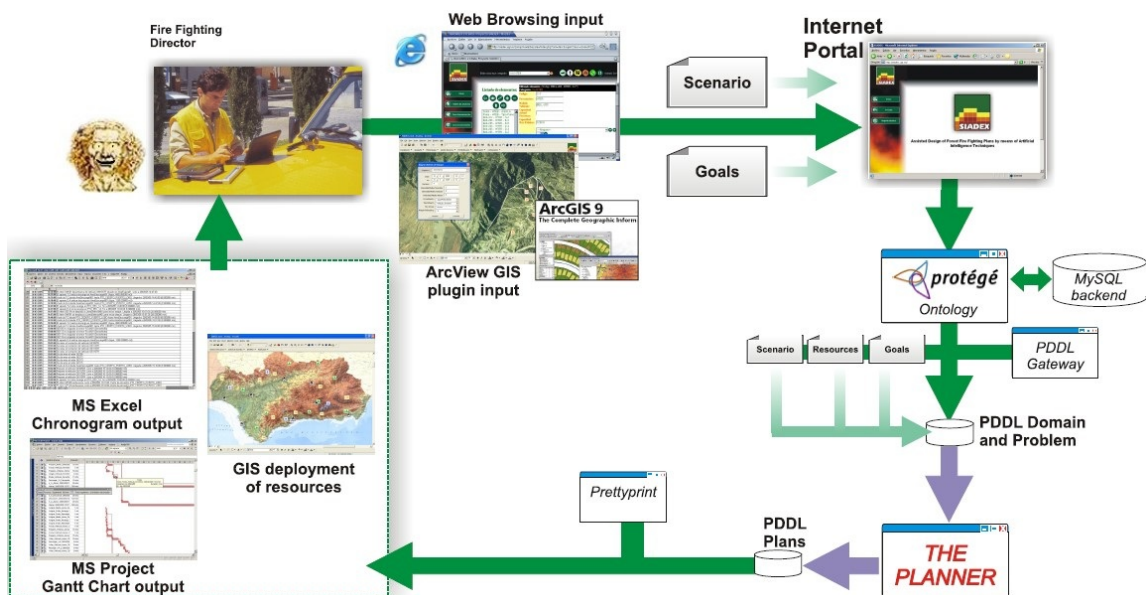


Abbildung 3.4: Der sog. *lifecycle* von SIADEx in einem Waldbrandszenario. (Quelle: [FCG+06])

In SIADEx wurde ein zustandsbasierter *Vorwärtsplaner* auf Basis des HTN-Planers SHOP2 (siehe [NAI+03]) integriert, dessen Vorgänger (J)SHOP auch schon im HICAP-Projekt eingesetzt wurde. Der Planer unterstützt an temporalen Konzepten die *Dauer* von Operatoren und die Angabe möglicher *Deadlines* in den Zielvorgaben. Die temporale Erweiterung der Planungsfunktionalität in SIADEx beruht auf dem von NAU [NAI+03] vorgeschlagenen *Multi-Timeline Preprocessing* (MTP) [NAI+03], einem speziellen Algorithmus, der eine zeitbehaftete PDDL (Planning Domain Definition Language) Operatorbeschreibung in eine zu SHOP2 konforme Darstellung umwandelt (siehe auch [CFG+06]). Abbildung 3.4 zeigt das Zusammenspiel aller Komponenten.

Eine weitere Schlüsselkomponente der Architektur stellt der Ontologie-Server BACAREX dar, in dem die Objekte der Planungsdomäne in Form von Individuen einer Wissensbasis repräsentiert sind [ACF+05a]. In BACAREX wurde die Wissensrepräsentation in vier verschiedene Ontologien aufgeteilt: *Domain Object*, *World & Object States*, *Temporal Constraints* und *Tasks*. In ihnen werden die Namen der Objekte der Domäne (wie z. B. Fahrzeuge, GIS-Daten und Orte), dynamische Eigenschaften (Prädikate/Fluents der Planungsdomäne), zeitliche Bedingungen (max. Flugdauer oder Einsatzdauer von Feuerwehrcräften) und die sog. *Standard Operation Protocols* (Task-Hierarchie) abgelegt. Der Ontologie-Server benutzt als Backend eine relationale Datenbank, wodurch eine persistente Datenspeicherung erreicht wird. Gleichzeitig werden durch die üblichen Logging- und Recovery-Mechanismen quasi „parallele“ Zugriffsmöglichkeit von verschiedenen Benutzern auf die Daten sichergestellt. In der Wissensbasis wurden wichtige Begriffe und offizielle Brandbekämpfungsvorschriften und Handlungsstrategien der Feuerwehr sowie geografische Informationen (GIS) über gefährdete Region abgelegt.

Als Inferenzmechanismen kommen in SIADEx *Abduktion* (Hypothese vom Einzelnen auf eine Regel und das Allgemeine) und *Deduktion* (Schluss vom Allgemeinen auf das Einzelne; Wenn-Dann-Aussagen) zum Einsatz. Die Deduktionsfunktionalität kann bei Bedarf durch spezielle Inferenz-Tasks im eigentlichen Dekompositionsprozess getriggert werden. Als Effekt dieser Inferenz-Tasks sollen Variablenbindungen oder *Assertions/Retracts* im jeweiligen Zustand resultieren.

Zugriff sollen alle Beteiligten über eine Web-Plattform erhalten, über die alle zur Verfügung stehenden Informationen eingesehen werden können. Ein Web-Browser und eine Internetverbindung stellen laut den Autoren die einzigen technischen Anforderungen an die Benutzer-PCs dar. Zur Repräsentation von Einsatzplänen wurden verschiedene den Nutzern vertraute Formate gängiger Office-Anwendungen verwendet.

Hybrider Planungsansatz für THW-Einsätze

Aus der Forschungsgruppe rund um BIUNDO und SCHATTENBERG (Institut für Künstliche Intelligenz der Universität Ulm) stammen Arbeiten rund um die Thematiken: (HTN)-Planung, Scheduling, Hybrid-Planung und ontologiebasierte Middleware für Planung und Scheduling [Sch08; SB02; SBG+09]. In [BS01] thematisieren BIUNDO & SCHATTENBERG den Einsatz der *Hybrid-Planung* als Basis eines Unterstützungssystems für das Krisenmanagement in der Domäne der Bundesanstalt Technisches Hilfswerk (THW). Unter Hybrid-Planung verstehen die Autoren die Kombination von HTN-Planung mit zustandsbasierter *Partial-Order-Planung* (POP). BIUNDO et al. sprechen hierbei von *Partial-Order Causal-Link* (POCL)-Planung.

Im Rahmen eines Meta-Projektes Namens PANDA (*Planning and Acting in a Network Decomposition Architecture*) wurden Fragen untersucht, die die Erstellung komplexer HTN-Planungsdomänen, die Unterstützung durch geeignete Werkzeuge, die Konsistenzprüfung komplexer Domänen und die Einbeziehung des Nutzers bei der manuellen Planverfeinerung und Reparatur betreffen. Das dabei entwickelte PANDA-System soll erstmals eine konsequente Integration von POCL-Plänen, HTN-Plänen und Scheduling realisieren [Sch08]. Führt die HTN-Planungskomponente zu einem noch nicht vollständig anwendbaren Plan, so soll in einem zweiten Schritt die „konventionelle“ POP zur Vervollständigung des Planfragmentes herangezogen werden. Offene Vorbedingungen sollen durch den Aufbau kausaler Beziehungen bzw. durch eine Einführung (abstrakter) Tasks in den Plan geschlossen werden (vgl. [BS01]).

PANDA hat als Anwendungsdomäne u. a. reale THW-Einsätze im Fokus, ähnlich den im Oderbruch-Hochwasser 1997 durchgeführten Maßnahmen. In solch einem Szenario sind

eine Vielzahl von qualitativ unterschiedlichen Aufgabenfeldern wie Logistik, Konstruktion, Organisation und Workflows zu managen, wobei laut Autoren von einer Beteiligung von bis zu 180 Gruppen, 1500 Helfern und einer Einsatzdauer von mehreren Wochen ausgegangen werden kann [Sch08]. Die Autoren kommen zu dem Schluss, dass es nicht selbstverständlich ist, für solch eine komplexe Domäne ein widerspruchsfreies und damit konsistentes Domänenmodell zu erstellen und zu administrieren.

SCHATTENBERG et al. schlagen daher – ähnlich dem Process-Panel in HICAP oder dem Domänen-Editor in CoSAR-TS – eine geeignete Tool-Unterstützung für die Modellierung konsistenter Domänenmodelle vor [SBB08]. Der Task-Editor soll den Benutzer über noch bestehende Inkonsistenzen und mögliche Fehler im Domänenmodell bereits während der Konfigurierung informieren. Der PANDA-Editor enthält Visualisierungstools (PANDA-Viz), mit denen der Planungserfolg visualisiert und eine Unterstützung für sog. „Plan-Entwicklungszyklen“ gegeben werden kann. Während bei herkömmlichen, meist kommandozeilenbasierten KI-Planern bei einem nichtlösbaren Planungsproblem der Grund nicht ersichtlich ist, soll PANDA-Viz durch geeignete Darstellung eines Planfragmentes die noch offenen Teile geeignet visualisieren. Ein Domänenexperte soll auf Basis dieses noch unvollständigen Planes eine manuelle Plan-Vervollständigung außerhalb der eigentlichen KI-Planungsroutine vornehmen können [Sch08]. Abbildung 3.5 zeigt einen beispielhaften Auszug aus einer Task-Hierarchie im PANDA-Projekt.

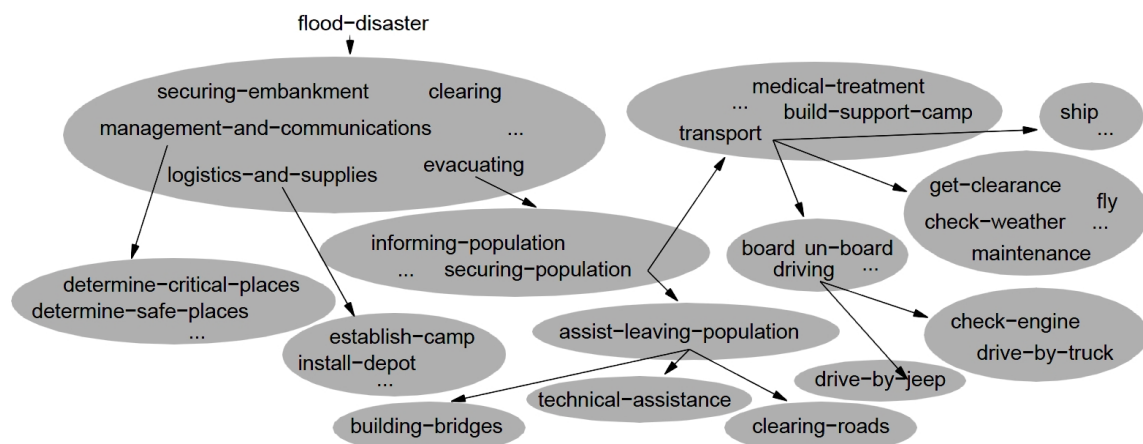


Abbildung 3.5: Auszug einer Task-Hierarchie aus Sicht des THW. (Quelle: [BS01])

Bezüglich der POCL-Planung stellt sich jedoch die Frage nach der Anwendbarkeit in der Praxis. Ohne explizite Ziele kann auch keine „pseudo“-Zielaktion angegeben werden, welche aber für eine POP benötigt wird. Obwohl die Autoren ausdrücklich auf den Einsatz ihrer Arbeit in der realen THW-Domäne hinweisen, erscheint das Domänenmodell konstruiert. Es bleibt der Eindruck, dass sich die Planungsdomäne eher den Anforderungen des vorgeschlagenen Planungsframeworks anpasst, als das Planungsframework den Anforderungen der realen THW-Domäne. Erfahrungs- und Evaluationsberichte bezüglich eines realen THW-Einsatzes sind nicht verfügbar.

3.4 Fazit und Motivation des Checklisten-Ansatzes

Die Bemühungen des BPM zielen hauptsächlich auf die Modellierung der Management-Prozesse bei Großschadenslagen ab. Wie sich jedoch zeigte, ist im BOS-Bereich eine Prozessmodellierung ungleich schwieriger als in klassischen BPM-Domänen. Gründe hierfür liegen in der hohen Komplexität und der Dynamik eines Einsatzgeschehens. PEINEL et al.

sehen die Komplexität der momentan zur Verfügung stehenden Modellierungsumgebungen als einen weiteren Hindernisgrund der Anwendung von BPM auf die BOS-Domäne [PR09]. Hinzu kommt, dass sich im BOS-Bereich kaum detaillierte Prozesse identifizieren lassen, da die ablaufenden Einsätze von Fall zu Fall unterschiedlich ausgeprägt sind. Falls Prozesse definiert sind, dann nur auf der recht abstrakten Ebene der Regelwerke (z. B. den Dienstvorschriften). Ein zu prozeduraler SOP-Charakter, wie er z. B. in [BLK11] vorgeschlagen wird, ist im BOS-Bereich eher hinderlich. Allen vorgestellten BPM-Ansätzen mangelt es an präzisen Beschreibungen dessen, wie die vorgeschlagenen Systeme aufgebaut sind und wie die Interaktion mit den Einsatzkräften genau abläuft – was natürlich auch der frühen Phase dieser Ansätze geschuldet ist. Hier lohnt es sich, die weiteren Fortschritte und zukünftige Praxiserfahrungen im Auge zu behalten.

Die KI-Ansätze gehen von der Annahme aus, dass es eine hinreichend große Menge definierter Prozesse gibt, sodass sich bei Angabe der Toplevel-Tasks automatisch ein kompletter „Lösungsplan“ für das Vorgehen bei einer speziellen Krise berechnen lässt. Dies scheint mit Blick auf die dynamische und heterogene BOS-Domäne nicht realistisch. Auch ist das Vorgehen nach festen Plänen in der BOS-Domäne wegen der hohen Dynamik kaum möglich. Wie die Erfahrungen im SPEEDUP-Projekt zeigten, ist zudem eine Erfassung aller Ressourcen (Einsatzkräfte und Einsatzmittel) in der für eine HTN-Planung notwendige Form unpraktisch oder aus Informationsmangel gar unmöglich. Hinzu kommt, dass die Domäne nicht statisch und nicht endlich ist. Auch sind die Zielvorgaben in Großschadensereignissen oft nicht ausreichend gut formulierte bzw. müssen ständig angepasst werden.

In dieser Arbeit soll als Alternative zur Planung und Prozessmodellierung ein anderer Weg eingeschlagen werden. Die vorliegende Arbeit konzentriert sich in den kommenden Kapiteln auf die Erarbeitung eines checklistenorientierten Ansatzes einer Handlungsassistenz bei der Anwendung von SOPs. In einigen der oben vorgestellten Arbeiten wurde bereits (direkt oder indirekt) das Thema Checklisten-Anwendung motiviert (vgl. z. B. „interactive SOPs“ in [BLK11], „smart Checklists“ in [PRW12b] oder „distributed and intelligent todo lists“ in [WTP06]). Keiner der gezeigten Ansätze geht jedoch auf Fragen der Funktion und Anwendung von Checklisten genauer ein. Ebenso wird der gewünschte Effekt einer Fehlervermeidung von keinem der genannten Ansätze thematisiert. In vielen der oben betrachteten Arbeiten bleibt unklar, inwieweit die Lösungsansätze den Praxisanforderungen gerecht werden.

Im Bereich der Sicherheitsforschung (als Teilbereich der Humanwissenschaften) sind Checklisten als kognitives Hilfsmittel bereits seit längerer Zeit ein Thema. Da Checkliste nicht gleich Checkliste ist, soll im kommenden Kapitel ein Blick auf relevante Aspekte der Human-Factors-Forschung zeigen, was genau hinter dem Konzept Checkliste steckt und welche bekannten Phänomene bei einer Computerunterstützung von Checklisten beachtet werden sollten.

Kapitel 4

Checklisten – eine Human Factors Perspektive

„No matter how expert you may be, well-designed checklists can improve outcomes.“ (STEVEN D. LEVITT)

In diesem Kapitel sollen Erkenntnisse aus dem seit den letzten Jahren an Bedeutung gewinnenden *Human-Factors*-Forschungszweig der Humanwissenschaften betrachtet werden, welche Fragen der Interaktion von Menschen in komplexer und dynamischer Arbeitsumgebung (auch komplexes, soziotechnisches System genannt) betreffen. Nachfolgend wird speziell auf das Checklisten-Prinzip eingegangen, das zur Handhabung von Komplexität und zur Einhaltung standardkonformer Arbeitsschritte verwendet wird.

4.1 Strategien der Fehlervermeidung & Handlungsassistenz

Menschliches Versagen ist ein zentrales Thema der *Human-Factors*-Forschung, die sich mit der menschlichen Komponente in komplexen Systemen beschäftigt. Sie leitet daraus u. a. Strategien zur Fehlervermeidung ab. In dieser Arbeit sind die Strategien zur Fehlervermeidung bei der Ausführung von Standardprozeduren von besonderem Interesse.

Bevor auf diese Strategien eingegangen wird, soll das Konzept „Fehler“, das bis jetzt eher allgemein gebraucht wurde, im Kontext der Human-Factors-Forschung und der Zielstellung dieser Arbeit präzisiert werden. Von RASMUSSEN [Ras82], NORMEN [Nor88] und REASON [Rea90] stammen grundlegende theoretische Arbeiten zur Thematik „menschliche Fehler“, aus denen entsprechende Fehlermodelle hervorgegangen sind. REASON (vgl. [Rea90] S. 9) definiert z. B. den Begriff Fehler (error) als: „a generic term to encompass all those occasions in which a planned sequence of mental or physical activities fails to achieve its intended outcome“. Beispielhaft orientiert sich die vorliegende Arbeit am Fehlermodell von REASON, der in seinem Buch „Human Errors“ [Rea90] zwei grundlegende Fehlertypen anhand der Ebenen *Planung*, *Speicherung* und *Ausführung* unterscheidet. REASON folgt in seiner Arbeit dem als „hybride Klassifikation“ bekannten Fehlermodell von NORMAN [Nor88], in dem kognitive und motorische Fehler-Aspekte eine Einteilung in zwei Arten von Fehlern begründen. In diesem Modell werden auf der Ausführungsebene Handlungen, die nicht planmäßig verlaufen, als *Patzer* oder *Schnitzer* (*slips*) bezeichnet. Während Pläne, die gemäß der Zielstellung in einer entsprechenden Situation nicht angemessen sind, von ihm als eigentliche *Fehler* (*mistakes*) bezeichnet werden (vgl. [Rea90] S. 255).

Folgt man obiger Einteilung, so wird deutlich, dass das Auslassen wichtiger, gebotener Handlungen (gemäß den SOPs) in erster Linie dem Fehlertyp Patzer (bzw. Schnitzer) zuzurechnen ist. Erst auf der Ebene der Evaluierung von SOPs kann deren Unangemessenheit gemäß der intendierten Zielstellung als ein Fehler im Sinne von REASON aufgefasst werden. Handlungspatzer treten nach REASON typischerweise bei routinierten Handlungen in vertrauten Arbeitsumgebungen immer dann auf, wenn die Aufmerksamkeit sinkt oder man durch externe Ereignisse abgelenkt ist.

In der weiteren Arbeit soll für das unbeabsichtigte sowie das beabsichtigte Auslassen von Handlungen der Einfachheit halber jedoch weiterhin die Bezeichnung „Fehler“ verwendet werden. Die Einteilung von REASON sollte dabei allerdings im Hinterkopf behalten werden. Weiterhin wird sich im Folgenden nur noch auf die Ausführungsfehler konzentriert.

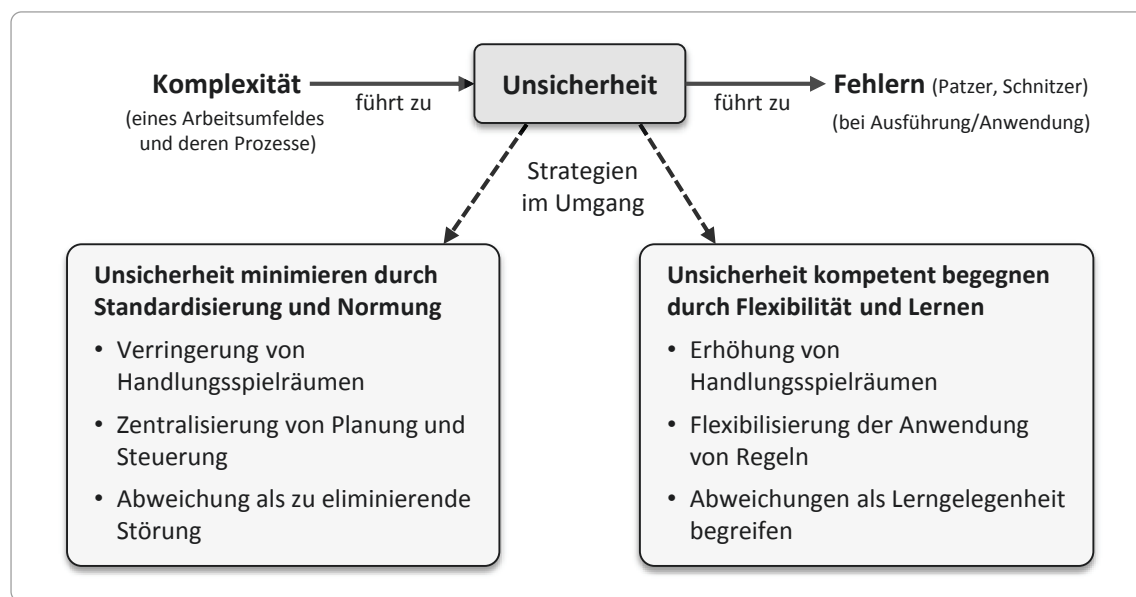


Abbildung 4.1: Grundlegende Strategien im Umgang mit Unsicherheit (Quelle: [Man09]).

Nicht nur die Gefahr von sinkender Aufmerksamkeit während Routinearbeiten hat die Human-Factors-Forschung als Ursache menschlicher Fehler identifiziert, sondern auch die Unsicherheit in den Arbeitsprozessen, die sich aus den Merkmalen komplexer Systeme ergibt.

MANSER [Man09] führt zwei grundlegende Tendenzen des Umgangs mit Unsicherheit in den Organisationen auf (siehe Abb. 4.1). Die Strategie der Standardisierung setzt auf eine Reduktion operativer Handlungsspielräume und verringert somit die subjektiv empfundene Komplexität der Prozesse aus Sicht des Handelnden. Somit wird die Unsicherheit für den Handelnden reduziert, wodurch letztendlich mögliche Abweichungen von der „best practice“ vermieden werden sollen.

In diesem Zusammenhang hat sich der Begriff „Komplexitätsreduktion“ etabliert, der jedoch etwas unglücklich gewählt ist. Er bedeutet nämlich nicht – wie der Begriff suggeriert – eine Vereinfachung des Arbeitsumfeldes und der Prozesse selbst, sondern vielmehr eine Abstraktion im Umgang mit einem komplexen Arbeitsumfeld.

Eine zur Standardisierung entgegengesetzte Strategie setzt auf eine bewusste Flexibilisierung. Gemeint ist eine bewusste Erweiterung von Handlungsspielräumen und eine flexible Anwendung von Regeln. Diese Strategie vertraut auf die menschlichen Problemlösefähigkeiten und sieht notwendige Abweichungen als eine Chance zum Lernen und zur flexiblen Anpassung an die Gegebenheiten an (vgl. z. B. [Man09]).

Beide Strategien haben ihre Vor- und Nachteile. Abstrahiert man Komplexität, so besteht immer die Gefahr, komplizierte Zusammenhänge zu weit zu vereinfachen, sodass mögliche kritische Punkte aus den Augen verloren werden. Auf der anderen Seite stößt die Strategie der Kompetenzerhöhung und Flexibilisierung in den Situationen an ihre Grenzen, in denen es gilt, Richtlinien (Gesetze, Vorschriften, SOPs) zu befolgen.

Die Strategie der flexiblen Regelanwendung erscheint auf den ersten Blick dem BOS-Bereich und den Merkmalen einer Großschadenslage angemessen. Wie in Kapitel 2 gezeigt wurde, ist im BOS-Bereich jedoch eine gewisse Standardisierung durch die Dienstvorschriften und Gesetze gegeben. Diese gilt es trotz einer notwendigen Flexibilität der Handlungen zu beachten.

Durch regelmäßiges Training von Großschadenslagen soll sichergestellt werden, dass jede Einsatzkraft gemäß den Vorschriften korrekt agiert oder zumindest nur soweit davon abweicht, wie es die spezielle Lage erfordert und es die Regeln noch erlauben. KLUGE et al. weisen in ihren Untersuchungen (siehe [KG12; KGB12]) jedoch darauf hin, dass der Trainingserfolg gerade für Handlungsrouninen, die längere Zeit nicht angewendet werden (z. B. die in einem MANV), in der Regel sehr bescheiden ausfällt. Als alternative Herangehensweise geht KLUGE daher davon aus, dass niemanden zugemutet werden sollte, „*sich alles auch über längere Phasen des Nichtgebrauchs zu merken*“ [KG12]. Vielmehr sollen sog. „Job-Aids“ als Arbeitshilfe die menschliche Fähigkeit, Informationen zu speichern und zu verarbeiten, erweiternd unterstützen. Laut ROSSETT & GAUTIER-DOWNES [RG91] sind diese kognitiven Hilfsmittel gerade für die Aufgaben sinnvoll, die unregelmäßig und selten ausgeführt werden, die komplex und umfassend sind, und in denen Fehler folgenschwere Konsequenzen haben können.

Ein pragmatischer Lösungsansatz zur Fehlervermeidung stammt von REASON. In Anbetracht der Feststellung, dass in der Nuklearindustrie viele Unfälle auf eine fehlerhafte Ausführung von Wartungsprozessen zurückzuführen waren, stellten sich REASON und sein Forschungsteam bereits in den 1990er Jahren die Frage, ob eine Art elektronischer Checklisten hierbei nicht hilfreich sein können (vgl. [Rea90], Kap. 8, S. 239 ff.). Er schlug hierzu ein computer-integriertes Checklisten-System vor, das einem Arbeiter bei komplexen und sicherheitskritischen Wartungsprozeduren assistierend zur Seite stehen soll. In seinem Labor entwickelten er und seine Kollegen hierfür ein System namens PIMA (*Portable Interactive Maintenance Auxiliary*) (vgl. [Rea90] S. 240 ff.), das auf einem frühen Laptop-Vorläufer (Epson PX-8) basierte. Laut REASON hatte dieses frühe System bereits die Fähigkeit, sich mit dem Überwachungssystem der Nuklearanlage zu verbinden und somit z. B. die Checklisten aufgrund der dort bereitgestellten Sensordaten automatisch anzupassen.

Bevor im nächsten Kapitel die Idee der elektronischen Checklisten aufgegriffen wird, soll im folgenden Abschnitt das Grundprinzip hinter Checklisten genauer vorgestellt werden, da es in der weiteren Arbeit eine Schlüsselrolle bei der Bereitstellung einer Computer-unterstützten Handlungsassistenz darstellt.

4.2 Checklisten als kognitives Hilfsmittel

Checklisten sind vielen von uns ein wohlvertrautes Konzept. Ohne groß über sie nachzudenken, nutzen wir sie in manch Alltagssituationen. Doch die Wenigsten von uns machen sich Gedanken über das Grundprinzip hinter den Checklisten und dessen korrekte Anwendung. Zu trivial scheint dieses Hilfsmittel zu sein. Doch sind Checklisten wirklich ein so einfaches und leicht anzuwendendes Konzept? In der Literatur und in der Praxis gehen die Meinungen darüber, *was* alles unter einer Checkliste zu verstehen ist und *ob*, *wo* und *wie* sie einzusetzen sind, weit auseinander.

4.2.1 Was sind Checklisten? – eine Arbeitsdefinition

Die Vorstellungen, was Checklisten genau sind und zu welchem Zweck sie eingesetzt werden, variieren von Anwendungsgebiet zu Anwendungsgebiet mehr oder weniger stark. Die folgenden Definitionsversuche aus dem WWW geben einen Eindruck über die vielfältigen Vorstellungen über Checklisten.

- „Eine Checkliste ist eine Sonderform einer Liste mit dem Ziel, alle relevanten Fragen zu einem Thema an zentraler Stelle zusammenzufassen. Sie bieten ein Grundgerüst an kritischen Punkten, die bei einer bestimmten Tätigkeit zu prüfen sind. Bei den Fragen handelt es sich um geschlossene Fragen [sic] die schnell und zweifelsfrei beantwortet werden können.“ (www.openthesaurus.de; abgerufen am 20.12.2013)
- „A checklist is a type of informational job aid used to reduce failure by compensating for potential limits of human memory and attention. It helps to ensure consistency and completeness in carrying out a task.“ (en.wikipedia.org; abgerufen am 20.12.2013)
- „Checklist: A list of tasks to be completed, names to be consulted, conditions to be verified and similar.“ (wiktioary.org; abgerufen am 20.12.2013)

Wissenschaftliche Publikationen zum Thema Checklisten finden sich jedoch kaum. Eine Ausnahme stellt der Luftfahrtbereich dar. So definieren z. B. M. R. GRÜNNINGER et al. Checklisten wie folgt. „A checklist is a device designed to support the pilots to avoid forgetting vital items and actions while following standard operating procedures“ ([GKB10]; S. 78). Allgemein lässt sich festhalten, dass als Checkliste (englisch *check-list*, aus: *check* = Kontrolle, prüfen und *list* = Liste) umgangssprachlich ein kognitives Hilfsmittel (englisch *mnemonic device*) bezeichnet wird, das einem Akteur bei einer Handlung als Kontrolle dient. Je nach Einsatzgebiet werden Checklisten auch synonym als Abhaklisten, Kontrolllisten oder auch Prüflisten bezeichnet (siehe z. B. [GLB86]). Im privaten Umfeld begegnen uns Checklisten z. B. in Form einer Einkaufsliste, einer einfachen Eselsbrücke oder einer *To-Do*-Liste. In kritischen Arbeitsbereichen, wie z. B. der Luftfahrt oder der Prozessindustrie (z. B. Chemie), nehmen sie eine besondere Schlüsselrolle bei der Fehlervermeidung ein.

In all diesen Anwendungen soll etwas – im Auge des Anwenders – Wichtiges auf ein mögliches und zu vermeidendes „Vergessen“ hin geprüft werden. Was dieses *Etwas* genau ist, lässt sich erst anhand einer konkreten Anwendung festmachen. Der Definitionsversuch von M. SCRIVEN ([Scr07]; S. 1) macht die prinzipielle Vielfältigkeit dessen, was überprüft werden soll, deutlich: „A checklist is [...] a list of factors, properties, aspects, components, criteria, tasks, or dimensions, the presence, referent, or amount of which are to be considered separately, in order to perform a certain task.“

Eine Checkliste kann als die große Schwester der Merkhilfe (Eselsbrücke) aufgefasst werden. Ein Beispiel für eine Eselsbrücke sind die *großen 5W*, die bei einer Notrufmeldung am Telefon als Informationen wichtig sind: **W**o geschah es? **W**as geschah? **W**ie viele Personen sind betroffen? **W**elche Art der Erkrankung/Verletzung liegt vor? **W**arten auf Rückfragen!

Eine Checkliste jedoch nur als eine bloße Auflistung von Punkten zu definieren, ist zumindest dann unzureichend, wenn man sich mit der Frage nach Kriterien einer erfolgreichen Anwendung beschäftigt. Hierzu gilt es, eine Reihe von Aspekten zu berücksichtigen, die allesamt entscheidend für eine erfolgreiche Erstellung, Anwendung und Evaluierung von Checklisten sind. Die angesprochenen Aspekte einer Checkliste lassen sich wie folgt als W-Fragen formulieren:

1. *Was* wird geprüft? – Diese Frage bezieht sich auf das bereits oben angesprochene *Etwas*, das es zu prüfen gilt. Hier besteht eine Relation zu kritischen Schritten einer dahinterliegenden komplexeren SOP. (Mehr hierzu im kommenden Abschnitt 4.3.)

2. *Wann* wird geprüft? – Wann eine Checkliste zu beachten ist und wann nicht, hängt von der jeweiligen Arbeitssituation des Anwenders ab. Für Checklisten sollte daher geregelt sein, wann sie anzuwenden sind und in welchen Situationen man von ihnen abweichen darf.
3. *Wer* prüft? – Checklisten sind kognitive Hilfsmittel für unterschiedliche Anwender. Hier stellt sich die Frage nach dem Hintergrundwissen, dem Ausbildungsgrad und der eingenommenen Rolle des jeweiligen Anwenders im korrespondierenden Prozess.
4. *Wozu* wird geprüft? – Welchen Zweck haben die einzelnen Items bzw. eine ganze Checkliste, d. h., auf welche Funktionen zielt die Checkliste ab?
5. *Wie* wird geprüft? – Wie die Anwendung in der Praxis zeigt, gilt es je nach Anwendungsphilosophie, unterschiedliche Abarbeitungsmodelle anzuwenden (siehe 4.2.3).
6. *Wo* wird geprüft? – Weiterhin spielt die Frage des Einsatzortes, d. h. die zu erwartende Arbeitsumgebung und Situation eine wesentliche Rolle bei der konkreten Ausgestaltung und der Anwendungsmethodik einer Checkliste.

Ein Großteil der Aspekte spiegelt sich in folgender Arbeitsdefinition wider:

Definition 1 (Checkliste) *Als Checkliste wird eine Sonderform einer Liste bezeichnet, deren Listenelemente (Items genannt) jeweils aus einem Freitext und einer optionalen Checkbox bestehen. Jedes Item steht für einen aus Sicht des Anwenders sicherheitskritischen Aspekt einer zur Checkliste korrespondierenden (Standard-)Prozedur. Darüber hinaus ist jeder Checkliste ein Gültigkeitsbereich, die Situation(en) ihrer Anwendung und ein adressierter Benutzerkreis zugeordnet.*

4.2.2 Funktionen, Einsatzbereiche und Zielgruppen

Checklisten helfen überall dort Handlungen korrekt (standardkonform) auszuführen, wo uns unsere tägliche Routine, z. B. durch Stress, steigende Komplexität oder aufgrund unbekannter Situationen, im Stich lässt. Sie sollen dabei helfen, Gedächtnisfehler zu vermeiden. Eine Checkliste ist somit ein kognitives Werkzeug zur Sicherstellung einer konsistenten Anwendung einer (momentanen) „best practice“ in einem Arbeitsgebiet. Ihre konsequente Anwendung soll zu einer koordinierten Vorgehensweise aller beteiligten Akteure führen [DW91]. Folgende Hauptfunktionen einer Checkliste lassen sich identifizieren:

- *Gedächtnisentlastung*: BUERSCAPER & ST. PIERRE [BP07] sehen als psychologische Kernfunktion von Checklisten vor allem die Gedächtnisentlastung bei der Durchführung sicherheitskritischer Handlungsketten an.
- *Sicherstellung eines Qualitätsstandards*: Individuelle Verhaltensabweichungen können durch explizite normative Vorgaben und deren Überprüfung reduziert oder gar vermieden werden. Dies garantiert eine gleichbleibende Qualität der Ausführung von Prozessen.
- *Handhaben von Komplexität*: Die Handhabung komplexer SOPs kann mittels Konzentration auf das Wesentliche deutlich überschaubarer für den Menschen gestaltet werden und somit durch Abstraktion (wo möglich) zur besseren Handhabung von Komplexität beitragen.

- *Reduktion von Unsicherheit*: Als psychologische Wirkung kann die Verringerung der Unsicherheit auf Seiten des Handelnden einen nicht zu unterschätzender positiven Effekt darstellen. (Sie kann jedoch auch kontraproduktiv wirken, siehe Abs. 4.2.5.)

Die oben genannten Hauptfunktionen tragen allesamt mehr oder weniger zu der gewünschten Minimierung des Fehlerrisikos auf der Handlungsebene bei (vgl. Abschnitt 4.1).

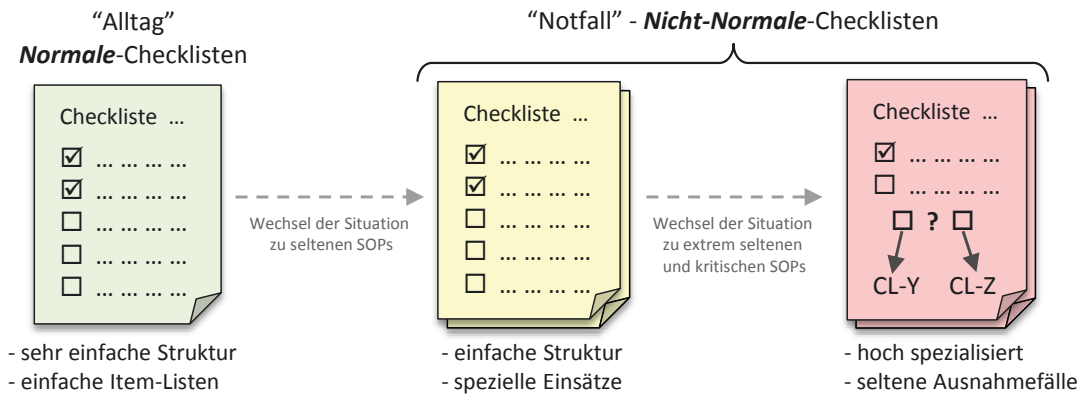


Abbildung 4.2: Checklisten für unterschiedliche Situations-Klassen.

In einigen Praxisfällen werden Checklisten dazu verwendet, Mitarbeiter jenseits von alltäglichen Routinehandlungen sicher durch komplexe und besonders sicherheitskritische Prozeduren zu führen. Die *abnormal-* und *emergency-checklists* in der Luftfahrt sind hierfür prominente Beispiele (siehe hierzu z. B. [DW90; DW93]). Je nach Anwendungsklasse der Prozeduren werden dort verschieden komplexe Checklisten angewandt; ausgehend von Standardprozeduren, über „Sonderprozeduren“ hin zu „Notfallprozeduren“ (siehe Abb. 4.2). Die Art der SOP bedingt hier die Ausgestaltung und vor allem den erforderlichen Komplexitäts- und Detaillierungsgrad der Checklisten.

Einige Autoren (siehe z. B. [BLK11]) sehen daher Checklisten primär als Hilfsmittel zur Entscheidungsunterstützung oder gar zur Prozesssteuerung an. In diesem Sinne sollen Checklisten Handlungsalternativen vorgeben und somit den Anwendern die Entscheidung über die korrekte Vorgehensweise abnehmen. Folgt man jedoch der Argumentation und der Fehler-Einteilung von REASON, so wird klar, dass Checklisten keine Hilfestellung in der Entscheidungsunterstützung und Prozesssteuerung geben können/sollten, wenn man deren Hauptfunktion, die Vermeidung von Ausführungsfehlern, ernst nimmt. Auch die Autoren der „Checklist for a Checklist“¹ geben zu bedenken: „A checklist is NOT a teaching tool or an algorithm“. Checklisten sollten also nicht alle Eventualitäten und alle denkbaren Prozessausprägungen wie in einem Workflow enthalten. GAWANDE schreibt hierzu: „Checklists are not comprehensive how-to guides [...]. They are quick and simple tools aimed to buttress the skills of expert professionals“ ([Gaw10], S. 128).

Da Checklisten nur als Gedächtnisstütze gedacht sind, sollte deren Einsatz als eine reine Handlungsanweisung (Prozesssteuerung) mit Vorsicht begegnet werden. Zumindest sollte diese Art der Anwendung nur in Ausnahmefällen angewendet werden, in denen keine Experten die SOPs ausführen bzw. in denen sehr seltene SOPs angewandt werden müssen.

Seit den späten 1960er Jahren finden Checklisten in der Luftfahrt Anwendung zur Sicherstellung einer richtigen Konfiguration der immer komplexer werdenden Flugzeuge. Heutzutage sind sie ein wichtiger Bestandteil aller sicherheitskritischen Prozeduren im Cockpit

¹ Siehe www.projectcheck.org (abgerufen am 20.12.2013) und Anhang Seite 137.

[DW90; DW93]. Checklisten spielen in der Luftfahrt jedoch eine gewisse Sonderrolle, da sie aufgrund der weiten Verbreitung und täglichen Anwendung ein fester Bestandteil der Cockpit-Handlungsroutinen selbst geworden sind. Hierin liegt wohl auch begründet, warum in der Luftfahrt Checkliste und SOP oft im gleichen Atemzug genannt werden (vgl. hierzu die Diskussion in Abschnitt 4.3).

In Sachen Sicherheitskultur und Anwendungskonzepte gilt die Luftfahrt als Vorreiter und Ideengeber für eine Reihe ähnlicher komplexer und sicherheitskritischer Arbeitsumgebungen. So finden Checklisten in weiteren sogenannten Hochsicherheitsbereichen wie z. B. der Chemischen Industrie, der Intensivmedizin, der Hochseeschifffahrt und dem Militär eine breite Anwendung. Überall dort, wo Menschen in komplexen (sozio-)technischen Systemen mit höchsten Sicherheitsanforderungen arbeiten, lassen sich Anwendungsbeispiele finden.

Qualität	Beispiel	Konsequenzen bei Fehlern
Bereiche mit sehr geringen Sicherheitsanforderungen	Privater Bereich, z.B. Einkaufsliste oder ToDo-List	Objektiv sehr geringe und eher unkritische Auswirkungen
Bereiche mit mäßigen Sicherheitsanforderungen	Wirtschaftsunternehmen / Industrie mit hohem Qualitätsanspruch (Risiko- und Qualitätsmanagement)	Bedrohend: Verdienstaustausch, Regressforderungen, Imageschaden, Personenschäden
Bereiche mit höchsten Sicherheitsanforderungen	Hochsicherheitsberufe wie z.B.: Luft- und Raumfahrt, Intensivmedizin, Chemische Industrie oder Militär	Katastrophal: Tote, Verletzte, große materielle Schäden, Zerstörung von Infrastruktur

Abbildung 4.3: Grade der Sicherheitsanforderungen.

Mit steigendem Ausmaß der Fehlerkonsequenzen steigt auch der Anspruch an die Fehlervermeidung (vgl. Abb. 4.3). Werden im privaten Bereich Punkte einer Checkliste vergessen, so sind die zu erwartenden Konsequenzen in der Regel wohl eher gering. In Hochsicherheitsbereichen hingegen können Fehler fatale Auswirkungen haben, sodass dem Checklisten-Konzept dort eine besondere Aufmerksamkeit zuteilwird. Checklisten finden dort ausnahmslos als kognitives Hilfsmittel für hochspezialisiertes und erfahrenes Personal (die Experten) Anwendung. WILLIAMSON weist in [Wil10] (S. 3) unter anderem drauf hin, dass „*checklists are used by experienced and qualified people and are not a substitute for training.*“ Checklisten nur zu Trainingszwecken einzusetzen, sollte vermieden werden.

Ein weiteres Einsatzgebiet stellt die Medizin dar. In der heutigen Zeit verlangen viele Arbeitsgebiete der Medizin den Ärzten und dem teils hochspezialisiertem Pflegepersonal immer mehr Spezialwissen über immer komplexer werdenden Techniken und Prozeduren ab. GAWANDE spricht hierbei von dem Phänomen der *super-specialization* [Gaw10]. Inspiriert durch den erfolgreichen Einsatz in der Luftfahrt, werden Checklisten daher seit ca. fünf Jahren vermehrt auch im Klinikalltag – speziell auf den Intensivtherapiestationen und in den Operationssälen – eingesetzt und wurden bereits in ausgewählten Kliniken evaluiert [HP06; Gaw10]. Checklisten dienen in der Medizin ausschließlich Experten zur Unterstützung. Jedoch stehen die Bemühungen von GAWANDE² oder PRONOVOST (siehe [Lan11]) trotz ihres bereits erreichten Erfolges erst am Anfang einer intensiveren Nutzung im medizinischen Bereich. Checklisten sind daher im Klinikbereich noch eher selten verbreitet und deren Verwendung ist üblicherweise auch nicht Pflicht (siehe Akzeptanzproblematik in Abschnitt 4.2.5).

² siehe z. B. Projektweiterführung unter <http://www.safesurgery2015.org> (abgerufen am 20.12.2013)

4.2.3 Items und Anwendungsmethoden

Bei den Items einer Checkliste handelt es sich um sicherheitskritische Aspekte (Schritte, Punkte, Sachverhalte) einer zur Checkliste korrespondierenden SOP. Darunter fallen jedoch nicht nur prozedurale Schritte wie z. B. der „Aufbau einer Verletztenversorgung“ oder die Aktion „Informiere X über die Einsatzlage“. Auch Aspekte, die nicht direkt Handlungen eines Prozesses entsprechen, können als Items in Frage kommen. Hierzu zählen allgemeine Hinweise, die sich auf die Gesamtheit des Prozesses selbst oder aber auf die Einbettung in andere Prozesse beziehen. Die Hinweise „Ruhe bewahren!“ oder „Weiteres Vorgehen mit der Einsatzleitung der Feuerwehr abstimmen!“ sind hierfür Beispiele.

Auch sind in der Praxis Items anzutreffen, die auf Dinge hinweisen, die ausdrücklich *nicht* getan werden sollen. Ein populäres Beispiel hierfür ist das Problem, dass Einsatzkräfte des Rettungsdienstes die Tendenz zum Abtransport von verletzten Personen haben, da dies ihrem korrekten Vorgehen im Regelbetrieb entspricht. In einer Großschadenslage soll jedoch genau dies zu Beginn unbedingt vermieden werden. „Keine Patienten abtransportieren!“ ist daher ein Beispiel für ein Item, das die Einsatzkraft daran erinnern soll, etwas nicht zu tun. Zusammenfassend sollen in dieser Arbeit grob drei Item-Klassen unterschieden werden:

1. SOP-Schritte, die nicht vergessen werden dürfen (die *To-Dos*),
2. allgemeine Hinweise für SOPs, die keinen direkten Aktionen entsprechen, und
3. Schritte, die es unbedingt zu vermeiden gilt (die *Not-To-Dos*).

Obige Item-Klassifikation soll (lediglich) verdeutlichen, dass Items für mehr als nur konkrete physische Handlungen stehen können. Jedoch sind die Grenzen zwischen den drei Klassifizierungen fließend. So lässt sich das Item „Ruhe bewahren!“ durchaus auch als ein Prozessschritt auffassen. Weiterhin lässt sich „Ruhe bewahren!“ zu „Nicht hektisch werden!“ umformulieren, was der gleichen Ursprungsentention entspricht, jedoch nun offenkundig in die Klasse der *Not-To-Dos* fällt. Das Abprüfen eines kritischen Aspektes kann somit durch verschiedene Item-Formulierungen realisiert werden. Um z. B. sicherzustellen, dass das obige Abtransport-Problem vermieden wird, kann prinzipiell die Form einer *Anweisung*: „Transportstopp aussprechen!“ oder „Keine Patienten abtransportieren!“, oder alternativ die einer *Frage*: „Wurde Transportstopp ausgesprochen?“ genutzt werden.

Weiterhin lassen sich verschiedene Fragekategorien für die Itemtexte einsetzen. Geschlossene Fragen können nur mit „ja“ oder „nein“ beantwortet werden und ermöglichen dadurch eine Verzweigung in z. B. entsprechende Folge-Checklisten, so wie es in Abb. 4.2 rechts angedeutet ist. Nicht alle Aspekte lassen sich mit geschlossenen Fragen abfragen. Mit den sog. offenen Fragen lassen sich Informationen über eine Situation vom Checklistenanwender abfragen. Hiermit kann beliebiger „Freitext“ als Antwortformat erfasst werden, der vor allem bisher zu Dokumentationszwecken genutzt wird. Offene Frageformen für Items sind in der Praxis eher selten anzutreffen. Im nächsten Kapitel eröffnet jedoch gerade diese Frageform eine nützliche Funktionalität in einem elektronischen Assistenzsystem.

Bei der Erstellung der Checklisten stellt sich die Frage nach einem geeigneten Vokabular für die Item-Texte. Zusätzlich zu der Anforderung an eine gute Lesbarkeit sollten die Texte möglichst präzise und unmissverständlich formuliert sein. Obwohl das Vokabular von der jeweiligen BOS-eigenen Organisationskultur und der adressierten SOP abhängt, sollten einige prinzipielle Empfehlungen bei der Wahl des verwendeten Vokabulars beachtet werden. Der Text der Checklisten sollte derart gestaltet werden, dass Missverständnisse bei der Interpretation durch den Endanwender möglichst auszuschließen sind.

Ein Blick auf die langjährigen Erfahrungen der Luftfahrt, in der der sog. *Checklisten Phraseologie* ein besonderer Stellenwert zukommt (vgl. DEGANI & WIENER [DW90]), scheint hierfür lohnenswert. Auch in der Literatur rund um die Thematik zur Schnittstelle Mensch-Maschine lassen sich grundlegende Empfehlungen finden, die ebenfalls auf die Checklistenstellung angewendet werden können. So empfiehlt z. B. KRYPTER [Kry72], den Umfang des Vokabulars möglichst so gering wie nötig zu halten, und WICKENS [Wic84], wichtige Wörter, deren Bedeutung den Anwendern klar ist, möglichst oft einzusetzen. In dieser Arbeit soll jedoch nicht weiter auf den Formulierungsprozess der Texte eingegangen werden. Für mehr Details zu dieser Thematik sei als Einstieg auf die obige Literatur ([Kry72; Wic84; DW90]) verwiesen.

Checklisten stellen in der Regel an den Benutzer keine besonderen Anforderungen. Jedoch variiert die Art und Weise ihrer Anwendung von Einsatzgebiet zu Einsatzgebiet bzw. von Organisation zu Organisation. DEGANI & WIENER [DW93] unterscheiden – orientiert an der Luftfahrt – prinzipiell zwei Durchführungsmethoden für Checklisten:

1. *Challenge-response* (oder *challenge-verification-response*). Die CL dient hier „nur“ als eine redundante Backup-Prozedur. Zum Beispiel konfiguriert der Pilot das Flugzeug aus dem Gedächtnis und prüft nachträglich (nach jedem Schritt bzw. nach einer gesamten Handlungssequenz), ob alle kritischen Punkte beachtet wurden. Hierbei handelt es sich um eine gewollte Redundanz: erst Gedächtnis, dann Überprüfung.
2. *Call-do-response* (oder *Do-list*). Die Items einer CL werden eines nach dem anderen „durchlaufen“. Diese Methode eliminiert die gewollte Redundanz aus Methode 1 – mit Vor- und Nachteilen (siehe Diskussion im Abschnitt 4.2.5).

In der Praxis sind beide Methoden in Kombination anzutreffen, wobei nach DEGANI & WIENER die Challenge-response Methode die gebräuchlichste Form der Anwendung in der Luftfahrt darstellt [DW93]. Zusätzlich weist GAWANDE [Gaw10], mit Blick auf die Anwendung von Checklisten in OP-Sälen, auf den wichtigen Aspekt der „lauten Aussprache“ jedes Items hin. Er spricht von *verbal checklist*, wodurch sichergestellt werden soll, dass sich alle im Raum der anstehenden Prozeduren mit all den zu erwartenden Gefahren bewusst werden. Im Cockpit eines Flugzeuges findet diese laute Aussprache ebenfalls Anwendung, welche zur gegenseitigen Überwachung eingesetzt wird und die Teamarbeit fördern soll.

GAWANDE weist in [Gaw10] und in Interviews darauf hin, dass die Abarbeitung einer Checkliste auf keinen Fall wie eine „*tick-box exercise*“ zu verstehen ist, wer dies trotzdem tut, wird der Grundidee und dem Potential von Checklisten nicht gerecht. Schlussendlich variiert die Checklisten-Philosophie von Organisation zu Organisation und spiegelt somit die unterschiedlichen Organisationskulturen wider. Organisationskultur beinhaltet viele Faktoren des gesamten operativen Konzepts der Organisation: traditionelle Methoden der Operationen, vorgegebenen Richtlinien, spezieller Führungsstil, Delegation von Verantwortung in der Befehlskette sowie Straf- und Disziplinarmaßnahmen.

Neben der Frage der geeigneten Anwendungsmethode kommt der Frage, wann (in welcher Situation) eine Checkliste anzuwenden ist, eine ebenso große Bedeutung zu. Diese korrespondiert mit der Frage, wann eine SOP einzuleiten ist. Wie CIMOLINO bezüglich der Standard-Einsatz-Regeln (SER) schreibt (siehe [CG05]), sollte in jeder SER (respektive in jeder SOP) beschrieben werden, wann sie anzuwenden ist. Der beste Zeitpunkt zur Initiierung einer Checkliste ist typischerweise der, wenn der Nutzer gerade im Begriff ist, einen anstehenden Arbeitsprozess zu beginnen, also gerade nicht mit anderen Aufgaben beschäftigt ist [Ros04; BP07]. Wie zum Beispiel vor dem take-off im Cockpit eines Flugzeuges, vor Beginn eines chirurgischen Eingriffs im OP oder bei der Fahrt eines Einsatzleiters zum Einsatzort.

4.2.4 Repräsentationsformen von Checklisten

Papierbasierte Checklisten (PCLs) gehören zu den verbreitetsten Repräsentationsformen (siehe Beispiele im Anhang, S. 138 ff.). Der Vorteil einer PCL liegt in ihrer Einfachheit und ihrer unkomplizierten Handhabung. Sie sind schnell erstellt und leicht zu vervielfältigen. Die Interaktion mit ihnen erfordert keinerlei technische Hilfsmittel noch spezielle Fachkenntnisse – lässt man das Wissen über die dahinter liegenden SOPs außer Acht.

Ihre Einfachheit kann in einigen Anwendungsgebieten jedoch auch nachteilig sein. So stoßen papierbasierte Checklisten in verteilten, dynamischen Umgebungen an ihre Grenzen, da ihre Anwendung lokal begrenzt und ihre Informationsdarbietung statisch ist. Zusätzlich stellt sich die Aufgabe, alle sich im Umlauf befindlichen PCLs aktuell und für alle Anwender zugreifbar zu halten. Hinzu kommt, dass PCLs schnell unübersichtlich werden, wenn die jeweilige SOP eine Realisierung mit mehr als einer Seite erforderlich macht. Zwei zu PCLs alternative Repräsentationsformen sollen im Folgenden skizziert werden.

Mechanische Checklisten: Sogenannte mechanische Checklisten sind eine seltene Form einer Checklistenrepräsentation, die z. B. in der Luftfahrt im Cockpit einiger Flugzeugtypen zum Einsatz kamen und kommen. Wegen ihrer technisch-bedingten Beschränktheit, berücksichtigen sie „normale“ Standardprozeduren mit wenigen Items, wie z. B. für die *Take-Off*- und *Landing*-Prozedur. Als Beispiel ist in Abbildung 4.4 eine mechanische CL in einer Boeing 757 zu sehen. Ein erfülltes Item quittiert der Pilot mittels eines einfachen mechanischen Kippschalters. Während zu Beginn alle entsprechenden Items der CL aufleuchten, erlischt nach jedem Umlegen des jeweiligen Schalters die Hintergrundbeleuchtung. Der Pilot hat somit jederzeit alle noch offenen Punkte im Blick, auch wenn er zwischenzeitlich die Aufmerksamkeit weg von der CL richten muss. Selbst wenn der Pilot die Abarbeitungsreihenfolge variiert, wird durch diese Art der Umsetzung ein Vergessen eines früheren Items verhindert.



Abbildung 4.4: Teil einer MCL (*Take-Off*) in einer *Boeing 757*.

Mechanische CLs stellen somit bereits eine einfache Möglichkeit dar, typische Fehler bei der Abarbeitung einer PCL vorzubeugen (vgl. hierzu Abschnitt 4.2.5). Jedoch sind sie in ihrer Flexibilität und in der Anzahl der Items deutlich beschränkt.



Abbildung 4.5: Elektronische Checkliste im Cockpit einer *Boeing 777*.

Elektronische Checklisten: In Zeiten zunehmender Computerisierung liegt die Idee nahe, elektronische Checklisten (ECLs) umzusetzen, die den Anwendern über einen Monitor dargeboten werden. Der Computer kann in diesem Fall nicht nur zur bloßen Anzeige der CLs genutzt werden, sondern darüber hinaus eine erweiterte Funktionalität durch eine Kopplung mit beliebigen Informationssystemen ermöglichen.

Mit der Einführung moderner sog. *Glascockpits* in der Luftfahrt wurden in einigen Flugzeugtypen auch ECLs eingeführt. In Abbildung 4.5 ist als Beispiel eine ECL des Flugzeugtyps Boeing 777 zu sehen, welche das Ergebnis jahrelanger Forschung und Entwicklung der *Boeing Flight Deck Research Group* in den späten 1980er Jahren war. Im Cockpit sollen ECLs zum einen Fehler bei der Nutzung von PCLs vermeiden helfen und zum anderen durch eine (teilweise) Kopplung des Bord-IT-Systems mit dem ECL-System ihren Einsatz weiter vereinfachen [Boo01a; Boo01b]. Darüber hinaus beinhalten sie alle *normal* und *not-normal* Checklisten des entsprechenden Flugzeugtyps. Vor allem die teils recht umfangreichen Ordner, in denen alle *not-normal* CLs üblicherweise abgelegt sind, können so eingespart werden. Boeing zog nach einer zehnjährigen Einsatzphase ein positives Resümee, sodass sie planten, weitere Flugzeugtypen zukünftig mit einem elektronischen Checklisten-System auszustatten [Ark06]. So ist z. B. der neue Flugzeugtyp Boeing 787 mit einer neuen überarbeiteten Version eines elektronischen Checklisten-Systems versehen [Bra10].

Auch im Klinikbereich wird die Anwendung elektronischer Checklisten seit einigen Jahren vereinzelt diskutiert [Par10], primär mit dem Ziel einer Anbindung an das klinikeigene IT-System. Es zeigte sich aber, dass mit steigender Komplexität solcher Systeme deren Akzeptanz und Bedienbarkeit typischerweise sinkt, wodurch dann der eigentliche Nutzen des Checklisten-Prinzips verloren geht. Nicht zuletzt aus diesen Gründen sind ECLs daher im Klinikalltag momentan kaum verbreitet [Par10].

4.2.5 Nachteile und Akzeptanzproblematik

Obwohl die Vorteile von Checklisten zur Absicherung kritischer Handlungen unbestritten sind, stehen ihnen einige bekannte Nachteile gegenüber. Vor allem die langjährige Praxiserfahrung in der Luftfahrt offenbarte, dass deren Einsatz auch zu unbeabsichtigten, negativen Effekten führen kann. Hinzu kommt die Tendenz einer geringen Akzeptanz von Checklisten bei Mitarbeitern verschiedenster Bereiche und Ausbildungsgrade. Nachstehend sollen beide kurz vorgestellt werden.

Bekannte Nachteile eines Checklisteneinsatzes

Die Standardisierung von Prozessen, und daher auch die Abarbeitung strikter Checklisten, kann neben den Vorteilen auch mangelnde Flexibilität in der Anpassung an außergewöhnliche Situationen mit sich führen [Gro04; HS98]. Langfristig kann dies auch zu einer Reduktion der Problemlösekompetenz der Benutzer führen, wodurch eine gewisse „Abhängigkeitsproblematik“ droht. Haben sich die Mitarbeiter erst einmal an die Abarbeitung einer Checkliste gewöhnt, so können sie ohne diese plötzlich große Probleme haben, ihre Arbeit korrekt durchzuführen.

Weiterhin ist bekannt, dass das Vorgehen gemäß zu starrer und detaillierter Checklisten, vor allem in dynamischen Anwendungsfällen, sogar handlungshemmend wirken kann. Die Entwicklung alternativer Lösungsstrategien kann dadurch gehemmt und die Handlungsfreiheit ungewollt eingeschränkt werden [BHL08]. An dieser Stelle sei jedoch angemerkt, dass dieser Effekt durchaus gewünscht sein kann (siehe Abschnitt 4.2.2).

BUERSCHAPER & ST. PIERRE weisen in [BP07] darauf hin, dass sich bei einer erfolgreichen Checklistenbearbeitung möglicherweise ein „falsches“ Sicherheitsgefühl einstellt, wodurch die Aufmerksamkeit sinkt. Der damit einhergehende Gefühlszustand kann negative

Folgen auf andere kognitive Prozesse der Anwender haben. Als Beispiel nennen die Autoren die Tendenz, bei vermeintlicher Sicherheit einen geringeren Erwartungshorizont aufzufächern, wodurch mögliche kritische Lageentwicklungen eventuell übersehen oder falsch bewertet werden. Zusätzlich kann es daher zu einem sinkenden Situationsverständnis (*Situation Awareness*, siehe z. B. [BH89]) kommen, da die Aufmerksamkeit für Dinge sinkt, die sich außerhalb der eigenen Checkliste befinden.

Weiterhin ist bekannt, dass Checklisten, wenn sie falsch eingesetzt werden, die Effektivität der dahinterliegenden Prozesse sogar beeinträchtigen können. In der Luftfahrt sind diesbezüglich eine Reihe von Problemen und Fehlerquellen bekannt (vgl. hierzu [Ros04]).

Subjektive Ablehnung

Checklisten als Hilfsmittel zur Fehlervermeidung werden in vielen Arbeitsbereichen, so auch im BOS-Bereich, kontrovers diskutiert. Wie sich im SPEEDUP-Projekt zeigte, stehen einige Führungskräfte dem Checklisten Einsatz skeptisch bis ablehnend gegenüber. Die Skepsis und die Ablehnung hat jedoch weniger objektive als persönliche und emotionale Gründe. So stellte GAWANDE fest, dass sich Mitarbeiter durch Checklisten oftmals bevormundet und in ihren Handlungsmöglichkeiten eingeschränkt fühlen [Gaw10]. Keiner der Praktiker will sich gern vorschreiben lassen, was er zu tun und zu lassen hat.

Schaut man in die Luftfahrt, so stellt man fest, dass obwohl Checklisten dort am längsten und erfolgreichsten eingesetzt werden, deren Einsatz teils als lästig und unnötig empfunden wird [Ros04]. Während einige Piloten nie ohne sie fliegen würden, sehen andere in ihrer Nutzung ein Zeichen von Schwäche. Die Fluggesellschaften haben jedoch Checklisten fest in ihre Sicherheitskultur verankert, sodass die Piloten CLs nutzen müssen.

Im Klinikbereich stößt die Einführung von Checklisten, trotz des nachgewiesenen positiven Effektes auf die Überlebensrate der Patienten [HP06], ebenfalls zu Beginn oftmals auf Ablehnung bei Ärzten und Pflegepersonal. Sie fragen sich, wie so ein Stück Papier mit ein paar Boxen ihre Arbeit verbessern kann? Die Fronten der Gegner und Befürworter scheinen verhärtet. GAWANDE identifiziert drei Hauptgründe, warum Checklisten in der Medizin schwer zu etablieren sind und warum ihnen wenig Aufmerksamkeit zuteil wird [Gaw10]:

1. Ärzte fühlten sich beleidigt, ihre Egos wurden verletzt.
2. Das Pflegepersonal sieht sich bereits mit anderen Formularen überlastet.
3. Solch „banale“ Forschung wird gern zu Gunsten der aufregenderen Themen ignoriert.

Papierbasierte Checklisten sind daher im Klinikbereich noch bei weitem nicht so verbreitet wie in der Luftfahrt.

4.3 Unterscheidung der Begriffe Checkliste und SOP

Obwohl oft in einem Atemzug genannt, stellen „Checklisten“ und „SOPs“ zwei unterschiedliche (wenn auch verwandte) Konzepte dar. Diese konzeptionelle Trennung wird in der Literatur kaum beachtet, was zu Missverständnissen in der praktischen Verwendung beider Begriffe führen kann. Während SOPs für mehr oder weniger detaillierte operationalisierte Richtlinien stehen, adressieren Checklisten in der Regel ausschließlich kritische Punkte bei der Ausführung dieser SOPs. BRABANT (siehe [Bra10], S.43) zitiert diesbezüglich einen Checklisten-Experten (E. MAHR) von Boeing: „A good checklist is precise and lists only critical steps in a concise way“.

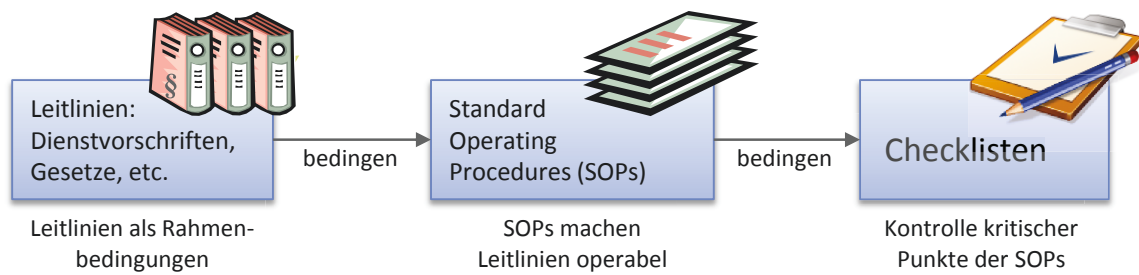


Abbildung 4.6: Zusammenhang zwischen „Leitlinien“, SOPs und Checklisten.

Die Vermischung der Begriffe SOP und Checkliste liegt offensichtlich nicht zuletzt darin begründet, dass eine klare Trennung aller drei in Abbildung 4.6 gezeigten Konzepte in der Praxis oft schwierig ist. So tragen manche konkreten Leitlinien den Charakter von SOPs, während in anderen Bereichen Checklisten als Repräsentationen komplexer und umfangreicher SOPs verstanden und eingesetzt werden. Checklisten dienen als kognitives Werkzeug zur Unterstützung der Erinnerungsleistung des Menschen und unterscheiden sich dadurch prinzipiell vom Konzept der SOPs. Checklisten repräsentieren in der Regel nur ganz bestimmte kritische Teilaspekte der dahinterliegenden SOPs. Damit die gewünschten positiven Effekte eines Checklisteneinsatzes auch erreicht werden, sollte eine Vermischung beider Konzepte vermieden werden.

In Anbetracht einer Erarbeitung der Anforderungen an ein IT-Assistenzsystem empfiehlt sich jedoch, eine strikte Trennung aller drei Konzepte vorzunehmen. Ein wesentliches Merkmal des in dieser Arbeit in Kapitel 5 vorgeschlagenen Lösungsansatz soll daher die explizite Trennung der beiden Konzepte *SOP* und *Checkliste* sein. Die in dieser Arbeit propagierte Sichtweise unterscheidet sich von der von BECKER et al. [BLK11] und von PEINEL et al. [PRW12b] (vgl. Abschnitt 3), in denen Checklisten als nur eine Möglichkeit der Darstellung von SOPs angesehen werden bzw. Checklisten einen Workflow (Plan) repräsentieren.

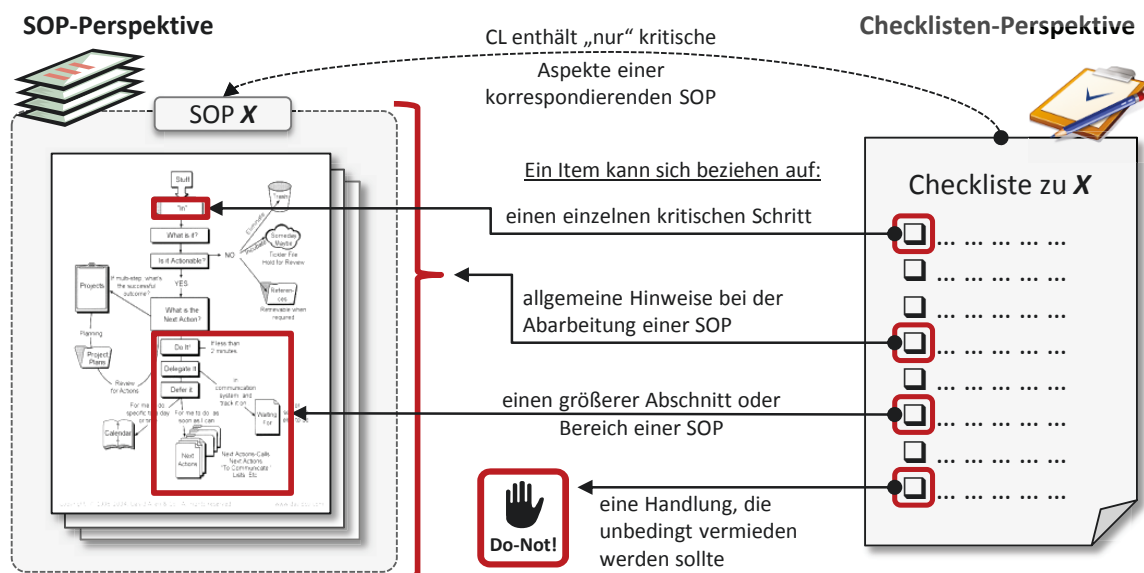


Abbildung 4.7: Arten der Korrespondenzen zwischen einer Checkliste und ihrer SOP.

Eine CL adressiert in der Regel nur die kritischen Aspekte einer SOP und keine detaillierte Prozessbeschreibung. Abbildung 4.7 illustriert die prinzipielle Vielfalt der Korresponden-

zen zwischen den Checklisten-Items und den entsprechenden Aspekten (Aufgaben, Schritte, Merkmale, Besonderheiten) einer zugehörigen SOP. Die komplette SOP soll und kann in der Regel nicht durch Checklisten abgebildet werden. Primär repräsentieren die Items kritische (atomare) Schritte (die *To-Dos*) einer SOP wie zum Beispiel „Sichtungszelt aufbauen!“. Zusätzlich können Items auf allgemeine Anordnungen (wie z. B. „Ruhe bewahren“) hinweisen sowie auf Aktionen, die auf keinen Fall bei der SOP-Anwendung durchgeführt werden sollten (die *Not-To-Dos*).

Spezialfall Luftfahrt: In der stark regulierten Luftfahrt werden die Begriffe SOP und Checkliste häufig synonym verwendet. Dies ist in den technischen und hochkomplexen Rahmenbedingungen begründet, die von den Piloten eine strikte Vorgehensweise abverlangen und die über die Jahre dazu geführt haben, dass sich die Abarbeitung der Checklisten selbst als ein wesentlicher Prozess (also als eine SOP) verankert hat. Viele SOPs werden in der Luftfahrt aus der Perspektive der Abarbeitung der entsprechenden Checklisten gesehen, obwohl auch dort komplexere SOPs überprüft werden. Einige hochkritische SOPs decken sich jedoch fast zu einhundert Prozent mit den entsprechenden Checklisten-Items. Dies ist jedoch die Ausnahme und nicht die Regel, sodass die Trennung beider Begriffe in anderen Anwendungsgebieten durchaus berücksichtigt werden sollte.

Fazit

Als eine Strategie des Umgangs mit Komplexität und der Vermeidung von Unsicherheit wurde die Standardisierung (von Prozessen) aufgeführt. Da in einer Großschadenslage bereits kleine Fehler zu fatalen Auswirkungen führen können, bietet sich im BOS-Bereich zur Fehlervermeidung auf den ersten Blick die Strategie der Standardisierung an. Wir haben in Kapitel 2 Dienstvorschriften als eine Möglichkeit dieser Bemühungen kennen gelernt. Jedoch wurde bisher auch deutlich, dass Dienstvorschriften allein noch zu viel Spielraum in der Ausgestaltung aufweisen. Die Einsatzkräfte müssen zum einen flexibel reagieren können und zum anderen jedoch auch die gültigen Richtlinien beachten. Sie sollen sich trotz Flexibilität in einem gewissen Rahmen (als eine Art „Leitplanken“) bewegen.

Wie kann ein elektronisches Assistenzsystem diesen beiden Ansprüchen genügen? Als Antwort auf diese Frage wird als Grundidee dieser Arbeit die Übertragung des Checklisten-Prinzips auf den BOS-Bereich vorgeschlagen. Dadurch dass Checklisten nur kritische Aspekte einer SOP berücksichtigen und „nur“ eine Erinnerungsfunktion anbieten, bleibt die gewünschte Flexibilität in den Handlungen erhalten. Die Einsatzkräfte können weiterhin frei agieren und sollen durch die Checklisten auf einen groben Handlungsrahmen als Leitplanken aufmerksam gemacht werden.

Die restliche Arbeit geht der Fragestellung nach, wie genau eine elektronische Umsetzung des Checklisten-Prinzips im BOS-Bereich realisiert werden kann und welche Merkmale ein entsprechendes Assistenzsystem aufweisen sollte. Hierzu wird im nächsten Kapitel ein Rahmenwerk eines reaktiven Assistenzsystems vorgestellt, das über die Fehlervermeidung und ein Prozessmonitoring hinaus, zu einer signifikanten Verbesserungen des gesamten Einsatzmanagements beitragen kann.

Kapitel 5

Intelligente elektronische Checklisten-Assistenz

Dieses Kapitel stellt ein Rahmenwerk und die Architektur eines intelligenten elektronischen Checklisten-Assistenzsystems vor. Das System kann, speziell konfiguriert für den BOS-Bereich, den Einsatzkräften eine neuartige Hilfestellung bieten, um Handlungsrouninen sicher und fehlerfreier anzuwenden. Darüber hinaus ermöglicht der vorgestellte Ansatz ein Monitoring der ausgeführten und sich in Bearbeitung befindlichen Handlungen für jede BOS bzw. für alle Organisationen mit ähnlichem Arbeitsumfeld.

5.1 Motivation & Anforderungen einer elektronischen Checklisten-Assistenz

Inwieweit eine IT-Unterstützung zur Harmonisierung des Einsatzmanagements und zur Fehlervermeidung beitragen kann, ist eine der Kernfragen dieser Arbeit. In der heutigen BOS-Praxis kommen zwar vereinzelt Computer bei der Administration und Koordinierung ausgewählter Aufgabenbereiche zum Einsatz (vgl. Abschnitt 2.4), eine direkte Hilfestellung bezüglich einer fehlerfreien Anwendung von Handlungsrouninen bieten diese jedoch bis heute nicht. Auch haben die bisherigen (meist akademischen) Lösungsansätze aus dem Bereich der Prozessmodellierung sowie der automatischen Handlungsplanung ihre deutlichen Schwächen, wenn man versucht, diese Konzepte eins zu eins auf die BOS-Domäne zu übertragen (vgl. Kapitel 3).

Doch was lässt sich tun, wenn sich die Prozesse im Vorfeld nicht zufriedenstellend erfassen lassen bzw. eine klassische (automatische) Handlungsplanung im Kontext einer Großschadenslage nicht praktikabel ist? Eine mögliche Antwort stellt die Anwendung von SOPs dar. SOPs können zu einer Harmonisierung von Handlungsrouninen beitragen. Jedoch sind sie im Vergleich zu Prozessen oder Handlungsplänen deutlich abstrakter formuliert und ermöglichen dadurch eine (gewollte) flexiblere Anwendung ihrer Handlungsbeschreibungen. REINWARTH definiert eine SOP als eine „*Beschreibung einer gewünschten Vorgehensweise bei der Erledigung bestimmter Aufgaben, die möglichst im Einklang mit operationell logischen Abläufen steht*“ ([Rei07], S. 13). Im SPEEDUP-Projekt zeigte sich frühzeitig, dass die angestrebte (formale) Modellierung derartiger SOPs nicht praktikabel ist, da die dort angewandten SOPs größtenteils nicht explizit vorliegen. Vielmehr lassen sich im BOS-Bereich kaum niedergeschriebene operationalisierte Handlungsrichtlinien in SOP-Form finden (vgl. Abschnitt 2). Dieser Einschätzung folgen auch BECKER et al. [BLK11] und PEINEL et

al. [PRW12a]. Doch wie lassen sich Fehler (Patzner und Schnitzer) mittels SOPs im BOS-Bereich vermeiden, wenn sich keine expliziten SOPs finden lassen?

Vergleicht man die Arbeitsbedingungen des BOS-Bereichs mit denen anderer Hochsicherheitsbereiche, in denen Checklisten seit langer Zeit erfolgreich eingesetzt werden, so lassen sich gemeinsame Charakteristiken feststellen, die für einen systematischen Checklisteneinsatz im BOS-Bereich sprechen. Hierzu zählen die Komplexität der Handlungsrouitnen, die Notwendigkeit einer reibungslosen Zusammenarbeit aller Einsatzkräfte (hier wird z. B. in ähnlichen Bereichen von einer Choreographie aller gesprochen [Gaw10]) und die stetig steigenden Anforderungen an die Einsatzkräfte. So müssen z. B. die Einsatzkräfte der Feuerwehr immer mehr Technik beherrschen und im Einsatzfall immer mehr Spezialwissen abrufen können.

Der BOS-Bereich weist jedoch überdies Merkmale auf, die eine Anwendung des Checklistenprinzips erschweren bzw. die eine spezielle Anpassung und neue Konzepte der Umsetzung erfordern. Folgende Hauptmerkmale eines komplexen, soziotechnischen Arbeitssystems einer Großschadenslage kommen hierbei erschwerend hinzu.

- *Dynamik der Einsatzlage.* Eine Großschadenslage ist eine hochgradig dynamische Arbeitsumgebung, deren Verlauf schwer vorhersagbar ist. Aufgrund dieser Unvorhersagbarkeit müssen die Einsatzziele, falls nötig, der Lageentwicklung angepasst werden. Einen detaillierten Einsatzplan vorzuhalten, ist daher nur für sehr beschränkte statische Einsatzlagen und in einer abstrakten Form sinnvoll.
- *Vernetztheit der Aufgaben.* Während z. B. im Cockpit oder im Operationssaal ein Team von Experten eine komplexe Arbeitsaufgabe bewältigt, wobei in der Regel „nur“ jeweils eine Checkliste aktiv ist, arbeiten in einem Großschadensfall verschiedene BOS parallel an sich teils bedingenden Aufgaben. Hier stellt sich die Aufgabe, die Vernetzung der SOPs mit zu berücksichtigen.
- *Ggf. geographische Verteilung.* Der Einsatzort kann sich über eine große Fläche erstrecken, wodurch eine direkte Face-to-Face-Kommunikation aller beteiligten Einsatzkräfte in der Regel nicht möglich ist. Der Frage nach einer Möglichkeit zum Informationsaustausch und Abgleich kommt daher zusätzlich eine große Bedeutung zu, wenn man als Ziel eine Koordination aller Einsatzkräfte verfolgt.

Um die oberen zusätzlichen Merkmale zu adressieren, bietet sich eine computerintegrierte Umsetzung von Checklisten an, deren Potential bis heute in der BOS-Praxis kaum ausgeschöpft wird (vgl. Kapitel 2). Zudem ergaben Anforderungsanalysen von Notfallmanagementsystemen aus dem zivilen und militärischen Bereich, dass eine elektronische Checklistenunterstützung mit zu den wichtigsten Funktionalitäten eines solchen Assistenzsystems zählen sollte (vgl. [RS08; LI09; WMY11]). Interviews mit Anwendern verschiedener BOS im Rahmen des SPEEDUP-Projektes ergaben, dass eine elektronische Checklistenunterstützung, als ein Hilfsmittel für Führungskräfte, durchaus gewünscht ist. Die Erkenntnisse von PEINEL et al. (siehe [PRW12a]) bestätigen diese Erfahrungen.

Thesen und Potential

Dass ECLs bezüglich PCLs ihre Vorteile besitzen, wurde bereits in der Luftfahrt erkannt. So kann im Gegensatz zu PCLs im Computer eine sehr große Anzahl von Checklisten hinterlegt und leicht recherchierbar gemacht werden. Für den BOS-Bereich zusätzlich interessant ist die Möglichkeit der Verteilung von Checklisten in einem geografisch verteilten Einsatzgebiet. Weiterhin können ECLs drei typischen Fehlern beim Umgang mit PCLs entgegenwirken ([PD91; Boo01a; Ark06]):

1. Vergessen (nach Unterbrechung oder Ablenkung) des aktuellen Items in der Liste.
2. Absichtliches (der Situation geschuldetes) Überspringen eines Items und dann vergessen, später zu diesem Item zurückzuspringen (sog. Wiederaufnahmefehler).
3. Item als „ausgeführt/beachtet“ markieren, obwohl dies nicht zutrifft. (Gilt für die Items, deren Erfüllung ein angeschlossenes IT-System automatisch überprüfen kann.)

Werden die besonderen Arbeitsbedingungen einer Großschadenslage bei der Entwicklung von elektronischen Checklisten berücksichtigt, so kann der Effekt der Gedächtnisentlastung auch im BOS-Bereich hilfreich genutzt werden. Dieser Kernansatz spiegelt sich in folgender Hauptthese dieser Arbeit wider:

These 1

Eine IT-unterstützte Umsetzung des Checklisten-Prinzips ist im BOS-Bereich besonders geeignet, um einen signifikanten Beitrag zur **Fehlervermeidung** und besseren Handhabung von Komplexität im Management einer Großschadenslage zu leisten.

Neben der Funktion der Fehlervermeidung können computerintegrierte Checklisten einen wesentlichen Beitrag zum Aufbau eines zentralen Lageüberblicks leisten. Jede Interaktion mit einer Checkliste eröffnet zugleich die Möglichkeit, Informationen über den Fortgang der Lagebewältigung zu erhalten. Dies können wichtige Informationen über den Verlauf kritischer Schritte einer SOP sein. Weiterhin ist es auch möglich, mit einem Item Informationen über die Einsatzlage vom Nutzer abzufragen bzw. zuvor annotierte Informationen beim Check eines Items automatisch zu einem Lagebild zusammenzuführen. Dieses Potential führt zu einer weiteren These dieser Arbeit:

These 2

Die Anwendung vernetzter computerintegrierter Checklisten kann für ein **Prozessmonitoring** genutzt werden, welches zusätzlich dem **Aufbau eines Lagebildes** dient und somit zu einem einheitlichen Lageverständnis (Situation Awareness) führt.

Wird den Einsatzkräften eine elektronische Checklisten-Assistenz zur Verfügung gestellt, so erschließen sich eine Vielzahl weiterer Zusatzfunktionalitäten, die in vielen Bereichen der Einsatzvorbereitung, -durchführung und -nachbereitung hilfreich sein können (siehe Kapitel 7). Bezüglich dieses zusätzlichen Praxispotentials ergibt sich eine allgemeinere, dritte These dieser Arbeit:

These 3

Eine elektronische Checklisten-Assistenz kann zu einer **qualitativen Verbesserung der gesamten Einsatzabwicklung** beitragen, die weit über die positiven Effekte der Fehlervermeidung und des Prozessmonitorings hinausreicht.

Grundlegende Arbeiten des im Folgenden vorgestellten Lösungsansatzes wurden bereits vom Autor in [KGK+10; WKK11; KWB12] einem Fachpublikum vorgestellt und diskutiert. In der BOS-Praxis existieren zurzeit keine vergleichbare checklistenorientierte IT-Unterstützung, die während einer Großschadenslage den Einsatzkräften eine Unterstützung vor Ort bietet. Aktuelle Arbeiten von PEINEL et al. [PRW12b] und BECKER et al.

[BLK11] zeigen hierfür zwar erste Lösungsansätze auf, skizzieren jedoch bislang deren genaue Umsetzung und vor allem Fragen des dynamischen Systemverhaltens nur recht vage (siehe Kap. 3).

Anforderungen an eine elektr. Checklisten-Assistenz für die BOS-Domäne

Nach der Analyse der BOS-Domäne in Kapitel 2, der Sichtung und Bewertung bisheriger Lösungsvorschläge in Kapitel 3 und der ausführlichen Betrachtung des Checklistenprinzips in Kapitel 4 werden im Folgenden die sich daraus herauskristallisierten Anforderungen an eine elektronische Checklisten-Assistenz zusammengefasst, die solch ein System leisten bzw. geeignet berücksichtigen sollte.

1. *Wahlfreiheit gewährleisten* – Keine Einschränkung bei der Wahl und Abarbeitung der hinterlegten CLs. Assistenz nur als Vorschlag, der jederzeit ablehnbar ist. Ein flexibler Wechsel zwischen verschiedenen Checklisten sollte immer möglich sein.
2. *Organisationsaufbau berücksichtigen* – Der Organisationsaufbau während einer Großschadenslage mit seinen unterschiedlichen Arbeitsbereichen sollte sich im Auswahlprozess der CLs widerspiegeln. Eine redundanzfreie Zuordnung der Checklisten zu den unterschiedlichen Arbeitsaufgaben sollte gewährleistet sein.
3. *Komplexität handhaben* – Damit bei komplexen SOPs die Übersichtlichkeit der zugehörigen CLs gewahrt bleibt, sollte das Anlegen und Anzeigen hierarchisch strukturierter CLs möglich sein.
4. *Dynamik berücksichtigen* – Reaktion auf die sich verändernde Einsatzlage (z. B. Vorschlag einer passenden Checkliste zu Gefahrgutlagen, wenn es eine Gefahrgutbeteiligung gibt). Ein Assistenzsystem sollte in seinem Verhalten die Dynamik einer Einsatzlage mit berücksichtigen und einen flexiblen Umgang mit den computerintegrierten Checklisten ermöglichen.
5. *Umgang mit Informationsmangel* – Ein CL-System sollte auch bei einem Mangel an (zuverlässigen) Informationen (Art der Gefährdung, Ausmaß, Anzahl Verletzter, örtliche Besonderheiten, Gefahren, Lageentwicklung) arbeitsfähig sein. D. h., für unbekannte Lagen kommen abstrakte CLs zum Einsatz, die bei wachsendem Wissensstand immer konkreter werden.
6. *Dokumentation* – Eine wichtige Forderung der Praxis ist das Protokollieren des gesamten Ablaufes einer Großschadenslage. Neben der rechtlichen Absicherung für jede BOS ist eine möglichst lückenlose Protokollierung die Grundlage einer jeden Einsatzbewertung. Eine elektronische Checklisten-Assistenz sollte daher auch einen wesentlichen Beitrag zur Einsatzdokumentation leisten.
7. *Dokumentenpool als Zusatzinformation* – Ein CL-System sollte die bereits existierenden elektronischen Dokumente genau an den Stellen eines Arbeitsablaufes zugreifbar machen, an denen sie beim Anwenden einer SOP nützlich sind.
8. *Einfache Bedienung* – Da im BOS-Bereich nicht davon auszugehen ist, dass alle Nutzer erfahren im Umgang mit Computern sind, und da sich die Arbeitsumgebungen und -bedingungen teilweise recht widrig gestalten (wechselnde Lichtverhältnisse, Regen, Schmutz, Kälte, Hitze, Stress), besteht die Notwendigkeit eines möglichst einfachen Benutzungskonzeptes.

9. *Einheitliches Vokabular* – Ein zentral verwaltetes Vokabular der Einsatzdomäne sollte als Quelle für eine einheitliche Begriffsverwendung bei den Texten dienen. Missverständnisse in den Begriffserklärungen sind zu vermeiden.
10. *Unterschiedliche Benutzergruppen* – Zusätzlich zu den Experten kommen in der Regel auch wenig trainierte bzw. unerfahrene Einsatzkräfte zum Einsatz. Eine elektronische Checklisten-Assistenz muss es auch diesen ermöglichen, die Checklisten mit ihren personalisierten Zusatzinformationen anzureichern und zu nutzen.

Lösungsansatz

Der folgende Teil der Arbeit stellt ein Rahmenwerk eines *intelligenten elektronischen Checklisten-Assistenzsystems* (im Folgenden kurz CASIE¹ genannt) vor. CASIE eröffnet ein neues Anwendungsszenario von Computerassistenz, dessen Besonderheit in dem bewussten Verzicht auf eine umfassende Prozessmodellierung zugunsten einer computerintegrierten Umsetzung von Checklisten liegt. Der gewählte Lösungsweg stellt somit einen Mittelweg zwischen einer vollständig normativen Handlungsanweisung und dem bloßen Vertrauen auf die menschlichen Problemlösefähigkeiten bei der Ausführung von Handlungen dar. Dabei bietet CASIE eine Checklisten-Assistenz auch für alternative soziotechnische Arbeitssysteme mit ähnlichen Merkmalen wie die des BOS-Bereiches. Obwohl das Rahmenwerk sich an den Anforderungen des BOS-Bereiches orientiert, adressiert es eine allgemeine Problemklasse. CASIE kann neben dem BOS-Bereich auch in alternativen sicherheitskritischen Arbeitsbereichen zu einer Harmonisierung der Handlungsabläufe und zur Verhinderung von Fehlern (z. B. Gedächtnisfehlern) beitragen. Hierzu bedarf es lediglich einer entsprechenden Konfiguration von CASIE für den entsprechenden Anwendungsbereich (siehe Abschnitt 6.3). Darüber hinaus bietet CASIE eine Vielzahl nützlicher Eigenschaften, die in Kapitel 7 erläutert werden.

Das Rahmenwerk beschreibt ein reaktives System, das ein „intelligentes“ Systemverhalten zeigt. Als intelligent wird hierbei die Fähigkeit des Systems verstanden, sich über die Arbeit mit den Checklisten einer sich ändernden Informationslage anzupassen. Das System überwacht den Fortschritt von Handlungsroutinen und gewinnt somit Wissen über seine Umwelt. Zum anderen wird das gewonnene Wissen dazu genutzt, Anwendungskriterien von Checklisten und deren Items zu überprüfen und somit eine dem aktuellen Wissensstand angepasste Checklisten-Assistenz zu bieten.

Der in dieser Arbeit vorgestellte Lösungsansatz hat jedoch nicht den Anspruch, eine Hilfestellung für eine detaillierte Einsatzplanung zu geben oder eine Grundlage für eine umfassende SOP-Modellierung zu sein. Beide Ansprüche sind hinsichtlich der in Kapitel 2 erarbeiteten Charakteristika der BOS-Domäne unangemessen. Vielmehr sollen die computerintegrierten Checklisten den Einsatzkräften eine einfache Assistenz zur Vermeidung von Handlungsfehlern an die Hand geben, wobei das Augenmerk nur auf den kritischen Punkten einer SOP liegen sollte. Die vorliegende Arbeit zeigt, dass ein Checklisten-System – frei nach dem Motto *keep it simple* – mit wenig viel erreichen kann. Vor allem die minimale Interaktion des Nutzers mit dem System, die sich im Wesentlichen auf das Aufrufen und Abhaken von Items beschränkt, ist den Einsatzkräften von PCL her gut vertraut und entspricht der Anforderung nach einem einfachen Benutzungskonzept. Zudem fördert eine einfaches Bedienkonzept die Akzeptanz des Computereinsatzes im BOS-Umfeld.

In der weiteren Arbeit wird vorausgesetzt, dass ein robustes Basiskommunikationssystem verfügbar ist, auf das die Funktionalität einer elektronischen Checklisten-Assistenz aufbau-

¹ Der Name CASIE wurde aus dem Akronym IECAS (intelligentes elektronisches Checklisten-Assistenz-System) abgeleitet und ist angelehnt an das irische Wort *casey* („umsichtig“, „wachsam“).

en kann. Wie genau solch ein robustes und flexibles Kommunikationssystem aufgebaut ist und wie es im Detail arbeitet, ist daher nicht Gegenstand dieser Arbeit. Das im Rahmen des SPEEDUP-Projektes entwickelte prototypische Kommunikationssystem ist zum Beispiel ein solches Basissystem, das gemäß den technischen und örtlichen Bedingungen eine robuste und flexible Datenkommunikation zwischen geografisch verteilten Einsatzkräften ermöglicht. Dessen prinzipielle Praxistauglichkeit konnte auf zahlreichen Übungen unter Beweis gestellt werden. Die allgemeinen Anforderungen und das prinzipielle Potential eines solchen Basissystems wurden in WUCHOLT et al. [WMY11] erarbeitet.

Auf die Aufgabe der Checklistenenerarbeitung (siehe Abschnitt 6.3) und der Umsetzung einer geeigneten Benutzerschnittstelle (siehe Abschnitt 6.1.1) wird nur am Rande eingegangen. Ohne Zweifel sind beides Problemstellungen, deren erfolgreiche Bearbeitung maßgeblich den Praxiserfolg eines hier beschriebenen Systems mitbestimmt. Da beide Thematiken jedoch eigenständige Forschungsgebiete betreffen, wird auf deren Bearbeitung zugunsten der Darstellung der Kernkomponenten und der Funktionsweise einer intelligenten elektronischen Checklisten-Assistenz verzichtet.

Der Autor ist sich bewusst, dass die Anwendung von IT im BOS-Bereich für die Anwender eine große Herausforderung darstellt. Des Weiteren ist davon auszugehen, dass sich neue Paradigmen und Technologien im öffentliche Bereich der BOS eher mittelfristig durchsetzen werden. Hinzu kommt, dass die Idee, in der BOS-Praxis verstärkt Computer einzusetzen, nicht nur auf Befürworter stößt. So besteht z. B. bei den Einsatzkräften die Befürchtung, sich von der IT abhängig zu machen und die Möglichkeit des flexiblen Handelns zu verlieren. Da der BOS-Bereich ein Arbeitsgebiet ist, das flexibles Handeln notwendig macht, darf diese Freiheit durch ein Assistenzsystem nicht eingeschränkt werden. Es ist daher unangebracht, zu regulierend in die Handlungsroutinen einzugreifen. Eine Handlungsassistenz, die lediglich grobe Leitplanken bietet, so wie das gute Checklisten tun sollten, ist der schwierigen Arbeitsumgebung und den Anforderungen einer Großschadenslage eher angemessen als eine zu detaillierte, starre Handlungsanweisung. Da Checklisten den Einsatzkräften jedoch nicht direkt ihre Entscheidungen abnehmen sollen, sondern vielmehr zur Absicherung und Entlastung dienen, können sich die Führungskräfte auf das Wesentliche (das Neue) einer Einsatzlage konzentrieren, ohne dabei die SOPs aus den Augen zu verlieren.

5.2 Grundlegende Konzepte und Designentscheidungen

Dieser Abschnitt stellt die Grundlagen der elektronischen Checklisten-Assistenz und die Designentscheidungen der CASIE-Architektur vor. Bevor der Aufbau und die Funktionsweise von intelligenten elektronischen Checklisten erläutert wird, wird zuvor das Konzept der *Rollen* eingeführt. Darauf folgend werden die wichtigsten Aspekte der Checklisten und Items erarbeitet und der Ansatz vorgestellt, wie Komplexität mittels Hierarchien gehandhabt werden kann. Dabei bezieht sich der Abschnitt auf verwandte Aspekte der HTN-Planung und auf Theorien der Handlungsregulation. Der Abschnitt erläutert die hierarchische Verknüpfung einzelner CLs und diskutiert die Möglichkeiten der Checklistenzuordnung zu den Einsatzlagen. Das Ende dieses Abschnittes motiviert den Einsatz eines standardisierten Vokabulars zur einheitlichen Begriffsverwendung.

5.2.1 Rollen und Rollenübernahme in einem Einsatz

Der Organisationsaufbau in einem Großschadensereignis ist unter anderem durch eine Aufteilung in unterschiedliche Arbeitsbereiche (z. B. in die sog. Einsatzabschnitte) charakte-

riert. Für diese sind bestimmte Aufgaben- und Verantwortungsbereiche definiert (vgl. Abschnitt 2). Jede Einsatzkraft nimmt in einem Großschadensereignis – wie auch im Alltagsbetrieb – eine bestimmte Position im Team ein, wie z. B. „Einsatzleiter Rettungsdienst“, „Leitender Notarzt (LNA)“ oder „Leiter Unterabschnitt Verletztenablage“.

Die Benennung jeder der Aufgaben- und Verantwortungsbereiche wird in dieser Arbeit als *Rolle* bezeichnet. Der Name einer Rolle stellt eine Art Signalwort dar, das die Einsatzkraft mit einem ganz bestimmten Aufgabenbereich verbindet. Um die Rollen geeignet im Computer zu repräsentieren, werden in dieser Arbeit neben den Namen weitere grundlegende Merkmale einer Rolle formalisiert. In CASIE findet folgende Definition einer Rolle Anwendung.

Definition 2 (Rolle) Die unterschiedlichen Positionen, die die Aufgaben in einem Arbeitsbereich bezüglich einer Einsatzkraftverteilung strukturieren und die eine Einsatzkraft übernehmen kann, heißen Rollen. Die Menge aller Rollen wird mit \mathcal{R} bezeichnet. Eine Rolle $r \in \mathcal{R}$ ist charakterisiert durch folgende Bestandteile:

- $name(r)$ – steht für einen eindeutigen Namen der Rolle r .
- $nc(r)$ – steht für die maximal zulässige Anzahl (Kardinalität) zuzuordnender Einsatzkräfte.
- $org(r)$ – steht für eine konkrete Organisation (BOS-Instanz), der die Rolle r zugeordnet ist und
- $desc(r)$ – steht für eine natürlichsprachliche Beschreibung der Verantwortungs- und Aufgabenbereiche der Rolle r .

Es gibt Rollen, die können mehreren Einsatzkräften zugeordnet werden, z. B. „Maschinist“ oder „Rettungsassistent“ (RA) oder aber Rollen, die nur ein einziges Mal besetzt werden, zum Beispiel die eines „Leitenden Notarztes“ (LNA). Durch die Angabe von nc kann dies genau festgelegt werden. In CASIE kann durch eine strikte Überwachung der Kardinalität sichergestellt werden, dass bestimmte Aktionen eines Aufgabenbereiches nicht versehentlich mehrfach eingeleitet werden. Die Angabe der Organisation (org) ist zur zusätzlichen Abgrenzung der Rollenkonzepte unterschiedlicher BOS notwendig, da es hier (wenn auch selten) zu Bedeutungsabweichungen bezüglich der Rollennamen kommen kann. Somit lassen sich Rollen verschiedener BOS in einer Großschadenslage zusammenführen und unterscheiden. Über das Merkmal $desc$ kann eine Rollenbeschreibung angegeben werden, die zum einen hilfreich bei der Modellierung aller Rollen im System ist und zum anderen als zusätzliche Informationsquelle im Einsatz (zur *runtime*) genutzt werden kann.

Beispiel 1 Sei $r1 \in \mathcal{R}$ die Rolle eines Organisatorischen Leiters Rettungsdienstes (OrgL) (vgl. Abschnitt 2). Dann kann die Rolleninstanz $r1$ z. B. folgende Ausprägung haben:

$$\begin{aligned} name(r1) &= \text{„Organisatorischer Leiter Rettungsdienst (OrgL)“}, \\ nc(r1) &= 1, \\ org(r1) &= \text{„Deutsches-Rotes-Kreuz Jena“}, \\ desc(r1) &= \text{„Der Organisatorische Leiter unterstützt den Leitenden Notarzt (LNA), indem er organisatorische Führungs- und Koordinationsaufgaben übernimmt. Er ist gegenüber dem Personal des Rettungsdienstes und den sonstigen zur rettungsdienstlichen Versorgung eingesetzten Kräften weisungsbefugt.“}. \end{aligned}$$

□

über eine zusätzliche Ausbildung als Führungskraft. Da nicht alle Mitarbeiter über spezielle Führungsqualifikationen verfügen, kommt es jedoch in der Praxis ab und zu vor, dass Mitarbeiter ohne entsprechende Qualifikation eine Führungsrolle – typischerweise zu Beginn einer Großschadenslage – kommissarisch einnehmen müssen, bis die entsprechend qualifizierten Einsatzkräfte eintreffen. Auch wird nicht zwangsläufig jede Einsatzkraft nach ihrer maximalen Qualifikation eingesetzt. Eine Rollenzuordnung spiegelt daher nicht zwangsläufig die Qualifikation einer Einsatzkraft wider, sondern lediglich deren eingenommene Rolle im Team. Das Konzept der Rolle ist daher von dem der Qualifikation zu unterscheiden.

5.2.2 Charakteristische Merkmale von Checklisten und SOPs

Um die Unterschiede (vgl. Abschnitt 4.3) und Gemeinsamkeiten der Konzepte SOP und Checkliste herauszuarbeiten, wird im Folgenden eine typische Struktur einer SOP definiert, von der dann die entsprechende Struktur von Checklisten abgeleitet wird. SOPs werden zwar in CASIE nicht direkt hinterlegt, jedoch verfolgt die Arbeit das Ziel, die Korrespondenz beider Konzepte auch formal repräsentieren zu können. In einem späteren Abschnitt (siehe Abschnitt 6.3.2) wird die sich daraus ergebende Möglichkeit einer Rückübersetzung von Checklisten zu (Teil-)Prozessmodellen einer SOP motiviert.

Eine SOP formal zu erfassen, wird nicht zuletzt dadurch erschwert, dass in der Praxis kein Konsens darüber besteht, welche Informationen eine SOP überhaupt beinhalten sollte und wie diese strukturiert sind. Vor allem aus dem angelsächsischen Raum stammt eine Vielzahl von Vorschlägen, wie SOPs zu erarbeiten sind und wie sie aufgebaut sein sollten (siehe hierzu z. B. [FEM99; Gru03; CG05; DHS06]). Diese Vorschläge unterscheiden sich mehr oder weniger stark voneinander, was auf die unterschiedlichen Einsatzgebiete und die vielfältigen Auffassungen davon, was alles unter einer SOP zu verstehen ist, zurückzuführen ist. Nichtsdestotrotz zeigen diese Arbeiten, dass SOPs strukturelle Invarianten besitzen sollten. Die vorliegende Arbeit orientiert sich an der Herangehensweise von CIMOLINO [CG05], welche sich an der Struktur sogenannter Verfahrensanweisungen aus dem Bereich des Qualitätsmanagements (QM) ausrichtet. Die folgende Definition soll den Anforderungen einer Checklistenableitung Genüge leisten:

Definition 3 (SOP) *Eine operationalisierte Arbeitsanweisung, die eine (gemäß den Vorschriften und Richtlinien) gewünschte Vorgehensweise bei der Erledigung bestimmter Aufgaben beschreibt, heißt SOP. Sei \mathcal{P} die Menge aller SOPs in der Domäne. Eine SOP $p \in \mathcal{P}$ wird durch folgende Merkmale charakterisiert:*

- $name(p)$ – steht für den eindeutigen Namen der SOP,
- $org(p)$ – bezeichnet die Organisation, in deren Einsatzbereich die SOP fällt,
- $pur(p)$ – steht für eine natürlichsprachliche Beschreibung des Zweckes der SOP,
- $r(p)$ – erläutert klar, für welche Rolle(n) (bzw. Rollengruppen) die SOP gültig ist,
- $app(p)$ – beschreibt, wann (in welcher Situation) die SOP angewendet werden sollte,
- $desc(p)$ – steht für eine (natürlichsprachliche) detaillierte Beschreibung der Prozedur,
- $notes(p)$ – steht (optional) für eine Menge von Hinweisen auf besonders zu beachtende Sachverhalte in der SOP, und
- $term(p)$ – steht (optional) für eine unmissverständliche Beschreibung wichtiger Begriffe und Abkürzungen, die in $desc$, pur und $notes$ relevant sind.

Die Beschreibung (*desc*) des Prozesses stellt den Kern einer SOP dar. Die Möglichkeiten einer geeigneten Formalisierung und Strukturierung sind hierbei so vielfältig wie die Anwendungsgebiete und die Organisationen selbst. Zusätzlich machen es regionale Eigenheiten sowie persönliche Vorlieben schwierig, eine einheitliche Struktur der Beschreibung *desc* festzulegen. In CASIE wird daher bewusst nicht der Versuch unternommen, diese Beschreibung detailliert zu formalisieren, so wie es teils in den in Kapitel 3 vorgestellten Arbeiten versucht wurde. Vielmehr soll obige Definition den Weg zu einer geeigneten Checklisten-Definition bereiten, in der die Korrespondenz zu den strukturellen Invarianten einer SOP Beachtung findet.

Beispiel 2 Als Beispiel p1 einer SOP dient das Vorgehen bei Auslösen einer Brandmeldeanlage aus der Feuerwehrpraxis (siehe [StadtVS06], S. 33 ff.). Es könnte dann z. B. für p1 gelten:

$name(p1) =$ „Brandmeldeanlagen“,
 $org(p1) =$ „Freiwillige Feuerwehr Villingen-Schwenningen“,
 $pur(p1) =$ „Vorgehensbeschreibung und Hinweise bei Auslösung einer Brandmeldeanlage für die taktische Grundeinheit der Gruppe“,
 $r(p1) =$ „Gruppenführer FW“,
 $app(p1) =$ „Bei Alarm einer Brandmeldeanlage“,
 $desc(p1) =$ (siehe [StadtVS06], S. 33-37),
 $notes(p1) =$ (siehe [StadtVS06], S. 33-37),
 $term(p1) =$ „BMA – Brandmeldeanlage; BMZ – Brandmelderzentrale; ÜE – Übertragungseinrichtung; FBF – Feuerwehr-Bedienfeld; FAT – Feuerwehr-Anzeigetableau; FSD – Feuerwehr Schlüssel Depot“ .

□

Beispiel 2 zeigt Teile einer SOP-Beschreibung aus dem Feuerwehrbereich. Das Merkmal $term(p1)$ beinhaltet in diesem Beispiel „nur“ eine Erläuterung der in der Beschreibung $desc(p1)$ und in den Hinweisen $notes(p1)$ verwendeten Fachbegriffe. Auf eine detaillierte Angabe von $desc$ und $notes$ wurde aus Platzgründen verzichtet und auf die entsprechende Quelle verwiesen. Zufälligerweise sind in der Beispiel-SOP die Hinweise ($notes$) in Form einer Checkliste (siehe Anhang, Seite 138) gegeben. Im Allgemeinen können diese Hinweise jedoch eine beliebige Form (Freitext, Stichpunkte oder Kombination aus beiden) aufweisen. Gerade mit Blick auf die Erarbeitung computerintegrierter Checklisten, die mehr bieten sollen als nur eine bloße Auflistung von Items, spielen die Metadaten (gemeint sind alle obigen Merkmale außer $desc$) einer SOP eine wesentliche Rolle dabei, die jeweiligen Checklisten zur aktuellen SOP zuzuordnen, und bei der Realisierung eines dynamischen und flexiblen Systemverhaltens.

Bevor auf die Items als wesentlichste Elemente einer Checkliste genauer eingegangen wird, soll im Folgenden deren Struktur unter Beachtung der zuvor erarbeiteten SOP-Korrespondenz definiert werden:

Definition 4 (Checkliste) Eine Liste kritischer Aspekte einer SOP heißt Checkliste. Eine Checkliste cl ist charakterisiert durch folgende Bestandteile:

- $name(cl)$ – steht für einen eindeutigen Namen von cl ,
- $org(cl)$ – steht für den Organisationsnamen, in deren Einsatzbereich cl fällt,

- $pur(cl)$ – steht für eine natürlichsprachliche Beschreibung des Zweckes von cl ,
- $\mathcal{R}(cl)$ – steht für die Menge von Rollen, in deren Kontext cl beachtet werden muss,
- $\mathcal{K}(cl)$ – steht (optional) für eine Menge von Anwendungskriterien, d. h. Kriterien der Umwelt (Einsatzlage), die die Anwendung der Checkliste indizieren,
- $\mathcal{I}(cl)$ – ist ein Paar (Φ, \prec) , wobei Φ für eine Menge kritischer „Aspekte“ (Items genannt) einer zu cl zugehörigen SOP und \prec für eine Menge von Ordnungsconstraints über Φ steht.
- $stat(cl)$ – steht für einen Bearbeitungszustand aus $\{\text{inactive, active, complete, canceled}\}$.

Die Merkmale $name$, org und pur entsprechen den gleichnamigen Merkmalen einer SOP (siehe Def. 3). Die Menge $\mathcal{R}(cl)$ aller für cl relevanten Rollen entspricht der Aufzählung der Rollen $r(p)$ einer SOP. Durch die Einführung der Menge \mathcal{R} soll ein expliziter Bezug auf die formalen Rollendefinitionen genommen werden. Weiterhin entspricht die Menge der Anwendungskriterien \mathcal{K} der Menge app einer SOP. Jedoch soll in CASIE, mit Blick auf eine computerverarbeitbare Situationsbeschreibung, die Angabe der Anwendungskriterien \mathcal{K} in einer formalen und präziseren Form als „nur“ in einer natürlichsprachlichen Beschreibung möglich sein. Näheres hierzu folgt im Abschnitt 6.3, in dem der Aufbau und die Nutzung einer formalen Wissensbasis thematisiert werden. Die Struktur \mathcal{I} , bestehend aus der Menge der Items und optional einer Menge von Ordnungsrelationen über den Items, stellt den Kern einer jeden Checkliste dar. Obwohl die Items typischerweise in Form einer Liste organisiert sind, bietet CASIE die Möglichkeit der Angabe einer partiellen Ordnung über den Items. Über die Angabe der partiellen Ordnung (\prec) kann eine gewünschte „Ausführungsreihenfolge“ der Items festgelegt werden. Dadurch ist es möglich, eine zeitlich-kausale Abhängigkeit kritischer SOP-Punkte zu repräsentieren. Das ermöglicht eine flexible Formulierung der Reihenfolge und der späteren Systemüberwachung der Item-Checks.

In Abgrenzung zu einer einfachen Abbildung von papierbasierten Checklisten in einem Computer wird im Folgenden unter einer *intelligenten elektronischen Checkliste* (kurz *intelligente CL* oder *ICL*) eine computerintegrierte Umsetzung einer (papierbasierten) Checkliste verstanden, die mittels einer grafischen Benutzerschnittstelle dem Benutzer eine Interaktion (Abhaken) mit der Liste ermöglicht (analog zu ECLs) und zusätzlich die folgenden Merkmale aufweist:

1. Ausführungsüberwachung von Items (Vergessen/Überspringen von Items verhindern),
2. Situationsorientierte, dynamische Anpassung der Items an die Einsatzlage,
3. Handhaben der Komplexität durch Hierarchien von Checklisten/Items und
4. Vernetztheit der Checklisten untereinander, auch über Rollengrenzen hinweg.

Eine ICL bietet daher mehr als eine bloße Speicherung einer statischen Liste von Items im Computer, die außer der Möglichkeit des Anzeigens und des Abhakens der Items keine weitere Funktionalität aufweist.

Jede Checkliste lässt sich in der Regel genau einer SOP zuordnen, unabhängig davon, ob die jeweilige SOP explizit (formal erarbeitet) oder implizit („in den Köpfen“) vorliegt. Dieser Zusammenhang soll in dieser Arbeit als *Prozesskontext*, kurz P_K , bezeichnet werden. Sei \mathcal{C} die Menge aller Checklisten, so lässt sich für jedes Element aus \mathcal{C} ein entsprechender Prozess aus \mathcal{P} zuordnen.

In der Praxis ist zu erwarten, dass, nicht zuletzt durch die für PCLs gewohnte und daher präferierte *Challenge-Verification-Response* Bearbeitungsmethode (vgl. Abschnitt 4.2.3), in der Regel von den Anwendern eine totale Ordnung der Items gewählt wird. Jedoch kann für ICLs der Freiheitsgrad bei der Angabe der Itemreihenfolge durchaus von Vorteil sein. So kann z. B. in Anwendungsgebieten mit einer Vielzahl zeitlich ausgedehnter und voneinander unabhängiger Aktionen deren parallele Abarbeitung sich in einem entsprechenden Freiheitsgrad in der vorgeschriebenen Reihenfolge der zugehörigen Items widerspiegeln. Zusätzlich kann das Problem, dass eine feste Itemreihenfolge von den Anwendern als zu einengend empfunden wird, somit für einige Einsatzfälle entschärft werden. Ob und wie eine Angabe einer Halbordnung der Items relevant ist, hängt jedoch einzig und allein von der jeweiligen zur CL korrespondierenden SOP ab.

Der Bearbeitungszustand einer ICL wird durch einen der vier Zustände *inactive*, *active*, *complete* und *canceled* repräsentiert. Neben *active* für „noch unvollständig bearbeitet“ und *complete* für „abgeschlossen“ lässt sich über den Status *canceled* ein gewollter Abbruch der Bearbeitung ausdrücken. ICLs, die noch keiner Einsatzkraft zur Bearbeitung zugeordnet wurden, gelten als *inactive* und werden im Folgenden auch als *CL-Schema* bezeichnet, da erst eine spätere Einsatzsituation die konkrete Instanz einer Checklistenhierarchie bestimmt. Die Erstellung dieser Schemata obliegt den jeweiligen BOS. Auf Fragen der Erarbeitung der ICL, speziell der Erarbeitung der kritischen Punkte einer SOP, wird in Abschnitt 6.3.2 eingegangen. Das Problem inkonsistenter Ordnungsconstraints, das bei einer unsachgemäßen Erstellung von ICLs auftreten kann, soll in dieser Arbeit nicht näher betrachtet werden. Vielmehr wird im Weiteren davon ausgegangen, dass die Item-Ordnung (↔) keine widersprüchlichen Elemente enthält.

Kommen wir nun zu den Items als wesentliche Bestandteile einer Checkliste. Obwohl jedes Item im Prinzip jeden erdenklichen, als kritisch erachteten Aspekt (Punkt, Sachverhalt) einer entsprechenden SOP repräsentieren kann (vgl. Abb. 4.7), lassen sich Invarianten in ihrer Ausprägung und ihren Intentionen erkennen. Nach der Sichtung typischer Checklisten aus der Praxis (siehe z. B. im Anhang, S. 138 ff.) und auf Basis der Analyse aus Kapitel 4 wird mit Blick auf die angestrebte Funktionalität des Assistenzsystems folgende Unterscheidung in drei Itemtypen gewählt:

Hinweis (*note-Item*) – soll den Anwender an einen allgemeinen Sachverhalt erinnern. Hinweise betreffen Meta-Aspekte der Abarbeitung, zu denen z. B. die bereits erwähnten *Not-To-Dos* zählen. Beispiele sind:

- Keine hohen Hindernisse in der Nähe des Landeplatzes. und Ruhe bewahren!

Aktion (*action-Item*) – steht für eine konkrete Aufgabe (einen Prozessschritt) in einer SOP und soll sicherstellen, dass diese Aufgabe ausgeführt bzw. delegiert wird. Beispiele hierfür sind:

- Rückmeldung an RLS über Lage! und Hubschrauberlandeplatz einrichten!

Hinter jedem dieser Items verbirgt sich eine konkrete bzw. abstrakte Aufgabe (*Task*). Für abstrakte (komplexe) Tasks, die auf eine eigene Sub-SOP referenzieren, kann (falls definiert) zur Konkretisierung eine eigens für diese Unteraufgabe zugeschnittene ICL bereitgestellt werden (siehe Abschnitt 5.2.3, S. 57 ff.).

Geschlossene/Offene Frage (*query-Item*) – Geschlossene Fragen sollte der Anwender schnell und zweifelsfrei (mit *ja* oder *nein*) beantworten können. Zum Beispiel:

- Anzahl der Notärzte ausreichend? ja/nein und Ist Gefahrgut involviert? ja/nein

Offene Fragen können hingegen beliebige Informationen über die Einsatzlage abfragen. Die Informationen können dann zur weiteren Steuerung der ICL-Darbietung bzw. zur Gewinnung eines Lagebildes genutzt werden. Ein Beispiel ist:

- Erste Schätzung über die Anzahl Verletzter: ...

Über offene Fragen können Quantitäten eines bestimmten Konzeptes (Anzahl Schwerverletzter oder Anzahl der Rettungsmittel vor Ort) abgefragt werden. Aber auch Fragen bezüglich qualitativer Eigenschaften von Konzepten sind möglich. So kann z.B. die Frage nach der Art eines Gefahrstoffes durch die Angabe seines Namens beantwortet werden. Denkbar sind weiterhin auch Fragen, die nur durch einen Freitext beantwortet werden können. Für diese ist jedoch eine spätere automatische Auswertung im Computer nicht ohne Weiteres möglich. Darüber hinaus haben Praxistests des SPEEDUP-Systems gezeigt, dass im Ernstfall wenig Zeit für das Eintippen von längeren Informationen in Freitextform bleibt.

Während bei klassischen PCLs die Checks und die niedergeschriebenen Antworten der Dokumentation und/oder zur Erinnerung für den Anwender dienen, erschließt sich in ICLs die Möglichkeit, gezielt Informationen über die Einsatzlage durch entsprechend vorbereitete ICLs für ein Computersystem bereitzustellen. Die Beachtung unterschiedlicher Itemtypen und deren Abarbeitungsstatus ermöglicht einen qualitativen Sprung einer Checklistenutzung, verglichen mit dem, was mit klassischen PCLs erreicht werden kann. Für die Einsatzleitung können wichtige Informationen über den Verlauf des Einsatzes mit Hilfe des Computers zu einem Lagebild verarbeitet werden. Um eine erweiterte Ausführungsüberwachung der Items zu erreichen, werden die Items in einer entsprechenden Struktur repräsentiert. Folgende Itemdefinition legt die Grundlage dafür, dass CASIE im Einsatz erweiterte qualitative Aussagen über den Bearbeitungsstand der Checklisten und somit den Verlauf des gesamten Einsatzes erhalten kann.

Definition 5 (Item) Die Listenelemente einer Checkliste heißen Items. Ein Item ϕ zeichnet sich durch folgende Merkmale aus:

- $text(\phi)$ – steht für einen stichpunkthaften Informationstext (für die Anzeige in der Benutzeroberfläche vorgesehen), der einem kritischen Aspekt einer SOP entspricht,
- $desc(\phi)$ – steht für eine (optionale) ausführlichere Erläuterung von $text(\phi)$,
- $res(\phi)$ – steht für eine Menge optionaler Zusatzressourcen (Dokumente, Tabellen, Abbildungen),
- $type(\phi)$ – steht für dessen Typ $\{action, note, query\}$,
- $stat(\phi)$ – steht für einen Bearbeitungsstatus $\{open, inprogress, done, cancel, fail, ignor\}$,
- $Sub(\phi)$ – steht (optional) für eine Menge von Sub-Checklisten, die je nach Erfüllung ihrer Anwendungskriterien ϕ entsprechend weiter konkretisieren können,
- $Pre(\phi)$ – steht (optional) für eine Menge an Vorbedingungen, und
- $Eff(\phi)$ – steht (optional) für eine Menge von Effekten (Merkmalen), die in der Einsatzlage explizit gelten, nachdem das Item bestätigt wurde.

Im Folgenden werden die einzelnen Strukturelemente der Definition 5 näher erläutert. Wenn im weiteren Verlauf dieser Arbeit von CLs oder ICLs die Rede ist, ist genau obere CL- und Itemstruktur gemeint.

Neben der klassischen Angabe eines Itemtextes (*text*) ermöglicht die Angabe von Zusatzinformationen (*desc*) und Dokumenten (*res*) selbige genau an den Stellen im Arbeitsablauf einer SOP zugänglich zu machen, an der sie im Einsatz auch relevant sind. Von besonderer Bedeutung in CASIE ist der Typ eines Items. Die in dieser Arbeit vorgenommene Klassifizierung der Items in *action*-, *note*- und *query*-Item ist das Ergebnis einer Analyse zahlreicher PCLs, die der Autor im Rahmen dieser Arbeit gesichtet hat. So können in CASIE Items angemessen berücksichtigt werden, die mit einer Aktion korrespondieren (*action*-Items wie z. B. „Hubschrauberlandeplatz einrichten!“), welche in der Regel eine gewisse Zeitspanne benötigt und gegebenenfalls auch fehlschlagen kann (siehe reaktives Systemverhalten, Abschnitt 6.2.2). Hingegen spiegeln die *note*-Items lediglich allgemeine Hinweise wider, so dass für diese Items keine zusätzlichen Funktionalitäten bezüglich des Systemverhaltens von CASIE notwendig sind. Unter dem Begriff *query*-Item sind all diejenigen Items zusammengefasst, in denen eine zusätzliche Nutzerinteraktion (neben dem Abhaken) in Form einer Dateneingabe modelliert wird. Mit CASIE ist es somit möglich, während der Arbeit mit einer Checkliste zugleich Informationen über die Einsatzlage in ein Computersystem aufzunehmen, weiterzuleiten und auszuwerten.

Weiterhin kann die Angabe des Itemtyps auf der Benutzeroberfläche dafür genutzt werden, die Items visuell nach ihrem Typ zu unterscheiden. Im Gesamtsystem können für die unterschiedlichen Typen computergestützte Dienste bereitgestellt werden. So kann z. B. zur Erfüllung des *action*-Items „Informiere IRL über Anzahl Verletzter!“ ein entsprechend zur Verfügung stehender elektronischer Dienst zur Informationsweitergabe (*information-propagation service*) genutzt werden. Oder es kann in der Gegenrichtung eine Informationsabfrage (*information-gathering service*) (z. B. „Anzahl der Unfallbetten im Klinikum X abfragen!“) automatisiert werden. In [KGK+10] wird hierzu bereits ein Ansatz der Einbindung semantischer Dienste für Aktionen in einem Krisenszenario vorgestellt.

Die *stat*-Eigenschaft repräsentiert den Zustand eines Items. Die Status *open* und *done* stehen für die noch leere (noch nicht bestätigte) bzw. „abgehakte“ Checkbox. Um eine zeitliche Ausdehnung und einen möglichen Fehlschlag einer Aktion (*action*-Items) zu repräsentieren, werden in CASIE zusätzlich die Zustände *inprogress* (noch in Bearbeitung) und *fail* (expliziter Fehlschlag) berücksichtigt. Die Ereignisse, die Zustandsübergänge von Checklisten und deren Items bewirken, werden im kommenden Abschnitt 6.2 genauer erläutert.

Eine Verknüpfung eines Items zu einer entsprechend konkretisierten Checkliste kann mittels der Eigenschaft *Sub* realisiert werden (siehe Abschnitt 6.2). Falls für ein Item ϕ keine Sub-Checkliste zur eventuellen Detaillierung angegeben ist, wird ϕ als *primitiv* bezeichnet, anderenfalls heißt ϕ *nicht-primitiv*. Synonym wird im Weiteren für *nicht-primitiv* auch von *abstrakten* Items gesprochen.

Durch die Menge der Vorbedingungen $Pre(\phi)$ und der Effekte $Eff(\phi)$ kann der Teil der Einsatzlage beschrieben werden, welcher für eine Beachtung gelten muss bzw. welcher sich durch eine erfolgreiche Bearbeitung des jeweiligen Items ergibt. Analog zu der Menge der Anwendungskriterien \mathcal{K} einer Checkliste sollen die Pre- und Effektmerkmale gemäß eines formalisierten Domänenvokabulars in einer computerverarbeitbaren Form abgelegt werden. Details hierzu werden im Abschnitt 6.3 erläutert.

Beispiel 3 Als Beispiel *i1* eines konkreten Items dient ein kritischer Schritt aus der SOP aus Beispiel 2. Es zeigt eine mögliche Ausprägung eines Items im Kontext einer Checkliste

zum Vorgehen beim Signaleingang einer Brandmeldeanlage (BMA) (vgl. Anhang, Seite 138):

```

name(i1) = „Entnahme Plan BMA“,
text(i1) = „Brandmeldeanlage (BMA) zurücksetzen.“,
desc(i1) = „BMA wird mit dem Taster 'BMZ rückstellen' im Feuerwehr-
Bedienfeld wieder in den Ruhezustand versetzt. Eine Rückstellung
darf erst nach Kontrolle der ausgelösten Melder erfolgen!“,
res(i1) = {'Abbildung Feuerwehr-Bedienfeld'},
type(i1) = action,
stat(i1) = open,
Sub(i1) = ∅,
Pre(i1) = ∅,
Eff(i1) = {bma_zurückgesetzt('BMA1')}.
    
```

□

Das Beispiel 3 zeigt, welche Arten von Informationen für ein Item in CASIE hinterlegt werden können. Die Referenz auf die Abbildung als zusätzliche Ressource zum Item ist im Beispiel nur angedeutet. Zu Beginn ist dem Item per *default* der Status *open* zugeordnet, damit wird angezeigt, dass dieser Punkt noch beachtet werden muss. Da die Menge *Sub(i1)* leer ist, handelt es sich bei *i1* um kein abstraktes Item. Als Effekt ist hinterlegt, dass die betreffende Brandmeldeanlage (BMA1) in den normalen Betriebszustand zurückgesetzt wurde. Effekte werden erst nach erfolgreicher Bestätigung der entsprechenden Items zum Aufbau und Update der Lageinformation genutzt. Näheres hierzu wird im Abschnitt 6.2 *Dynamik* erläutert.

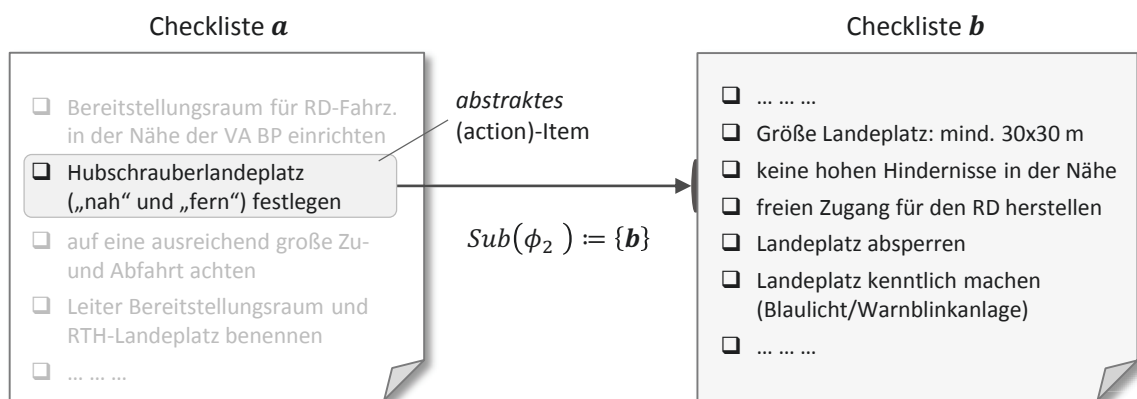


Abbildung 5.2: Beispiel einer Konkretisierung eines abstrakten Items.

5.2.3 Komplexität handhaben durch Hierarchien

Die Umsetzung von PCLs ist in der Regel aus Platz- und Handhabbarkeitsgründen auf eine oder wenige A4-Seite(n) beschränkt. Einfache SOPs lassen sich somit überschaubar darstellen. PCLs stoßen jedoch bei komplexeren SOPs schnell an ihre Grenzen. Eine bekannte und effektive Methode zur Handhabung von Komplexität ist die Abstraktion. Angewandt auf komplexe SOPs heißt das, dass bei Bedarf Teilaspekte einer SOP in Form eines (abstrakten) Items zusammengefasst werden. Doch was tun, wenn die SOPs schwer abstrahiert werden

können, da dadurch wichtige Informationen verloren gehen oder Schritte ausgelassen werden? Eine Lösung liegt darin, komplexe SOPs in einzelne, unabhängige Bestandteile zu zerlegen und für jeden dieser Teilbereiche jeweils eine eigene, detailliertere CL umzusetzen.

Durch den Computereinsatz erschließt CASIE die Möglichkeit einer hierarchisch vernetzten Gesamtverwaltung dieser Checklisten. In CASIE lassen sich zusammengehörige Teile einer SOP als ein sogenanntes *abstraktes* Item (auch *nicht-primitives* Item genannt) in die übergeordnete ICL dieser SOP aufnehmen (vgl. Abschnitt 5.2). Bei Bedarf kann der Anwender über die abstrakten Items Zugriff auf die entsprechenden referenzierten Sub-CLs erhalten.

In Abbildung 5.2 ist als Beispiel eines abstrakten Items und einer möglichen Konkretisierung die Aufgabe, einen Hubschrauberlandeplatz („nah“ und „fern“) einzurichten, illustriert (angelehnt an [PM01], S. 92). Der Einfachheit halber besitzt im Beispiel die Menge $Sub(\phi_2)$ mit der Checkliste **b** nur ein einziges Element. Im Allgemeinen können jedoch für ein abstraktes Item mehrere alternative Realisierungen angegeben werden.

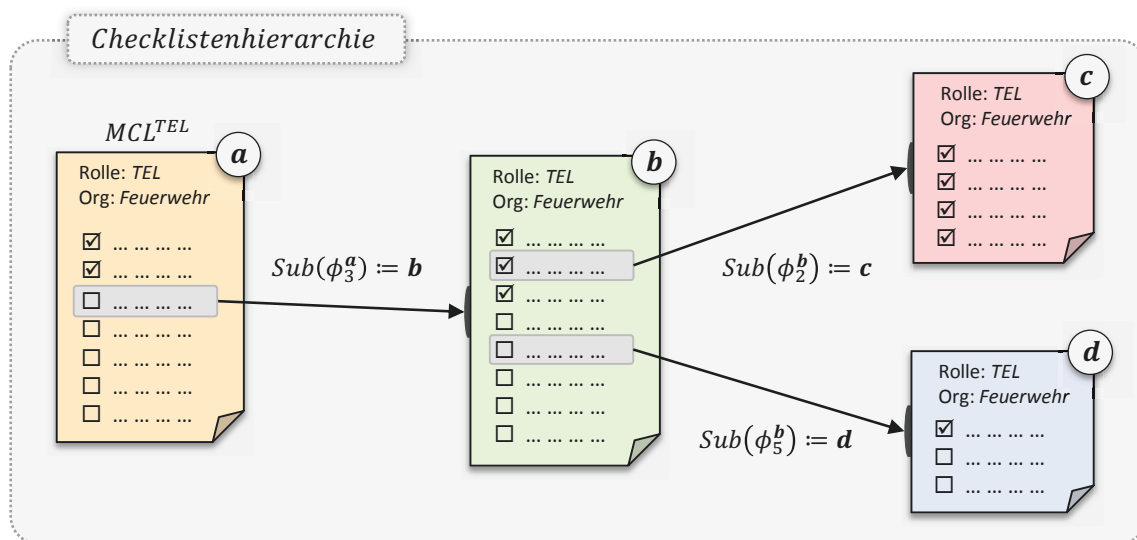


Abbildung 5.3: Schema einer Konkretisierung abstrakter Items.

Die Methode der Abstraktion ist im Prinzip für alle drei in dieser Arbeit unterschiedenen Typen von Items anwendbar. Neben den *action*-Items eignen sich auch *note*-Items zur Abstraktion für komplexere Hinweise. Aber auch auf *query*-Items lässt sich das Prinzip anwenden. Somit können z. B. allgemeine Fragen über spezielle Bereiche einer Einsatzlage auf eine Sub-Checkliste verweisen, mit der sich detailliertere Angaben über die Lage vom Benutzer abfragen lassen.

Die Abbildung 5.3 zeigt ein Beispielschema einer sich durch Abstraktion ergebenden *Checklistenhierarchie* (CLH). Jedes abstrakte Item ϕ referenziert dabei auf eine Menge $Sub(\phi)$ möglicher Sub-Checklisten als mögliche Konkretisierung. Hier wurde ebenfalls als Vereinfachung eine jeweils einelementige Menge $Sub(\phi)$ angenommen. Hierdurch ergibt sich eine Hierarchie von Checklisten, die der Struktur eines Baums ähnelt. Ausgehend von einer Wurzel an oberster (abstrakter) Stelle, fächert sich somit ein Baum auf, dessen Blattknoten aus Checklisten bestehen, deren Items alle primitiv sind. Alle inneren CL-Knoten besitzen hingegen abstrakte Items. Die Checklistenhierarchie stellt das Prinzip einer Konkretisierung abstrakter Items einer Checkliste **a** dar. Der Aufbau einer solchen CLH wird in Abschnitt 6.2 genauer erläutert.

Der Einfachheit halber wurde in der Abbildung 5.3 als Ordnungsrelation der Items eine Totalordnung angenommen. Die Ordnung der Items entspricht in diesem Fall einer Sequenz von Items. In den kommenden Beispielen wird, falls nicht anders angegeben, immer von einer Totalordnung der Checklistenitems ausgegangen.

Mastercheckliste und Rollenübergang

Die Checkliste an oberster Position einer CLH nimmt in CASIE einen besonderen Stellenwert ein. Sie wird im weiteren *Mastercheckliste* (MCL) einer CLH genannt (siehe Abb. 5.3, links) und dient der jeweiligen Einsatzkraft als ein Ausgangspunkt der Checklistenbearbeitung. Für eine MCL gilt, dass keine andere Checkliste auf sie als Sub-Checkliste referenziert. Des Weiteren sind über eine MCL alle relevanten Sub-Checklisten der Rolle zu erreichen.

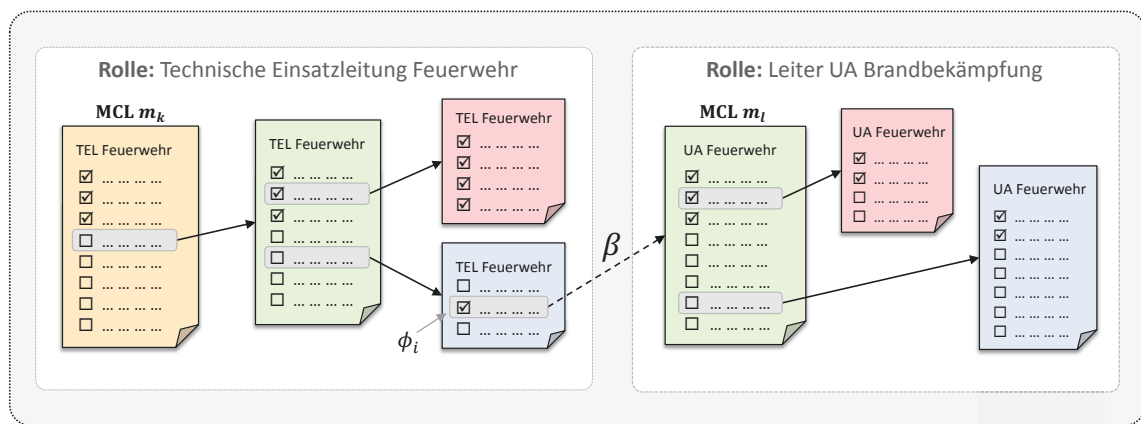


Abbildung 5.4: Rollenbegrenzung in einer Checklistenhierarchie.

Die Designentscheidung in CASIE, dass einem aktiven Rolleninhaber genau eine MCL zugeordnet wird, beruht auf der Annahme, dass jede MCL eine (abstrakte) Zusammenstellung aller kritischen Aspekte des gesamten Aufgabenbereiches der entsprechenden Rolle umfasst. Diese Sichtweise soll zur *runtime* eine initiale Zuordnung der Checklisten gemäß der Aufgabenbereiche ermöglichen (siehe kommenden Abschnitt 5.2.5).

Vor allem für Einsatzkräfte der oberen Führungsebene ist eine volle Expandierung einer Mastercheckliste nur dann sinnvoll, wenn die Items der unteren Ebene noch in das Aufgabengebiet der eingenommenen Rolle fallen. Je höher die Führungsinstanz (z. B. Gesamteinsatzleiter) desto mehr Aufgabenbereiche und untergeordnete Führungskräfte (z. B. Unterabschnittsleiter) sind dieser in der Regel unterstellt. Details der Arbeiten im Unterabschnitt sind für die oberen Führungsebenen meist nicht von Belang. Wesentlich ist zu wissen, ob und ggf. wann eine Aufgabe erfüllt wurde. Eine Besonderheit des CASIE-Konzeptes liegt in der Beachtung aller durch die Einsatzkräfte unterschiedlich eingenommenen Rollen. D. h., die Konkretisierung der abstrakten Items soll nur bis auf die Ebenen erfolgen, deren Items noch in den Verantwortungsbereich des Anwenders der MCL fallen. Abbildung 5.4 skizziert diese Rollengrenzen.

Ausgehend von der MCL m_k zeigt obige Abbildung 5.4 eine Checklistenhierarchie, die sich über zwei Rollenbereiche erstreckt. Der hervorgehobene Rollenübergang, der aus β berechnet ist, korrespondiert mit einem entsprechenden Wechsel des Verantwortungsbereiches gemäß einer zugrunde liegenden Organisations- und Führungsstruktur (vgl. Abschnitt 5.2.5, unten): Der Verweis vom abstraktem Item ϕ_i zur MCL m_l korrespondiert im Beispiel mit einem Wechsel des Verantwortungsbereiches vom Technischen Einsatzleiter hin zum

Unterabschnittsleiter Brandbekämpfung. Dem Item ϕ_i kommt in CASIE daher zur *runtime* eine besondere Bedeutung zu.

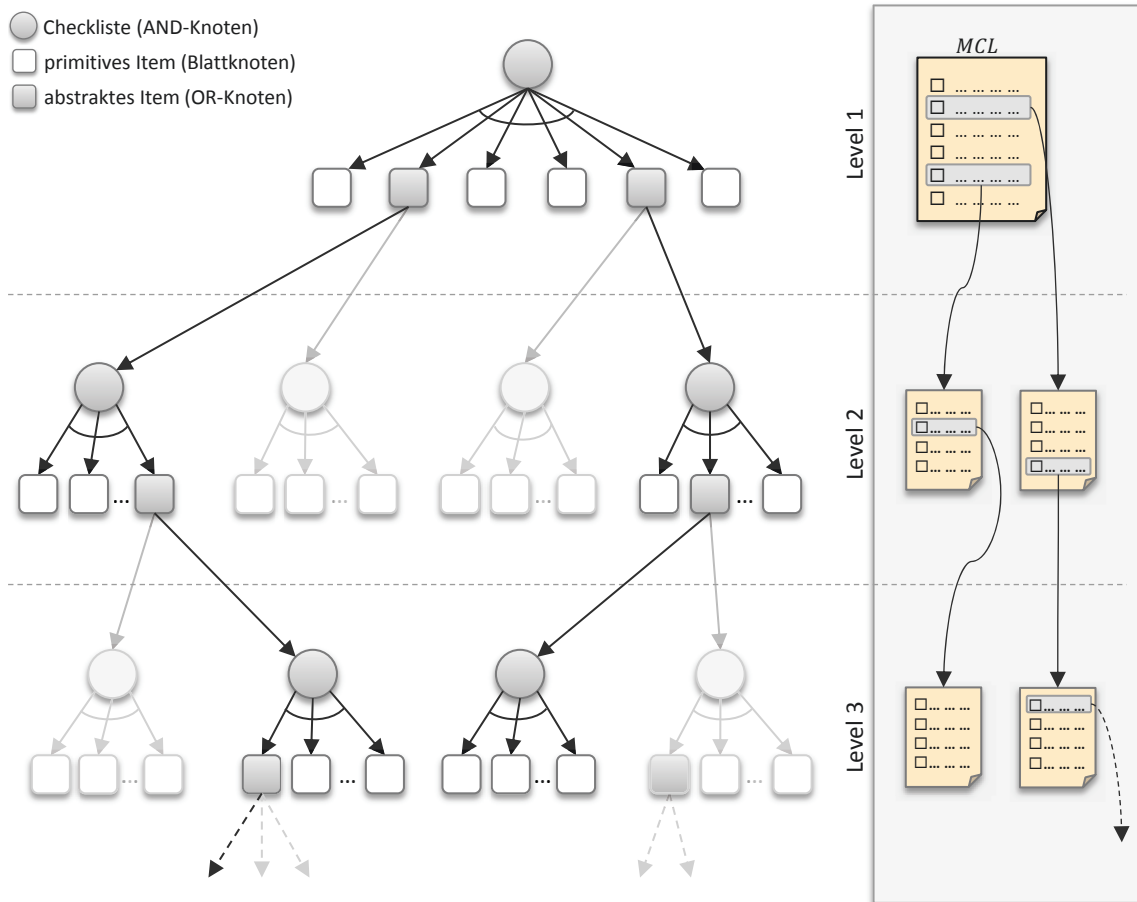


Abbildung 5.5: Entwicklung einer CLH als AND/OR-Graph.

Erweiterung um alternative Sub-Checklisten

Wie die BOS-Praxis zeigt, kann es für bestimmte SOPs durchaus sinnvoll sein, je nach den Erfordernissen der Lage alternative Vorgehensweisen für ein und dieselbe Zielstellung zu unterscheiden. So kann sich z. B. die SOP hinter der Aufgabenstellung 'Behandlungsplatz etablieren' für eine Einsatzlage *mit* und *ohne* Gefahrgutbeteiligung wesentlich unterscheiden. In Anbetracht des Prozesskontextes von Checklisten sollte daher obige Anforderung auch bei einem Checklisten-Assistenzsystem mit beachtet werden.

In Analogie zu der Struktur von AND/OR-Graphen, welche unter anderem in der KI-Forschung im Umfeld von Problemreduktion (siehe z. B. [HJW94], S. 99 ff.) zu finden sind, spannt sich in CASIE ein Suchraum bezüglich der zu beachtenden nicht-primitiven Items auf. Abbildung 5.5 zeigt links das Schema einer möglichen MCL-Expansion in Form eines AND/OR-Graphen. Eine der potentiell möglichen Konkretisierungen ist (dick) hervorgehoben. Die sich daraus ergebende CLH ist in der Abbildung auf der rechten Seite angedeutet. Jede Checkliste mit ihren Items steht für einen AND-Knoten. Jedes abstrakte Item, für das mehr als eine alternative Konkretisierung angegeben ist, stellt einen OR-Knoten dar. Die Blattknoten als Saum des Graphen sind die konkreten Items, die beachtet werden müssen.

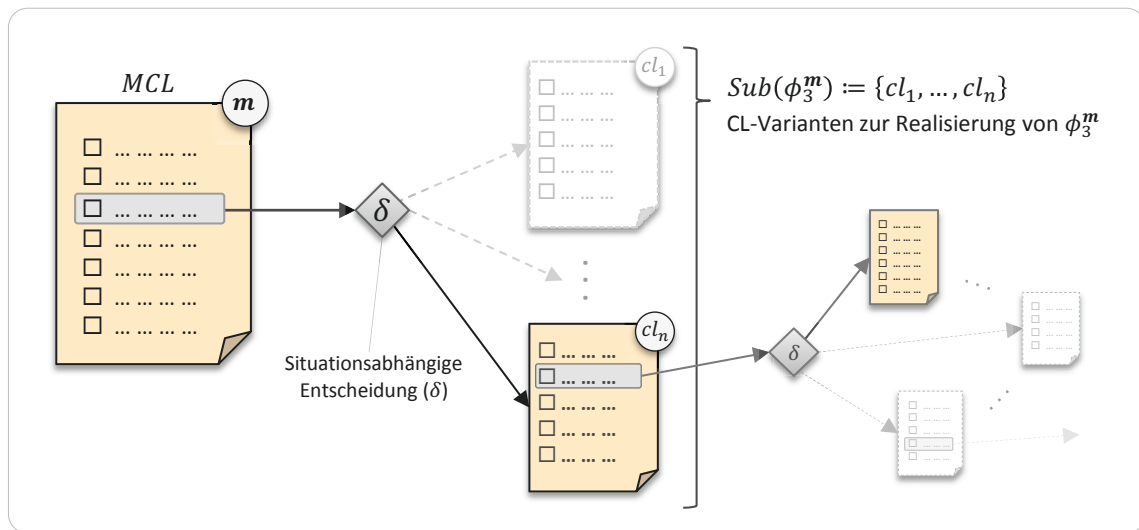


Abbildung 5.6: Grundprinzip der Angabe von alternativen Sub-Checklisten.

Eine Möglichkeit, CASIE für unterschiedliche SOP-Ausprägungen zu konfigurieren, besteht darin, für ein abstraktes Item alternative Sub-CLs als Realisierung anzugeben. Obwohl theoretisch einem abstrakten Item ϕ mittels $Sub(\phi) = \{cl_1, \dots, cl_n\}$ beliebig viele Checklisten zugeordnet werden können, ist in der BOS-Praxis zu erwarten, dass nur sehr wenige Alternativen in Frage kommen. Dieser Mechanismus kann jedoch, neben der Beachtung unterschiedlicher Einsatzlagen, auch zur Realisierung von personalisierten Checklisten genutzt werden (siehe Kapitel 7).

Die Abbildung 5.6 zeigt schematisch das Grundprinzip der alternativen Sub-CLs. Zu sehen ist eine MCL, die ein abstraktes Item enthält. Zur *runtime* muss geklärt werden, welche der möglichen Varianten aus $\{cl_1, \dots, cl_n\}$ als Konkretisierung eines abstrakten Items in Frage kommen. Dieser Entscheidungsschritt wurde in der Abbildung mittels δ kenntlich gemacht. Die Entscheidung darüber, welche Sub-Checkliste in der jeweiligen Situation angewandt werden muss, obliegt in erster Linie der Einschätzung des Anwenders. Als Assistenzsystem kann jedoch auch CASIE die Gültigkeit der Anwendungskriterien überprüfen und die Zuordnung automatisch vornehmen, falls nur eine Checkliste als Konkretisierung in Frage kommt. Hierzu wird überprüft, für welche Checkliste alle Anwendungskriterien der Menge $\mathcal{K}(cl)$ erfüllt sind – vorausgesetzt, es lassen sich dementsprechende Informationen über die Gültigkeit der jeweiligen Anwendungskriterien ermitteln (siehe Abschnitt 6.2, *Dynamik*).

Es obliegt dem Anwender, die Varianten so zu konfigurieren, dass sich die Anwendungskriterien gegenseitig ausschließen. Das heißt, es muss klar geregelt sein, welche Sub-CL aus dem Pool der Varianten im konkreten Einsatz anzuwenden ist.

Die Abbildungen 5.7 und 5.8 (siehe nächste Seite) skizzieren beispielhaft eine mögliche Checklistenhierarchie. Ausgehend von der MCL cl_1 für die Rolle des Gesamteinsatzleiters (TEL) ist ein Auszug einer CLH skizziert, die sich über 4 weitere Rollenbereiche erstreckt. Zu sehen sind Beispiele realer Checklisten (in Anlehnung an [PM01], S. 83 ff.) für die Rollen 'Gesamteinsatzleiter (TEL)', 'Unterabschnittsführer Hubschrauberlandeplatz (HLP)', 'OrgL', 'Leiter Behandlungsplatz (BP)' und 'Medizinische Leiter Behandlungsplatz'. Obwohl in den beiden Abbildungen viel Wert auf Authentizität gelegt wurde, stellen die Items lediglich Musterbeispiele ohne Anspruch auf Korrektheit der Iteminhalte dar.

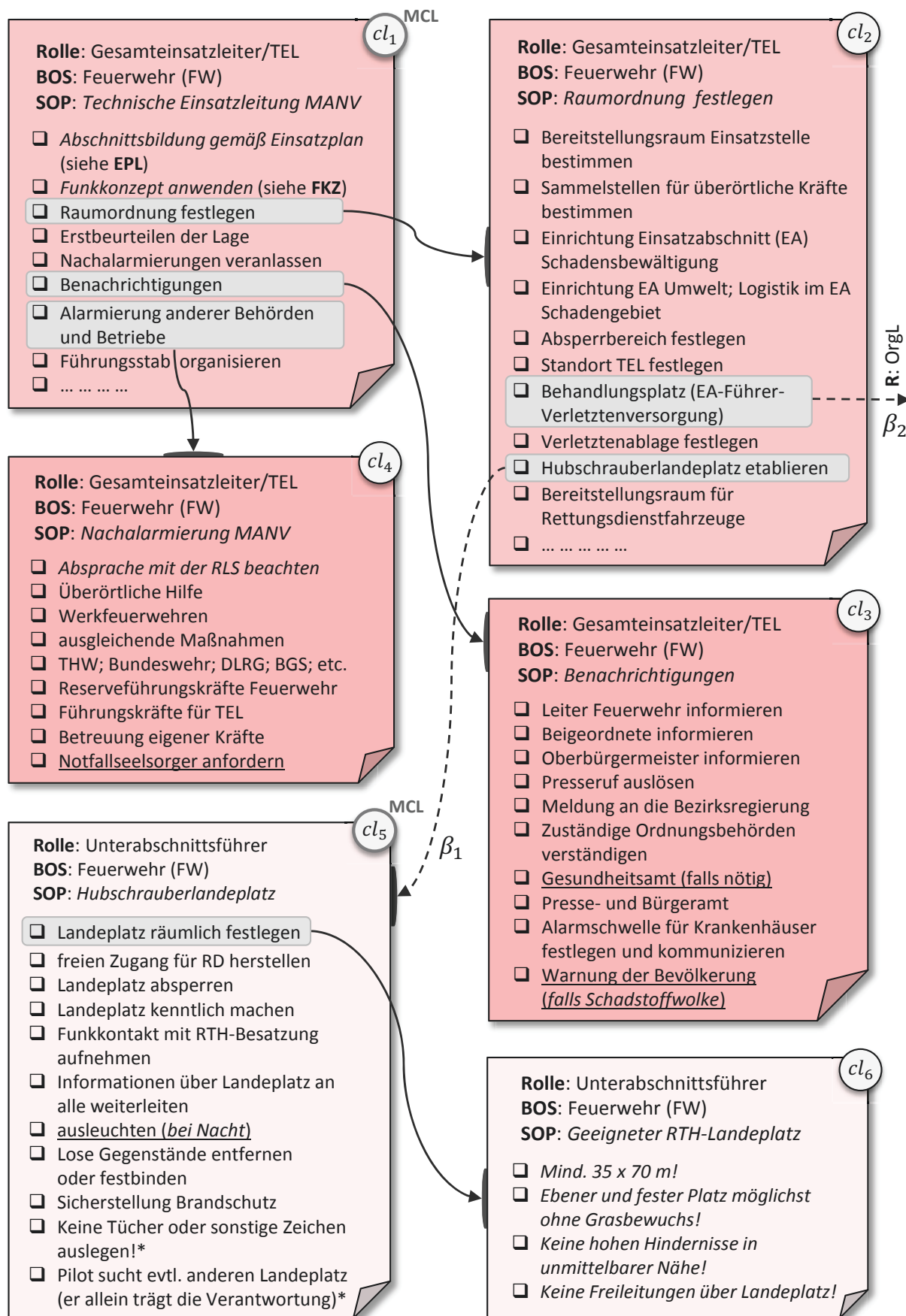


Abbildung 5.7: Ausprägung einer Checklistenhierarchie (1/2).

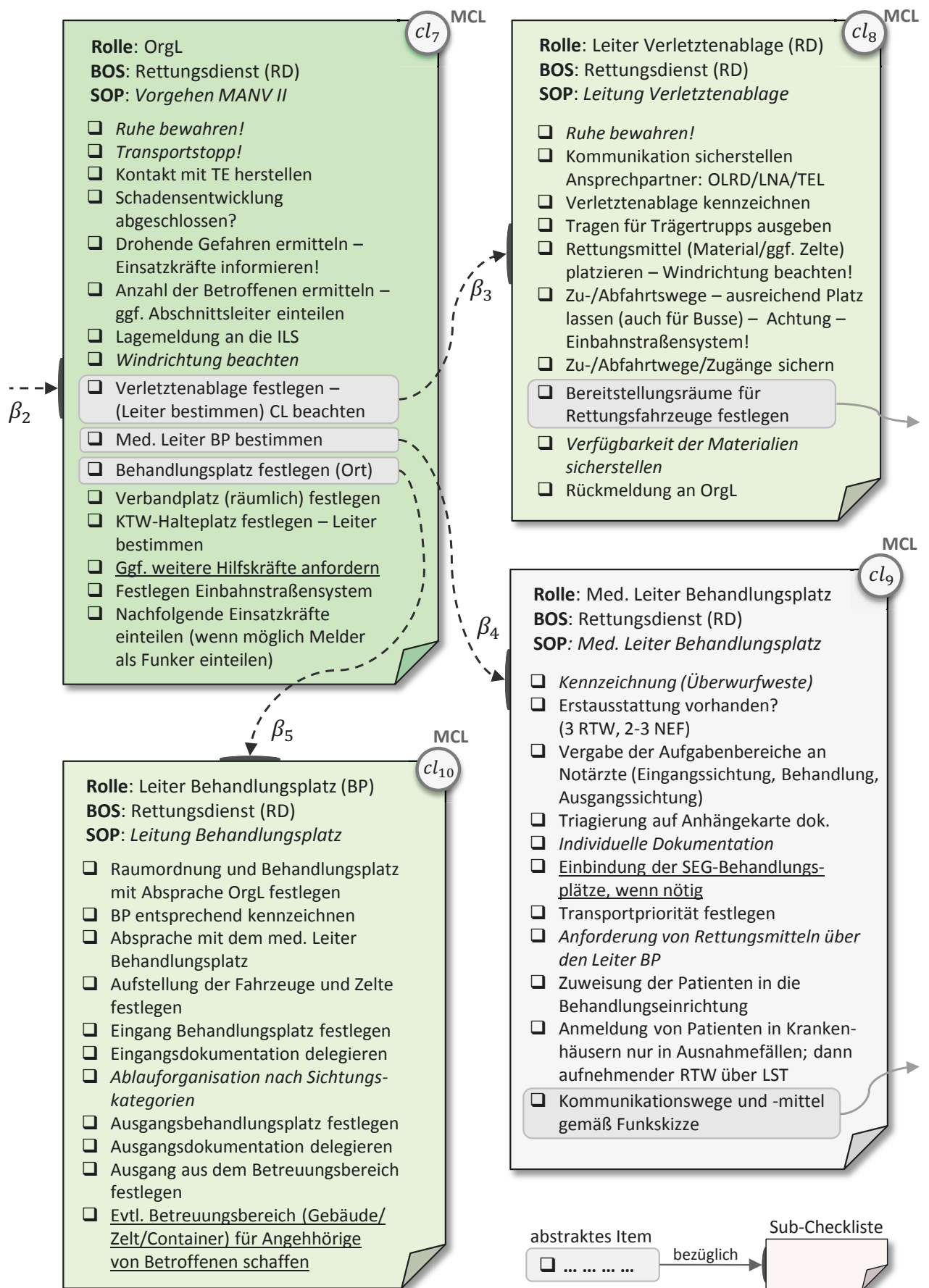


Abbildung 5.8: Ausprägung einer Checklistenhierarchie (2/2).

Weiterhin ist in den Abbildungen 5.7 und 5.8 im Kopf einer Checkliste die jeweilige Rolle der Einsatzkraft, ihre Organisationszugehörigkeit und der Prozesskontext $P_K(cl)$ benannt. Weitere Merkmale wie mögliche Anwendungskriterien $\mathcal{K}(cl)$ oder der Zustand $stat(cl)$ sind in der Skizze einfachheitshalber nicht abgetragen. Auch eine eventuell vorliegende partielle Ordnung der Items ist in der gewählten Serialisierung nicht mehr als solche zu erkennen.

Zur besseren Unterscheidung sind im Beispiel die *note*-Items kursiv dargestellt und die *query*-Items durch ein Fragezeichen am Ende des Textes gekennzeichnet. Items, die nur in bestimmten Situationen beachtet werden müssen (d. h., die Menge $Pre()$ eines Items enthält mindestens eine Vorbedingung), werden *bedingte* Items genannt und sind in der Abbildung mittels Unterstreichung als solche kenntlich gemacht. Bei einem bedingten Item handelt es sich nicht um einen zusätzlich eingeführten Itemtyp, sondern um eine (optionale) Qualität der bereits definierten *note*-, *action*- und *query*-Items. Vorbedingungen lassen sich in natürlichsprachlicher Form angeben. Der Hinweis, dass bei einer Schadstoffwolke („falls Schadstoffwolke“, siehe Abbildung 5.7, cl_3) die Bevölkerung zu warnen ist, ist ein Beispiel eines bedingten Items, das nur beachtet werden muss, falls die Vorbedingung erfüllt ist. Bedingte Items ermöglichen die Konfiguration einer Checkliste für ein möglichst breites Spektrum an Anwendungsszenarien. Auch für diese spezielle Qualität von Items kann CASIE automatisch die Gültigkeit der Bedingungen prüfen. Voraussetzung ist auch hier, dass die Vorbedingungen in einer computerverarbeitbaren Form gegeben sind und der momentane systeminterne Wissensstand über die Einsatzlage eine Schlussfolgerung ermöglicht. Ist z. B. in der Wissensbasis bereits bekannt, dass es sich bei dem Einsatz um einen Großbrand handelt, der zudem in einem Düngemittelbetrieb ausgebrochen ist, so kann CASIE automatisch den Schluss ziehen, dass eine Schadstoffwolke droht. Hierfür muss das System zuvor mit einer entsprechenden Regel (Axiom) konfiguriert werden. In dem kommenden Abschnitt 6.2 und dem Kapitel 7 wird genauer auf die Thematik der bedingten Items eingegangen.

Für einzelne abstrakte Items (grau hinterlegt) sind Beispiele möglicher Konkretisierungen zu sehen. So sind z. B. für die drei abstrakten Items von cl_1 deren jeweilige Konkretisierung (cl_2, \dots, cl_4) als Sub-Checklisten zu sehen. Weiterhin sind Beispiele für Rollenübergänge β_1, \dots, β_5 gegeben. Die Konkretisierung abstrakter Items, welche im Verantwortungsbereich einer anderen Rolle liegen, ist durch gestrichelte Pfeile gekennzeichnet. Die Checkliste cl_2 des 'Technischen Einsatzleiters' beinhaltet z. B. zwei abstrakte Items, deren jeweilige Konkretisierung auf die Sub-Checkliste cl_5 bzw. cl_7 verweist. cl_7 des 'OrgL' verweist wiederum auf drei Sub-Checklisten (cl_8, \dots, cl_{10}), welche jeweils im Verantwortungsbereich von Abschnittsleitern liegen, die dem 'OrgL' unterstellt sind. (Dass der 'Medizinische Leiter Behandlungsplatz' in der Praxis immer dem leitenden Notarzt unterstellt ist, wurde in der Abbildung 5.8 zu Gunsten der Übersichtlichkeit ausgeblendet.)

5.2.4 HTN-Planung und Theorien der Handlungsregulation

Die Berechnung einer Checklistenhierarchie orientiert sich an der sogenannten Reduktion eines hierarchischen Tasknetzwerks in der HTN-Planung (vgl. [GNT04]), deren Grundlagen 1975 von SACERDOTIS als Lösungsansatz komplexer Probleme beschrieben wurden (vgl. [Sac75]). Während SACERDOTIS in seinen Arbeiten statt von HTN noch von „prozeduralen Netzen“ sprach und eher die hierarchische Organisation und zielorientierte Wiederverwendung von Programmprozeduren adressierte, beschäftigt sich die heutige HTN-Planung allgemein mit der Perspektive weltverändernder Handlungen in einer Planungsumgebung. Ein wesentliches Konzept in der HTN-Planung sind die sog. Methoden. HTN-Methoden sind stark problembezogen und können als eine Art *Standard Operating Procedure* aufgefasst werden [GNT04]. Domänenexperten formulieren in ihnen ihre Standardvorgehensweise bei

der Bearbeitung von mehr oder weniger abstrakten wiederkehrenden Aufgaben. Wie in den HTN-Methoden, so wird auch in den Checklisten jeweils eine „best practice“ als Lösung für eine abstrakte Aufgabe beschrieben, wenn auch bei letzteren ausschließlich kritische Aspekte einer übergeordneten SOP adressiert werden. Im Prinzip lässt sich der Aufbau einer CLH mit der Aufgabe einer Task-Dekomposition bei der HTN-Planung vergleichen.

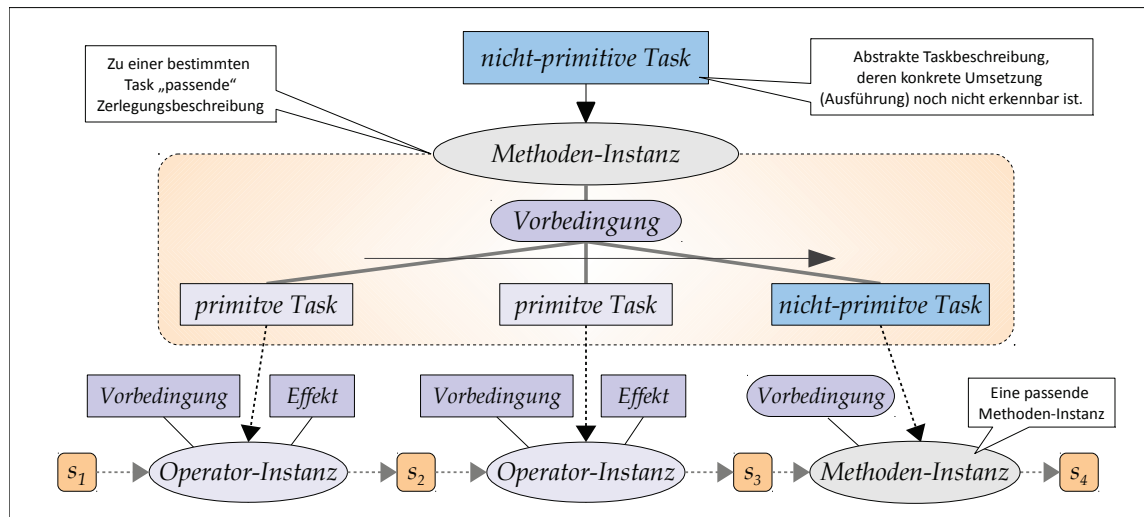


Abbildung 5.9: Das Prinzip der Task-Dekomposition.

Die MCL in CASIE ähneln dem Konzept eines Start-Task-Netzwerk w eines HTN-Planungsproblems $P = (s_0, w, O, M)$ (vgl. hierzu [GNT04], S. 236 ff.). Wie eine komplexe Task realisiert werden kann, wird durch eine oder mehrere zuvor konfigurierte Methoden vorgegeben. Abbildung 5.9 zeigt, wie das Prinzip der Task-Dekomposition (auch Taskreduktion genannt) bei der Plansuche angewendet wird. Zu Beginn steht eine vorgegebene nicht-primitive Task, die erfüllt werden soll. Für solch eine Task kann es mehrere vorgegebene Methoden als Realisierung geben. Kann z. B. zwischen zwei Methoden für ein und dieselbe Task gewählt werden, so entscheiden die in der jeweiligen Methode angebbaren Vorbedingungen über die Wahl der anzuwendenden Methode. Nur wenn die Vorbedingungen der jeweiligen Methode im momentanen Zustand gelten, versucht der Planer, die Methode zur weiteren Konkretisierung der Task anzuwenden. Beginnend vom Startzustand s_1 wird durch Hintereinanderausführung der Aktionen (Operator-Instanzen) der Zielzustand s_4 erreicht. Im Zustand s_4 wurde der potentielle Weltzustand durch Anwendung der Aktionen dahingehend geändert, dass nun alle Merkmale, die eine erfolgreiche Taskerfüllung charakterisieren, gelten. In Abbildung 5.10 ist eine vereinfachte Darstellung eines Teils einer möglichen Planungsdomäne als AND/OR-Graph dargestellt.

Die Itemstruktur \mathcal{I} einer MCL lässt sich mit der eines Task-Netzwerks w vergleichen. Bei der HTN-Planung besteht das Problem in einer Reduktion von w im Startzustand s_0 , unter Beachtung der Domänenmodellierung (bestehend aus den Methoden M und den Operatoren O). Hingegen stellt sich in CASIE die Aufgabe einer Reduktion der Itemstruktur \mathcal{I} der MCL. Für CASIE gelten daher im Vergleich zur HTN-Planung folgende Besonderheiten/Unterschiede:

1. In einer Großschadenslage lassen sich aufgrund des zu Beginn herrschenden Informationsmangels kaum konkrete Planungsprobleme formulieren. Wenn überhaupt, dann sind die Ziele nur sehr abstrakt gegeben („Leben retten“, „Schaden abwenden“, „Behandlungsplatz aufbauen“). Bei CASIE handelt es sich aus Planungssicht nicht um

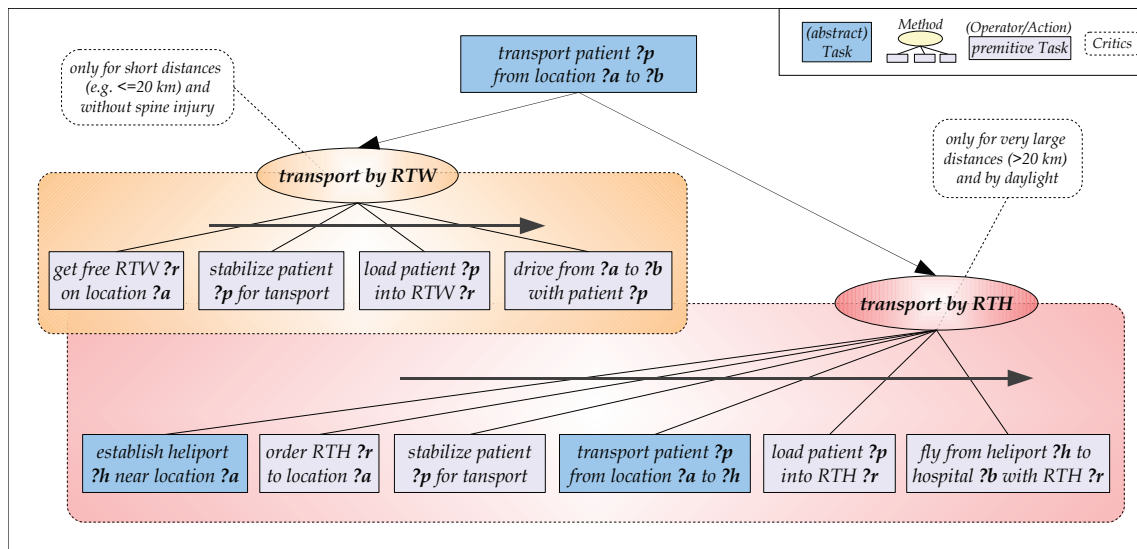


Abbildung 5.10: Beispielhafter Auszug einer HTN-Planungs-Domäne als AND/OR-Graph.

eine Vorplanung, sondern vielmehr um eine Art *on-demand* CLs-Konkretisierung mit Prozessmonitoring.

2. Die Items in CASIE entsprechen im Gegensatz zu den HTN-Operatoren nicht zwangsläufig konkreten Aktionen einer SOP. D. h., die meisten Items entsprechen abstrakten Tasks. Es ist zu erwarten, dass nur für die wenigsten Items eine korrekte Angabe der Vor- und Nachbedingungen möglich (sinnvoll) ist. Auch steht die Instantiierung von Operatoren zu Aktionen (durch Ressourcenzuweisung) in CASIE nicht im Fokus. CASIE arbeitet vielmehr auf einer abstrakteren Ebene, da umfassende Informationen über alle Ressourcen in der Regel nicht vorliegen.
3. In CASIE müssen aufgrund des Multiagentenszenarios zusätzlich zu dem Aufbau der Checklistenhierarchie(n) die jeweiligen eingenommene Rollen und die möglichen Rollenübergänge mit beachtet werden. Des Weiteren gilt es zusätzlich zum Paradigma der Taskreduktion in der HTN-Planung, den Level der jeweiligen Sub-CLs im Ergebnis einer CLH mit zu berücksichtigen.

Obige Diskussion soll aufzeigen, dass die Anwendung des HTN-Planungsparadigmas im Prinzip für die in CASIE benötigte Konkretisierung von MCLs geeignet ist. Jedoch liegt der Fokus bei CASIE eher auf der Verzahnung zwischen Taskreduktion und Ausführung zur Einsatzzeit, und weniger auf der Berechnung eines in allen Einzelheiten konkretisierten Planes. Die KI bietet jedoch mit der HTN-Planungstheorie ein Verarbeitungsmodell menschlicher Handlungsplanung, das als Rahmenwerk für den Umgang mit abstrakten Aufgabenbeschreibungen geeignet ist.

Abstraktionen und Hierarchien finden sich ebenfalls in den sozialpsychologischen Handlungstheorien (auch Handlungsregulationstheorien genannt) wieder. Dies untermauert die Angemessenheit der hierarchischen Checklistenstruktur und zeigt, dass der CASIE-Ansatz der menschlichen Herangehensweise an komplexe Problemstellungen entspricht. Im Folgenden werden zwei populäre Handlungstheorien aufgeführt, die die Analogie zum Vorgehen der CLH bzw. HTN-Taskreduktion deutlich machen.

Die Handlungstheorien leisten einen Beitrag zur Erklärung und Vorhersage sozialer Interaktionen und sozialen Verhaltens [FG97]. Sie versuchen zu klären, wie elementare

Handlungen im Gesamtkontext von abstrakten übergeordneten Aufgaben zu verstehen sind [FG97]. So gibt es unter anderem relevante kognitions-, motivations-, persönlichkeits-, entwicklungs- und arbeitspsychologische Ansätze [Bos07]. In allen handlungspsychologischen Ansätzen kommt die Interaktion des tätigen Menschen mit anderen Menschen und seiner Umwelt, d. h. die Wechselwirkung mit seiner sozialen und physikalischen Umwelt zum Ausdruck [Amm05]. Die Handlungspsychologie geht davon aus, dass sich der Mensch aktiv, zielgerichtet, denkend und planend mit seiner Umwelt auseinandersetzt ([Wie94] S. 73f.). Durch Pläne werden Abfolgen von menschlichen Handlungen festgelegt, die auf ein bestimmtes Ziel hin ausgerichtet sind. Dieser Sichtweise entspricht prinzipiell die KI-Planung, jedoch wird in der sozialpsychologischen Literatur der Begriff „Planung“ teils sehr allgemein für jegliche zielgerichteten Bemühungen verwendet.

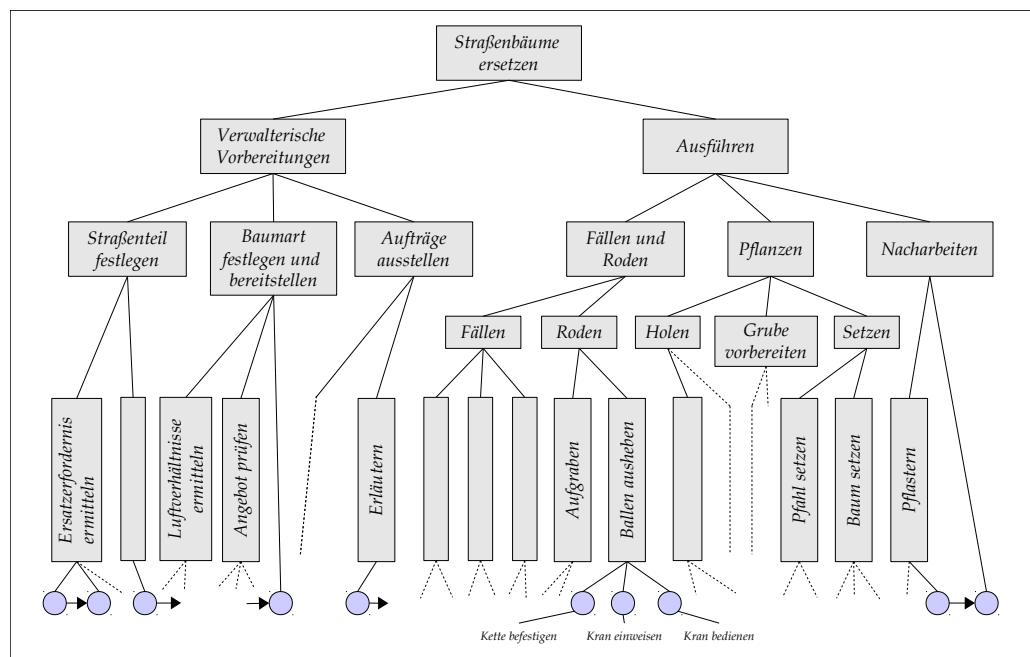


Abbildung 5.11: Beispiel einer hierarchischen Aufgabendekodierung nach HACKER [Hac97].

Wie HACKER in [Hac97] schreibt, besteht „das Kernproblem bei der psychologischen Analyse des Handelns darin“, ein „theoretisches Vakuum zwischen Erkenntnis und Handeln zu füllen.“ Gemeint ist hier die Frage, „wie Handlungen durch die interne Repräsentation der Umwelt im Organismus reguliert werden“. Die Handlungspsychologie versucht, menschliches Handeln zu erklären, indem sie ihre Aufmerksamkeit auf die handlungssteuernden Pläne und die kognitiven Strukturen beim Planen richtet.

Aus dem Bereich der Arbeitspsychologie stammt ein Modell der Regulation von Arbeitstätigkeiten, das HACKER unter anderem in seinem Buch „Allgemeine Arbeitspsychologie. Psychische Regulation von Arbeitstätigkeiten“ [Hac97] vorstellt. HACKER untersucht die Bildung von regulativen Funktionseinheiten und erarbeitet in diesem Zusammenhang ein hierarchisches Modell der Regulation von Arbeitstätigkeiten. Komplexere Aufgaben werden dabei in Unteraufgaben unterteilt. Die in Abbildung 5.11 gezeigte Aufgabendekodierung entspricht der einer Taskreduktion in der HTN-Planung. Die Kreise auf der unteren Ebene stellen die auszuführenden elementaren Handlungen (Aktionen in der Planung) dar. Die angedeutete Sequenz dieser Handlungen spiegelt hier einen (total-geordneten) Plan wider. Wie HACKER schreibt, entspricht „der Hierarchie der Zielsetzung“ eine „Hierarchie der Pläne“ [Hac97]. Dies entspricht der Vorstellung der HTN-Planung von HTN-Methoden

als eine Art „Teilpläne“ zur Lösung von untergeordneten Zielen. Weiterhin empfiehlt er als allgemeineren Begriff für „Plan“ den Begriff „Aktionsprogramm“ zu nutzen, wodurch erneut die Nähe zum Planbegriff der KI-Planung deutlich wird.

Die Aufgabekodierung trifft nicht nur auf individuelle Arbeitstätigkeiten zu, sondern gilt auch für Tätigkeiten von Gruppen und umfassenderen organisatorischen Einheiten. Dieser Ansatz lässt sich gut auf die Zusammenarbeit der unterschiedlichen BOS in einem MANV anwenden. HACKER fasst die in Abbildung 5.11 angedeuteten Teilaufgaben hierzu zu Blöcken bezüglich ihrer organisationalen (betrieblich-abteilungsbezogenen) Arbeitsteilung bzw. Arbeitskooperation zusammen. Besonders interessant sind hierbei die Schnittstellen und der notwendige Informationsaustausch zwischen den Blöcken, die eine reibungslose Zusammenarbeit zwischen den Organisationen sicherstellen. Bezogen auf die HTN-Planung korrespondiert diese Gruppierung mit dem Konzept der Methodendefinition.

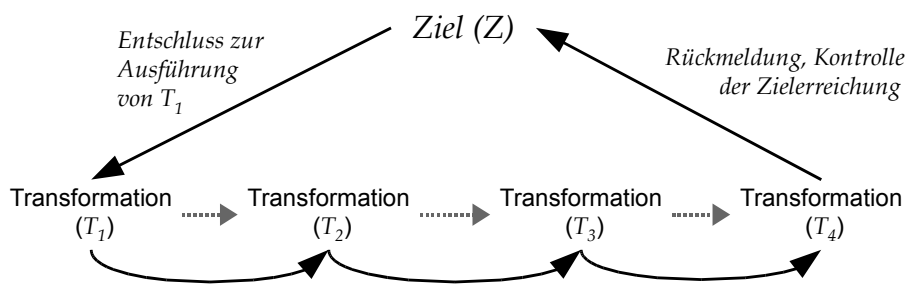


Abbildung 5.12: Die Zyklische Einheit nach VOLPERT 1982 [FG97].

Eine weitere Handlungstheorie stellt das Modell der *hierarchisch-sequentiellen Handlungsorganisation* von VOLPERT [Vol83] dar. In ihr besteht die Ausgangssituation in einem hierarchisch übergeordneten komplexen Ziel, das absteigend über mehrere Hierarchieebenen in einfachere Teilziele aufgeteilt wird, bis eine Sequenz elementarer umweltveränderlicher Operationen realisiert ist. Ein Basiskonzept dieser Handlungstheorie ist die *Zyklische Einheit* (siehe Abbildung 5.12). Das Erfüllen eines Ziels wird über sequenzielle Transformationen derart realisiert, dass nach Beendigung der letzten Transformation das gewünschte Ziel erreicht ist. Im Idealfall hat der erfolgreiche Abschluss aller einzelnen Transformationen der Handlungssequenz das Erreichen des gewünschten Ziels zur Folge.

Die Möglichkeit einer Rückmeldung im Fehlerfall spiegelt sich in dem vorgesehenen Zyklus dieser Einheit wider. Begründet auf äußere Einflüsse, die das erfolgreiche Anwenden der Handlungsstrategie vereiteln können, ist hier eine Erfolgskontrolle nach der letzten Transformation vorgesehen. Somit kann überprüft werden, ob die Handlungssequenz das gewünschte Ziel erreicht. Falls das Ziel nicht erreicht wurde, sieht das Modell eine erneute Anwendung der Sequenz vor. Dieser Zyklus wird solange durchlaufen, bis das Ziel erreicht ist. Ist die Handlungssequenz jedoch fehlerhaft, d. h. prinzipiell nicht zielführend, dann nützt eine ständige Wiederholung hier nichts. Vielmehr sollte in den meisten Situationen, bei denen eine zuvor gewählte Handlungssequenz fehlschlug, eine Neueinschätzung aller Umstände zu einer abgeänderten Handlungssequenz führen. Der Fehlschlag kann prinzipiell zwei Ursachen haben: (1) die Handlungssequenz ist prinzipiell nicht durchführbar und (2) die Handlungssequenz ist in der momentanen Umgebung (mit ihren aktuellen Merkmalsausprägungen) nicht durchführbar. Im 2. Fall wäre eine erneute Wiederholung durchaus sinnvoll, während im 1. Fall eine neue/abgeänderte Handlungssequenz gefunden werden muss (Re-Planung in der KI-Planung, vgl. z. B. [NK95]).

Die Transformationen können wiederum selbst eine zyklische Einheit darstellen (siehe Abb. 5.13). Jede Transformation kann als Teilziel der übergeordneten Transformation bzw.

des übergeordneten Ziels angesehen werden. Zum einen kann die Aufspaltung des Hauptziels in Teilziele als Abstraktionsreduktion gesehen werden, bis hin zu konkret ausführbaren Handlungen (*Top-down-Sichtweise*). Auf der anderen Seite ist jedes Teilziel ein Bestandteil eines komplexeren Oberziels, dessen Beschreibung auf dem Weg von „unten nach oben“ abstrahiert wird (*Bottom-up-Sichtweise*). Dies ähnelt einer Baumstruktur, die über eine bestimmte Anzahl von Ebenen hinweg die Handlungsorganisation darstellt. Das abstrakte „Hauptziel“ stellt die Wurzel des Baumes da.

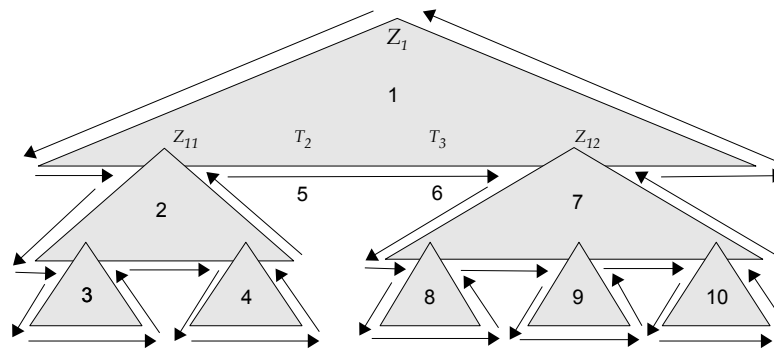


Abbildung 5.13: Das hierarchisch-sequenzielle Modell der Handlungsregulation [FG97].

Das in Abbildung 5.13 gezeigte Modell entspricht prinzipiell der Arbeitsweise der *Ordered-Task-Decomposition* des SHOP2-Planers (vgl. [NMAC+01]). Eine unabhängige Teilzieldekomposition ist jedoch in der Praxis nicht immer möglich. Bei der separaten Erfüllung von Teilzielen besteht die Gefahr, dass die Lösung eines Teilzieles eine bereits zuvor erreichte Lösung eines anderen Teilzieles (zum Teil) wieder zerstören kann. Dieses Problem ist in der KI-Planung unter der sog. *Sussman-Anomalie* [Sus75] bekannt, die beispielhaft die Problematik einer unbedachten Zielzerlegung aufzeigt. In der HTN-Planung muss diese unabhängige Teilzieldekomposition durch den Methodenmodellierer sichergestellt werden.

5.2.5 Checklisten-Auswahlprozess und -Zuordnung

Analog zu klassischen PCLs obliegt es in erster Linie der Einsatzkraft, zu entscheiden, welche Checkliste in welcher Situation zur Hand genommen werden muss. Neben der Ablage aller Checklisten in einem entsprechenden Repository (siehe Abschnitt 6.1), aus dem dann die Einsatzkraft zur Laufzeit die passende Checkliste manuell auswählt, stellt sich in CASIE besonders die Frage danach, inwiefern dieser Auswahlprozess vom Computer unterstützt werden kann. Ziel ist es, die zum Einsatz und zur Rolle passende(n) ICL(s) anzubieten bzw. die Menge der zur Auswahl stehenden ICLs möglichst gering und damit überschaubar zu halten. Zwei Problemstellungen sind hierbei zu unterscheiden:

1. *Zu Beginn eines Einsatzes:* Wie lässt sich zu Beginn (bei beschränkter Faktenlage) eine Auswahl passender Checklisten realisieren? Mit anderen Worten, nach welchen Kriterien und durch welche Automatismen lässt sich erreichen, dass das System möglichst nur für die Einsatzlage relevante Checklisten anbietet?
2. *Während eines Einsatzes:* Wie ist es möglich, auf sich ändernde Einsatzmerkmale mit entsprechend angepassten Checklisten zu reagieren bzw. die Notwendigkeit der Bearbeitung neuer Checklisten automatisch abzuleiten?

Die zweite Problemstellung adressiert das in CASIE angestrebte dynamische und intelligente Gesamtsystemverhalten, auf das in den kommenden Abschnitten (siehe Abschnitt 6.2)

genauer eingegangen wird. An dieser Stelle soll vorerst die erste Fragestellung, nämlich die der Zuordnung von Checklisten zu entsprechenden Aufgaben einer Einsatzsituation, behandelt werden. Zur Beantwortung der ersten Frage wurden im Rahmen dieser Arbeit folgende drei Herangehensweisen in Betracht gezogen:

1. *Einsatzstichworte*: Zuordnung der MCLs über eine Hierarchie von sog. Einsatzstichworten, welche jedem Einsatztyp ein entsprechendes Schlüsselwort zuordnet.
2. *Aufgaben/Tasks*: Zuordnung zu entsprechenden Aufgaben (Tasks), deren Bearbeitung durch die Situation oder durch eine Handlungsaufforderung eines Vorgesetzten notwendig werden.
3. *Rollen und Organisationsstruktur*: Zuordnung mit Hilfe der anzuwendenden Organisationsstruktur des jeweiligen Einsatzes, durch die die Einsatzrollen bestimmt werden.

Einsatzstichworte. Auf den ersten Blick bietet sich eine MCL-Zuordnung anhand einer zuvor erstellten Einsatzklassifikation (z. B. Gebäudebrand, Gefahrgutunfall, MANV) an. Hierbei besteht der Ansatz darin, eine Hierarchie von sog. Einsatzstichworten (vgl. Abschnitt 2) zu nutzen – so wie sie in Rettungsleitstellen bekannt sind, um hinter jedem Einsatztyp die zu beachtenden Checklisten zu hinterlegen. Diese Möglichkeit der Klassifizierung ist jedoch nur in Rettungsleitstellen geläufig, wo sie zur Auswahl der zu alarmierenden Kräfte dient. Die Verknüpfung zu entsprechenden zugehörigen SOPs ist in der Regel nicht damit verbunden. Zudem sind die Listen der Einsatzstichworte von Region zu Region stark individuell ausgeprägt bzw. deren Verwendung nicht in jedem Einsatzgebiet verbreitet. Obgleich eine erste Einschränkung der in Frage kommenden CLs durch Einsatzstichworte erreicht werden kann, bleibt der Rollenaspekt noch unbeachtet. CLs anhand von Einsatzstichworten zu organisieren ist daher eher ungeeignet.

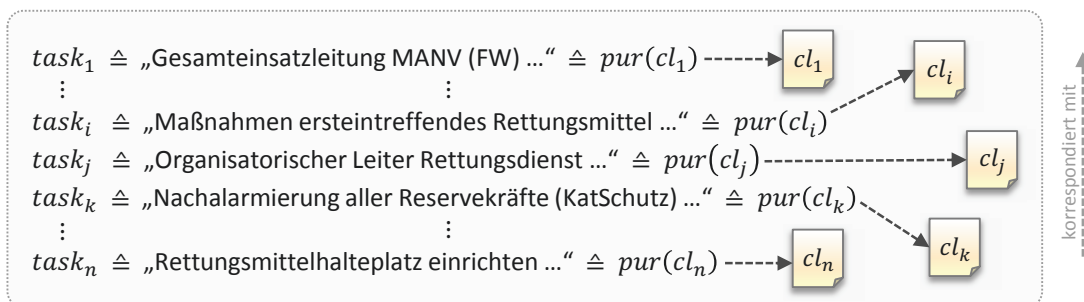


Abbildung 5.14: Zuordnung der Aufgaben/Tasks über die Beschreibungen zu den Checklisten.

Aufgaben/Tasks. Ausgehend von den jeweiligen Aufgabengebieten lassen sich gemäß der SOPs alle Checklisten anhand ihrer adressierten Aufgabenstellungen (gemäß SOP Beschreibung) organisieren. Abbildung 5.14 skizziert eine mögliche 1:1-Zuordnung der Tasks (1...n) aus einer Liste zu den entsprechenden Checklisten. Die Beschreibung $pur(cl)$ einer Checkliste cl beschreibt zugleich die dahinterstehende Task. Sind für alle Tasks auch entsprechende CLs hinterlegt, so könnte die Einsatzkraft, je nach Aufgaben, Zugriff auf eine passende Checkliste anhand einer Task aus der Tasklist erhalten. Eine automatische Vorauswahl ist jedoch ohne konkretes Wissen über die Einsatzlage schwer möglich bzw. hängt von der Entscheidung ab, welche Organisationsstruktur etabliert wird. Als taskorientierte Sicht auf die Checklisten ist diese Variante jedoch geeignet.

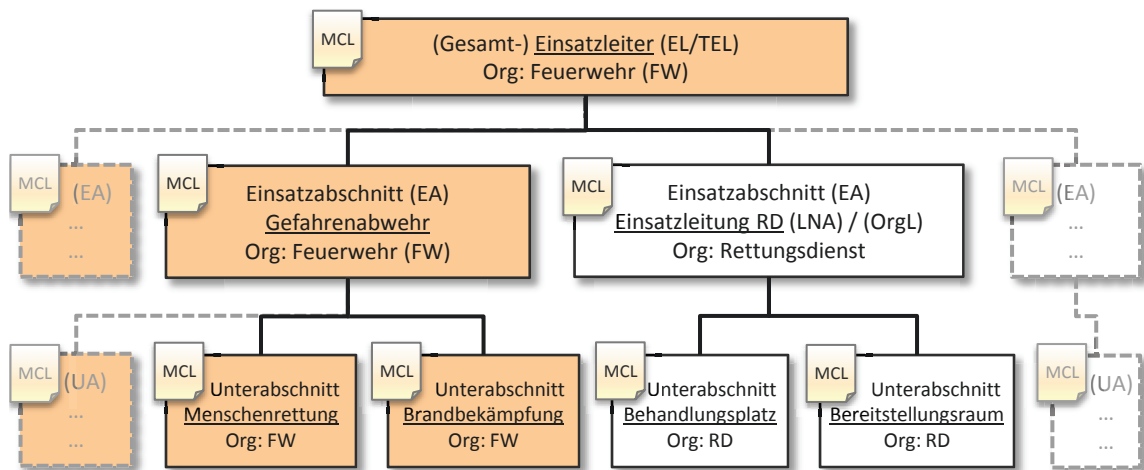


Abbildung 5.15: Auszug eines MANV-Organisationsaufbaus.

Rollen und Organisationsstruktur. In Abschnitt 5.2.1 wurde bereits die enge Verbindung von SOPs zur Rollenverteilung angesprochen. Da die Rolle einer Einsatzkraft auch deren Aufgabengebiet bestimmt (zumindest stark eingrenzt), wird in CASIE die Organisation der ICLs anhand der definierten Rollen favorisiert. Wählt eine Einsatzkraft ihre zutreffende Rolle aus, so kann sie die prinzipiell für sie in Frage kommende ICL selektieren oder die Menge aller relevanten ICLs auf eine überschaubare Menge einschränken. Bei der Konfiguration des Systems empfiehlt es sich, für jede definierte Rolle eine entsprechende MCL als Startpunkt (Einstiegspunkt) festzulegen.

Die Abbildung 5.15 zeigt hierzu ein Beispiel einer Rollenverteilung anhand der Gliederung der Einsatzabschnitte und der Führungshierarchie. Zu sehen ist ein Auszug einer möglichen etablierten Führungsorganisation mit den unterschiedlichen Aufgabenbereichen wie z. B. Gesamteinsatzleitung, Einsatzabschnitte (EA) und Unterabschnitte (UA). Abbildung 5.15 kann jedoch nur den Teil der Rollenstruktur visualisieren, der dem Verantwortungsbereich der Führungskräfte entspricht. Verschiedene Unterabschnitte oder mehrfach eingenommene Rollen (wie z. B. die eines Notarztes (NA) oder Truppführers (TF)) müssen zusätzlich berücksichtigt werden.

Es ist zu beachten, dass eine Rolle in der Regel zwar eine spezielle Qualifikation der Einsatzkraft voraussetzt, die Qualifikation jedoch nicht zwingend mit der Rolle gleichgesetzt werden darf. Zum Beispiel kann ein Arzt mit der Qualifikation zum LNA durchaus einem anderen, die Rolle des LNA innehabenden Notarzt untergeordnet sein. Beispiel hierfür ist die Rolle 'Eingangssichtung Behandlungsplatz', die unter der Leitung eines 'Leitenden Notarztes' (LNA) arbeitet. Die Frage, wann welche ICL beachtet werden muss und wie deren möglichen abstrakten Items konkretisiert werden, ist über die Rolle hinaus abhängig von der jeweiligen vorgefundenen und sich ergebenden Einsatzsituation.

5.2.6 Standardisiertes Vokabular

Für die Problemstellung dieser Arbeit stellt sich die Frage, wie eine Computerassistenz bei der Definition und Eingrenzung eines Grundwortschatzes für die Erstellungszeit (*offline*) und für die Nutzung eines Assistenzsystems im Einsatz (*online*) genutzt werden kann. Hierfür wird für CASIE als eines der zentralen Elemente der Einsatz einer formalen Wissensrepräsentation vorgeschlagen (vgl. Abschnitt 6.1). Drei Verwendungsmöglichkeiten einer formalen Wissensrepräsentation kommen für CASIE zum Einsatz:

1. *Erstellung (offline)* – Während der Erstellung der Checklisten-texte kann die Terminologie der Anwendungsdomäne (in dieser Arbeit beispielhaft an der BOS-Domäne gezeigt, vgl. Abschnitt 6.3.1) bei der Wahl der zu verwendenden Fachbegriffe zu Rate gezogen werden. Dies befördert eine Vereinheitlichung im Sprachgebrauch, wodurch das Risiko eventueller Fehlinterpretationen im Einsatz verringert wird. Zum Beispiel sollte immer dann, wenn von Orten die Rede ist, an denen Einsatzkräfte und Einsatzmittel für den Einsatz vorgehalten werden, der Name „Bereitstellungsraum“ verwendet werden. Abbildung 5.16 zeigt eine entsprechende Konzeptdefinition aus der BOS-Ontologie. (Zu sehen ist ein Ausschnitt der BOS-Ontologie ([BOS-O]), ediert mit dem Ontologieeditor Protégé, vgl. hierzu Abschnitt 6.3.)

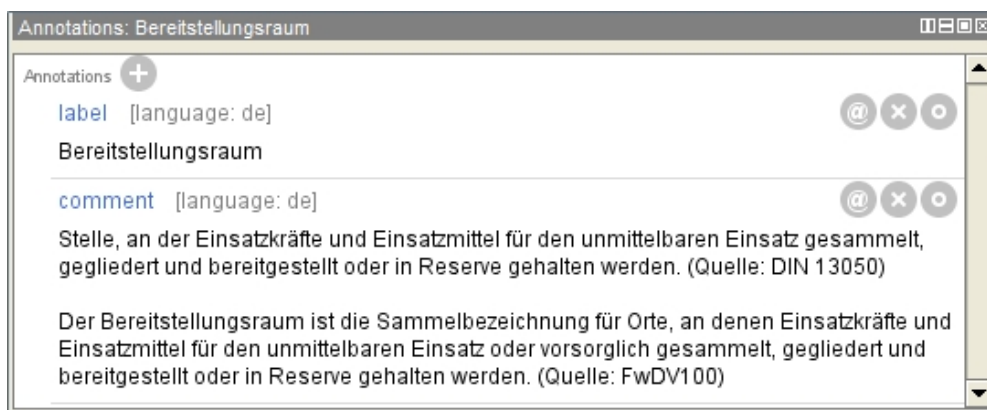


Abbildung 5.16: Annotation des Konzeptes „Bereitstellungsraum“ (siehe [BOS-O]).

2. *Informationsquelle (online)* – Da für die Itemtexte in der Benutzerschnittstelle (UI) in der Regel wenig Platz zur Verfügung steht, müssen die Texte eher stichpunktartig gehalten werden. Darunter kann das Verständnis der Aussagen leiden. Eine formale Wissensbasis kann nun auf Begriffsebene eine Quelle für Zusatzinformationen sein, die ein Nutzer bei Bedarf im UI bezüglich einer Klärung eines Begriffes (ähnlich eines interaktiven Thesaurus) konsultieren kann. Diese Assistenz ist vor allem mit Blick auf die wenig trainierten und unerfahrenen Einsatzkräfte von Vorteil. Die in Abbildung 5.16 unter *comment* hinterlegte Begriffsbeschreibung des KB-Konzeptes „Bereitstellungsraum“ kann beispielsweise bei Bedarf dem Anwender als Information eingeblendet werden.
3. *Wissensakquise und Reasoning (online)* – Beobachtete Fakten der Einsatzlage können gemäß der Terminologie der Wissensbasis abgespeichert werden. Dabei entsteht eine formale Repräsentation der Einsatzlage, die ein automatisches Ableiten von Wissen erlaubt (intelligentes Systemverhalten). Das Problem, wie in der Praxis das Wissen über eine Einsatzlage gemäß der Terminologie abgespeichert werden kann, wird in CASIE pragmatisch durch den Einsatz der ICLs gelöst. Durch die Arbeit mit den ICLs werden z. B. bereits bei den Rollenübernahmen die Namen der entsprechenden Einsatzkräfte den jeweiligen formal definierten Führungskonzepten zugeordnet. Die automatische Auswertung der Item-Nachbedingungen und die manuell abgefragten *query*-Items sind weitere Quellen für eine Wissensakquise. Ein Beispiel der Reasoningfunktionalität ist in Abschnitt 6.1.4 erläutert.

Selbst wenn das verwendete Vokabular sich an einer Anwendungsontologie orientiert, sind in der BOS-Praxis immer noch Missverständnisse bezüglich der Deutung einzelner Fachbegriffe zu erwarten. Da in einer Großschadenslage typischerweise mit unterschiedlich qualifi-

zierten Benutzergruppen zu rechnen ist, ist auch ein differenziert ausgeprägtes Verständnis für BOS-eigene und für BOS-übergreifende Fachtermini bei den Einsatzkräften zu erwarten (vgl. WUCHOLT et al. [WYM+11]). Die Verwendung einer einheitlichen Terminologie als Informationsquelle im Einsatz (Punkt 2.) ist daher besonders hilfreich zur Vorbeugung von begrifflichen Missverständnissen im BOS-Bereich. Wie solch eine Nachschlagefunktionalität in einem UI geeignet umgesetzt werden kann, soll nicht Gegenstand dieser Arbeit sein.

Im Rahmen des SPEEDUP-Projekts konnte diesbezüglich bereits eine Methode erarbeitet werden (siehe WUCHOLT et al. [WYM+11]), welche mit einer Analyse der inter- und intraorganisationalen Kommunikation in einem MANV-Szenario beginnt. Begleitet durch Beobachtungen (Übungen), Dokumentenanalysen (Dienstvorschriften, Regelwerke, Fachbücher) und durch das Mittel (narrativer) Interviews, lässt sich so eine erste präformale Modellierung des Wissensbereiches erreichen. Diese dient dann in einem nächsten Schritt einem sog. *Knowledge Engineer* als Ausgangspunkt einer formalen Modellierung. Die Ontologie selbst wird wiederum Bestandteil einer sog. Wissensbasis (WB) in einem wissensbasierten System (*Knowledge Base System*) sein (vgl. Abb. 5.17). Die Begriffe *Wissensbasis* (WB), *Wissensdatenbank* und *Knowledge Base* (KB) werden in dieser Arbeit synonym verwendet.

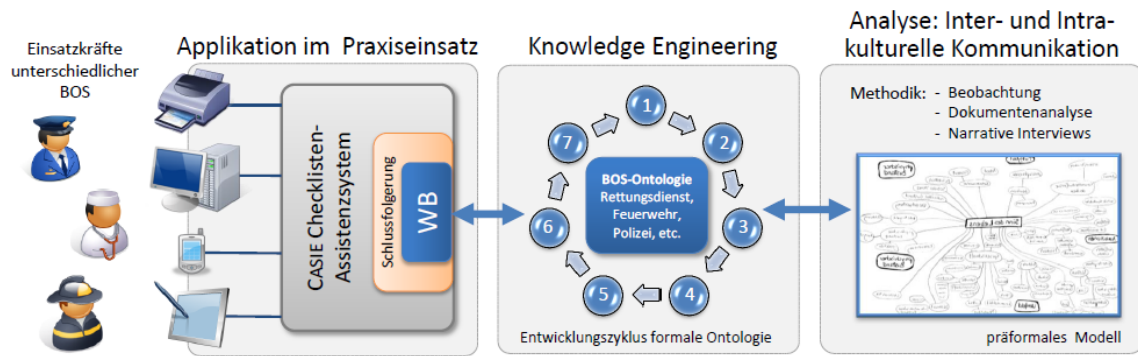


Abbildung 5.17: Kulturanalyse und Knowledge Engineering.

Abbildung 5.17 skizziert die drei Schritte von der präformalen Modellierung hin zu der Nutzung in einer wissensbasierten Softwarekomponente. Die im Rahmen dieser Arbeit erstellte BOS-Ontologie dient ausschließlich als Beispiel einer konkreten Realisierung und muss für einen realen Einsatz noch auf die jeweiligen lokalen Besonderheiten und der benötigten Anforderung hin erweitert/abgeändert werden.

Kapitel 6

Technische Realisierung

In diesem Kapitel werden das CASIE-Architekturkonzept und das Zusammenspiel der einzelnen Komponenten dargestellt. Als Voraussetzung wird angenommen, dass bereits ein geeignetes (robustes) Kommunikationsnetzwerk existiert, worauf CASIE mit seiner Funktionalität aufsetzen kann.

6.1 Komponenten der Casie-Architektur

In Abbildung 6.1 ist das vorgeschlagene Architekturkonzept von CASIE zu sehen.

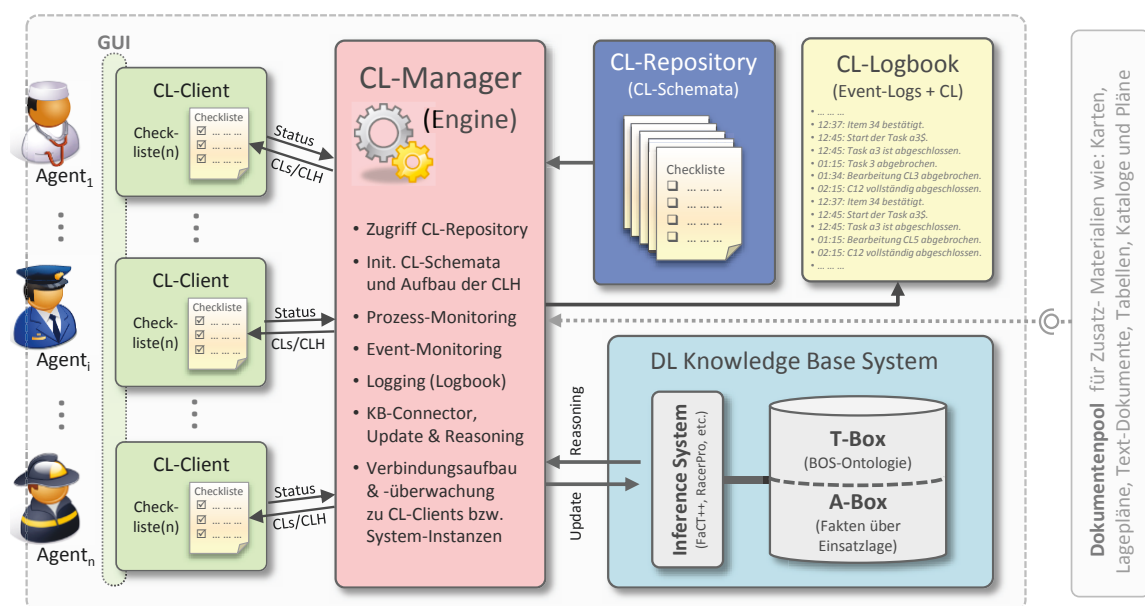


Abbildung 6.1: Meta-Architekturskizze der wichtigsten CASIE-Komponenten.

Eine frühe Version dieses Architekturkonzeptes wurde bereits 2010 auf dem *International Workshop on Emergency Management through Service Oriented Architectures* (vgl. [KGK+10]) vorgestellt. Die hier gezeigte Variante stellt eine Verfeinerung des frühen Konzeptes dar, wobei die automatische Planungskomponente durch eine Checklisten-Assistenz (CL-Manager) spezialisiert wurde. Die Servicekomponente aus [KGK+10] (S. 5 ff.), welche eine automatische Ausführung ausgesuchter Items ermöglicht, wurde in dieser Arbeit ausgearbeitet. Prinzipiell kann, je nach Typ und Aufgabe der Items, auch eine Servicearchitektur,

z. B. durch Bereitstellung von sog. *information gathering*- und *information propagation*-Services, die Arbeit der Einsatzkräfte erleichtern.

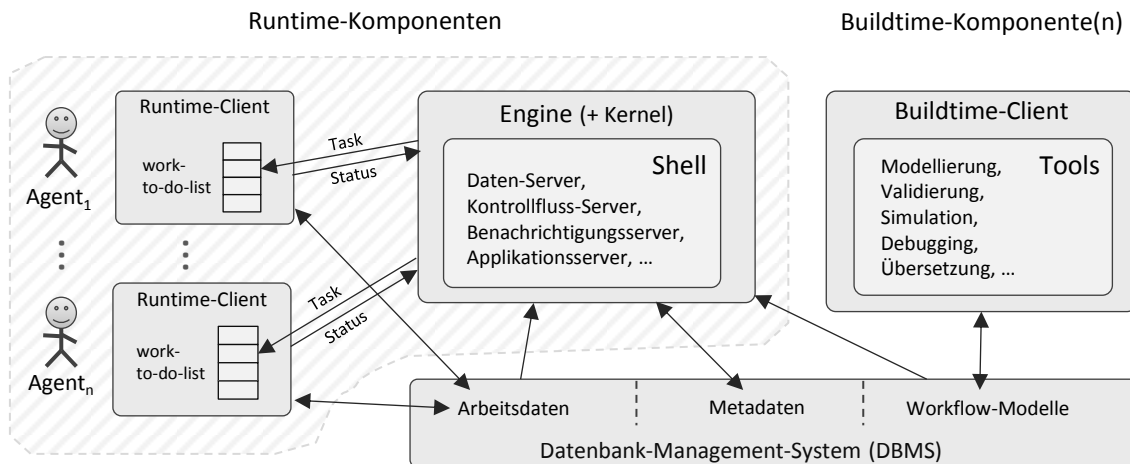


Abbildung 6.2: Architektur eines WfMS (nach KLAUSNER [Kla01]).

Obwohl Checklisten in der Regel keine (reinen) Workflows¹ darstellen (vgl. die Diskussion in Abschnitt 3 und zu Beginn des Kapitels 5), kann die Architektur von CASIE mit der eines Workflow-Management-Systems (WfMS) (vgl. z. B. KLAUSNER [Kla01], S. 5 ff.) verglichen werden. Abbildung 6.2 zeigt den Aufbau eines WfMS, wobei der zu CASIE korrespondierende Bereich schraffiert gekennzeichnet ist. Vor allem das Konzept der *work-to-do-lists* (vgl. Abb. 6.2) ist mit dem der Itemlisten in CASIE vergleichbar. Auch die *distributed and intelligent to-do lists* des Ansatzes von WICKLER et al. (siehe [WTP06]) beruhen auf einem ähnlichen Prinzip. Im Gegensatz zu einem WfMS steht in CASIE jedoch keine automatische, sondern eine manuelle Ausführung von Aufgaben im Zentrum. Es gilt, in einer Multiagentenumgebung physische Aktionen auszuführen und die Einsatzkräfte dabei assistierend mittels Checklisten zu „begleiten“. Eine vollständige automatische Ausführung(-skontrolle) wird durch CASIE nicht realisiert.

Dokumentenpool Die in der Architekturgrafik (Abb. 6.1) rechts außen angedeutete Komponente des zusätzlichen Dokumentenpools (Karten, Lagepläne, Text-Dokumente, Tabellen, Kataloge, Pläne) trägt der Praxisanforderung Rechnung, eine Vielzahl von möglichen bzw. bereits verfügbaren elektronischen Dokumenten genau an den Stellen zugreifbar zu machen, an denen sie als Hilfestellung auch wirklich angebracht sind. Die Realisierung einer geeigneten Zugriffsmöglichkeit auf die verfügbaren Dokumente wird vorausgesetzt. Ein Verweis auf die jeweiligen Zusatzressourcen wird in CASIE über das Definieren eines entsprechenden Merkmals $res(\phi) := \{Ress1, Ress2, \dots\}$ einer CL bzw. eines Items realisiert.

Im Folgenden werden die einzelnen CASIE-Architekturkomponenten mit ihren Funktionen vorgestellt. Wie diese in einem reaktiven und wissensbasierten System zusammenspielen, wird im Anschlussabschnitt 6.2 erläutert.

¹ Als Workflow (-Instanz) wird nach JABLONSKI [Jab95], „eine ausführbare Beschreibung bzw. Abbild eines Geschäftsprozesses“ verstanden.

6.1.1 CL-Client, Benutzerschnittstelle und Endgeräte

Der sog. *Checklisten-Client* (kurz *CL-Client*) stellt als eine der Runtime-Komponenten die Funktionalitäten der ICL den Einsatzkräften bereit. Jeder CL-Client kommuniziert dabei mit dem CL-Manager, von dem er mit Daten versorgt wird und der die Dynamik steuert. Abbildung 6.1 abstrahiert das Detail der technischen Aufteilung der Architekturkomponenten auf die eingesetzten Endgeräte. Die Abbildung lässt bewusst offen, ob jedes Endgerät, auf dem der CL-Client läuft, auch zugleich alle restlichen Komponenten bereitstellen muss. Ob sich der CL-Client und der CL-Manager auf ein und demselben Rechner befinden oder aber auf unterschiedliche Rechner im Netzwerk verteilt sind, hängt von der konkret gewählten Umsetzung der CASIE-Architektur ab. Aus der Perspektive der Komponentenverteilung kommen folgende zwei Grundvarianten der Architekturumsetzung in Betracht:

1. *Stand-Alone-Variante* – Jedes Endgerät verfügt über eine komplette Implementierung aller CASIE-Komponenten in einer einzigen Applikation bzw. auf einem Endgerät.
2. *Network-Variante* – Jedes Endgerät benötigt „nur“ einen CL-Client und eine Netzwerkanschlussschnittstelle. Über das Netzwerk kann sich der Client an einen verfügbaren CL-Manager anmelden und seine Dienste in Anspruch nehmen.

Welche der beiden Varianten in der Praxis einsetzbar ist, hängt von der jeweiligen Implementierung und der zu erwartenden Netzwerkabdeckung ab. Im SPEEDUP-Projekt hat sich eine Variante der ersten Lösungen als vorteilhaft herausgestellt (siehe Abschnitt 6.2), bei der auf jedem Endgerät eine komplette Instanz aller Architekturelemente des SPEEDUP-Demonstrators untergebracht wurde. Dies hat den Vorteil, dass das System unabhängig von einer bestehenden Netzwerkverbindung autonom arbeiten kann. Kann ein System während einer Großschadenslage eine Verbindung zu einem zweiten System aufbauen, so können diese dann ihre bis dahin aggregierten Daten abgleichen. Dies ist eine Anforderung, die sich aus der Einsatzcharakteristik einer Großschadenslage direkt ableitet und daher auch für die konkrete CASIE-Implementierung beachtet werden sollte.

Für die Stand-Alone-Variante (und den Datenaustausch in der Network-Variante) ist die Anwendung verschiedener Netzwerktopologien möglich. Beispielsweise kann ein zentraler Server alle CL-Clients bedienen (überwachen) oder aber ein dediziertes Endgerät übernimmt bei bestehender Netzwerkverbindung zusätzlich die Aufgabe des Servers für alle CL-Clients, die im Einsatz sind.

In der Network-Variante wird die Verbindung einzelner CL-Clients untereinander (bzw. mit einer zentralen Verwaltungskomponente) über ein zuvor etabliertes Kommunikationsnetzwerk realisiert. In diesem Fall benötigt der Nutzer als Mindestanforderung nur einen auf das eingesetzte Endgerät (PC, Notebook, Tablet) zugeschnittenen Client und eine Netzwerkverbindung. Gemäß des klassischen *Client/Server*-Prinzips [Gei95] nutzt der Client dann die von einem dedizierten CL-Manager innerhalb eines Netzwerkes bereitgestellten Funktionalitäten (siehe CL-Manager weiter unten).

Der BOS-Bereich ist ein stark heterogener Bereich, was sich auch in der vorhandenen IT-Unterstützung widerspiegelt. Ziel einer konkreten CASIE-Implementierung sollte daher eine möglichst hohe Plattformunabhängigkeit bezüglich der eingesetzten Systeme sowie eine breite Unterstützung möglicher Geräteklassen sein. Abbildung 6.3 skizziert von Tablet-PCs über normale Desktop-PCs bis hin zu Smartphones ein Spektrum möglicher Geräteklassen. Weiterhin ist an eine Druckeranbindung als „Rückfallebene“ und/oder zu *Back-Up*- und Dokumentationszwecken als notwendiges Kriterium für einen Praxiseinsatz zu denken.

Obwohl die Gestaltung einer geeigneten grafischen Benutzerschnittstelle (engl. *graphical user interface*, kurz GUI) nicht Zielstellung dieser Arbeit ist, sollen im Folgenden einige allgemeine Aspekte und Anforderungen an diese zusammengefasst werden.

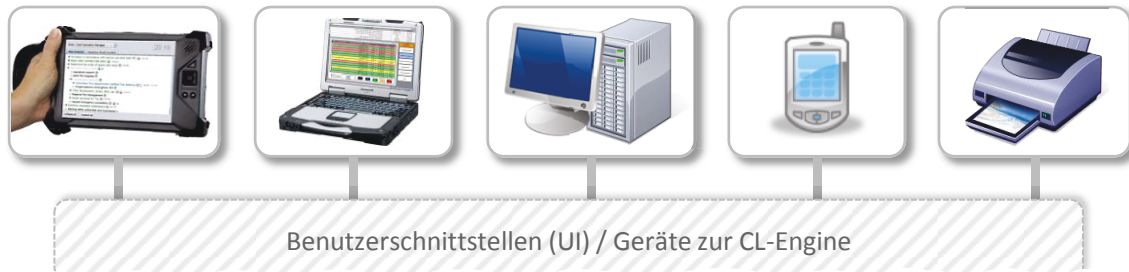


Abbildung 6.3: Mögliche Klassen von Endgeräten.

Für einer CASIE-Implementierung stellt sich die Aufgabe, jeweils eine geeignete GUI für die Buildtime- und eine für die Runtime-Komponenten zu realisieren. Für die Modellierung der Domänenontologie (mit Rollen und Organisationsstruktur) kann auf vorhandene Werkzeuge (wie z. B. den Ontologieeditor Protégé) zurückgegriffen werden. Für die Konfiguration der Checklisten bedarf es hingegen neuer Softwarewerkzeuge, die auf die jeweiligen Anwender zugeschnitten sind. PEINEL et al. schlagen hierzu z. B. in [PRW12b] einen webbasierten Editor vor, mit dem Einsatzkräfte SOPs bzw. Checklisten in elektronischer Form ablegen und organisieren können.

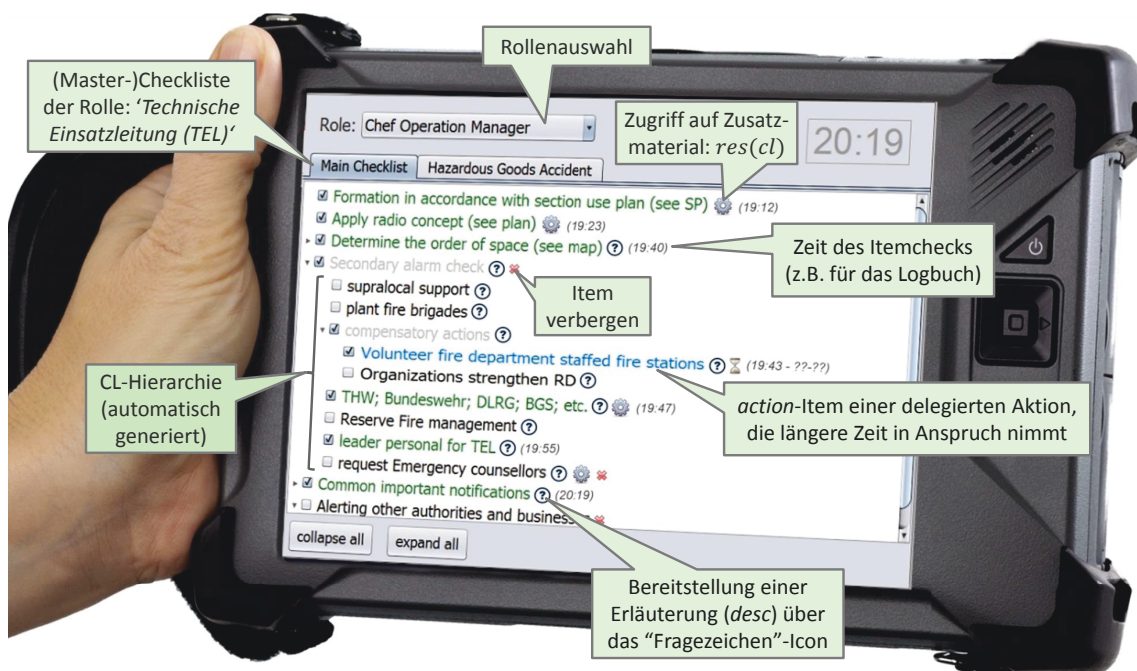


Abbildung 6.4: Eigene Studie eines möglichen GUI (vgl. [KWB12]).

Wie die Erfahrungen des SPEEDUP-Projekts zeigten, besteht bereits bei der Konfiguration eines solchen Systems ein sehr hoher *Usability*-Anspruch an die eingesetzten Softwarewerkzeuge. Da zudem im BOS-Bereich nicht davon auszugehen ist, dass alle Nutzer erfahren im Umgang mit Computern sind, und da sich die Arbeitsumgebungen und -bedingungen teilweise recht widrig gestalten (wechselnde Lichtverhältnisse, Regen, Schmutz, Kälte, Hitze,

Stress), leitet sich die Notwendigkeit einer möglichst einfach zu bedienenden Benutzerschnittstelle ab. Nicht zuletzt um der Akzeptanzproblematik entgegen zu wirken, sollten die Werkzeuge den Einsatzkräften ermöglichen, ihr eigenes CL-Repository möglichst einfach einzupflegen und zu warten. Erfahrungen der SPEEDUP-Praxis zeigten darüber hinaus, dass sich die Führungskräfte eine einheitliche Benutzerschnittstelle für die Buildtime- und Runtime-Komponenten wünschen. Eines der Designziele sollte daher darin liegen, die notwendige Interaktion mit dem System auf ein Minimum zu beschränken.

Die Verknüpfung der einzelnen CLs einer CLH lässt sich in einem GUI auf mindestens zwei Arten visualisieren. So können z. B. ausgehend von einer MCL alle Items der Sub-CLs (der kompletten Hierarchie) in einer Baumstruktur visualisiert werden (siehe Beispiel in Abb. 6.4). Oder es können alle Checklisten einer Hierarchie jeweils separat dargestellt werden. Die Darbietung aller Items für eine MCL, in Form einer flachen Liste, wird bei großen und komplexen Hierarchien zu unübersichtlich und sollte daher keine Option sein. Eine mögliche Itemreihenfolge sollte im Benutzerinterface der Ordnungsconstraints (\prec) folgen. Für jede partielle Ordnung existieren mehrere Möglichkeiten der seriellen Darstellungen. Abbildung 6.4 zeigt eine Möglichkeit einer Umsetzung, die im Rahmen dieser Arbeit prototypisch entwickelt wurde. Bedingte Items sind in der Abbildung nicht beachtet.

Welche Darstellungsformen konkret den Kriterien einer angestrebten Benutzerfreundlichkeit entsprechen, kann letztendlich nur eine Evaluierung verschiedener Konzepte in der Praxis zeigen. In dieser Arbeit soll der sich dadurch eröffnende eigene Themenkomplex nicht weiter vertieft werden. Vielmehr wird im Folgenden vorausgesetzt, dass geeignete Werkzeuge zur Konfiguration der Checklisten und der Ontologie vorhanden sind und von den BOS-Mitarbeitern (mit oder ohne Hilfe von Experten) bedient werden können.

6.1.2 CL-Manager

Der *CL-Manager* stellt die zentrale Steuereinheit in CASIE dar. Je nachdem, ob ein CL-Client für eine Einsatzkraft isoliert genutzt wird (Single-User-Modus, kurz *SUM*) oder aber im Datenaustausch mit weiteren Clients (Multi-User-Modus, kurz *MUM*) betrieben wird, steht ein eingeschränkter bzw. der volle Funktionsumfang dem Nutzer bereit. Folgende Funktionalitäten stellt der CL-Manager einem CL-Clients zur Verfügung.

- *Zugriff auf Repository* – Der CL-Manager ermöglicht den CL-Clients einen wahlfreien Zugriff auf alle hinterlegten (relevanten) CL-Schemata (siehe nächsten Unterabschnitt).
- *Prozess-Monitoring* – Der CL-Manager überwacht den Status aller Checklisten (von der Aktivierung bis zum Ende der Bearbeitung) und kann somit einen Überblick über alle abgeschlossenen, abgebrochenen und noch laufenden Prozesse gemäß des Prozesskontextes der ICL liefern. Dies wird durch die Überwachung aller Client-Ereignisse und aller externen Ereignisse (ausgelöst durch andere CASIE-Instanzen im Multi-User-Mode) erreicht.
- *KB-Connector* – Alle über den Client erfassten Informationen über die Einsatzlage werden vom CL-Manager erfasst und zur Erweiterung der KB genutzt. Zu den Informationen zählen die Effekte der bestätigten Items bzw. bereits abgeschlossener Checklisten. Ein automatisches Schließen über die Informationen für situationsangepasste ICL wird so erst ermöglicht.
- *Logging/Dokumentation* – Durch das automatische Speichern von Informationen über die Statuswechsel der aktiven CL und deren Items ist der CL-Manager in der Lage,

einen *Logging*-Mechanismus zu Dokumentationszwecken bereitzustellen (siehe nächsten Abschnitt).

- *Network-Manager* – Der Verbindungsaufbau und die Überwachung der Ereignisse anderer Clients ermöglichen eine einsatzübergreifende Zusammenführung und Beachtung aller durch die Arbeit mit den Checklisten gewonnenen Informationen.

Auf die einzelnen Funktionsbereiche, die das reaktive Systemverhalten von CASIE bestimmen, wird im kommenden Abschnitt 6.2 genauer eingegangen.

6.1.3 CL-Repository & -Logbook

Im *Checklisten-Repository* ist die Menge aller erarbeiteten Checklisten hinterlegt. Hierzu können die Checklisten jeweils lose in einer geeigneten Datenstruktur organisiert werden. Da selbst bei einer größeren Anzahl vorgehaltener Checklisten der zu erwartende Datenumfang relativ gering ausfällt, bedarf es keiner besonderen Lösungen für große Datenmengen. Die Domänenexperten können mittels eines Build-Time-Editors (welcher an dieser Stelle nicht näher erläutert wird) ihre eigenen CLs *erstellen*, *ändern* und *löschen*. Die Speicherung über das Dateisystem der Plattform ist hierbei ausreichend. Eine einfache Möglichkeit stellt z. B. eine Menge von Dateien dar, bei der jede Datei für eine einzelne Checkliste steht. Die Checklisten können zentral vorgehalten werden, was eine Pflege und Aktualisierung erleichtert. Als Build-Time-Komponente kann die Erstellung der CLs z. B. über eine Tabellenkalkulationsanwendung gängiger Office-Produkte erfolgen. Dies ermöglicht den Einsatzkräften die Konfiguration des Repositories unter Verwendung gängiger Softwareumgebungen.

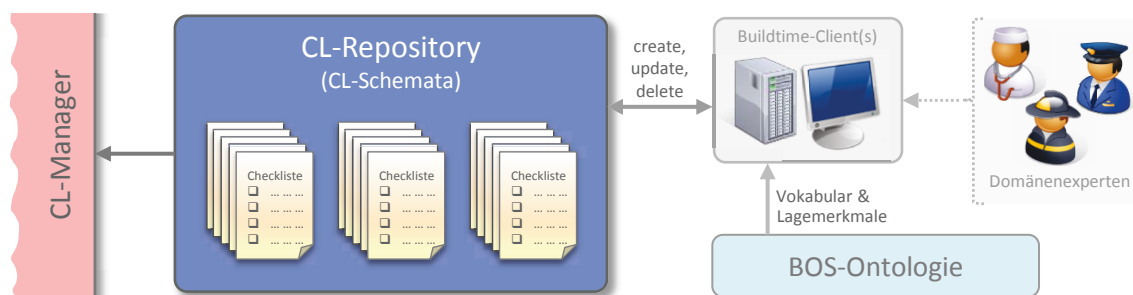


Abbildung 6.5: Die Komponente des CL-Repository.

Die Komponente *CL-Logbook* steht für eine digitale Variante eines Einsatztagebuches, in der Informationen über die Lageentwicklung bezüglich der laufenden und bereits erledigten Handlungen protokolliert werden können. Der CL-Manager aggregiert hierzu die Informationen über die vom CL-Client registrierten Benutzeraktionen. Die so gewonnene Historie der Aufgabenabarbeitung leistet einen entscheidenden Beitrag zur Pflicht der Einsatzdokumentation, welche im BOS-Bereich einen hohen Stellenwert einnimmt (siehe Abschnitt 7.3).

Jedem im Einsatz aktiven CL-Client kann sein eigenes Einsatztagebuch zugeordnet werden. Zusätzlich können durch das Anlegen verschiedener Eintragungsschemata (Log-Modus) (siehe Abb. 6.6) eine auf die jeweiligen Ansprüche des Anwenders zugeschnittene Form und der gewünschte Detaillierungsgrad angegeben werden. Die Aggregation aller Informationen kann mittels einer dedizierten Logging-Komponente des CL-Manager erfolgen, welche alle relevanten Informationen z. B. in Form eines einfachen Logfiles im CL-Client abgespeichert. Tabelle 6.1 zeigt für jeden der möglichen internen Zustandsübergänge einen entsprechenden Beispieleintrag. Die genaue Ausprägung der Einträge im Logbuch lässt sich je nach

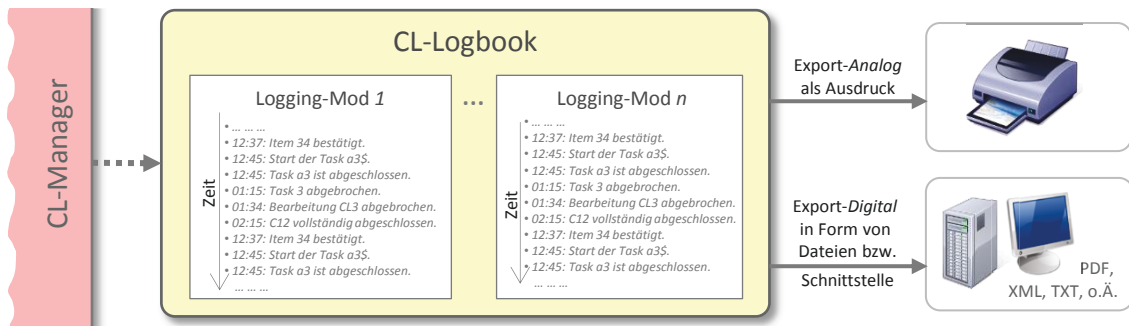


Abbildung 6.6: Die Komponente des CL-Logbooks.

Wunsch frei gestalten. Die Tabelle soll ein Gefühl dafür vermitteln, welche Möglichkeiten eine automatische Loggingfunktion in CASIE bietet.

Event	Beispiel-Schema für Einträge eines Logbook
$init(cl)$	$time$ Uhr: Start der Checkliste $name(cl)$.
$check(\phi)$	$time$ Uhr: Item ϕ bestätigt.
$start(\phi)$	$time$ Uhr: Start der Task (<i>action-Item</i>) $name(\phi)$.
$finish(\phi)$	$time$ Uhr: Task (<i>action-Item</i>) $name(\phi)$ ist abgeschlossen.
$abandon(\phi)$	$time$ Uhr: Task (<i>action-Item</i>) $name(\phi)$ abgebrochen.
$cancel(cl)$	$time$ Uhr: Bearbeitung von $name(cl)$ abgebrochen.
$finish(cl)$	$time$ Uhr: $name(cl)$ vollständig abgeschlossen.
$login(w)$	$time$ Uhr: w ist am CL-Client mit der ID id angemeldet.
$fulfil(w)$	$time$ Uhr: w übernimmt Rolle $name(r)$.

Tabelle 6.1: Mögliches Schema der Logbook-Einträge.

Die beiden letzten Einträge in der Tabelle 6.1 beziehen sich bereits auf die Interaktion eines Nutzers mit dem System selbst. Im kommenden Abschnitt 6.2 werden die Loggingfunktionalität und die Events näher erläutert.

6.1.4 Knowledge-Base (System) / BOS-Ontologie

In CASIE soll der Wissensbereich der BOS-Domäne in Form einer formalen Ontologie modelliert werden. Unter einer Ontologie wird in der Informatik eine „*explicit specification of a conceptualization*“ [Gru93] eines Wissensbereiches verstanden. In der jüngeren Zeit wurde der formale Aspekt einer solchen Spezifikation herausgehoben und obige Definition zu „*explicit and formal specification of a conceptualization*“ erweitert [AH04]. Die Begriffe BOS-Ontologie, Anwendungsontologie, Domänenontologie und Ontologie werden in der weiteren Arbeit, wenn nicht explizit eingeschränkt, synonym verwendet. Ein wesentliches Merkmal der CASIE-Architektur liegt daher darin, die formale Spezifikation des ontologischen BOS-Domänenwissens im Rechner geeignet zu repräsentieren und als Wissensbasis zu nutzen. Das so modellierte Wissen ist konform seiner Bedeutung maschinell verarbeitbar (*machine processable*), wodurch sich vor allem die Möglichkeit ergibt, automatisch über die Menge der vorhandenen Fakten der Wissensbasis schließen zu können. Der oft statt *machine processable* unsauber verwendete Ausdruck *machine understandable* sollte vermieden

werden, da er ein wirkliches Verstehen im Sinne des menschlichen Verstandes suggeriert, wovon jedoch im Rechner keine Rede sein kann.

Konkret wird CASIE durch den Aufbau einer Wissensbasis in die Lage versetzt, die Gültigkeit der Anwendungskriterien von ICLs und der Vorbedingungen der Items automatisch zu evaluieren. Falls die Qualität und die Quantität der Datenlage es ermöglichen, kann diese zu einer intelligenten, lagekonformen Anpassung der ICLs genutzt werden (siehe Abschnitt 6.2).

Ein weiterer Vorteil dieser Art der Modellierung liegt darin, dass Änderungen am Domänenmodell (in der Regel) ohne eine Anpassung der Programmlogik möglich sind, da das Wissen über die Anwendungsdomäne unabhängig von der jeweiligen Programmlogik (CL-Manager) realisiert ist. In der KI werden Systeme, in denen deklaratives Wissen einer Domäne von der Programmlogik getrennt wird, als *wissensbasierte Systeme* oder auch *Knowledge Base Systems* bezeichnet. Die Abbildung 6.7 skizziert den Aufbau einer Wissensbasis (*Knowledge Base*) als die zentrale Komponente eines solchen Systems.

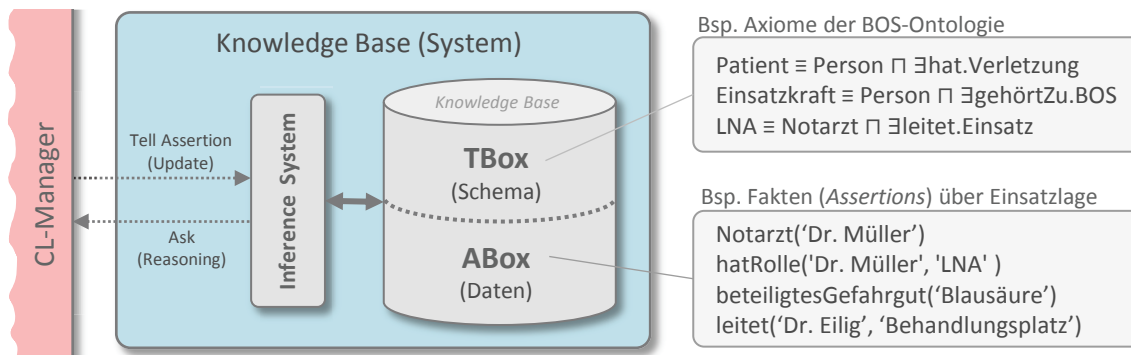


Abbildung 6.7: Architektur eines *Knowledge Base Systems* (KBS) (nach BAADER [BCM+03]).

Eine formale Wissensbasis besteht aus einer *TBox* (das T steht für *Terminology*) und einer *ABox* (das A steht für *Assertions*) [BCM+07]. Die *TBox* enthält axiomatische Definitionen (terminologisches Wissen) aller Konzepte und deren Beziehungen untereinander – in dieser Arbeit z. B. bereitgestellt durch eine BOS-Ontologie. Sie stellt das Vokabular des Anwendungsgebietes bereit und legt fest, welche Relationen zwischen den Instanzen der Konzeptklassen bestehen. In der *ABox* wird mit den Sprachmitteln der *TBox* Faktenwissen (*Assertions*) über Individuen (Objekte der Welt) abgelegt. Die *ABox* beinhaltet Informationen über einen konkreten Zustand der Welt (der Anwendungsdomäne). Um die Wissensbasis nutzen zu können, bedarf es der Anwendung entsprechender Schlussfolgerungssysteme (Reasoner). Das Ziel der logischen Inferenz ist es, zu entscheiden, ob aus einer gegebenen Wissensbasis *KB* die Gültigkeit einer gegebene Formel φ ableitbar (logisch inferierbar) ist, geschrieben $KB \models \varphi$. Alle Komponenten zusammen, stellen ein KB-System dar. Die KB-Komponente stellt eine sog. Tell/Ask-Schnittstelle zum CL-Manager bereit, über die Fakten der Welt der KB mitgeteilt werden können (Tell) und über die Anfragen über die Gültigkeit einer Formel φ gestellt werden können (Ask) [BCM+07]. Abbildung 6.7 stellt dies schematisch dar. Auf der rechten Seite sind jeweils Beispiele von Axiomen bzw. Assertions gegeben. Das Axiom $\text{Patient} \sqsubseteq \text{Person} \sqcap \exists \text{hat.Verletzung}$ definiert z. B. ein Konzept *Patient*, zu dem Individuen zählen, die eine Person sind und zugleich eine Verletzung aufweisen. Eine konkrete Beobachtung, z. B. dass Dr. Müller die Rolle des LNA übernommen hat, wird in der *ABox* als Assertion $\text{hatRolle}(\text{'Dr. Müller'}, \text{'LNA'})$ repräsentiert.

Als Sprachmittel zur terminologischen Wissensrepräsentation hat sich in der KI-Forschung der Einsatz von Beschreibungslogik (*Description Logic*, DL) etabliert. DL steht für ei-

ne ganze Familie von Formalismen zur Wissensrepräsentation (*Knowledge Representation*, KR), deren Ursprünge in der Forschung zu *semantischen Netzen* und *Frames* (siehe MINSKY, [Min75]) liegen. Frühere objektzentrierte Wissensrepräsentationssysteme (siehe z. B. SCHMOLZE et al. [SBI85]) verfügten noch nicht über eine formale Semantik. Dies führte zur Entwicklung entsprechender Sprachformalismen wie DLs, die ausdrucksstärker als die Aussagenlogik (*propositional logic*), aber noch nicht so komplex wie die volle Prädikatenlogik 1. Stufe (PL1) (*first-order predicate logic*, FOL) sind, sodass einige interessante Entscheidungsprobleme (und damit Schlussfolgerungen) effektiv berechnet werden können (vgl. z. B. [BHS08]). DL stellt wegen ihrer Beschränkung auf binäre Prädikate und den Verzicht auf Funktionen eine entscheidbare Untermenge der PL1 dar.

Ein einfaches Beispiel soll den prinzipiellen Zusammenhang von PL1 und DL verdeutlichen. Angenommen wir wollen folgenden Sachverhalt modellieren: „*Ein Leitender Notarzt ist ein Notarzt, der einen Einsatz leitet*“. Formalisiert in PL1 lautet er, wie folgt

$$\forall x. [\text{LeitenderNotarzt}(x) \rightarrow \text{Notarzt}(x) \wedge \exists y. [\text{leitet}(x, y) \wedge \text{Einsatz}(y)]] .$$

Die Syntax der Description Logic erlaubt, den gleichen Sachverhalt einfacher und in einer prägnanten, variablenfreien Syntax zu notieren:

$$\text{LeitenderNotarzt} \sqsubseteq \text{Notarzt} \sqcap \exists \text{leitet.} \text{Einsatz} .$$

Im Beispiel wird das Konzept *LeitenderNotarzt* durch Zuhilfenahme bereits definierter Konzepte (*Notarzt* und *Einsatz*) und deren Relationen (hier z. B. *leitet*) neu definiert. Eine Ontologie lässt sich als eine Sammlung von DL-Axiomen obiger Beispielstruktur verstehen. Das zentrale Element jeder Ontologie stellen die Konzepte (auch Klassen genannt) und die Relationen (auch Rollen oder Eigenschaften genannt) zwischen den Konzepten der Anwendungsdomäne dar [NM01]. (Dieser Rollenbegriff darf nicht mit dem in CASIE eingeführten Begriff der Rolle einer Einsatzkraft (vgl. Def. 2) verwechselt werden.) Zum Beispiel kann die Klasse *Patient* alle Verletzten und Erkrankten in einem MANV repräsentieren. Ein konkreter Patient ist diesbezüglich ein Individuum, eine *Instanz* dieser Klasse. Je nach Modellierung und Anforderung können Konzepte auch abstrakte Dinge wie z. B. eine *Schadenslage* oder ein *Gefahrenpotential* repräsentieren. Spezialisierungen zu Konzepten können in Form von Subklassen realisiert werden. Zum Beispiel wäre das Konzept *Verletzter* eine Subklasse (*sub-class*) von *Patient* und dieses wiederum von *Person*. Durch Relationen werden Eigenschaften von Klassen und Beziehungen zwischen konkreten Instanzen beschrieben. So kann z. B. eine Relation *hatDiagnoseGestellt* zwischen der Klasse *Patient* und *Notarzt* definiert werden. Ein konkreter Patient kann so später in Beziehung zu seinem diagnostizierenden Notarzt gebracht werden. Es handelt sich in diesem Beispiel um eine binäre Relation *hatDiagnostiziert(Notarzt,Patient)*, deren konkrete Ausprägung z. B. *hatDiagnostiziert('Dr. Eilig', 'Max Mustermann')* lauten könnte. Eine ausführliche Abhandlung zum Thema Ontologien und Beschreibungslogik lässt sich z. B. in [SS04] bzw. [BCM+07] finden und soll nicht Gegenstand dieser Arbeit sein.

Die Wahl einer konkreten Ontologiesprache

Bezüglich des theoretischen Hintergrundes von DL und KBS stellt sich die Frage der Umsetzung in einem entsprechenden Softwaresystem. Eine konkrete Realisierung von DL stellt eine Variante der *Web Ontology Language (OWL)* dar, für die mittlerweile eine Vielzahl hilfreicher Softwarewerkzeuge existieren. Die Sprache ist eine Weiterentwicklung der Ontologiesprache DAML+OIL (siehe [DAML]). Bei der Entwicklung der Sprache hatte das Forschungsgebiet der Beschreibungslogiken einen großen Einfluss, vor allem was die formale

Semantik, die Wahl der Sprachkonstruktoren und die Integration der Datentypen und spezieller Werte anbelangt [HPM+07]. OWL ist eine semantische Auszeichnungssprache (*Markup-Sprache*), die ursprünglich zur Veröffentlichung und zum Austausch von Ontologien im *World Wide Web* (WWW) entwickelt wurde. Durch eine relativ gute Tool-Unterstützung und die theoretische DL-Basis ist OWL auch außerhalb des WWW-Umfeldes als Mittel der Wahl zur Ontologieentwicklung weit verbreitet. Mit OWL können Klassen (*Classes*), Eigenschaften (*Properties*) und Instanzen (*Individuals*) beschrieben werden. Die Klassen stehen für die Konzepte, welche spezielle Eigenschaften besitzen können. Instanzen sind Individuen einer oder mehrerer Klassen. Die OWL-Spezifikation² unterscheidet zwischen den drei zunehmend komplexen Sprachversionen (auch *profiles* genannt) *OWL-Lite*, *OWL-DL* und *OWL-Full*. Da im Rahmen dieser Arbeit nur OWL-DL relevant ist, soll auf die beiden anderen Profile nicht weiter eingegangen werden. OWL-DL („DL“ steht hier für *Description Logic*) stellt einen ausdrucksstarken Sprachformalismus dar und ist äquivalent zu der Beschreibungslogik $\mathcal{SHOIN}(\mathbf{D})$ [HPM+07].

Der praktische Einsatz von Beschreibungslogiken ließ in vielen Anwendungsbereichen den Wunsch aufkommen, Schlussfolgerungen über spezielle Domänen mit einer fixen (*concrete*) Semantik für ausgewählte Konzepte, wie *Integer*, *Real* oder *String*, nutzen zu können. Ziel war es, konkrete Qualitäten (wie z. B. Gewicht, Temperatur oder Entfernung) für reale Objekte der Welt einfacher zu modellieren und verarbeiten zu können. Um dies zu ermöglichen, wurden die DL-Reasoner um die Möglichkeit des Schließens über diese sogenannten *Concrete Domains* (kurz \mathbf{D}) erweitert. Für weitere Details sei z. B. auf [BH91; HMW01; Lut03] verwiesen. Folgende Auflistung soll speziell für $\mathcal{SHOIN}(\mathbf{D})$ die charakteristischen Sprachbestandteile wiedergeben (siehe Description Logic Complexity Navigator [DLCN]):

- \mathcal{S} – steht für transitive Rollen (Rollen-Axiom): $Trans(R)$.
- \mathcal{H} – steht für Rollen-Hierarchie (Rollen-Axiom): $R \subseteq S$.
- \mathcal{O} – steht für sog. *Nominale* (abgeschlossene Klassen), d. h. für eine Menge von Individuen a_1, \dots, a_n ist $\{a_1, \dots, a_n\}$ ein eigenes Konzept.
- \mathcal{I} – steht für inverse Rollen: R^- .
- \mathcal{N} – steht für eine (*unqualified*) *number restrictions*: $(\geq n R), (\leq n R)$.
- \mathbf{D} – steht für *Concrete Domains*.

Obige Sprachbestandteile ermöglichen es, ein sehr detailliertes Domänenmodell zu erstellen. Die im Rahmen dieser Arbeit beispielhaft umgesetzte BOS-Ontologie (siehe [BOS-O]) nutzt direkt (oder indirekt) all diese Sprachbestandteile, indem es die entsprechenden OWL-Konstrukte verwendet. In den Abbildungen 6.8 und 6.9 wird die Äquivalenz ausgewählter OWL-Konstrukte (linke Spalte) mit der entsprechenden korrespondierenden DL-Syntax aufgezeigt, ergänzt um eine jeweilige FOL-Variante bzw. Beispiele.

Die Praxiserfahrungen mit OWL führten in jüngster Zeit zu dem Wunsch nach Erweiterung des alten OWL-Standards und schließlich zu einer überarbeiteten OWL-2-Spezifikation [OWL2]. In OWL 2 wurden zusätzliche ontologische Axiome zur Steigerung der Ausdruckskraft eingeführt. Des Weiteren wurden nichtlogische Erweiterungen, z. B. bei Syntax und Metadaten, und eine Überarbeitung der OWL-Varianten (Lite/DL/Full) vorgenommen. Durch die Erweiterungen hat sich auch die DL-Basis von $\mathcal{SHOIN}(\mathcal{D})$ zu $\mathcal{SROIQ}(\mathcal{D})$ geändert (erweitert). An dieser Stelle soll auf eine detaillierte Vorstellung von OWL 2 verzichtet werden (siehe [OWL2]).

² siehe <http://www.w3.org/2004/OWL/> (abgerufen am 01.12.2013)

Constructor	DL Syntax	Example	FOL Syntax
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human \sqcap Male	$C_1(x) \wedge \dots \wedge C_n(x)$
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor \sqcup Lawyer	$C_1(x) \vee \dots \vee C_n(x)$
complementOf	$\neg C$	\neg Male	$\neg C(x)$
oneOf	$\{x_1\} \sqcup \dots \sqcup \{x_n\}$	{john} \sqcup {mary}	$x = x_1 \vee \dots \vee x = x_n$
allValuesFrom	$\forall P.C$	\forall hasChild.Doctor	$\forall y.P(x, y) \rightarrow C(y)$
someValuesFrom	$\exists P.C$	\exists hasChild.Lawyer	$\exists y.P(x, y) \wedge C(y)$
maxCardinality	$\leq nP$	≤ 1 hasChild	$\exists^{\leq n} y.P(x, y)$
minCardinality	$\geq nP$	≥ 2 hasChild	$\exists^{\geq n} y.P(x, y)$

Abbildung 6.8: OWL-Konstruktoren vs. DL (Quelle: [BHS08]).

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human \sqsubseteq Animal \sqcap Biped
equivalentClass	$C_1 \equiv C_2$	Man \equiv Human \sqcap Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$ Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} \equiv {G_W_Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg \{x_2\}$	{john} $\sqsubseteq \neg$ {peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter \sqsubseteq hasChild
equivalentProperty	$P_1 \equiv P_2$	cost \equiv price
inverseOf	$P_1 \equiv P_2^-$	hasChild \equiv hasParent $^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor $^+$ \sqsubseteq ancestor
functionalProperty	$\top \sqsubseteq \leq 1P$	$\top \sqsubseteq \leq 1$ hasMother
inverseFunctionalProperty	$\top \sqsubseteq \leq 1P^-$	$\top \sqsubseteq \leq 1$ hasSSN $^-$

Abbildung 6.9: OWL-Axiome vs. DL (Quelle: [BHS08]).

Für diese Arbeit relevant ist jedoch die Tatsache, dass ab OWL 2 eine explizite Angabe negierter Relationen (z. B. \neg hatDiagnostiziert('Dr. Eilig', 'Max Mustermann')) möglich ist. Für CASIE ist dies vor allem bei der Aufnahme von Beobachtungen der Fakten über die Welt von Interesse, bei denen ein explizites Nichtzutreffen eines bestimmten Merkmales im System aufgenommen werden soll. Zum Beispiel kann der Fakt, dass kein Gefahrgut bei einem Großbrand mit beteiligt ist (\neg beteiligtGefahrgut('MANV-A7')), somit in CASIE explizit repräsentiert werden. Dieser Fakt hat mehr Aussagekraft als das bloße Nichtwissen, ob Gefahrgut beteiligt ist oder nicht. Folgende zwei Punkte sprechen zusätzlich für den Einsatz von OWL-DL als Repräsentationssprache in CASIE:

1. *Entwicklungstool-Support.* Da OWL in der jüngsten Zeit als quasi de facto Standard für die Ontologieentwicklung auch außerhalb des Einsatzes in der *Semantic Web Vision* annoncierte, existieren mittlerweile einige mächtige Softwarewerkzeuge zur Entwicklung und Integration von OWL-Ontologien. Zu nennen sei hier z. B. der Ontologieeditor *Protégé* [Protégé] und die *OWL API* [HB11].
2. *Formale Semantik.* Da sich die wohldefinierte Syntax der Sprachelemente von OWL-DL auf die entsprechende Syntax der Beschreibungslogik $\mathcal{SHOIN}(\mathbf{D})$ abbilden lässt, können Schlussfolgerungsmechanismen der Beschreibungslogiktheorie auf die Ontologie angewandt werden. Dies ist nur möglich, da DL über eine formale Semantik verfügt. Des Weiteren existiert eine Reihe von Reasonern wie z. B. [Racer], [FaCT++] und [Pellet], die als Beschreibungslogikdialekt OWL-DL unterstützen.

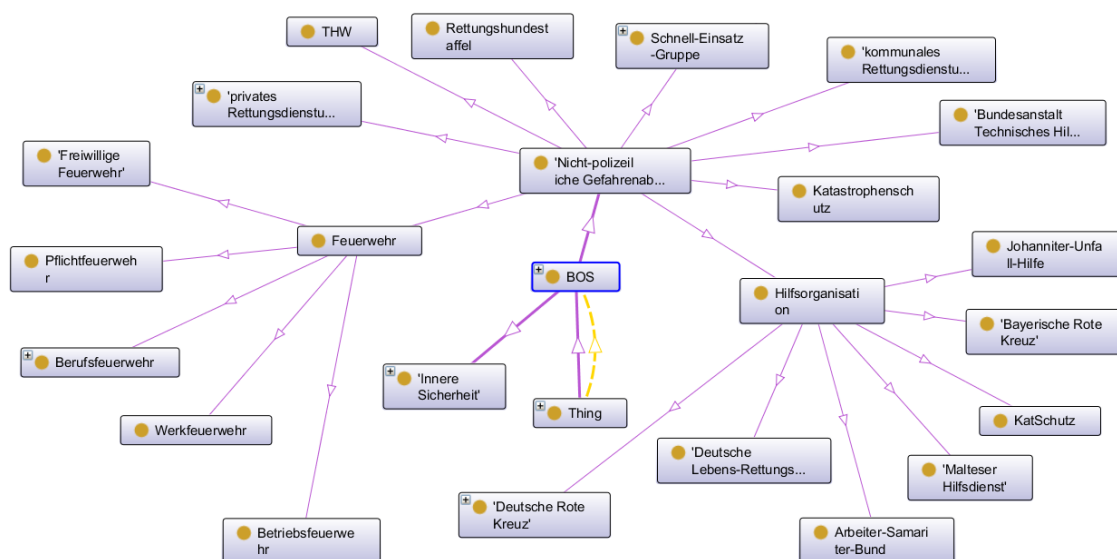


Abbildung 6.10: TBox-Auszug der BOS-Definitionen aus [BOS-O].

Modellierung und Reasoning-Fähigkeiten

In diesem Unterabschnitt wird das Grundprinzip hinter der Reasoning-Funktionalität und den dazu notwendigen TBox-Axiomen anhand eines einfachen Beispiels illustriert. Stellen wir uns vor, wir wollen in Erfahrung bringen, welche BOS alle in einer Großschadenslage rettungsdienstliche Aufgaben inne haben. D. h., wir wollen wissen, welche BOS bei der Behandlung von Patienten involviert sind, um z. B. nur ihnen wichtige Erkenntnisse über einen Giftstoff zukommen zu lassen. Dies stellt ein Beispiel einer Competency Question dar (vgl. Schritt 1 im vorhergehenden Abschnitt), welche die Ontologie in der Lage sein sollte, zu beantworten. Vorausgesetzt, es sind alle konkreten BOS am Einsatzort ihrem allgemeinen BOS-Typ zugeordnet, ließe sich obige CQ durch Aufzählung aller Individuen erreichen, die der Klasse *Hilfsorganisation* zugeordnet sind (vgl. Abb. 6.10).

Konkret handelt es sich hierbei um eine zusammengesetzte Reasoningaufgabe (vgl. hierzu [BCM+03; BCM+07]). In einem ersten Schritt muss mittels Klassifikation (*Classification*) das Hasse-Diagramm der Subsumtions-Relationen $\sqsubseteq_{\mathcal{T}}$ bezüglich einer gegebenen TBox \mathcal{T} berechnet werden. Somit erhält man im oberen Beispiel all die Klassen, die direkt per „is-a“-Relation zur Klasse *Hilfsorganisation* zugeordnet sind, und zusätzlich jene, die indirekt aufgrund entsprechender TBox-Axiome ebenfalls unter die gesuchte Klasse subsumiert werden. In einem zweiten Schritt werden über die Menge der Individuen diejenigen berechnet, die in eine der gesuchten Klassen fallen. Letztere Aufgabe ist ein Inferenzproblem bezüglich Zusicherungen in der ABox. Alle Individuen einer Klasse zu ermitteln, wird über *Retrieval* erreicht (siehe [BCM+03]). *Retrieval* bezeichnet die Berechnung der Menge $I_{\mathcal{A},\mathcal{T}}(C)$ von Individuennamen $a \in \mathcal{A}$, für die gilt: $\mathcal{A} \models_{\mathcal{T}} C(a)$. Vorausgesetzt, gegeben ist eine ABox \mathcal{A} , eine TBox \mathcal{T} und ein Konzept C .

In der Praxis übernehmen jedoch nicht nur Hilfsorganisationen wie das DRK oder der ASB rettungsdienstliche Aufgaben. Auch Feuerwehren und die Bundeswehr können hierfür in Frage kommen (vgl. Kapitel 2). Die Abfrage nur auf die Klasse *Hilfsorganisation* zu beschränken, ist daher ungeeignet. Eine Erweiterung auf die gesamte Klasse der BOS ist hingegen zu weitreichend, schließlich kann es auch Hilfsorganisationen oder Feuerwehren in einem Einsatz geben, die nicht in die Patientenbehandlung involviert sind. Die Definition

einer Klasse der Rettungsdienstorganisation ist hierfür eine Lösung, andere Lösungen sind ebenfalls denkbar.

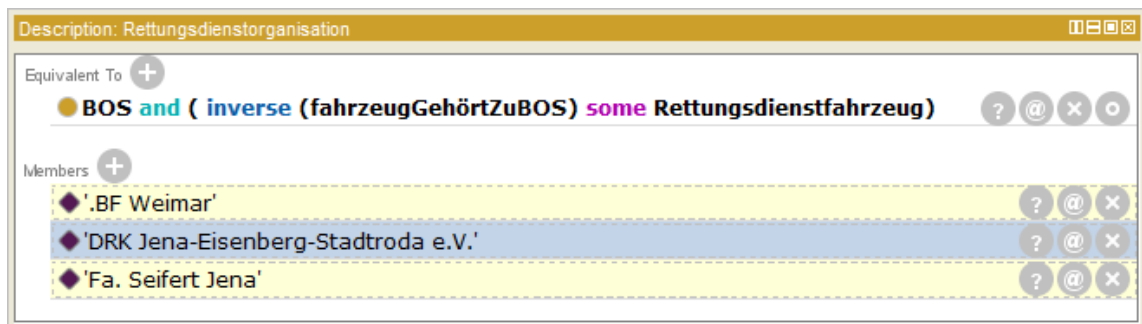


Abbildung 6.11: Beispiel eines Reasoningergebnisses (Protégé).

Folgendes TBox-Axiom definiert eine neue benannte Klasse auf Basis von BOS, die einen Fahrzeugtyp der Klasse Rettungsdienstfahrzeug am Einsatzort angemeldet haben.

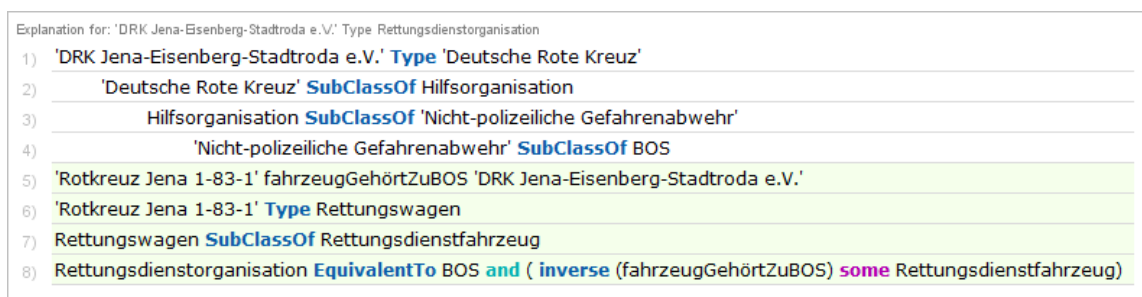
$$\text{Rettungsdienstorganisation} \doteq \text{BOS} \sqcap \exists \text{fahrzeugGehörtZuBOS}^- . \text{Rettungsdienstfahrzeug}.$$


Abbildung 6.12: Explanation Schlussfolgerungsergebnis (Protégé).

Abbildung 6.11 zeigt ein entsprechendes Reasoningergebnis, so wie es mittels Protégé auf Basis eines Reasoneres (hier *HermiT* OWL Reasoner, siehe [MSH09]) berechnet und angezeigt wird. Das Axiom selbst ist in der *Manchester Syntax* zu sehen, welche eine alternative Schreibweise der DL-Syntax darstellt. Das Beispiel stammt aus der BOS-Ontologie [BOS-O], in der einige Individuen als Beispiel abgelegt sind. Zu sehen sind drei Individuen, die als Members der Klasse Rettungsdienstorganisation ermittelt wurden.

Eine Begründung, warum z. B. das 'DRK Jena-Eisenberg-Stadtroda e.V.' mit zu den gesuchten BOS zählt, kann in Protégé durch einen Click auf das Fragezeichen-Icon angezeigt werden. Abbildung 6.12 listet alle Fakten auf, die in ihrer Gesamtheit zur Ableitung geführt haben.

Die Stärken logischen Schließens können bereits während der Modellierungsphase genutzt werden. Dadurch können ungewollte Beziehungen (Relationen) und Inkonsistenzen aufgedeckt werden. Automatische Schlussfolgerungen sind für folgende Anwendungsbereiche hilfreich:

1. Es können Modellierungsfehler z. B. anhand widersprüchlich definierte Konzepte der TBox erkannt werden.

2. Die Struktur der Konzepthierarchie kann mittels Schlussfolgerung bezüglich der Subklassenbeziehungen (*is-a-Relation*) zur besseren Darstellung explizit gemacht werden. Die Darstellung der Konzepthierarchie in Abbildung 6.19 ist ein Beispiel hierfür.
3. Schlussfolgerungen können ungewollte Redundanzen finden, in dem z. B. zwei äquivalente Konzepte erkannt werden, die unter Umständen so nicht beabsichtigt waren.

Der hier vorgeschlagene Ansatz geht weit über die Bereitstellung eines bloßen Glossars, wie er z. B. in PEINEL et al [PRW12b] vorgeschlagen wird, hinaus. So kann das Domänenwissen (Punkt 3.) als Schema genutzt werden, um ihm die Informationen einer taskorientierten Datenakquise während eines Einsatzes zuzuordnen. Was genau unter taskorientierter Datenakquise zu verstehen ist und wie diese in CASIE zur Gewinnung einer Lagerepräsentation genutzt werden kann, wird im Unterabschnitt 7.3 erläutert.

6.2 Dynamik / reaktives Systemverhalten

Unter Zuhilfenahme der oben erarbeiteten Konzepte und Definitionen werden im Folgenden die wesentlichen Aspekte des dynamischen Systemverhaltens von CASIE erläutert. Eine den Anforderungen entsprechende Konfiguration (siehe Abschnitt 6.3) des Systems (Checklisten, Rollendefinitionen, Domänenmodell in der Wissensbasis) soll hierbei als gegeben vorausgesetzt werden. Vorerst wird sich auf die reine Anwendungsdynamik im Einsatz konzentriert.

6.2.1 Zustandsübergänge einer Checkliste und deren Items

Wesentliche Bestandteile der Modellierung des dynamischen Verhaltens reaktiver Systeme sind deren Zustände, in denen sich das System befinden kann, und die als Transitionen bezeichneten Zustandsübergänge. Über ein Transitionssystem (englisch *state machine*) können die dynamischen Aspekte eines solchen Systems spezifiziert werden. Ziel ist es hierbei, die Zustände der Welt (speziell der Bearbeitungsstand der ICLs) durch interne Systemzustände in CASIE zu repräsentieren.

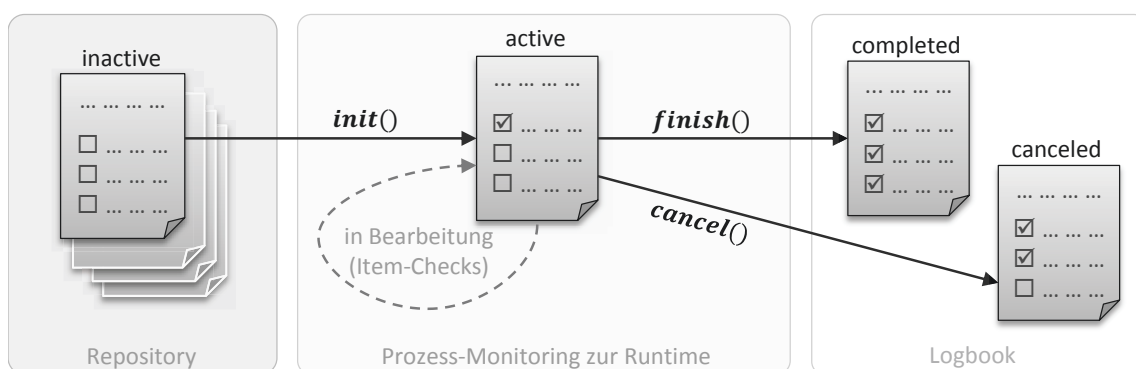


Abbildung 6.13: Diagramm der möglichen Zustandsübergänge einer ICL.

Für CASIE sind folgende drei Ereignisse definiert, die einen Zustandsübergang einer Checkliste bewirken:

- init()* – Die Initiierung einer Checkliste aus dem Repository zeigt an, dass die zugehörige SOP begonnen wurde. Der Zustand der ICL wechselt von inactive auf active. Das Ereignis *init()* wird entweder durch eine aktive Auswahl einer Checkliste durch den

Benutzer selbst ausgelöst, durch Bestätigung eines automatischen Vorschlages von Checklisteninstanzen, welche dem Nutzer dargeboten werden, oder im Rahmen des automatischen Aufbaus eines CLH durch CASIE selbst (siehe Abschnitt 6.2.2).

finish() – Alle Items wurden vom Nutzer bestätigt (siehe „Item-Zustände“ weiter unten), oder der Nutzer deklariert die Checkliste als „abgeschlossen“ (Status *completed*), obwohl noch offene Items vorhanden sind. Die Aufgabenstellung der zugehörigen SOP wurde den Gegebenheiten entsprechend erfüllt.

cancel() – Expliziter Abbruch der Bearbeitung durch den Benutzer, bevor alle Items bearbeitet/beachtet wurden, d. h., die Aufgabenstellung der zugehörigen SOP ist nicht mehr aktuell oder konnte nicht vollständig abgeschlossen werden. Gründe für einen Abbruch sind der aktuellen Einsatzlage geschuldet und spielen an dieser Stelle keine weitere Rolle.

In Abbildung 6.13 sind die möglichen Zustände und deren Übergänge durch die jeweiligen Ereignisse illustriert. Die Initiierung einer ICL ordnet eine entsprechende CL-Instanz einer Einsatzkraft zur Bearbeitung zu. Die ICL ist solange im Zustand *active* und wird somit in CASIE überwacht, wie keines der Ereignisse *finish()* bzw. *cancel()* eintritt.

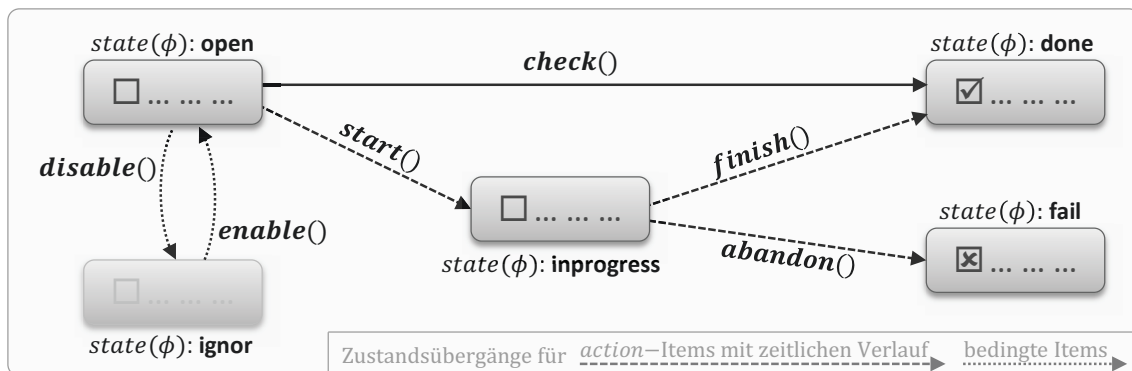


Abbildung 6.14: Diagramm der Zustandsübergänge eines Items.

Auf der Ebene der Items werden die fünf Zustände *open*, *inprogress*, *fail*, *done* und *ignore* unterschieden (vgl. Def. 5). Um den für *action*-Items typischen zeitlichen Verlauf repräsentieren zu können, sind in CASIE speziell für diesen Itemtyp die Zustände *inprogress* und *fail* definiert. Nachfolgend sind die möglichen Ereignisse und deren Bedeutungen aufgeführt, die zu einem Zustandsübergang eines Items ϕ führen können:

check() – Durch das Abhaken eines Items wird dessen erfolgreicher Abschluss, d. h. dessen Berücksichtigung quittiert. Der Zustand wechselt von *open* zu *done*. *check()* ist für alle drei Item-Typen (*note*, *query*, *action*) definiert.

start() – Ist der Nutzer an einem bestimmten Prozessschritt angelangt, für den ein *action*-Item in der zugehörigen Checkliste definiert wurde, so kann er den Start (Ereignis *start()*) dieses Schrittes bestätigen.

finish() – Der Nutzer bestätigt den erfolgreichen Abschluss eines zuvor gestarteten *action*-Items. Der Zustand des Items wechselt von *inprogress* zu *done*.

abandon() – Ist eine begonnene Aktion (repräsentiert durch den Zustand *inprogress*) nicht abschließbar und muss daher aufgegeben werden, so quittiert dies der Nutzer explizit. Das dadurch ausgelöste Ereignis *abandon()* setzt ϕ auf den Zustand *fail*.

disable() – Sind die unter $Pre(\phi)$ angegebenen Vorbedingungen eines bedingten Items ϕ nicht gegeben, so wird dieses auf den Status **ignor** gesetzt. Die Elemente aus $Pre(\phi)$ können entweder manuell durch die Einsatzkraft oder aber (je nach Type) automatisch durch CASIE evaluiert werden.

enable() – Wird für ein bedingtes Item ϕ im Status **ignor** (automatisch oder durch die Einsatzkraft) festgestellt, dass im Verlauf eines Einsatzes alle Vorbedingungen erfüllt sind, wird ϕ wieder auf den Status **open** gesetzt.

Die Abbildung 6.14 zeigt alle definierten Ereignisse, die einen Zustandsübergang eines Items bewirken können. Die Rechtecke repräsentieren alle möglichen Zustände eines Items, während die Kanten die Zustandsübergänge darstellen. Zusätzlich zu dem bereits beschriebenen Systemverhalten werden die Informationen, die mittels eines *query*-Items vom Nutzer abgefragt werden können, bzw. die bei der Itemdefinition zuvor annotierten Effekte mit in die lokale Lagerepräsentation durch entsprechende ABox-Einträge in die \mathcal{KB} aufgenommen. Die Möglichkeit, bedingte Items (siehe S. 64) zu definieren (konfigurieren) trägt der Beobachtung Rechnung, dass in vielen Einsätzen eine Vielzahl von zuvor bedachten Items keine Rolle mehr spielen. Diese Items werden dem Nutzer zwar zu Beginn präsentiert, können jedoch manuell oder automatisch ausgeblendet werden, wodurch eine übersichtlichere Darstellung komplexer CLH erreicht werden kann.

Event	Effekt(e)/Systemverhalten
<i>check()</i>	Statuswechsel zu done und alle Effekte werden als Assertions in die ABox geschrieben.
<i>start()</i>	Statuswechsel zu inprogress und entsprechender Logbucheintrag.
<i>abandon()</i>	Statuswechsel zu fail und Logbucheintrag und Löschen der Aufgabe aus der Liste der laufenden Tasks.
<i>finish()</i>	Statuswechsel zu done und Löschen der Aufgabe aus der Liste der laufenden Tasks.
<i>disable()</i>	Statuswechsel zu ignor , Item wird zur Überwachungsliste hinzugefügt.
<i>enable()</i>	Statuswechsel zu open und Löschen des Items aus der Überwachungsliste.

Tabelle 6.2: Mögliche Item-Events und das resultierende Systemverhalten.

Die Tabelle 6.2 fasst die Zustandsübergänge und das jeweils daraus resultierende Systemverhalten zusammen.

6.2.2 (Intelligentes) reaktives Systemverhalten

In Abhängigkeit von der sich ändernden Einsatzsituation müssen unterschiedliche Aufgaben durch die Einsatzkräfte erfüllt werden. Ein CL-Assistenzsystem sollte in seinem Verhalten die Dynamik einer Einsatzlage mitberücksichtigen. CASIE erfüllt diese Forderung dahingehend, dass aus den durch die ICL-Bearbeitung gewonnenen Informationen eine entsprechende ICL-Anpassung vorgenommen wird.

Die Abbildung 6.15 illustriert die in Abschnitt 6.1.2 eingeführten Betriebsmodi SUM und MUM. Während im SUM nur lokale, auf den jeweiligen Runtime-Client beruhende Events und Informationen berücksichtigt werden, können im MUM die (externen) Events und Lageinformationen (z. B. die Anzahl der Verletzten oder das erfolgreiche Einrichten

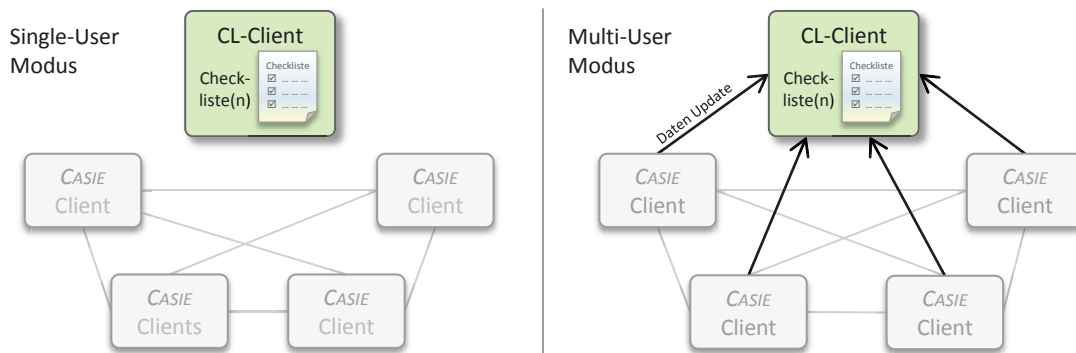


Abbildung 6.15: Singel-User- vs. Multi-User-Modus.

eines Behandlungsplatzes) anderer Nutzer zusammengeführt und gegebenenfalls mitberücksichtigt werden.

Das intelligente reaktive Systemverhalten von CASIE wird hierbei durch das Zusammenspiel zweier grundlegender Mechanismen erreicht: zum einen durch das Auswerten der Interaktionen zwischen der Einsatzkraft und dem CL-Client sowie der Statuswechsel und/oder KB-Updates andere CL-Clients im Netzwerk; und zum anderen durch eine darauf folgende interne Überprüfungsroutine, welche die durch die Nutzeraktion gewonnenen Informationen im System weiterverarbeitet und ggf. mit der Anpassung bzw. der Darbietung der ICLs reagiert.

In den Definitionen der ICLs (Def. 4, S. 52) und der Items (Def. 5, S. 52) sind durch $\mathcal{K}(cl)$ bzw. $Pre(\phi)$ zwei Stellen zur Angabe von Anwendungsbedingungen (*Constraints*) vorgesehen. Bei den Bedingungen handelt es sich um eine Menge formaler bzw. natürlichsprachlich repräsentierter (Anwendungs-)Kriterien, die, wie folgt, in CASIE Anwendung finden.

1. Bei manueller Auswahl einer Checkliste durch den Benutzer und deren Initialisierung kann davon ausgegangen werden, dass die Anwendungskriterien der ausgewählten Checkliste in der aktuellen Einsatzlage gelten. Diese Informationen können als Fakten über die Einsatzlage zur Lagerepräsentation hinzugefügt werden.
2. Wird im Zuge eines Überprüfungszyklus der Anwendungskriterien aller noch nicht initiierten Checklisten festgestellt, dass die Kriterien für eine ICL erfüllt sind, so wird die entsprechende ICL der Einsatzkraft als „zu beachtend“ automatisch vorgeschlagen.

Im Folgenden soll das Systemverhalten aus der Perspektive einer einzelnen Führungs- bzw. Einsatzkraft (im Folgenden auch Nutzer genannt) beschrieben werden. Mit dem SUM und dem MUM werden jedoch beide Facetten der Anwendung gemeinsam behandelt. Die Abbildung 6.16 (nächste Seite) skizziert hierzu in Form eines Aktivitätsdiagramms das reaktive Systemverhalten von CASIE. Die Abbildung stellt die dynamischen Aspekte des modellierten Systems dar. Die einzelnen Aktivitäten sind in die logischen Bereiche *Anmelden*, *Auswahl & CL-Hierarchie*, *Bearbeitung/Update* und *Abmelden/Dokumentation* zusammengefasst dargestellt. Die grau (hell) hinterlegten Aktivitäten stehen für manuelle Interaktionen des Nutzers mit dem System, während die rot (dunklen) hinterlegten (voll-) automatisch durch CASIE realisiert werden. Die schraffiert hinterlegten Aktivitäten stehen für eine Mischform beider, d. h. für eine manuelle Aktion des Nutzers mit semiautomatischer Assistenz des Systems.

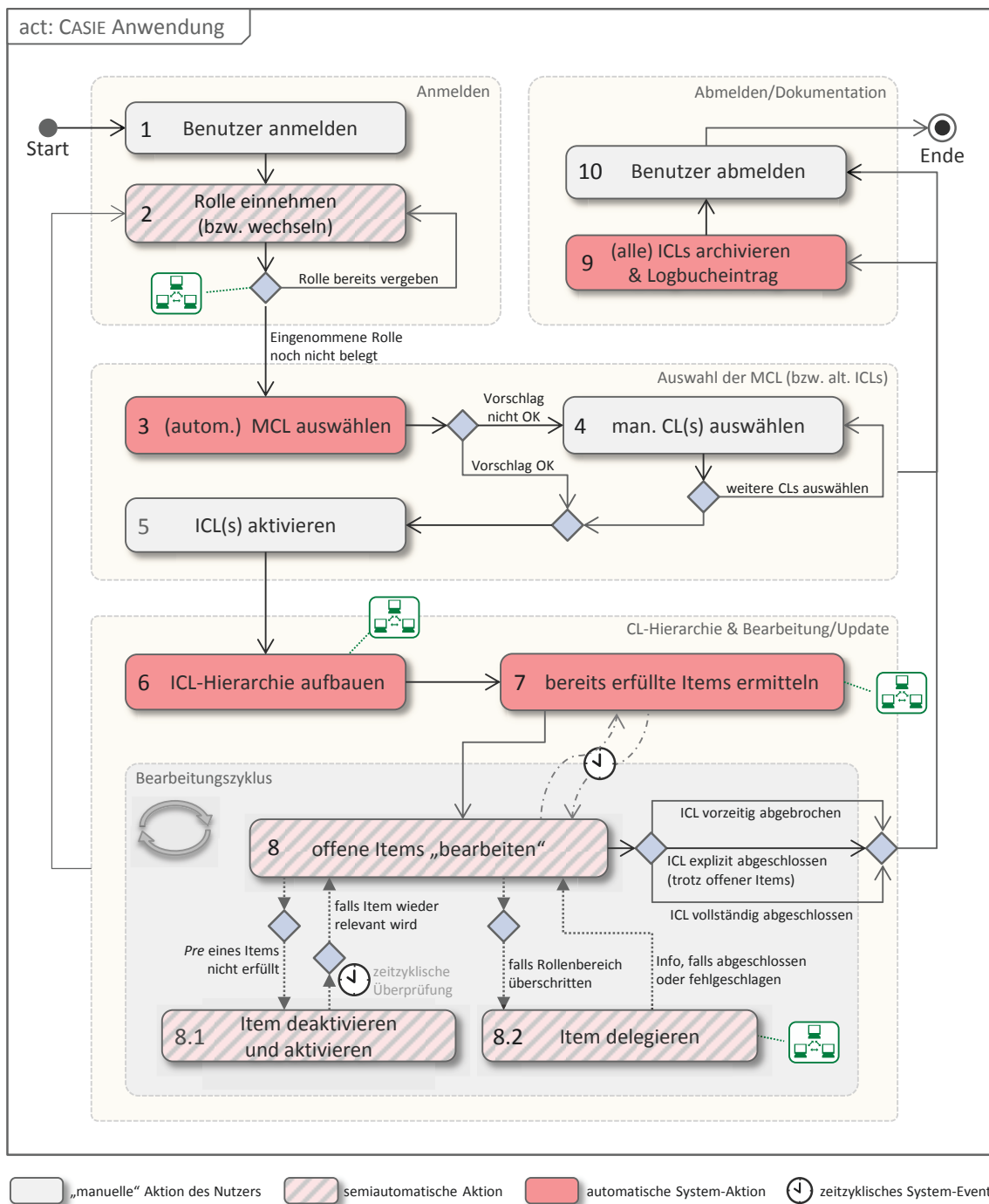



Abbildung 6.16: Aktivitätsdiagramm des reaktiven Systemverhaltens von CASIE.

Anmelden

Zu Beginn (1) muss sich der Nutzer am CL-Client mit seinem Namen anmelden. Im SUM spielt der Name zwar noch keine Rolle, jedoch kann im MUM dann nach Bedarf eine Liste aller aktiven CL-Clients mit deren Nutzernamen erstellt werden. Danach wählt der Nutzer diejenige Rolle aus (2), die er übernehmen möchte. CASIE kann dem Nutzer hierfür eine Liste aller Rollen präsentieren, für die entsprechende ICLs hinterlegt sind. Die Liste lässt sich auf Basis aller ICL-Definitionen des Repository leicht ermitteln. Diese semiautoma-

tische Systemunterstützung ist in der Abbildung 6.16 durch die Schraffierung des Aktivitätsknotens gekennzeichnet. Im Prinzip sollte der Nutzer jede beliebige Rolle frei wählen können. Als Assistenz kann jedoch in der MUM-Variante auf Basis der Rollendefinitionen und deren Kardinalitäten (*nc*) eine ungewollt redundante Rollenübernahme erkannt und dem Nutzer angezeigt werden. Das Netzwerksymbol  in der Abbildung 6.16 zeigt an, dass ein Mehrwert nur dann zur Verfügung steht, wenn eine Verbindung mehrerer CL-Clients (bzw. CASIE-Systeme) untereinander besteht.

Auswahl der MCL bzw. alternativer ICLs

Nach erfolgreicher Rollenübernahme wird die für die Rolle hinterlegte MCL automatisch aus dem Repository ermittelt (3). Existiert eine für den Nutzer passende, personalisierte MCL-Variante (siehe Abschnitt 6.3), so wird diese anstatt der allgemeinen MCL ausgewählt. Zur weiteren Detaillierung können auch alle dazugehörigen Sub-CLs gelistet werden. Sind zu den ICLs entsprechende formale Vorbedingungen definiert, so prüft CASIE automatisch, ob diese mit dem momentanen Wissenstand der Wissensbasis (KB) jeweils als gültig ableitbar sind ($\Sigma \models_{\mathcal{T}} \mathcal{K}$). Falls die Anwendungskriterien einer ICL bzw. MCL nicht erfüllt sind, so kann dies dem Nutzer entsprechend kenntlich gemacht werden. Ist der Nutzer mit der angebotenen Auswahl nicht zufrieden, wählt er alternativ eine beliebige Menge von ICLs, passend zu oder abweichend von seiner Rolle aus (5). Ist der Nutzer hingegen mit der MCL einverstanden, was als Regelfall zu erwarten ist, kann er diesen Vorschlag übernehmen und somit den Bearbeitungsbeginn der MCL aktivieren (4). Der Status der jeweils ausgewählten MCL (bzw. der alternativ gewählten ICLs) wechselt hierbei von *inactive* auf *active*. Der CL-Manager überwacht ab nun die Bearbeitung der ICLs, bis diese abgeschlossen bzw. explizit abgebrochen werden. Hierzu werden alle aktiven ICLs im CL-Manager in eine sog. *Watch-List* (Beobachtungsliste) aufgenommen.

Initialer Aufbau der CL-Hierarchie & -Bearbeitung

Mit der Aktivierung einer MCL erfolgt der Aufbau der entsprechenden CL-Hierarchie (6). Die CL-Hierarchie ist als eine automatisch generierte Sicht einer Menge verknüpfter ICLs zu verstehen, wodurch ein Wurzelbaum entsteht, dessen Wurzel eine MCL darstellt. Abstrakte Items werden dabei nur soweit durch Sub-CLs konkretisiert, wie es die Auswertung der Anwendungsbedingungen gestattet. Der Einfachheit halber wird davon ausgegangen, dass der Nutzer nur die für ihn passende MCL auswählt. Für eventuelle alternative ICLs gilt jeweils die analoge Vorgehensweise.

Um eine initiale CL-Hierarchie zu berechnen, werden in einem ersten Schritt alle zur Rolle passenden ICLs aus dem Repository gelesen. Danach wird die in jedem Sub-CL-Verweis (falls vorhanden) angegebene ICL mit dem abstrakten Item verknüpft. Die so entstehende Datenstruktur muss bei Bedarf zusätzlich die verschiedenen alternativen Sub-CLs, die für ein abstraktes Item angegeben sein können, mitbeachten. Alle so ermittelten Sub-CLs, die exklusiv (ohne Alternativen) mit einem Item verknüpft sind, werden ebenfalls mit in die *Watch-List* aufgenommen und auf Status *active* gesetzt.

Zu jedem Satz an CL-Alternativen wird dann überprüft, welche Checklisten ein abstraktes Item in der gegenwärtigen Lage konkretisiert, d. h., welche Ausprägung die CL-Hierarchie in der jeweiligen Lage konkret einnimmt. Analog zur Überprüfung der Anwendungskriterien für ICLs wird hierzu in jedem Entscheidungsschritt δ (vgl. Abb. 5.6, S. 61) eine passende Sub-CLs gewählt. Die Konkretisierung kann entweder automatisch oder auf Basis der ausgewerteten Anwendungsbedingungen getroffen werden – vorausgesetzt, die ICLs sind mit entsprechenden formalen Anwendungskriterien \mathcal{K} annotiert worden und es lässt

sich aus der Wissensbasis schließen, dass die Anwendungsbedingungen gelten ($\Sigma \models_{\mathcal{T}} \mathcal{K}$). Oder aber der Nutzer trifft die Entscheidung selbst, indem er eine seiner Meinung nach passende Sub-CL aus der Liste der möglichen Alternativen auswählt.

In einem weiteren Schritt wird automatisch für jedes Item aller aktiven ICLs überprüft, ob das Item bereits erfüllt wurde (7), d. h., ob anhand der möglicherweise bereits vorliegenden Informationen zur Einsatzlage deren Effektmenge $Eff(\phi)$ (falls angegeben) als erfüllt abgeleitet werden kann oder nicht. Falls ja, so wird dessen Status automatisch auf **done** gesetzt. Eine entsprechende visuelle Hervorhebung kann dies dem Nutzer kenntlich machen.

Die CL-Bearbeitung (8) stellt die Hauptaktivität des Anwenders dar. Dieser Bearbeitungszyklus besteht im Wesentlichen aus dem Abhaken der einzelnen Items. Da der Verlauf dieser Aktivität von der jeweiligen Einsatzlage und den konkreten ICLs abhängt, lässt sich dieser Teil im Aktivitätsdiagramm nur abstrakt darstellen. Neben dem allgemeinen Abhaken von Items lassen sich zwei mögliche Unteraktionen nennen. Zum einen kann CASIE die Gültigkeit von Item-Vorbedingungen automatisch überprüfen. Lässt sich z. B. aus der KB ableiten, dass kein Gefahrgut beteiligt ist, so können all diejenigen bedingten Items ausgeblendet werden (8.1), die nur für den Fall einer Gefahrgutbeteiligung beachtet werden müssen. Tritt während des Einsatzes Gefahrgut auf und wird dies der KB bekannt gemacht, so kann CASIE automatisch die entsprechenden Items wieder in der Liste anzeigen. Bei der zweiten Unteraktion handelt es sich um eine Delegation (8.2) eines Items zu einem anderen Nutzer, da dessen Bearbeitungsverantwortung in einer untergeordneten Rolle liegt. Diese Funktionalität steht jedoch nur zur Verfügung, falls ein weiterer CL-Client im Netz verfügbar ist, an dem sich der entsprechende Rolleninhaber angemeldet hat. Zusätzlich müssen diese Verknüpfungspunkte zuvor bei der Konfiguration des Systems explizit angegeben werden (siehe Abschnitt 6.3).

Abmelden/Dokumentation

Nachdem der Nutzer alle ICLs abgeschlossen hat (Event *finish()*) bzw. die Bearbeitung explizit abgebrochen hat (Event *cancel()*), werden die ICLs automatisch mittels der Logbook-Komponente archiviert (9). Das explizite Abmelden (10) des Nutzers am CL-Client archiviert alle noch offenen (Status **aktive**) ICLs und versetzt den CL-Client in den Stand-by-Modus. Ein Abmelden ist prinzipiell zu jeder Zeit möglich, ohne dabei die Informationen der bis dato Bearbeiteten ICLs zu verlieren. Die Vorteile, die solch eine automatische Dokumentation für die BOS hat, werden im Abschnitt 7.3 erläutert.

6.3 Konfiguration/Wartung

Um CASIE für den Einsatz in einem Anwendungsbereich zu konfigurieren, bedarf es zwei sich teils ergänzender Modellierungen. Zum einen muss ein entsprechendes terminologisches Domänenmodell erstellt und zum anderen muss das prozedurale Wissen der SOPs in Form der ICLs erfasst werden. Da jede Organisation ihre eigenen spezifischen Anforderungen und Ausprägungen von Arbeitsprozessen aufweist, wird jede Organisation das System mit ihren eigenen Checklisten füllen müssen. Während die Benutzung von CASIE kein spezielles Know-how von einer Einsatzkraft abverlangt, ist bei der Konfiguration des Systems, der Ontologierstellung und der Verknüpfung der Checklisten und der Merkmale zur Situationsbeschreibung ein tiefgreifendes Fachwissen erforderlich. Die jeweils regional ausgeprägte BOS-Ontologie kann daher nur mit Hilfe eines Ontologieexperten in enger Kooperation mit den Anwendern erstellt werden.

Im Folgenden werden die Erstellung einer Domänenontologie erläutert und Hinweise für den Aufbau eines CL-Repository und die Personalisierung von Checklisten gegeben.

6.3.1 Entwicklung einer BOS-Ontologie

Dieser Abschnitt beschreibt die Entwicklung einer BOS-Ontologie als beispielhaftes Domänenmodell und gibt am Ende ein Beispiel dafür, inwieweit die Mächtigkeit der Reasoning-funktionalität von einer geeigneten Modellierung abhängig ist.

Formale Ontologien sollen einen Konsens der Begriffsverwendung einer speziellen Domäne repräsentieren. Dieser Thematik widmet sich der Forschungsbereich des *Ontologie Engineering* aus dem bis heute eine Vielzahl von Vorschlägen, Techniken und Methoden zur Ontologieerstellung hervorgegangen sind. So gibt z. B. FERNÁNDEZ-LÓPEZ in [Fer99] einen kurzen Überblick über verschiedene Methoden, während GÓMEZ-PÉREZ et al. in ihrem Buch „Ontological Engineering“ [GFC04] einen umfassenden Einstieg in das Fachgebiet geben. Ohne auf die Vor- und Nachteile der unterschiedlichen Herangehensweisen genauer einzugehen, werden in dieser Arbeit die Einzelschritte der Ontologiemodellierung beispielhaft anhand der von NOY & MCGUINNESS in [NM01] vorgeschlagenen *Knowledge-Engineering Methodology* vorgestellt. Die Autoren schlagen vor, den Entwicklungszyklus in sieben Schritte zu unterteilen (siehe Abbildung 6.17).

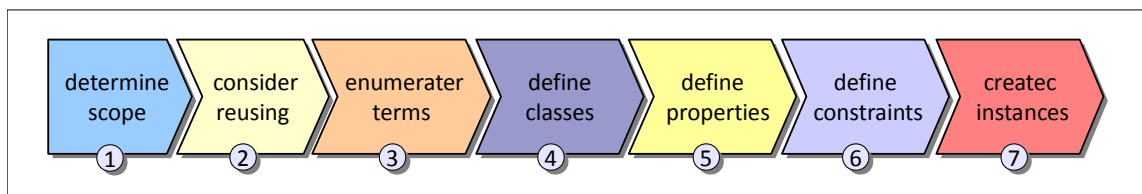


Abbildung 6.17: Schritte der Ontologieentwicklung nach MCGUINNESS et al. [NM01].

Während der Evaluierungs- und Anpassungsphase der Ontologie werden je nach Bedarf einzelne Schritte erneut durchlaufen. Es ist bei der Ontologieentwicklung ratsam, einige wichtige Grundsätze nicht aus dem Blick zu verlieren. NOY und MCGUINNESS (siehe [NM01]) geben dem Entwickler drei Leitsätze mit auf den Weg: (1) „*There is no one correct way to model a domain – there are always viable alternatives. The best solution [...] depends on the application that you have in mind.*“ (2) „*Ontology development is necessarily an iterative process.*“ (3) „*Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest. These are most likely to be nouns (objects) or verbs (relationships) in sentences that describe your domain.*“

Die Erstellung einer Ontologie ist als ein kreativer Prozess zu verstehen. Zwei unterschiedliche Modellierer werden aufgrund ihrer unterschiedlichen Fähigkeiten und Sichtweisen auch unterschiedliche Ontologien für ein und dieselbe Domäne entwickeln. Die Güte der Ergebnisse lässt sich nur durch den Grad der erfolgreichen Anwendbarkeit in der zu erzielenden Anwendung einschätzen („*the proof is in the pudding*“ [GF95]). Die im Rahmen dieser Arbeit modellierte BOS-Ontologie (siehe [BOS-O]) ist daher als eine mögliche Beispielmmodellierung der BOS-Domäne anzusehen.

Protégé als Tool zur Ontologieentwicklung

Eine Schlüsselrolle für die erfolgreiche Modellierung einer Ontologie nehmen spezielle Software-Tools ein. Obwohl sich die Idee des Semantic Web wachsender Beliebtheit erfreut, existieren bisher nur wenige Programme zur Unterstützung einer Ontologieentwicklung.

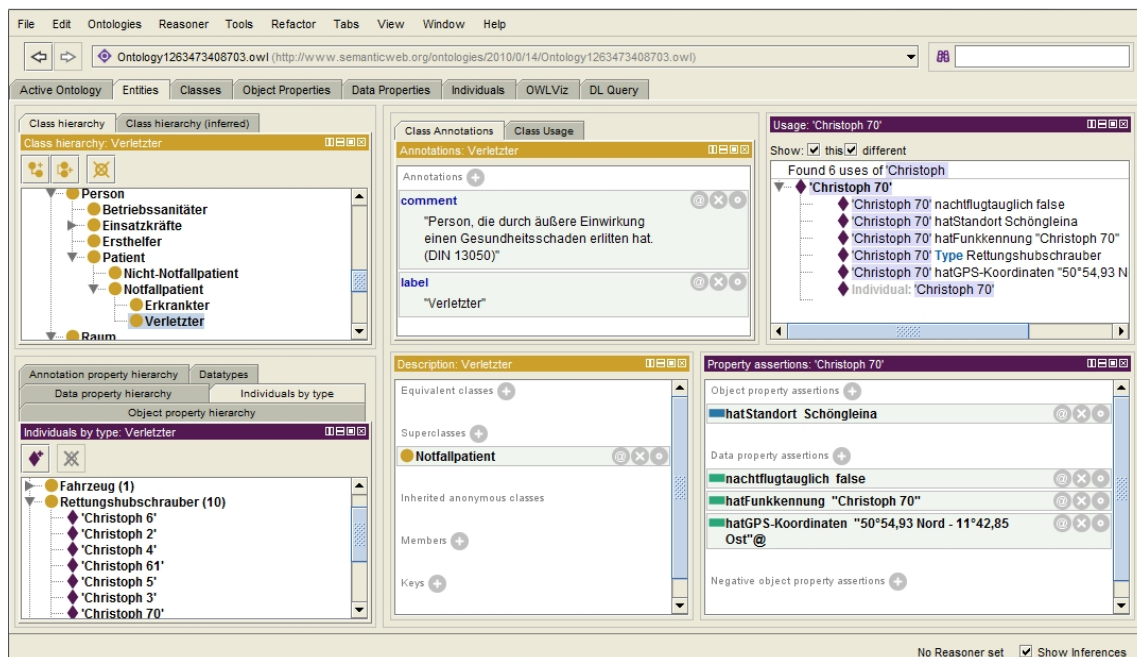


Abbildung 6.18: Das GUI von Protégé (4.*) mit geladener BOS-Ontologie [BOS-O].

Speziell für die junge Ontologiesprache OWL2 stehen dem Entwickler momentan wenige Tools zur Verfügung. Hinsichtlich der Entwicklungsumgebungen nimmt das *Protégé*-Projekt³ eine führende Stellung ein. Nicht zuletzt sind die Editoren *OilEd* [BHG+01] von der University Manchester und *OntoStudio* (ehemals *OntoEdit*) der Firma *semafora systems* zwei weitere Vertreter für Ontologieeditoren.

Bei *Protégé* handelt es sich um ein Open-Source-Projekt. Das mit einem *Graphical User Interface (GUI)* ausgestattete mächtige Entwicklungstool (Abb. 6.18) gilt momentan als de facto Standard zur Ontologieentwicklung mit OWL2 und stellt darüber hinaus eine standardisierte Schnittstelle zu verschiedenen Inferenzmaschinen bereit. *Protégé* ist in Java implementiert und somit plattformunabhängig einsetzbar. Weiterhin ist eine Ontologie-Validierung schon während der Erstellung möglich, wodurch syntaktische Modellierungsfehler bereits zu Beginn erkannt und behoben werden können.

Die Benutzeroberfläche bietet eine Reihe von Registerkarten, mit deren Hilfe die Erstellung bzw. das Editieren von Ontologien ermöglicht wird. Hierbei sind die wichtigsten Registerkarten *OWL-Classes*, *Properties* und *Individuals*. Für den Aufbau einer Wissensbasis werden über den *OWL-Classes*-Tab die verschiedenen benötigten Klassen definiert. Über den *Properties*-Tab können die Eigenschaften, die eine Klasse bzw. eine spätere Instanz der Klasse haben soll, festgelegt werden. Zur Definition des Werte- (*Range*) und Definitionsbereichs (*Domain*) einer Property werden die zuvor eingeführten Klassendefinitionen (bzw. die zur Verfügung stehenden Datentypen) genutzt. Auf der Reiterkarte *Individuals* können im Anschluss beliebig viele Instanzen einer Klasse angelegt und die durch die Properties zuvor bestimmten möglichen Attributwerte festgelegt werden. Es wird an dieser Stelle auch von *Wissensakquise* gesprochen, da das Anlegen von Instanzen eine konkrete Repräsentation von Individuen eines Wissensbereiches anhand einer zuvor definierten Ontologie entspricht.

Ob eine Ontologie mit Protégé 3.* oder mit Protégé 4.* entwickelt wird, hängt ganz von den jeweiligen Anforderungen ihrer Verwendung ab (vgl. WANG [Wan06]). Da im Rahmen

³ Homepage des Protégé Projekts: <http://www.protege.stanford.edu>. (abgerufen am 01.12.2013)

dieser Arbeit die Modellierung der korrekten Zusammenhänge der Begrifflichkeiten mehr im Vordergrund steht als die Akquise umfangreicher Datenmengen, wurde sich für die OWL-Variante von Protégé entschieden.

Die einzelnen Schritte der BOS-Ontologieentwicklung im Detail

Im Folgenden werden die einzelnen Schritte der Ontologieentwicklung gemäß der vorgeschlagenen Herangehensweise von NOY & MCGUINNESS [NM01] Schritt für Schritt betrachtet. Die Entwicklungsschritte werden beispielhaft anhand der BOS-Ontologie illustriert. Vereinzelt werden dabei einige spezifische Modellierungsdetails aufgeführt, die die eingesetzten Werkzeuge bzw. die formale Ontologiesprache betreffen.

Schritt 1: Festlegen des Wissensbereiches

Im ersten Schritt müssen Fragen geklärt werden wie: Welches Anwendungsgebiet soll modelliert werden und was soll *nicht* modelliert werden? Wie soll die Ontologie genutzt werden? D. h., welche Fragen über die Domäne soll die Ontologie beantworten? Und nicht zuletzt: Wer wird die Ontologie nutzen und administrieren? Diese Fragestellungen sollen folgend genauer betrachtet werden.

Welches Anwendungsgebiet soll modelliert werden? Die Beantwortung dieser Frage ist so einfach wie schwierig zugleich. Die Herausforderung besteht darin, einen Mittelweg zwischen einer einfachen Ontologie und ihrer gleichzeitigen Ausdruckskraft und Vollständigkeit bezüglich ihres geplanten Einsatzgebietes zu finden. Im Rahmen dieser Arbeit wurde sich bei der Umsetzung einer beispielhaften BOS-Ontologie auf die Modellierung der wesentlichen Konzepte einer Großschadenslage und einem Satz beispielhafter Relationen konzentriert.

Welche Fragen über die Domäne soll die Ontologie beantworten? Das problemorientierte Erarbeiten sogenannter *Competency Questions* (CQ) dient dazu, den Einsatzbereich der Ontologie zu bestimmen (vgl. [GF95; UG96]). Um eine Ontologie in einer realen Anwendung hilfreich nutzen zu können, sollte sie einen Reasoner dazu befähigen, die in den CQ informell formulierten Anforderungen zu erfüllen. D. h., die CQs sollen deutlich machen, welche Fragen durch den Einsatz der Ontologie mit einem Reasoner durch die Anwendung überhaupt beantwortet werden können. Die CQs stellen eine Orientierung für die weiteren Schritte der Ontologieentwicklung und des Anwendungssystems selbst dar. Es ist davon auszugehen, dass sich die CQs über den Entwicklungszyklus der Ontologie ändern werden und daher die Ontologie gemäß der neuen Anforderungen angepasst werden muss. Eine beispielhafte Menge von informellen Competency Questions, die sich in CASIE ergeben, lautet wie folgt:

- Welche Rollen sind momentan durch wen belegt, und welche SOPs sind aktuell durch wen in Bearbeitung?
- Gibt es Rollenkonflikte durch irrtümlich mehrfach belegte Rollen?
- Um welches Schadensereignis handelt es sich?
- Gibt es Redundanzen in der Patientenerfassung?
- Welche Organisationen mit Rettungsdienstaufgaben sind im Einsatz involviert?

Wer wird die Ontologie nutzen und wie? Dieser Aspekt wird (nach MCGUINNESS [NM01]) oft vernachlässigt, obwohl dies eine wichtige Fragestellung hinsichtlich des Praxiserfolgs einer ontologiebasierten Anwendung ist. Wie in Abschnitt 5.2.6 gezeigt, soll die BOS-Ontologie in CASIE zum einen als Diskussionsgrundlage für die Modellierung relevanter Aspekte einer Großschadenslage dienen, und zwar bereits im Vorfeld einer solchen. In einem zweiten Schritt dient die Ontologie einer Wissensbasis als Terminologie, wodurch logische Schlussfolgerungen über eine konkrete Einsatzsituation und über Aktionen in CASIE möglich sind.

Wer wird die Ontologie bei Bedarf anpassen? Die Frage nach der Anpassung der Ontologie ist ein zweischneidiges Schwert. Zum einen liegt der Vorteil der Nutzung einer Wissensbasis gerade darin, dass die Modellierung von Wissen und die Einordnung von Fakten der Anwendungsdomäne flexibler gehandhabt werden kann. So kann die Ontologie bei Bedarf angepasst werden, ohne dass eine oft aufwändige Änderung der Logik des Anwendungsprogramms erforderlich wird. Auf der anderen Seite bedarf es eines tiefgreifenden Grundverständnisses der einer Ontologie zugrunde liegenden Theorie sowie der Syntax und der formalen Semantik der verwendeten Ontologiesprache. Es lässt sich der Standpunkt vertreten, dass die Ontologie nach ihrer Erstellung nicht mehr geändert werden kann und soll.

Das SPEEDUP-Projekt hat gezeigt, dass ein großer Anpassungsbedarf solcher Systeme an die lokalen Gegebenheiten besteht, was nicht selten den Eingriff in die grundlegende Modellierung zur Folge hat. Ziel von CASIE ist jedoch, ein möglichst flexibles Rahmenwerk zu erarbeiten, das eine Anpassung der Ontologie an sich ändernde Gegebenheiten unterstützt. Eine Anpassung der Ontologie wird immer durch einen Ontologieexperten erfolgen müssen.

Schritt 2: Integration bereits bestehender Ontologien prüfen

Existiert bereits für einen Teilbereich eine formale Ontologie, sei es eine selbst entwickelte oder eine „fremde“ Ontologie, so ist es ratsam, abzuklären, ob diese Ontologie genutzt werden kann. Lässt sich eine bereits vorhandene Ontologie integrieren, hilft dies, die Entwicklungsarbeit zu erleichtern. Darüber hinaus kann es Vorteile haben, sich auf eine bereits etablierte (Quasi-)Standardontologie zu beziehen.

Zwei Beispiele sollen dies verdeutlichen. Angenommen es soll eine Ontologie für einen Buchshop erstellt werden, so bietet es sich an, die bereits existierende Ontologie des *Dublin Core Metadata Standards* mit bei der Ontologieentwicklung zu berücksichtigen. Soll hingegen eine Ontologie zur Beschreibung von Web-Services erstellt werden, so sollte die Nutzung der bereits dafür etablierten *OWL-S*-Ontologie in Betracht gezogen werden [MBM+07].

Es ist jedoch eine Konsequenz aus Schritt 1, dass es recht unwahrscheinlich ist, eine frei verfügbare Ontologie zu finden, die sich auch mit der gleichen Anwendungsdomäne und vor allem mit den gleichen CQs deckt. Je spezieller das Anwendungsgebiet, desto spezieller wird auch eine Ontologie zu diesem Gebiet ausfallen.

Zwar wurden in den letzten Jahren einige Ontologien im Bereich Krisenmanagement entwickelt (vgl. hierzu z. B. die Auflistung von LIU et al. [LSB13]), doch sind diese nicht auf die deutschen BOS übertragbar. Obwohl die modellierten Domänen der in Kapitel 3 vorgestellten Forschungsarbeiten den BOS-Bereich ähneln, sind deren Unterschiede jedoch so groß, dass eine Nutzung für CASIE nicht möglich ist. Dies liegt nicht zuletzt an den deutlichen länderspezifischen Unterschieden im *Rescue Management* und den damit einhergehenden Konzepten. Selbst die deutschen Projekte, in denen Ontologien im BOS-Bereich Teil der

Forschungsarbeit waren (vgl. z. B. SHARE [KPS+08]), decken sich nicht mit den Anforderungen in CASIE.

Schritt 3: Zusammentragen der wichtigsten Konzepte

Ist die Domäne abgesteckt, werden im nächsten Schritt die wichtigsten Begriffe des Anwendungsbereiches zusammengetragen, welche primär Kandidaten für die zu modellierenden Konzepte darstellen. Ob die Aufnahme eines Konzeptes sinnvoll ist, hängt direkt von der konkreten Verwendung der Ontologie in der entsprechenden Anwendung (siehe *Competency Questions*) ab.

Natürlich sagt eine bloße Aufzählung der wichtigsten Begriffe noch nichts über den Zusammenhang und die Eigenheiten der Domäne aus. Hierfür bedarf es eines ausreichenden Domänenverständnisses und einer weiterführenden Modellierung in den folgenden Schritten. Auch sind nicht alle der ermittelten Begriffe zugleich sinnvolle Kandidaten für Klassendefinitionen. So werden z. B. substantivierte Tätigkeitswörter zumeist als Relationen umgesetzt (siehe Schritt 5).

Nachfolgend werden beispielhaft einige relevante Begriffe der BOS-Domäne aufgeführt. Für eine komplette Auflistung aller potentiell relevanten Begriffe und Konzepte sei auf die BOS-Ontologie verwiesen.

- *Spezielle Pläne*: Alarmierungsplan, Ausrückeordnung, Objektplan, Hydrantenplan, Evakuierungsplan und Lageplan.
- *Organisationen*: Hilfsorganisation, Polizei, Feuerwehr, Berufsfeuerwehr, Werkfeuerwehr, Rettungsdienst und Rettungsleitstelle sowie konkrete BOS: DRK, THW, JUH, ASB, DLRG und BRK.
- *Örtlichkeiten*: Einsatzort, Standort, GPS-Koordinaten, räumliche Einsatzabschnitte, Anfahrtswege, Objektnamen, Behandlungsplatz, Hubschrauberlandeplatz und Bereitstellungsraum.
- *Organisationsstruktur und Rollen*: Technische Einsatzleitung, Stab, Krisenstab, AAO, EL, Notarzt, OrgL, Abschnittsleiter, Einsatzleiter, Gruppenführer, Zugführer, Truppführer, Rettungsassistent, Rettungshelfer, Rettungssanitäter und Polizeiführer.
- *Tätigkeiten*: Alarmieren, Aufbauen, Anfordern, Melden der Lage, Treffen von Entscheidungen, Befehlen, Dokumentieren, Sichten, Absprechen, Datenerfassung, Behandlung, Bergung, Rettung, Berichten, Angriff, Ermitteln, Aggregieren, Zuweisen, Transportieren und Informieren.

Für die BOS-Ontologie konnte für die Definition einiger Begriffe die DIN-Norm 13050 „Rettungswesen – Begriffe“ herangezogen werden. Insgesamt sind derzeit ca. 350 Konzepte der Domäne in der Ontologie erfasst. Alle Konzepte wurden unter Zuhilfenahme der Annotation-Property `label` benannt und mit einem erklärenden Kommentar über die Annotation-Property (`comment`) annotiert. Falls vorhanden, wurde zu jedem Konzept dessen geläufige Abkürzung mit erfasst, z. B. LNA für Leitenden Notarzt oder GrFü für einen Gruppenführer der Feuerwehr. Zum Zweck der Ontologiedokumentation können diese Beschreibungen automatisch aus der Ontologie extrahiert und als Bestandteil der Klassenbeschreibungen mit in die Dokumentation übernommen werden.

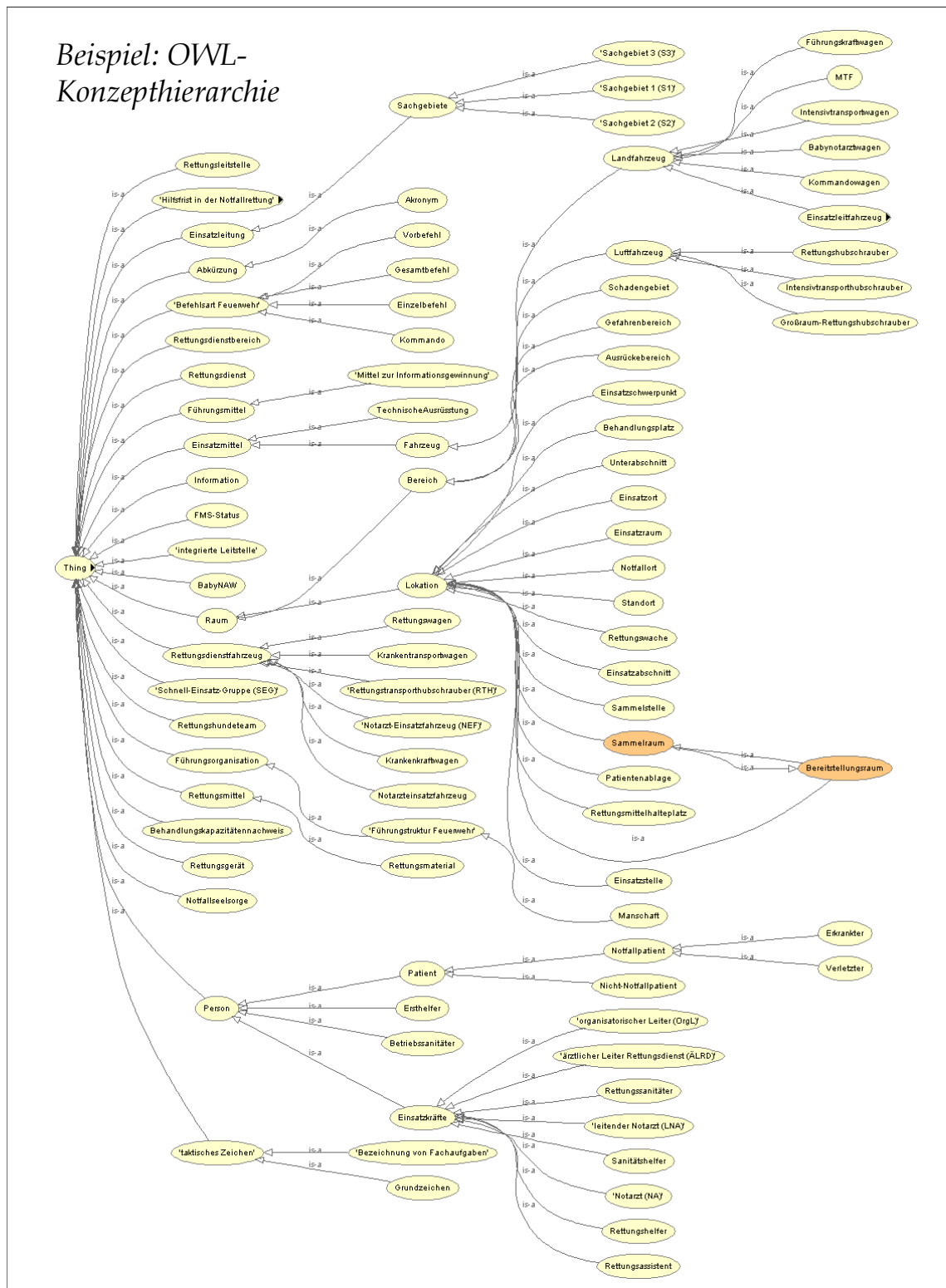


Abbildung 6.19: Beispielhafter Ausschnitt aus der BOS-Ontologie [BOS-O].

Schritt 4: Definition der Konzepte und der Konzepthierarchie

Zur richtigen Definition der Konzepte, und speziell der Konzepthierarchie (siehe Abb. als Beispiel 6.19), existieren unterschiedliche Ansätze und Empfehlungen. Ein Ansatz besteht

darin, mit der Definition der allgemeineren Konzepte zu beginnen (*Top-down*) und in der Folge speziellere Konzepte diesen unterzuordnen. Als entgegengesetzter Ansatz werden spezielle Konzepte zu allgemeineren Konzepten zusammengefasst (*Bottom-up*). Beide Vorgehensweisen können je nach Anforderung kombiniert werden. USCHULD & GRUNINGER weisen darauf hin, dass keine Methode prinzipiell besser ist als die andere. Vielmehr hängt es von den Fähigkeiten des Modellierers selbst und seiner Sicht auf die Domäne ab, welche Methode für ihn die geeignetste und praktikabelste ist [UG96].

In Abbildung 6.19 ist mit den Klassen nur eine Facette der Ontologiestruktur sichtbar. Durch eine Ontologie wird in der Regel mehr als eine bloße Taxonomie von Konzepten, so wie die Abbildung vielleicht suggeriert, repräsentiert. Eine vollständige Abbildung einer komplexen Ontologie wäre an dieser Stelle weitaus umfangreicher und unübersichtlicher.

Schritt 5: Festlegen der Relationen

Nachdem die wichtigsten Konzepte bestimmt wurden, gilt es, deren Relationen (auch Beziehungen, Rollen oder Eigenschaften genannt) untereinander zu formalisieren. Einige in Schritt 3 identifizierte Begriffe lassen sich zum Beispiel als Relation⁴ zwischen Konzepten interpretieren und stellen somit Kandidaten für eine zu definierende Relation dar. So erscheint z. B. der Begriff *Retten* (aus der DIN 13050) als Konzeptklasse wenig sinnvoll, da er vielmehr eine Beziehung zwischen zwei Konzeptklassen beschreibt (Einsatzkräfte *retten* Patienten), und sollte daher als Relation definiert werden.

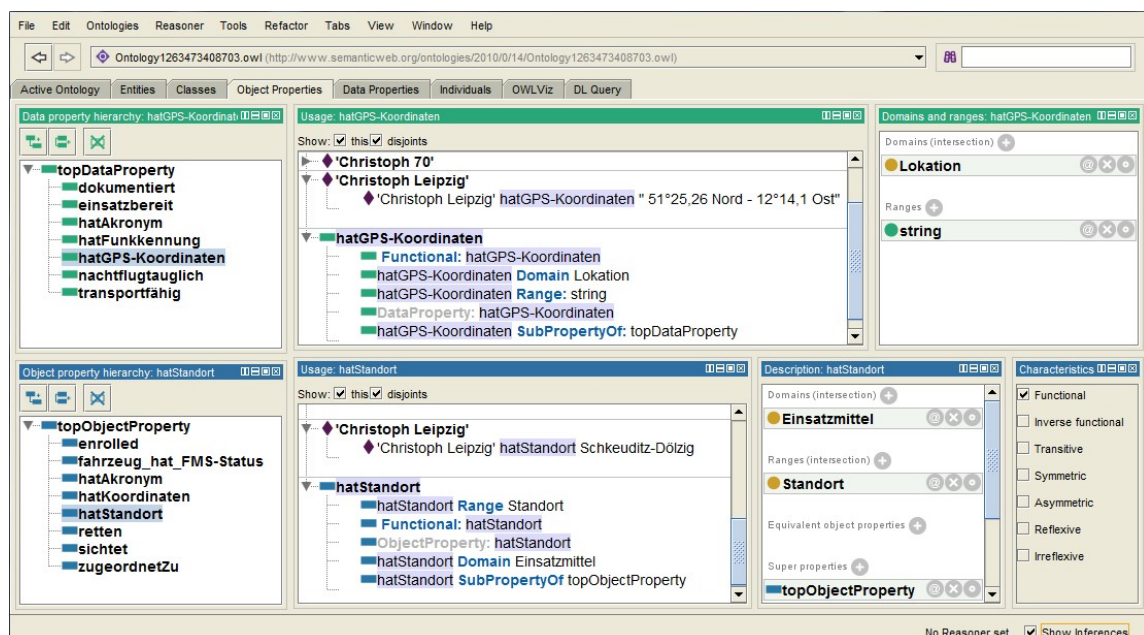


Abbildung 6.20: Auszug einer Rollendefinition der BOS-Ontologie in Protégé.

Relationen zwischen Konzeptklassen werden durch die Angabe ihres Definitionsbereichs (domain) und des Wertebereiches (range) genauer spezifiziert. In dem eben genannten Beispiel (*retten*) wurde daher als Definitionsbereich das Konzept Einsatzkraft und als Wertebereich das Konzept Patient zugeordnet. Abbildung 6.20 zeigt beispielhaft einen Auszug aus der mittels Protégé modellierten Relationendefinitionen. Im oberen Bereich sind die Datatype-Properties und im unteren Bereich die Object-Properties zu sehen.

⁴ Da der Begriff *Rolle* in dieser Arbeit bereits belegt ist, wird im Weiteren der Begriff *Relation* verwendet.

Schritt 6: Festlegen der Restriktionen über die Relationen

Die in Schritt 5 definierten Rollen unterliegen Restriktionen bezüglich des Datentyps und der möglichen Anzahl der Werte (*cardinality* des Wertebereiches). Die Restriktionen der Relationen beziehen sich auf die Festlegung des Definitionsbereiches und des Wertebereiches. Zum Beispiel hat die Relation `hatName(Ort,String)` als Definitionsbereich Individuen der Klasse `Ort` und als Wertebereich Individuen des Typs `String`, während die Relation `istFrei(RTW,Boolean)` einen RTW mit einem Wahrheitswert (*true* oder *false*) in Beziehung setzt.

Obwohl `String` und `Boolean` prinzipiell zwei Konzepte der TBox darstellen, werden solche elementaren Datentypen in konkreten KB-Systemen jedoch separat behandelt und nicht extra als neue Konzepte in die Ontologie aufgenommen. Hierbei handelt es sich um ein Beispiel des bereits angesprochenen *SHOIN(D)*-Sprachmerkmals (*concrete domain, D*). Mehr Informationen zum Thema finden sich im Überblick von LUTZ [Lut03].

Analog zur Kardinalitätsbeschränkung im *Entity-Relationship-Modell* (vgl. [Che76]) lässt sich die Anzahl der möglichen Ausprägungen für eine spezielle Relation festlegen. Es kann grob zwischen *Single*-Kardinalität (höchstens ein Wert ist erlaubt) und *Multiple*-Kardinalität (mehrere Werte sind erlaubt) unterschieden werden. Zum Beispiel wird durch die Relation `hatFunkkenner(RTW,Funkkenner)` ein Rettungswagen mit einer (seiner) ganz speziellen Funkkennung in Beziehung gesetzt. Hierbei handelt es sich um eine 1-Kardinalität, da mehrere Funkkenner für ein Rettungsfahrzeug ausgeschlossen sind.

Da einem Rettungsfahrzeug immer ein Funkkenner zugewiesen ist, handelt es sich bei diesem Beispiel genau genommen um einen funktionalen Zusammenhang, der sich z. B. mit OWL-Sprachmitteln *owl:FunctionalProperty* auch als solcher definieren lässt. Die Relation `hatAlsMitglied(Org,Person)` hingegen erlaubt die Zuordnung mehrerer Personen zu einer Organisation und ist somit ein Beispiel für eine *Multiple*-Kardinalität.

Schritt 7: „Bevölkern“ der Ontologie mit Instanzen

In der Regel beinhaltet die formale Spezifikation einer Konzeptualisierung noch keine konkreten Symbole für Individuen. Erst in einer konkreten Anwendung werden Individuen den Konzepten und Relationen als *Assertions* über einen Weltzustand hinzugefügt. Dieser Vorgang wird in der Literatur auch als Vorgang der *Ontologie Population* bezeichnet und entspricht der Beschreibung eines bestimmten Weltzustands der Domäne.

Ob Ontologien prinzipiell individuenfrei sein müssen, wird selbst unter Experten teils kontrovers diskutiert. Da Ontologien in der Praxis jedoch in der Regel kein reiner Selbstzweck sind, sondern immer einer ganz bestimmten Anwendung dienen, entscheidet die Art der Anwendung darüber, ob es sinnvoll ist, Individuendefinitionen bereits in die Ontologie einfließen zu lassen. So können z. B. (invariante) örtliche Gegebenheiten wie `istBenachbart(Ort1,Ort2)` als konstante individuenbezogene Aussagen einer Domäne in die Ontologie aufgenommen werden. Die unterschiedlichen Triagierungsstufen bei der Verletztensichtung sind ein weiteres Beispiel für invariante Fakten, die bereits im Vorfeld in der Ontologie als Instanzen des Konzeptes „Triagierungsstufe“ erfasst werden können, da sie sich während eines Einsatzes nicht ändern werden.

Die Ontologie sollte einerseits möglichst alle relevanten Sachverhalte der Welt widerspiegeln und andererseits kompakt genug sein, um noch effizient Schlussfolgerungen ziehen zu können. Von den konkreten Objekten der Domäne (teils *Ressourcen* genannt) wie Personen (Einsatzkräfte, Helfer, Patienten) und Rettungsmittel sollen in CASIE nur solche beachtet werden, die für einen Ablauf und die Koordinierung von Checklisten notwendig sind. Eine

vollständige Aufnahme aller Ressourcen (Einsatzkräfte, technische Geräte, Verbrauchsmaterialien, etc.) ist nicht praktikabel umsetzbar und soll nicht realisiert werden.

6.3.2 Entwicklung von ICLs

Ist die Erstellung und Nutzung simpler Checklisten in unserem Alltag noch recht einfach und überschaubar, so steht man bei sicherheitskritischen Einsatzgebieten schon vor einer größeren Herausforderung. Welche kritischen Schritte lassen sich denn bei welchen Routinehandlungen identifizieren? Wie präsentiere ich diese Punkte geeignet, damit möglichst alle Mitarbeiter den Sinn dahinter korrekt interpretieren?

Das Verbalisieren und das Formulieren passender Checklisten (Methoden) ist eine recht schwierige Aufgabe, vor allem dann, wenn es sich um Handlungen handelt, denen sich die Akteure nicht in Gänze bewusst sind. Sie würden in einer entsprechenden Situation durchaus richtig handeln, können ihre Handlungsschritte im Vorfeld jedoch schwer formulieren bzw. nicht vollständig begründen. Geeignete Checklisten für die unterschiedlichen SOPs zu erstellen, kann nur individuell von den jeweiligen lokalen BOS und unter Berücksichtigung ihrer speziellen regionalen Besonderheiten umgesetzt werden. Im Rahmen dieser Arbeit soll trotzdem versucht werden, die einzelnen Schritte der CL-Erarbeitung aufzuzeigen. Abbildung 6.21 zeigt hierfür ein mögliches Ablaufschema.

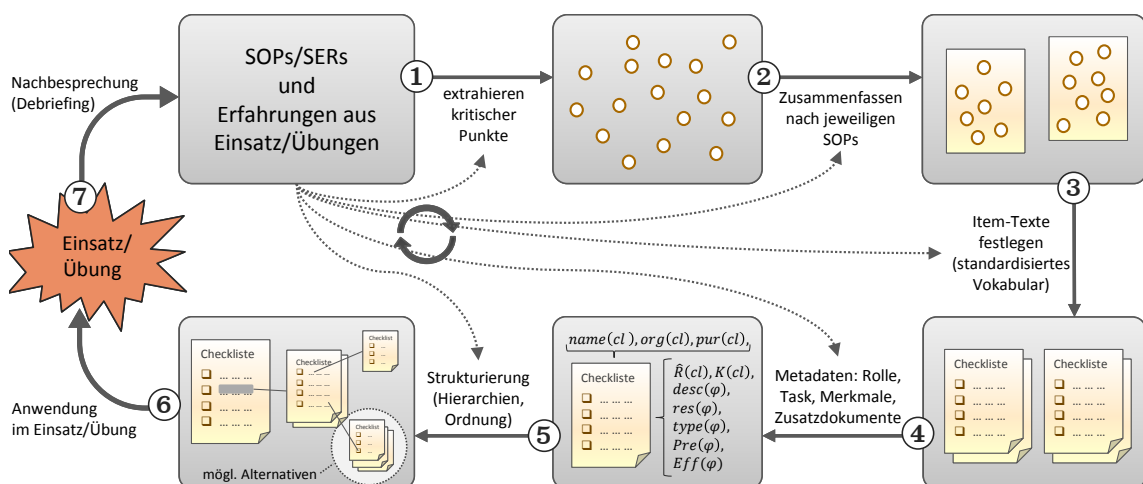


Abbildung 6.21: Von SOPs zu Checklisten und zurück.

① **Kritische Punkte ermitteln** In einem ersten Schritt gilt es, basierend auf den Erfahrungen vorangegangener Einsätze (und Übungen), die kritischen Punkte zu identifizieren, die oft vergessen werden, deren Beachtung aber essentiell für die jeweiligen SOPs sind. Jede Organisation muss anhand ihrer spezifischen SOPs selbst für sich entscheiden, für welche Bereiche eine Fehlerminimierung angestrebt wird. Weiterhin sind die gültigen Ordnungen, Regelwerke, Gesetze und Dienstvorschriften zu beachten. Aus ihnen können sich ebenfalls kritische Punkte ergeben, deren Einhaltung im Ernstfall durch die CLs sichergestellt werden soll.

② **Kritische Punkte gruppieren** Wurde ein Satz kritischer Punkte notiert, so müssen diese zunächst gemäß der zu beachtenden SOPs bzw. Rollenverantwortungen gruppiert werden. Hierbei muss zuvor geklärt werden, für welche der Punkte eine Überprüfung mittels Checklisten überhaupt sinnvoll und praktikabel ist. Bei Bedarf lassen sich ebenfalls

individuelle Wünsche und die jeweiligen Erfahrungen konkreter Einsatzkräfte mit berücksichtigen.

③ **Itemtexte und -ordnung festlegen** Die bis jetzt ungeordneten kritischen Punkten müssen zum einen (falls erforderlich) gemäß der SOP-Korrespondenz geordnet werden und zum anderen in eine geeignete Itemstruktur überführt werden (vgl. Definition, S. 55). Die Ordnung der Items einer ICL wird durch das Paar $\mathcal{I}(cl) = (\Phi, \prec)$ bestimmt, mit Φ als Menge der Items und \prec als Ordnungsconstraints über Φ (vgl. Def. 4, S. 52). Für jedes Item ϕ einer ICL wird dann ein stichpunkthafter Informationstext $text(\phi)$ erstellt. Siehe hierzu die Diskussion zur Checklisten-Phraseologie auf Seite 36 ff.

Das durch die BOS-Ontologie bereitgestellte standardisierte Vokabular sollte zudem eine wichtige Quelle für verwendete Fachbegriffe sein. Im Idealfall bedient man sich bei der Formulierung der Itemtexte strikt der Bezeichnungen der Fachkonzepte, die in der Ontologie berücksichtigt wurden.

Die Verbindung der Fachbegriffe mit den jeweiligen Namen der TBox-Konzepte entspricht einer semantischen Annotation ausgewählter Itemtext-Elemente, womit sich die Möglichkeit einer Erklärungsfunktionalität für weniger routinierteren Einsatzkräfte erschließt.

④ **Metadaten, Anwendungskriterien und Effekte** In einem weiteren Schritt können gemäß der Strukturdefinitionen von Checklisten und Items, deren Merkmale an den CLs und Items angegeben werden. Auf Seiten der Checkliste selbst sollten folgende Angaben gemacht werden: ein eindeutiger Name $name(cl)$, der BOS-Name $org(cl)$, in deren Einsatzbereich cl fällt, eine natürlichsprachliche Beschreibung $pur(cl)$ des Zweckes, die Menge $\widehat{\mathcal{R}}(cl)$ von zuständigen Rollen und die Menge $\mathcal{K}(cl)$ von Anwendungskriterien.

Für jedes Item ϕ der Liste können eine ausführliche Beschreibung $desc(\phi)$, eventuell vorhandenen Zusatzressourcen $res(\phi)$, der Typ des Items und mögliche Vor- und Nachbedingungen angegeben werden. Items mit Vorbedingungen werden bedingte Items genannt. Der Anwender/Ersteller einer ICL kann Vorbedingungen auf zwei Weisen definieren:

1. Er kann die Vorbedingungen als Anwendungskriterien vom Typ „Freitext“ angeben. Diese Bedingungen können von CASIE im GUI nur dem Nutzer angezeigt werden. Eine automatische Überprüfung der Gültigkeit ist nicht möglich. Die Gültigkeit der angegebenen Bedingungen kann jedoch vom Nutzer während einer CASIE-Anwendung abgefragt (erfragt) werden. (Vorausgesetzt in der GUI wird eine entsprechende Funktionalität bereitgestellt.)
2. Er kann aus dem Satz von zuvor in der Ontologie definierten Merkmalen wählen und somit die Grundlage für eine spätere automatische Auswertung geben. Jedoch sollte in den Fällen, in denen die \mathcal{KB} als Ergebnis der Auswertung „*unknown*“ liefert, ebenfalls der Nutzer über die eventuelle Gültigkeit der Vorbedingungen befragt werden.

Nicht für jedes bedingte Item wird sich in der Praxis auch ein geeignetes Merkmal finden lassen, das zuvor auch in der Ontologie berücksichtigt wurde. Oftmals ist es wohl sinnvoller, die Vorbedingungen in natürlichsprachlicher Form abzulegen und dem Nutzer im Einsatz selbst entscheiden zu lassen. Bedingte Items, für die das System keine Gültigkeit feststellen kann, können (automatisch) zur Erhöhung der Übersichtlichkeit von der jeweiligen Checkliste ausgeblendet werden. Falls während der Bearbeitungszeit der Checkliste die Vorbedingungen plötzlich erfüllt sind, kann dieses Item automatisch als noch offen wieder eingeblendet werden.

⑤ **Sub-CLs und Alternativen** Für jedes abstrakte Item ϕ werden über die Menge $Sub(\phi)$ alle potentiell möglichen Sub-CLs angegeben. Diese Verknüpfungen sind die Basis der automatischen Berechnung der Checklistenhierarchien. Analog zu den HTN-Methoden geben sie an, welche CLs die jeweiligen abstrakten Items konkretisieren können (siehe Abschnitt 5.2.3).

⑥ **Anwenden in der Praxis** Das Ergebnis des ersten Durchlaufs des Zyklus kann dann im Ernstfall oder in einer Übung angewandt werden. Neben dem Einsatz einer konkreten CASIE-Implementierung können die CLs aus dem CL-Repository auch als einfache PCLs eingesetzt werden. Selbst in diesem Fall lassen sich noch fehlende Punkte oder Unstimmigkeiten aufdecken.

⑦ **Evaluierung/Debriefing** In Analogie zur Entwicklung von SOPs [Gru03] müssen nach der Erarbeitung der Checklisten Evaluationsmechanismen erarbeitet und etabliert werden, die die Effektivität der Checklisten und deren korrekte Anwendung überprüfen.

Personalisierte Checklisten

Aufgrund der Heterogenität der Qualifikationen aller Einsatzkräfte liegt der Wunsch nahe, Checklisten zusätzlich zu personalisieren (vgl. Anforderungen, S. 46 ff.). Gemeint ist hiermit, dass z. B. ein hoch qualifizierter und erfahrener Notarzt, der die Rolle des Leitenden Notarztes einnimmt, deutlich weniger und detailliertere Items bevorzugt als etwa ein kaum erfahrener Kollege. Letzterer wird unter Umständen über mehr Assistenz (erweiterte CLs und detailliertere Items, mehr Zusatzmaterialien, persönliche Anmerkungen) sehr dankbar sein. Um diese Zusatzfunktionalität zu realisieren, bedarf es lediglich einer entsprechenden Abfrage im GUI und ein um die personalisierten Checklisten erweitertes CL-Repository. Je nachdem, unter welchem Namen (ID) sich eine Einsatzkraft am CL-Client anmeldet und welche Rolle sie übernommen hat, wird die zuvor personalisierte Checkliste aktiviert. Diese Zusatzfunktionalität kann recht einfach durch entsprechend abgelegte CL-Varianten realisiert werden, so wie sie weiter oben im Abschnitt 5.2.3 vorgestellt wurden. Die Menge der Anwendungsbedingungen \mathcal{K} muss nun lediglich um ein Merkmal erweitert werden, das Auskunft über den jeweiligen Rolleninhaber gibt.

Eine unsachgemäße Nutzung solch einer Personalisierung birgt jedoch auch Gefahren. Volle Freiheit bei der Personalisierung könnte dazu führen, dass der gewünschte Effekt, ausschließlich auf kritische Punkte hinzuweisen, verloren geht, wenn diese (leichtsinnig) von einer personalisierten Checkliste entfernt werden. Hingegen weitgehend unproblematisch ist die Erweiterung um zusätzliche Items oder Informationen (persönliche Anmerkungen und Ergänzungen). Aber auch hier können zu viele Informationen dazu führen, dass die CL zu umfangreich wird, was sich wiederum negativ auf deren Anwendung auswirken kann (siehe Diskussion in Abschnitt 4.2.5). Inwieweit diese Funktionalität genutzt wird und bis zu welcher Granularität diese umgesetzt wird, liegt bei jeder Organisation selbst.

Von CLs zu SOPs

In der vorliegenden Arbeit (Kapitel 4 und 5) wurde der Verzicht auf eine komplexe und allumfassende Prozessmodellierung zugunsten des Checklisten-Konzeptes motiviert. Begründet wurde dies vor allem durch zwei Feststellungen: (1) Eine Großschadenslage ist hochgradig dynamisch und im Detail nicht planbar. (2) Auch die Teile der SOPs, welche sich in jeder Großschadenslage gleichen, sind schwer zu erfassen und zu modellieren. Es mangelt hierfür zum einen an geeigneten und einfach zu bedienenden Softwarewerkzeugen,

und zum anderen sind diese SOPs als solche kaum in der deutschen BOS-Kultur verankert. Obgleich sich SOPs sehr wohl in der Praxis herausbilden, sind diese zumeist nur implizit in den Köpfen der Anwender gegeben.

Trotz der oben genannten Schwierigkeiten ist im BOS-Bereich der Versuch einer (teilweisen) Prozessfassung durchaus ein lohnenswertes Ziel. Eine Evaluierung von Handlungsstrategien und eine Qualitätssicherung kann somit ermöglicht werden. Zum Beispiel beschreibt ARSENOVA in [Ars08] die Praxiserfahrung, dass ein Prozessmodell den Einsatzkräften der Feuerwehr einen ersten Überblick über ihre Prozesse gibt und somit bereits bei der Diskussion der Einsatzvorbereitung dienlich ist. Es bleibt die Frage, wie solch ein Prozessmodell zu erarbeiten ist. Der Autor stimmt mit PEINEL et al. [PR09] überein, dass heutige BPM-Techniken nicht direkt 1:1 im BOS-Bereich angewendet werden können (siehe Diskussion in Kapitel 3). Vielmehr bedarf es alternativer Ansätze der Modellierung.

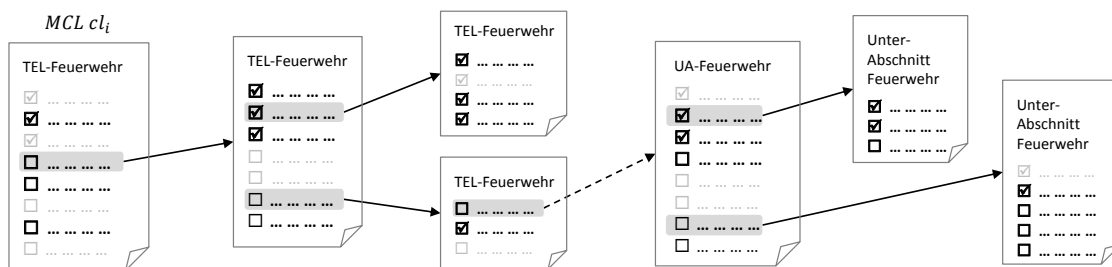


Abbildung 6.22: Beispielschema eines Prozessskeletts.

Mit CASIE erschließt sich hierfür ein pragmatischer, alternativer Ansatz. Ausgehend von der Menge der Checklisten im CL-Repository, lässt sich theoretisch zu jeder Checkliste ein entsprechendes Prozess-Skelett ermitteln. Dieses besteht „nur“ aus kritischen Tasks einer (oder mehrerer) SOPs. In der Praxis ist es jedoch ausreichend, die Checklisten der höchsten Hierarchieebene (MCLs) zu konkretisieren. Berücksichtigt man dabei ausschließlich die definierten *action*-Items, so erhält man, in Abhängigkeit der Anwendungskriterien der jeweiligen CLs, ein sog. Task-Netzwerk. Abbildung 6.22 skizziert solch ein sich ergebendes Prozess-Skelett. In einem zweiten Schritt lässt sich daraus ein Prozessmodell in der gewünschten Modellierungstechnik erstellen. So lässt sich leicht eine Umwandlung in einen gängigen Modellierungsformalismus, wie z. B. in den, der Ereignisgesteuerten Prozessketten (siehe SCHEER [KNS92]), vornehmen. Man erhält somit ein grobes Schema der in einer Großschadenslage ablaufenden Prozesse, was z. B. nützlich für eine Prozessdokumentation sein kann.

Durch die Definition der Checklisten erhält man gleichzeitig Informationen über alle potentiellen Aufgaben und deren Prozesse. Checklisten können daher Einblick auf die dahinterliegenden impliziten SOPs geben. Genau dieser Effekt kann dazu beitragen, dass jede BOS sich ihrer implizit vorhandenen SOPs bewusst(er) wird und diese somit reflektieren, evaluieren und verbessern kann. Auch wenn diese Sicht auf die Prozesse noch lückenhaft ist, da sie sich nur auf kritische Aspekte beziehen, können die Prozessskelette einen Ausgangspunkt für eine detailliertere Modellierung sein.

Kapitel 7

Vorteile der CASIE-Anwendung in der Praxis

In diesem Kapitel wird der praktische Nutzen von CASIE zusammengefasst. Darüber hinaus werden eine Reihe zusätzlicher Vorteile eines solchen Assistenzsystemes erläutert und Ausblicke auf zukünftige Erweiterungen gegeben. Das Anwendungsfeld der BOS dient dabei an einigen Stellen als Illustration allgemeiner Systemmerkmale.

7.1 Beitrag zur Fehlervermeidung

Der in erster Linie durch CASIE erreichte Praxisnutzen liegt in der Fehlervermeidung bei der Anwendung von standardisierten Handlungsrouinen. Das liegt vor allem an den Hauptfunktionen von Checklisten (vgl. Kapitel 4):

- *Sicherstellung eines Qualitätsstandards für Standardprozeduren,*
- *Handhaben komplexer Prozeduren durch Abstraktion,*
- *Reduktion der Unsicherheit der Einsatzkräfte und*
- *Gedächtnisentlastung in Stresssituationen,*

die jede für sich genommen das Risiko des Auftretens von groben Fehlern (Putzer oder Schnitzer) minimieren können (vgl. Abschnitt 4.1). Zusätzlich ist zu erwarten, dass im Zuge der Konfiguration der Checklisten, sich jede BOS auch mit der Qualität ihrer eigenen SOPs kritisch auseinandersetzt. Der von REASON [Rea90] als „eigentlicher Fehler“ bezeichnete Problematik, unangemessene Pläne für bestimmte Zielstellungen anzuwenden, kann somit ebenfalls entgegengewirkt werden, da bereits bei der Konfiguration offensichtliche Fehler im Ablauf erkannt und behoben werden können. In der Praxis ist daher zu erwarten, dass CASIE dazu beitragen kann, die beiden grundlegenden Fehlertypen „Fehler“ und „Putzer“ auf den von REASON unterschiedenen Handlungsebenen *Planung*, *Speicherung* und *Ausführung* zu vermeiden (vgl. Abschnitt 4.1).

Natürlich sind obige Effekte auch bereits mit klassischen, papierbasierten Checklisten zu erreichen. Wie in dieser Arbeit in Abschnitt 5.1 gezeigt wurde, besitzen Großschadenslagen jedoch besondere Merkmale, bei denen der Einsatz papierbasierter Checklisten an seine Grenzen stößt. Zu ihnen zählen die zu erwartenden unterschiedlichen Benutzergruppen, die geographische Verteilung der Einsatzkräfte, die Vernetztheit der Aufgaben unterschiedlicher SOPs und BOS sowie die Dynamik der Einsatzlage und der Informationsmangel.

Computerintegrierte Checklisten können zum einen den bekannten Fehlern (wie z. B. dem *Vergessen von Items* oder dem *Wiederaufnahmefehler*) bei der Ausführung von PCL entgegenwirken. Zum anderen kann CASIE den besonderen Merkmalen einer Großschadenslage gerecht werden. So kann im Gegensatz zu PCLs den Einsatzkräften eine theoretisch unbeschränkte Zahl an unterschiedlichsten ICLs leicht zugänglich gemacht werden.

Alle eben genannten Vorteile von CASIE zeigen, dass eine IT-unterstützte Umsetzung des Checklisten-Prinzips im BOS-Bereich besonders geeignet ist, um einen signifikanten Beitrag zur Fehlervermeidung beizutragen. Dies bestätigt die These 1 dieser Arbeit.

CASIE dient in erster Linie den Einsatzleitern verschiedener BOS zur Bereitstellung und Anwendungsassistenz von elektronischen Checklisten. Im Prinzip kann jedoch jede beliebige Einsatzkraft, die über ein entsprechendes Endgerät mit einem CL-Client ausgestattet ist, ebenfalls durch die Definition eigener Checklisten von CASIE profitieren.

Das Anwenderspektrum sowie die Anwendungsfelder von CASIE sind prinzipiell unbeschränkt. Die freie Konfigurierbarkeit des CL-Repositories und der formalen Anwendungsontologie ermöglicht es den Anwendern, ein Assistenzsystem zu etablieren, das genau auf ihre speziellen Anforderungen zugeschnitten ist.

7.2 Prozessmonitoring

Besteht eine Verbindung zwischen mehreren CL-Clients am Einsatzort, so kann mittels der aktuellen Bearbeitungszustände aller ICLs ein Überblick über alle laufenden und abgeschlossenen SOPs abgeleitet werden. Neben den ICLs mit ihren Status `inactive`, `active`, `completed` und `anceled`, sind deren *action*-Items hierbei von besonderem Interesse, da sie für bestimmte Handlungen/Aufgaben der zu den ICLs korrespondierenden SOPs stehen. Auf Basis des für *action*-Items speziell berücksichtigten Status `inprogress` können auch alle noch nicht abgeschlossenen, aber bereits gestarteten Aktionen berücksichtigt werden. Zusätzlich können auch abgebrochene ICLs oder Items als solche mittels der Status `cancel` bzw. `abandon` berücksichtigt werden, wodurch CASIE eine detailliertere Sicht auf die Einsatzlage bereitstellt, als es nur unter Berücksichtigung der bestätigten Items möglich ist. Im Endeffekt ermöglicht dies einen qualitativen Sprung im gesamten Einsatzmanagement. Prinzipiell lassen sich durch die Komponente des Prozessmonitorings folgende Fragen an die Einsatzlage beantworten.

1. Welche Rollen sind im Einsatz aktiv und wer hat diese übernommen?
2. Welche ICLs, und daher welche SOPs, sind gerade in Bearbeitung?
3. Welche ICLs wurden bereits abgeschlossen bzw. explizit abgebrochen?
4. Wie ist der jeweilige Bearbeitungszustand der aktiven ICLs? Dies umfasst alle noch offenen, abgeschlossenen und delegierte Items.

Abbildung 7.1 zeigt das Schema der durch CASIE möglichen Prozessüberwachung. Die vertikalen Pfeile stehen für Statusabfragen, in denen, über alle aktiven Rollenbereiche hinweg, die Bearbeitungszustände aller aktiven CLs aggregiert werden können. Aufgrund des Prozesskontextes jeder ICL kann daher auch der Fortschritt aller eingeleiteten SOPs überwacht werden. Die CASIE-Komponente des Prozessmonitorings kann somit einen Beitrag dazu leisten, dass Führungskräfte einen umfassenden Überblick über alle laufenden Prozesse erhalten. Somit kann für die Einschätzung der eigenen Lage zeitnah auch der Verlauf der zu den CLs korrespondierende SOPs herangezogen werden.

Wenngleich über das Monitoring „nur“ die in den ICLs berücksichtigten SOPs beachtet werden, kann solch eine Funktionalität einen signifikanten Beitrag zur allgemeinen *Situation Awareness* (siehe S. 111) leisten. Die Qualität der auf durch das Monitoring fußenden Entscheidungen kann dadurch wesentlich verbessert werden. Dies untermauert die These 2 dieser Arbeit (vgl. S. 45), welche besagt, dass durch den Einsatz einer Checklisten-Assistenz zugleich ein Monitoring der laufenden Prozesse erreicht werden kann.

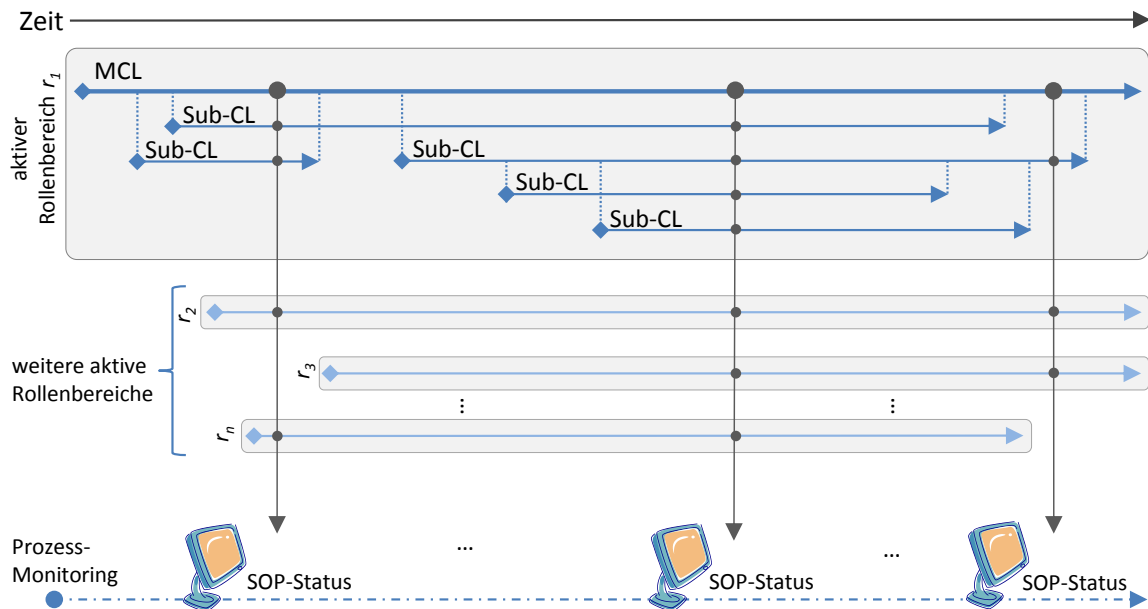


Abbildung 7.1: Schema des Prozessmonitorings.

Obwohl die Informationen aller laufenden Prozesse an jedem beliebigen CL-Client angezeigt werden können, bietet es sich an, die Rollenstruktur und somit die Organisationsstruktur (vgl. Abb. 5.15) des Einsatzes zu beachten. Über die Auswertung der Rollenhierarchie kann der Zugriff auf die einzelnen Status der CL-Clients gesteuert und gefiltert werden.

Wie eine konkrete GUI-Umsetzung eines Prozessmonitoring realisiert werden kann, soll nicht Gegenstand dieser Arbeit sein. Mit der *Task-Agenda* des HICAP-Projektes (siehe Abb. 3.3, S. 22) und mit dem *Process Panel* des CoSAR-TS-Projektes (siehe Abb. 3.3, S. 23 und [TDS02]) wurden bereits mögliche Umsetzungen vorgestellt, an denen sich orientiert werden kann. Diese Lösungen berücksichtigen jedoch noch keine Rollen Aspekte und zielen ausschließlich auf eine Ausführungüberwachung von Plänen ab.

7.3 Zusätzliche Vorteile und Ausblick

Im Folgenden wird eine Reihe weiterer Vorteile beschrieben, die CASIE in der Praxis bietet.

Taskorientierte Datenakquise

Wie nicht zuletzt die Erfahrungen des SPEEDUP-Projekts zeigten, besteht vor allem in der Chaosphase, aber auch im weiteren Verlauf einer Großschadenslage ein Mangel an (vertrauenswürdigen) Informationen über die Einsatzlage. Dies liegt zum einen darin begründet, dass eine Einsatzlage zu Beginn typischerweise noch weitgehend unklar ist und erst erkundet werden muss. Zum anderen auch darin, dass wichtige Informationen zwar verteilt vorliegen, diese jedoch noch nicht an die richtigen Stellen weitergeleitet bzw. zu einem

einheitlichen Lagebild aufbereitet wurden. Eine Führungskraft, auf welcher Ebene auch immer, wird daher nicht selten auf Basis eines unvollständigen bzw. veralteten Lagebildes wichtige Entscheidungen treffen müssen.

Um den Führungskräften möglichst zeitnah die Informationen an die Hand zu geben, die sie auch benötigen, liegt die Idee nahe, relevante Daten bereits frühzeitig mit Hilfe eines IT-Systems aufzunehmen und die sich daraus ergebenden Informationen weiterzuleiten und auszuwerten. Wie das SPEEDUP-Projekt gezeigt hat, kann bereits ein relativ einfaches IT-System, das Informationen über die Anzahl der Verletzten und deren Triage-Daten sammelt, den Einsatzkräften äußerst hilfreich sein. Ein allgemeiner Einsatz computerbasierter Informationsplattformen stößt im BOS-Bereich jedoch auf eine grundlegende Schwierigkeit: Woher kommen die Informationen über die Einsatzlage und deren momentanen Abarbeitungsstand? *Wer* soll diese *wann* und *wie* in ein Computersystem einpflegen? Wie werden diese gespeichert und wie kann auf diese zugegriffen werden? Einsatzübungen im Umfeld des SPEEDUP-Projektes zeigten, dass den Führungskräften vor Ort in der Regel kaum Zeit bleibt, umfangreiche Eingaben über eine Tastatur in ein IT-System vorzunehmen. Zusätzlich verschärft sich das Problem durch widrige technische Gegebenheiten wie z. B. schlechter bedienbare „Tastaturen“ von Tablet-PCs oder kleineren mobilen Endgeräten.

Folgt man der Zielstellung dieser Arbeit, nämlich den Führungskräften eine elektronische Checklistenassistentz an die Hand zu geben, die auf einer minimalen Nutzerinteraktion beruht (im Wesentlichen das Abhaken von Items), so erschließt sich quasi ganz nebenbei ein pragmatischer Lösungsbeitrag für obige Problemstellung. Informationen des Bearbeitungsstatus der ICLs können dazu genutzt werden, um daraus Rückschlüsse über die „eigene Lage“ (vgl. Kapitel 2) ziehen zu können. Darüber hinaus können *query*-Items für eine taskorientierte Datenakquise herangezogen werden. CASIE kann Informationen gemäß folgender prinzipieller Fragestellungen bereitstellen:

1. *Welche Rollen sind aktiv und wer hat sie übernommen?* – Die Rollenzuordnung der ICLs ermöglicht es, Informationen darüber zu erhalten, welche Rollen durch wen übernommen wurden. Dies kann zur eigenen Lageeinschätzung genutzt werden und z. B. bei einer möglichen Ablösung eines Rolleninhabers durch eine andere Einsatzkraft Klarheit über die etablierte, aktuelle Rollenstruktur geben.
2. *Wie ist der Bearbeitungsstand der einzelnen Aufgaben?* – Durch die Interaktion mit den ICLs können in CASIE automatisch Informationen über offene, laufende und bereits abgeschlossene Aufgaben gesammelt werden. Realisiert wird diese Funktionalität über das Prozessmonitoring der CL-Manager-Komponente (siehe Abschnitt 6.1).
3. *Welche Lagemerkmale sind bisher sicher bekannt?* – Zusätzlich kann jeder Statuswechsel eines Items (oder ganzer ICLs) dazu führen, dass neue Fakten in die Wissensbasis aufgenommen werden. Das einheitliche Datenmodell, das durch die BOS-Ontologie realisiert wird, garantiert eine semantisch korrekte Speicherung und Verwendung der Informationen. Darüber hinaus kann die Wissensbasis von CASIE als Informationsquelle für ein externes Informations- und Managementsystem dienen.

Eine rechnerinterne Repräsentation der Einsatzlage wird jedoch zwangsläufig immer hinter einer wirklichen Einsatzlage hinterherhinken. Der Versuch, eine exakte Lagerepräsentation im Computer abzubilden, wird in dieser Arbeit daher nicht verfolgt; vielmehr soll sich nur auf die Einsatzmerkmale konzentriert werden, die aufgrund der Interaktionen mit den ICLs (Auswahl, Abhaken, Beenden) dem System bekannt gemacht wurden. Durch CASIE können jedoch nur Informationen gesammelt werden, welche zuvor im Kontext der vordefinierten

Rollenstrukturen und der Aufgaben sowie der damit verknüpften Ergebnisse (Effekte) berücksichtigt wurden. Nichtsdestotrotz bietet CASIE einen innovativen und pragmatischen Lösungsbeitrag zur Frage, wie die Informationen überhaupt in ein Computersystem zur Weiterverarbeitung gelangen können.

Beitrag zur Steigerung der Situation Awareness

Die Informationen, die mittels CASIE erhoben werden, können zum Aufbau eines besseren Verständnisses über eine Einsatzlage genutzt werden. Der besondere Vorteil von CASIE liegt darin, dass Einsatzkräfte aller Organisationen Einblick in laufende Aktionen anderer Teams erhalten können, was zusätzlich zu einem besseren Verständnis der Problemstellungen der jeweils anderen BOS führen kann.

Sich der richtigen Einsatzsituation bewusst zu sein, spielt im Management einer Großschadenslage eine besonders wichtige Rolle. In der Literatur wird hierfür der Begriff der *Situation Awareness* (SA) verwendet. Im *Crew Resource Management* (CRM) der Human-Factors-Forschung in der Luftfahrt nimmt die SA einen prominenten Stellenwert ein (vgl. z. B. [BH89]). Da eine Großschadenslage ähnliche Charakteristika aufweist, lohnt sich ein Blick auf das Konzept der SA auch im Kontext des BOS-Bereiches. Aufgrund einer vergleichbaren Komplexität der Arbeitsbereiche greift z. B. WAGNER in [Wag06] das Konzept der SA auf und propagiert eine Übertragung auf den Bereich der Notfallmedizin.

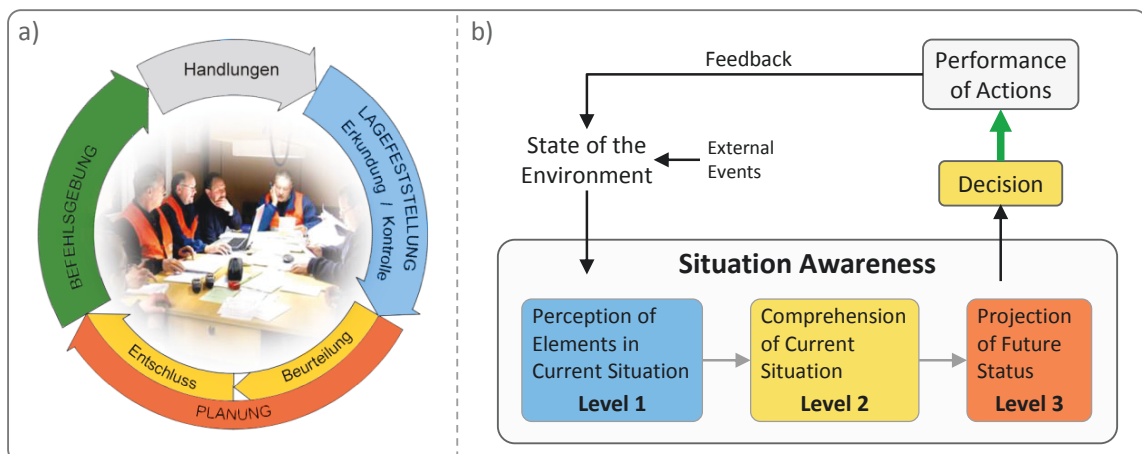


Abbildung 7.2: a) Führungsvorgang vs. b) Situation Awareness im Entscheidungsprozess.

Um den Beitrag von CASIE zur SA in einer Großschadenslage zu illustrieren, sei beispielhaft das populäre Modell des SA-Konzeptes im dynamischen Entscheidungsprozess von ENDSLEY (siehe [End00]) aufgeführt. Abbildung 7.2 (nach [End00], S. 3) zeigt hierzu unter b) das (vereinfachte¹) SA-Modell nach ENDSLEY. Das in Kapitel 2.3 vorgestellte Modell des Führungsvorganges (siehe Abbildung 2.3) laut [FwDV100], das auf der linken Seite der Abbildung dem SA-Modell gegenübergestellt ist, zeigt zudem, dass beide Modelle denselben Zyklus aus leicht unterschiedlichen Perspektiven darstellen. Der Aufbau der SA wird demnach in drei aufeinander folgende Schritte (Level) unterteilt. Die Wahrnehmung der Umweltmerkmale im ersten Level ist der fundamentalste Schritt zum Aufbau einer SA. Darauf aufbauend folgt in Level zwei die Interpretation der Informationen, die zum Aufbau eines gesamten Lageverständnisses führt. In Level drei wird aus den vorhandenen

¹ Im Bild wurden die *Individual Factors* und die *Task/System Factors* ausgespart, da hier die Überschneidung mit dem Modell des Führungsvorgangs (Abb. 7.2 a) deutlich gemacht werden soll.

Informationen über die Lage deren zukünftige Entwicklung abgeschätzt. Das Ergebnis aller drei Level bildet die Grundlage für die zu treffenden Entscheidungen.

Eines der Hauptprobleme für die Führungskräfte in einer Großschadenslage ist der Mangel an zuverlässigen Informationen. Indem CASIE Informationen über die Einsatzlage (vgl. „State of the Environment“ in Abb. 7.2) automatisch über die Arbeit mit den ICLs aggregieren kann, was vor allem in geografisch weit verteilten Einsatzsituationen ohne CASIE schwer bzw. zeitverzögert möglich ist, leistet das System einen entscheidenden Beitrag für den ersten Schritt der „Perception Of Elements In Current Situation“ im SA-Modell.

Wird CASIE in der Praxis von allen beteiligten BOS genutzt, so kann es dazu beitragen, das Verständnis für die jeweils anderen agierenden BOS zu erhöhen. Da in einer Großschadenslage alle BOS an einer übergeordneten Lagebewältigung zusammenarbeiten, existieren in einigen Bereichen und zu bestimmten Zeitpunkten Überschneidungen in den Arbeitsschritten. Die Arbeiten der unterschiedlichen BOS müssen dann situationsbedingt ineinandergreifen. Deutlich wird dies z. B. bei einem MANV mit chemisch kontaminierten Patienten. Für die Dekontaminierung der Verletzten ist die Feuerwehr zuständig und erst nach erfolgter Dekontaminierung kann und soll der Rettungsdienst z. B. mit seinem Sichtungszelt aktiv werden. Weiterhin ist es beispielsweise möglich, dass aufgrund einer durch die Feuerwehr festgestellten Gefahrgutbeteiligung am Einsatzort (erkannt durch die Beantwortung eines entsprechenden *query*-Items) CASIE dahingehend reagiert, dass alle anderen beteiligten BOS automatisch die passenden ICLs (falls definiert) für den Umgang mit Gefahrgutlagen eingespielt (vorgeschlagen) bekommen. In Kombination mit einer diesbezüglichen Meldung in der GUI wird die Aufmerksamkeit der Einsatzkräfte dadurch zeitnah auf diesen neuen Sachverhalt gelenkt.

Automatische Protokollierung

Eine wichtige Forderung der Praxis ist das Protokollieren des gesamten Ablaufs einer Großschadenslage. Neben der rechtlichen Absicherung ist eine möglichst lückenlose Protokollierung für jede BOS die Grundlage einer Einsatzevaluierung. Das Führen eines sog. Einsatzlogbuchs (oder auch Einsatztagebuch genannt) gestaltet sich jedoch in der Praxis oft als schwierig bzw. wird zugunsten „dringenderer“ Handlungen vernachlässigt. Während das Führen eines Logbuchs im Krisenstab für einen extra dafür abgestellten Mitarbeiter noch einfach ist, gestaltet sich eine Protokollierung für eine Führungskraft am Einsatzort schon problematischer. Aus eigener Erfahrung des Autors werden nicht selten Einträge in ein Einsatztagebuch erst nach einem Einsatz aus dem Gedächtnis verfasst, ergänzt oder geändert, worunter die Genauigkeit der Angaben leidet.

Auch ein elektronisches Logbuch, das dem Nutzer den Eintrag von Freitext ermöglicht, so wie es z. B. im Rahmen des SPEEDUP-Projektes umgesetzt wurde, unterliegt obiger Problematik. So konnte der Autor selbst erfahren, wie unhandlich sich das Führen eines elektronischen Logbuchs in einer stressigen Einsatzlage gestalten kann. Obwohl die Benutzerfreundlichkeit stark von der jeweils eingesetzten Hardware bzw. des GUI abhängt, bleibt im Einsatz meist kaum die Zeit, um längere Sätze oder Stichpunkte korrekt und in Ruhe einzutippen. Hinzu kommt, dass in der Praxis die Eingabe von Text in ein GUI nicht von jeder Einsatzkraft gleich gut bewältigt werden kann.

Mit CASIE erschließt sich ein alternativer Beitrag zu obiger Problemstellung (siehe Abschnitt 6.2). Da die Interaktionen des Nutzers mit den ICLs minimal ausfallen (vgl. Abschnitt 6.1.1), wird diese Problematik weitgehend vermieden. Die Idee besteht hierbei darin, dass in der Praxis bereits das Aktivieren einer ICL und das einfache Abhaken von Items einen wichtigen Beitrag zur Protokollierung leisten kann. Je nach Statuswechsel der Items (oder ganzer ICLs) kann ein entsprechend vorbereiteter (oder automatisch aggregierter)

Text in ein lokales/globales Einsatzlogbuch mit einem Klick (bzw. Fingerdruck am Touchscreen) eingetragen werden. Somit lässt sich bequem festhalten, *wer was wann* festgestellt bzw. bearbeitet hat. Auch die Übernahme einer Rolle durch eine Einsatzkraft und der Wechsel der Person zu einer neuen Rolle sind wichtige Informationen, die in einer Dokumentation mit erfasst werden sollten. Eine Einsatznachbereitung und Rekonstruktion kann dadurch maßgeblich verbessert werden.

Kontextabhängige Bereitstellung von Dokumenten

In vielen BOS existiert eine Vielzahl an Dokumenten zur Einsatzvorbereitung und Einsatzplanung. Diese reichen von Alarmierungsplänen, Kartenmaterialien und Funkplänen bis hin zu Rahmenplänen für spezielle Szenarien (z. B. MANV) oder bestimmten Objekten (z. B. Chemiefabrik) des Einzugsgebietes. Wie die Erfahrungen in SPEEDUP zeigten, werden diese Dokumente nicht selten zentral bei der Einsatzleitung (z. B. auf einem Notebook im Einsatzleitfahrzeug) vorgehalten. Damit stellt sich die Frage des organisierten Ablegens dieser Dokumente.

Je nach Lage und aktuellem Bearbeitungsstand der Checklisten ermöglicht CASIE das Hinterlegen dieser Zusatzinformationen in elektronischer Form. Diese Materialien können den Einsatzkräften kontextabhängig, d. h. zur jeweiligen SOP oder Task über die GUI des CL-Clients entsprechend zugänglich gemacht werden. In Abschnitt 6.1.1 wurde hierfür beispielhaft eine entsprechende GUI-Komponente gezeigt.

Assistenzsystem für dynamische Lageentwicklungen

Während beim Start von CASIE keinerlei Wissen über die aktuelle Einsatzsituation in der KB vorliegt, wird durch die Interaktion des Nutzers mit dem CL-Client im Laufe eines Einsatzes sukzessiv das Wissen über die Lage erweitert. Zu Beginn eines jeden Einsatzes bedarf es einer initialen Benutzerinteraktion. Als Erstes meldet sich eine Einsatzkraft am CL-Client durch die Auswahl der eingenommenen Rolle an. Danach hat sie Zugriff auf alle hinterlegten CL-Schemata. Je nach Informationslage kann hierbei bereits eine Filterung nach Relevanz automatisch angeboten werden. Im Laufe des Einsatzes passt sich das System mit Hilfe der Informationen, die über die Abarbeitung der Checklisten gewonnen wurde, bestmöglich an die Einsatzsituation an. Die Zunahme an faktischem Wissen manifestiert sich über die Zeit in der Wissensbasis und dient der fortlaufenden, automatischen Überprüfung aller Vorbedingungen der CL. Ändert sich etwas in den Vorbedingungen der aktiven Checklisten, kann der Nutzer darüber automatisch informiert werden. Sind plötzlich Kriterien für eine CL-Anwendung erfüllt, die noch nicht initiiert wurden, so kann der passende Rolleninhaber (falls vorhanden) darüber informiert werden, dass die Situation die Bearbeitung einer Checkliste erfordert. Wenn es laut einem Item z. B. notwendig ist, einen Gerätewagen „Gefahrgut“ zur Gefahrstofferkundung anzufordern, dies aber bereits durch eine z. B. übergeordnete Führungsinstanz erfolgreich geschehen ist, so kann dieser Punkt automatisch in der ICL als „erledigt“ markiert werden. Dies führt zu einer direkten Entlastung der Einsatzkräfte und zu einem koordinierterem Vorgehen.

Organisationsübergreifende Checklisten

Wie die Erfahrungen im SPEEDUP-Projekt zeigten, verläuft die Koordination und das Zusammenspiel aller beteiligten BOS in einem MANV oft nicht in der Art, wie es gewünscht ist. Dies kann zu zusätzlichen Reibungsverlusten bei einem koordinierten Vorgehen führen. Zum Beispiel wird in der Praxis um die Nutzung der richtigen Funkkanäle diskutiert,

wichtige Ressourcen werden eigenmächtig an falschen Stellen eingesetzt und damit gebunden oder es werden Informationen nicht oder verspätet an andere BOS weitergeleitet. Die bereits oben angesprochene Situation Awareness ist zumeist auf Aspekte beschränkt, die „nur“ für die eigenen BOS wichtig sind. Mit CASIE bietet sich die Möglichkeit der Erstellung BOS-übergreifender Checklisten. Vorausgesetzt, die einzelnen CL-Clients sind über ein etabliertes Kommunikationssystem am Einsatzort miteinander verbunden. Die Informationen über initiierte und abgeschlossene Aufgaben (repräsentiert durch *action*-Items) können so in einer organisationsübergreifenden Checkliste aggregiert werden. Diese sollte als eine Art Meta-Checkliste kritische Aspekte beim Zusammentreffen und Arbeiten von BOS mit unterschiedlichen Kompetenz- und Aufgabenbereichen widerspiegeln.

Der Einsatz von Checklisten in organisationsübergreifenden Prozessen ist ein gänzlich neuer Ansatz, der im heutigen Management einer Großschadenslage noch nicht angewandt wird. Die Schwierigkeiten dieser Art der CL-Nutzung liegen jedoch weniger im technischen Bereich. So kann CASIE diese Art von Checklisten ohne Weiteres durch z. B. die Einführung einer Meta-Rolle und der ihr zugeordneten Checklisten abbilden. Vielmehr stellt neben der Akzeptanzproblematik die Erstellung solcher organisationsübergreifender Checklisten eine besondere und neuartige Herausforderung dar. Es gilt, viele Interessen und Meinungen unter einen Hut zu bringen. So weisen die in einer Großschadenslage involvierten BOS neben unterschiedlichen Zuständigkeiten und Aufgabengebieten jeweils abweichende und ganz spezifische Arbeits- und Kommunikationsstile auf (vgl. hierzu [SMS10]). All dies gilt es zu berücksichtigen.

Neue Items zur Laufzeit einfügen

Bis jetzt wurde davon ausgegangen, dass alle kritischen Aspekte der SOPs im Vorfeld (offline) in Form von CLs erarbeitet werden. Die Einsatzkräfte können daher zur Laufzeit (online) „nur“ an zuvor auch berücksichtigte Aspekte erinnert werden. Erst im Zuge einer Einsatznachbesprechung und Evaluierung können noch fehlende bzw. neue Aspekte erkannt und die jeweiligen CLs für einen zukünftigen Einsatz erweitert werden. In Interviews (im Rahmen des SPEEDUP-Projektes) äußerten Führungskräfte der Feuerwehr den Wunsch, Checklisten nicht erst nach einem Einsatz zu editieren/erweitern, sondern diese Anpassung bereits während einer Übung oder gar im Einsatz (wenn es die Umstände erlauben) vornehmen zu können.

Aus Sicht der Erarbeitung und ständigen Verbesserung der Checklisten ist diese Funktionalität durch eine Erweiterung im GUI und im CL-Manager realisierbar. In der Praxis ist es wünschenswert, dass ein Anwender bereits während der Bearbeitung einer Checkliste neue Items an einer beliebigen Position hinzufügen kann. Je nachdem, welche Rolle und welche Aufgabe das neu eingefügte Item betrifft, wird ein Update des entsprechenden CL-Schemas im CL-Repository vorgenommen.

Analog zur Hinzunahme von Items kann prinzipiell auch das Löschen und das Anpassen von Items zur Laufzeit ermöglicht werden. Diese sollten jedoch aufgrund der sich daraus ergebenden Problematiken (Update laufender CLs, mögliche Verwirrung der Einsatzkräfte) erst bei zukünftigen Einsätzen berücksichtigt werden. In der Praxis kann ein Update auch dazu genutzt werden, allen Beteiligten die sich zur Laufzeit neu ergebenden kritischen Aspekte einer Einsatzlage „einzuspielen“, welche nur für den aktuell laufenden Einsatz gelten sollen. Zu realisieren ist dies z. B. durch die Einführung eines neuen Itemtypes (z. B. *temp*-Item) in CASIE, der nur im aktuellen Einsatz der Checklisten aktiv ist und nicht mit in das CL-Schema aufgenommen wird. In gewisser Weise kommt dies einer Informationsfunktion nahe, ähnlich einer Chatfunktion (vgl. „Instrument for communication“ in [WMY11]) in einem Einsatzmanagementsystem. Jedoch mit dem Unterschied, dass sich

die Einsatzkräfte nur auf eine Art Assistenzsystem konzentrieren müssen und die Informationen rollenabhängig verbreitet werden können. Somit ist es mit CASIE möglich, basierend auf sich ändernde Spezialanforderungen der aktuellen Situation, kritische Punkte, die sich neu ergeben, ad hoc während der Laufzeit zu definieren und allen Beteiligten bekannt zu machen.

Alternative Einsatzgebiete

Neben den klassischen Einsatzgebieten von Checklisten in Hochsicherheitsbereichen kann die hier vorgestellte CASIE-Architektur auch in ähnlich charakteristischen Arbeitsbereichen hilfreich eingesetzt werden. Gemeint sind alternative Einsatzgebiete, in denen (verteilte) Prozesse in einem beliebigen Notfallszenario fehlerfrei auszuführen sind. Das Arbeitsfeld des *Business Continuity Management* (BCM) ist ein Beispielbereich (vgl. STARKE [Sta10], S. 52 ff.), in dem die Sicherstellung fehlerfreier SOP-Anwendungen im Zuge eines Qualitäts- und/oder Risikomanagements anhand des Checklistenprinzips möglich ist.

Jedes größere Industrie- oder Wirtschaftsunternehmen bereitet sich auf mögliche Notfälle und Krisen vor. Die Etablierung eines BCM wird vielerorts sogar von übergeordneten Institutionen verpflichtend vorgeschrieben. Im Rahmen des BCM werden u. a. entsprechende Krisenmanagementpläne erarbeitet. Sind diese erst einmal vorhanden, stellt sich die Frage: Wie kann sichergestellt werden, dass alle Mitarbeiter im Ernstfall Kenntnis über die aktuellen Pläne erhalten und darüber hinaus auch innerhalb der vorgegebenen „Leitplanken“ handeln? Durch die prinzipielle Domänenunabhängigkeit kann die CASIE-Architektur für jedes denkbare Einsatzgebiet konfiguriert werden. Somit kann CASIE auch für diese Art von Notfall- und Krisenprozessen hilfreich eingesetzt werden.

Neben Krisenszenarien ist CASIE auch zur Sicherstellung der Prozessqualität einsetzbar. Bei der Einführung eines Qualitätsmanagementsystems (QMS) (z. B. gemäß der ISO-9000 Normenreihe) stellt sich die Frage, wie die Einhaltung der dort zugesicherten und dokumentierten Prozesse über die Zeit sichergestellt werden kann. Es gilt, Abweichungen von den SOPs möglichst zu vermeiden. Ein gebräuchliches Vorgehen im Qualitätsmanagement stellt die Standardisierung dar. Konkret werden detaillierte Verfahrensanweisungen für bestimmte Arbeitsgänge erarbeitet. Diese Anweisungen sind jedoch eher zur Schulung als zur Assistenz in der Praxis gedacht. Sie können allein noch nicht sicherstellen, dass die Mitarbeiter nichts Wichtiges vergessen. Der in dieser Arbeit erarbeitete Lösungsansatz ist zur Überprüfung dieser Prozesse geeignet und kann somit mit Blick auf ein QMS ein wichtiges Hilfsmittel zur Qualitätssicherung darstellen.

Kapitel 8

Zusammenfassung

Die vorliegende Arbeit beantwortet die Frage, inwieweit eine Computerunterstützung in komplexen Arbeitsbereichen zu einer Harmonisierung der Handlungsabläufe und zur Verhinderung von Fehlern (z. B. Gedächtnisfehlern) beitragen kann. Eine Analyse des Arbeitsbereiches deutscher *Behörden und Organisationen mit Sicherheitsaufgaben* (BOS) zeigt, dass vor allem der Informationsmangel und die nicht selten auftretenden Handlungsfehler wesentliche Grundprobleme in einer Großschadenslage darstellen. Die Untersuchung existierender Forschungsarbeiten zum Thema ergab, dass sich die dort vorgeschlagenen Lösungsansätze nur schwer in den Arbeitsprozess deutscher BOS übertragen lassen. Es mangelt ihnen an einem praktikablen, d. h. den Anforderungen einer Großschadenslage angepassten Interaktionskonzept mit den Einsatzkräften, das eine Computerassistenz am Einsatzort ermöglicht. Der in dieser Arbeit vorgestellte Lösungsansatz bietet genau dies. Motiviert durch den großen Erfolg von Checklisten (CL) in der Luftfahrt und in der Intensivmedizin, schlägt die Arbeit eine systematische Übertragung des Checklisten-Prinzips auf den deutschen BOS-Bereich vor. Die Dynamik der Einsatzlage, die Vernetztheit der Aufgaben und die ggf. geographische Verteilung der Einsatzkräfte am Einsatzort erschweren jedoch eine 1:1-Übertragung des CL-Prinzips in den BOS-Bereich. Um diese Merkmale zu adressieren, schlägt die Arbeit eine computerintegrierte Umsetzung vernetzter proaktiver Checklisten vor. Dadurch werden zum einen die Probleme vermieden, die beim Einsatz papierbasierter Checklisten zu verzeichnen sind, und zum anderen erschließen sich damit weitere nützliche Einsatzgebiete einer Computerassistenz, deren Potential bis heute in der BOS-Praxis kaum ausgeschöpft wird (vgl. Kapitel 2). Um alle Anforderungen an ein Checklisten-system für den BOS-Bereich zu adressieren, wird in dieser Arbeit ein Rahmenwerk eines *intelligenten elektronischen Checklisten-Assistenzsystems* (CASIE) erarbeitet und bewertet. Die Funktionalität der computerintegrierten Checklisten geht dabei weit über eine bloße elektronische Abbildung hinaus. Vielmehr werden beim Abhaken der Items einer Checkliste wichtige Informationen über die Einsatzlage aggregiert. In Abhängigkeit der so aufgebauten Informationslage wird im Gegenzug die Checklistendarbietung selbst dynamisch gesteuert. Die dadurch entstehenden erweiterten elektronischen Checklisten werden in CASIE *intelligente elektronische Checklisten* (ICLs) genannt.

Die Arbeit zeigt, dass neben dem Haupteffekt der Fehlervermeidung (**These 1**, siehe S. 45) CASIE zusätzlich zur Prozessüberwachung der Standardhandlungsroutinen (*Standard Operation Procedures*, SOPs) und zum Aufbau eines einheitlichen Lagebildes dienen kann (**These 2**, S. 45). Darüber hinaus wird dargelegt, welchen zusätzlichen Nutzen CASIE im BOS-Alltag und während einer Großschadenslage den Einsatzkräften bieten kann (**These 3**, S. 45). Inwieweit CASIE die zu Beginn von Kapitel 5 herausgearbeiteten Anforderun-

gen an eine elektronische Checklisten-Assistenz erfüllt, ist im Folgenden Punkt für Punkt zusammengefasst.

Wahlfreiheit gewährleisten: CASIE ermöglicht den Einsatzkräften zu jeder Zeit einen wahlfreien Zugriff auf alle vorkonfigurierten ICLs. Die automatischen Vorgaben, die CASIE auf Basis der zuvor von der Einsatzkraft eingenommenen Rolle (Aufgabenbereich) anbietet, können jederzeit überstimmt werden. Der Nutzer kann das Assistenzangebot von CASIE nutzen, muss es aber nicht. Ein flexibler Wechsel zwischen verschiedenen Checklisten ist dem Anwender zu jeder Zeit möglich. Ruft er jedoch z. B. eine Checkliste auf, die nicht in sein durch die Rolle bestimmten Aufgabenbereich fällt, so kann CASIE ihn auf diesen Sachverhalt hinweisen. Dadurch, dass Checklisten nur kritische Aspekte einer SOP berücksichtigen und „nur“ eine Erinnerungsfunktion anbieten, bleibt die gewünschte Flexibilität in den Handlungen erhalten. Die Anwender können weiterhin frei agieren und werden durch die Checklisten lediglich auf einen groben Handlungsrahmen aufmerksam gemacht.

Organisationsaufbau berücksichtigen: Da in CASIE explizite Rollenzuordnungen unterstützt werden, ist es möglich, die Checklisten gemäß eines beliebigen (in der Ontologie) vorkonfigurierten Organisationsaufbaus auf die unterschiedlichen Zuständigkeitsbereiche zu organisieren und zu verteilen. Darüber hinaus kann CASIE erkennen, ob ein anderer Nutzer genau dieselbe Checkliste bereits bearbeitet, und dies dem Nutzer mitteilen. Einem versehentlichen Mehrfachausführen von SOPs kann so effektiv vorgebeugt werden, wodurch die These 1 untermauert wird.

Komplexität handhaben: Die Möglichkeit zur Aufteilung und Hierarchisierung von Checklisten trägt in CASIE dazu bei, dass komplexe Checklisten mittels der Angabe von Sub-Checklisten für die Einsatzkraft handhabbarer (übersichtlicher) gestaltet werden können. Komplexe SOPs können somit aufgabengerecht in kleinere aufgeteilt und im Einsatz wieder automatisch zu einer CL-Hierarchie verbunden werden. Die Berechnung einer Checklistenhierarchie orientiert sich an der sogenannten Reduktion eines hierarchischen Tasknetzwerks in der HTN-Planung.

Dynamik berücksichtigen: Die Dynamik einer Einsatzlage wird in CASIE wie folgt berücksichtigt. (1) Es werden automatisch ICLs vorgeschlagen, wenn deren Anwendungsbedingungen erfüllt sind. (2) Im Gegenzug wird darauf hingewiesen, wenn ursprünglich geltende Anwendungsbedingungen nicht mehr gelten. (3) Ereignisse anderer Nutzer können mitbeachtet werden (z. B. Item wurde bereits an einer anderer Stelle erfüllt).

Problem des Informationsmangels: Weil Checklisten-Items zum einen mit Nachbedingungen annotiert und zum anderen Daten an geeigneter Stelle über die *query*-Items abgefragt und im System aufgenommen werden können, trägt CASIE zur Verbesserung des Aufbaus eines einheitlichen Lagebildes bei. Da des Weiteren die Informationen, die sich über die ICLs gewinnen lassen, Konzepten aus der Ontologie zugeordnet sind, ist deren Bedeutung klar definiert. Darüber hinaus gibt das Prozessmonitoring einen Überblick über die „eigene Lage“ der Einsatzkräfte. Dies untermauert die These 2.

Dokumentation & Backup: Der in CASIE vorgesehene automatische Logbookmechanismus erfüllt die Praxisanforderung einer Dokumentationspflicht, indem detaillierte Informationen über die Bearbeitung der SOPs automatisch protokolliert und festgehalten werden können. Die Frage, wer was wann eingeleitet und mit welchem Erfolg abgeschlossen hat, lässt sich somit leicht und zu jeder Zeit beantworten.

Zugriff/Organisation von Zusatzressourcen: Die in der BOS-Praxis vorhandenen elektronischen Dokumente (Gebäudepläne, Einsatzpläne, Funkschemata und Telefonnummernregister) können in CASIE den Einsatzkräften, gemäß ihres Aufgabenkontextes, direkt an den Stellen des Arbeitsprozesses automatisch zugänglich gemacht werden, an denen sie

eine Hilfestellung bieten. Zusatzinformationen zu einem Item oder auch zu einer ganzen Checkliste können so ebenfalls leicht über die vernetzten Checklisten bereitgestellt werden.

Einfache Bedienung: Eine der Stärken von CASIE liegt in der Art und Weise, wie die Anwender mit dem System interagieren. Sie arbeiten zur Einsatzzeit ausschließlich mit den Checklisten, die für sie ein wohlvertrautes bzw. leicht zu verstehendes Konzept darstellen. Dadurch wird zur Laufzeit im Wesentlichen vom Nutzer nur eine minimale Interaktion mit dem System erwartet. Lediglich die Eingabe konkreter Daten bei *query*-Items verlangt vom Nutzer mehr Aufwand.

Einheitliches Vokabular: Die Forderung, für die Texte der Items und die Beschreibungen ein möglichst einheitliches Vokabular zu verwenden, unterstützt CASIE dahingehend, dass den Einsatzkräften durch die Formulierung einer BOS-Ontologie solch ein einheitliches Vokabular zur Verfügung steht, das für die Erstellung der Texte zu den Items herangezogen werden kann. Letztendlich obliegt es jedoch den Einsatzkräften selbst, eine ausreichende Konsistenz bei der Wahl der Begrifflichkeiten einzuhalten.

Unterschiedliche Benutzergruppen: Dadurch, dass CASIE durch die Einsatzkräfte selbst frei konfigurierbar ist, d. h., jeder Nutzer ICLs seiner Wahl im Repository hinterlegen kann, steht auch einer Personalisierung der einzelnen Checklisten nichts im Wege. Damit unterschiedliche ICLs für ein und dieselbe SOP im Einsatz auch entsprechend zugreifbar sind, bedarf es lediglich einer zuvor hinterlegten Nutzerkennung. Die dafür benötigten Informationen sind im Wesentlichen bereits dadurch gegeben, dass sich jeder Nutzer an seinem CL-Client explizit mit seinem Namen anmeldet und so automatisch die für ihn hinterlegten Checklisten dargeboten bekommt.

Als Ergebnis der Arbeit steht interessierten Anwendern mit CASIE ein Rahmenwerk eines neuartigen Checklisten-Assistenzsystems zur Verfügung. Erst durch die ICLs, als elektronische Checklisten, erschließt sich ein Mehrwert gegenüber klassischen papierbasierten Checklisten. Neuartig ist vor allem der Ansatz, computerintegrierte Checklisten untereinander zu vernetzen und mit einer formalen Wissensbasis zu kombinieren, um ein maschinelles Schließen über Fakten einer Einsatzlage zu ermöglichen. Dadurch können Informationen über die Einsatzlage gesammelt, protokolliert und vom Computer automatisch entsprechend ihrer Bedeutung ausgewertet werden.

Um ein für die Problemstellung angemessenes Rahmenwerk zu erarbeiten, wurden in dieser Arbeit folgende Aufgabengebiete bearbeitet:

1. Umfassende Analyse des Checklistenprinzips aus einer Human-Factors-Perspektive
2. Erarbeitung der Anforderungen an eine elektronische Checklisten-Assistenz
3. Entwicklung eines technologischen Rahmenwerkes (Modellierung, Architektur, Technologieauswahl, Dynamik, Konfiguration von Checklisten als prozedurale Wissensrepräsentation)
4. Entwicklung (beispielhaft) einer formalen BOS-Ontologie [BOS-O] als terminologische Wissensrepräsentation

Damit eine konkrete CASIE-Instanz in der Praxis erfolgreich genutzt werden kann, gilt es aus informatischer Sicht, noch eine Reihe von weiteren Herausforderungen zu meistern. Neben der Realisierung eines Basiskommunikationssystems, das für die Anwendung von CASIE vorausgesetzt wurde, werden geeignete Softwarewerkzeuge benötigt, die dem Anwender das Erstellen und die Wartung der ICL-Beschreibungen auf eine bequeme Art und

Weise ermöglichen. Zusätzlich bedarf es weiterer Forschung hinsichtlich geeigneter Benutzerschnittstellen, die das volle Potential von CASIE ausschöpfen und den Anforderungen eines Einsatzes in einer Großschadenslage genügen müssen. Eine weitere noch offene Fragestellung betrifft den Abgleich der individuellen Wissensbasen unterschiedlicher CL-Clients. Hier gilt es, mögliche Inkonsistenzen und versteckte Redundanzen zu vermeiden bzw. geeignet zu behandeln. Des Weiteren sind die heutigen Programmierschnittstellen zur Ontologieunterstützung (siehe z. B. [OWL-API]), zum Reasoning (z. B. [FaCT++]) und zur Realisierung einer Wissensbasis noch unhandlich und erfordern viel Know-how, um sie zu beherrschen und geeignet miteinander in einer Praxisanwendung zu verbinden.

Die Bereitstellung eines technischen Assistenzsystems allein garantiert jedoch noch nicht, dass der angestrebte Haupteffekt der Fehlervermeidung erreicht wird. Damit die Lösung nicht zum Problem wird, bedarf es weiterer Untersuchungen, wie ein Assistenzsystem wie CASIE in der Praxis genau eingeführt wird und für welche Bereiche ICLs geeignet sind und für welche nicht. Hierfür müssen zusätzlich zu den technischen Problemstellungen folgende Aufgaben gemeistert werden:

- Erarbeitung einer Fehlerkultur und unterschiedlicher Anwendungsphilosophien,
- Lösungsansätze für die Akzeptanzproblematik,
- Beachten der Arbeitsumgebung bei der Anwendung von Checklisten,
- Aspekte der Fehlbenutzung und deren Auswirkungen,
- Kulturwandel bei Nutzung von Checklisten im Arbeitsprozess,
- mögliche Effekte bei Abweichung von Checklisten und
- die Erarbeitung geeigneter Evaluationskriterien und -konzepte.

Vor allem der Checklisten Einsatz für Nichtexperten stellt ein kaum untersuchtes Forschungsgebiet dar, da er von der Einsatzphilosophie anderer Hochsicherheitsbereiche abweicht, in denen Checklisten ausschließlich die Experten bei komplexen Arbeitsschritten unterstützen. Jedoch ist gerade in Anbetracht wenig trainierter Einsatzkräfte eine Lockerung dieser Forderung sinnvoll. Immer dort, wo Mitarbeiter zum Einsatz kommen, die (aus unterschiedlichen Gründen) unsicher in dem sind, *was* und *wie* sie etwas zu tun haben, können Checklisten zur Orientierung beitragen. Es gilt aber, Situationen vorzubeugen, in denen Einsatzkräfte, ohne selbst zu überlegen, blind den Checklisten folgen: „Ja, aber laut Checkliste sollte ich doch ...“. Solch eine Reaktion offenbart, dass das Prinzip hinter den Checklisten entweder falsch vermittelt wurde oder aber Checklisten als ausschließlich normatives Mittel zur Prozesssteuerung gebraucht werden. In beiden Fällen ist zu bezweifeln, dass der Effekt der Fehlervermeidung wirklich eintritt. Dem gilt es, durch ein geeignetes Training und eine begleitende Einführung der Checklisten in den Praxisbetrieb vorzubeugen.

Der zunehmende Einsatz von Computern im BOS-Bereich ermöglicht es, zukünftig neuartige Assistenzsoftware am Einsatzort einzusetzen. Mit CASIE stellt diese Arbeit ein Rahmenwerk bereit, in dem genau solch eine neuartige Assistenz beschrieben wird. Die Arbeit stellt einen Ausgangspunkt für eine Reihe weiterführender (interdisziplinärer) Forschungsarbeiten dar. Weiterhin soll sie die BOS-Einsatzkräfte dazu motivieren, eine elektronische Checklisten-Assistenz auf Basis des CASIE-Rahmenwerkes in ihrem Bereich einzusetzen, zu testen und zu evaluieren. In Anbetracht der vielen Vorteile, die CASIE in der Praxis bieten

kann, und der immer komplexer werdenden Arbeitsumgebungen sollte jede Organisation mit Sicherheitsaufgaben daran interessiert sein, solch ein System im Praxisbetrieb einzusetzen. So wie die Luftfahrt aus ihren Fehlern gelernt hat, so sollten auch die BOS den Einsatz neuartiger Technologie zur Fehlervermeidung bedenken. Die benötigten Grundtechnologien sind bereits vorhanden, sie müssen „nur“ geeignet kombiniert und richtig eingesetzt werden, dann lässt sich in der hier fokussierten Anwendungsdomäne Leben retten. Auch wenn der Weg zu einer erfolgreichen Etablierung einer elektronische Checklistenassistenz mühsam erscheint, die Aussicht auf nur *ein* gerettetes Menschenleben sollte Motivation genug sein, diesen Weg zu gehen.

Literatur

- [ACF+05a] Marc de la ASUNCIÓN, Luis CASTILLO, Juan FDEZ-OLIVARES u. a. „Knowledge and plan execution management in planning fire fighting operations“. In: *Planning, Scheduling and Constraint Satisfaction: From Theory to Practice*. IOS PRESS, 2005, S. 149–158 (siehe S. 26).
- [ACF+05b] Marc de la ASUNCIÓN, Luis CASTILLO, Juan FDEZ-OLIVARES u. a. „SIADEx: An interactive knowledge-based planner for decision support in forest fire fighting“. In: *AI Communications* 18.4 (2005). Volume 18, Issue 4 (December 2005), Binding Environmental Sciences and AI, S. 257–268 (siehe S. 21, 25).
- [AGP04] Marc de la ASUNCIÓN, Oscar GARCÍA-PÉREZ und Francisco PALAO. *SIADEx: A Real World Planning Approach for Forest Fire Fighting Introduction*. 2004 (siehe S. 25).
- [AH04] Grigoris ANTONIOU und Frank HARMELEN. *A Semantic Web Primer (Cooperative Information Systems)*. The MIT Press, 2004 (siehe S. 81).
- [Amm05] Stefanie AMMON. „Commitment, Leistungsmotivation, Kontrollüberzeugung und erlebter Tätigkeitsspielraum von Beschäftigten in Unternehmen und Behörden im Vergleich“. Diss. Fachbereich Kultur- und Sozialwissenschaften der FernUniversität in Hagen, 2005 (siehe S. 67).
- [Ark06] Debby ARKELL. „From safe to safer: Boeing’s Electronic Checklist marks 10 years of enhanced safety for pilots and travelers“. In: *Boeing Frontiers (online)* 04.11 (2006). http://www.boeing.com/news/frontiers/archive/2006/april/i_ca3.html (abgerufen am 20.12.2013) (siehe S. 39, 44).
- [Ars08] Emilija ARSENOVA. *Unterstützung der Prozessmodellierung im Notfallmanagement*. Gesellschaft für Informatik (Ed.) Informatiktage 2008. B-IT Bonn-Aachen, International Center for Information Technology, Bonn, Gesellschaft für Informatik. 2008 (siehe S. 19, 106).
- [Bar05] Martin BARTONITZ. „BPMS, BPML, BPEL, BPMN, WMS, XPD, Äm, Alles ist so schön bunt hier“. In: *PROJECT CONSULT Newsletter* (2005), S. 8–13 (siehe S. 18).
- [BBD+05] Stefan BROCKMANN, Christoph BRODESSER, Bernd DOMRES u. a. *Gefahrenabwehr bei Großveranstaltungen*. Hrsg. von Klaus MAURER und Hanno PETER. Verlagsgesellschaft Stumpf + Kossendey, Edewecht, Wien, 2005 (siehe S. 10).
- [BBK10] SCHUTZKOMMISSION BEIM BUNDESMINISTERIUM DES INNERN, BUNDESAMT FÜR BEVÖLKERUNGSSCHUTZ UND KATASTROPHENHILFE, BONN. *Katastrophenmedizin – Leitfaden für die ärztliche Versorgung im Katastrophenfall*. Bonifatius GmbH, Druck – Buch – Verlag, Paderborn. 5. völlig überarbeitete Auflage, München 2010. 2010 (siehe S. 10).
- [BCM+03] Franz BAADER, Diego CALVANESE, Deborah MCGUINNESS u. a. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge, UK: Cambridge University Press, 2003 (siehe S. 82, 86).
- [BCM+07] Franz BAADER, Diego CALVANESE, Deborah MCGUINNESS u. a. *The Description Logic Handbook: Theory, Implementation and Applications (2nd Edition)*. Cambridge, UK: Cambridge University Press, 2007 (siehe S. 82, 83, 86).

- [BH89] D. B. BERINGER und P. A. HANCOCK. „Exploring situational awareness: A review and the effects of stress on rectilinear normalisation.“ In: *In Proceedings of the Fifth International Symposium on Aviation Psychology*. Bd. 2. Columbus: Ohio State University. 1989, S. 646–651 (siehe S. 40, 111).
- [BH91] Franz BAADER und P. HANSCHKE. „A Scheme for Integrating Concrete Domains into Concept Languages“. In: *Proceedings of the 12th International Joint Conference on Artificial Intelligence, IJCAI-91*. Sydney (Australia), 1991, S. 452–457 (siehe S. 84).
- [BHG+01] Sean BECHHOFFER, Ian HORROCKS, Carole GOBLE u.a. „OilEd: a Reasonable Ontology Editor for the Semantic Web“. In: *Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence*. Lecture Notes in Computer Science 2174. Vienna: Springer-Verlag, 2001, S. 396–408 (siehe S. 96).
- [BHL08] Petra BADKE-SCHAUB, Gesine HOFINGER und Kristina LAUCHE, Hrsg. *Human Factors: Psychologie sicheren Handelns in Risikobranchen*. 1. Aufl. Springer Medizin Verl., Heidelberg, 2008 (siehe S. 1, 39).
- [BHS08] Franz BAADER, Ian HORROCKS und Ulrike SATTLE. „Description Logics“. In: Hrsg. von V. Lifschitz Edited by F. van HARMELEN und B. PORTER. *Handbook of Knowledge Representation*. Elsevier B. V., 2008. Kap. 3, S. 135–176 (siehe S. 83, 85).
- [BLK11] T. BECKER, B.-S. LEE und R. KOCH. „Effiziente Entscheidungsunterstützung im Krisenfall durch interaktive Standard Operating Procedures.“ In: *Software Engineering (Workshops)*. Hrsg. von Ralf REUSSNER, Alexander PRETSCHNER und Stefan JÄHNICHEN. Bd. 184. LNI. GI, 2011, S. 217–224 (siehe S. 20, 28, 34, 41, 43, 46).
- [BMBF] Bundesministerium für Bildung und Forschung (BMBF). *Bewilligte Projekte aus dem Themenfeld Schutz und Rettung von Menschen der Programmlinie „Szenariensorientierte Sicherheitsforschung“*. <http://www.bmbf.de/de/13091.php> (abgerufen am 20.12.2013). 2010 (siehe S. 2).
- [Boo01a] Daniel BOORMAN. „Safety Benefits of Electronic Checklists: An analysis of Commercial Transport Accidents“. In: *Boeing Commercial Airplanes* (2001). Boeing Commercial Airplanes, Seattle, Washington (siehe S. 39, 44).
- [Boo01b] Daniel BOORMAN. „Today’s electronic checklists reduce likelihood of crew errors and help prevent mishaps.“ In: *ICAO Journal, International Civil Aviation Organization* 56.1 (2001), S. 17–20 (siehe S. 39).
- [BOS-O] Uwe KRÜGER. *Beispielmodellierung einer terminologischen Wissensrepräsentation der Domäne deutscher BOS (Behörden und Organisationen mit Sicherheitsaufgaben)*. <http://sourceforge.net/projects/bos-ontology/> (abgerufen am 20.12.2013). 2013 (siehe S. 50, 72, 84, 86, 87, 95, 96, 100, 119).
- [Bos07] Christiane BOSOLD. *Polizeiliche Übergriffe: Aspekte der Identität als Erklärungsfaktoren polizeilicher Übergriffsintentionen – Eine handlungspsychologische Perspektive*. Nomos Verlag, Baden-Baden, 2007 (siehe S. 67).
- [BP07] Cornelius BUERSCHAPER und Michale St. PIERRE. „Teamarbeit in der Anästhesie – Entwicklung einer Checkliste“. In: Hrsg. von Stefan STROHSCHNEIDER. Bd. 2. *Menschen in kritischen Situationen – Schriftenreihe der Plattform Menschen in komplexen Arbeitswelten e. V.* Verlag für Polizei-Wissenschaft, 2007. Kap. 3, S. 25–38 (siehe S. 33, 37, 39).
- [Bra10] Tom BRABANT. „The simple genius of checklists - Boeing’s experience using checklists is helping many other professions“. In: *BOEING FRONTIERS* October (2010), S. 42–43 (siehe S. 39, 40).
- [BS01] Susanne BIUNDO und Bernd SCHATTENBERG. „From Abstract Crisis to Concrete Relief - A Preliminary Report on Combining State Abstraction and HTN Planning“. In: *Proceedings of the 6th European Conference on Planning (ECP-01)*. Springer Verlag, 2001 (siehe S. 21, 26, 27).

- [Bür05] Richard G. BÜRMAN. *Effizientes Prozessmanagement im Katastrophenschutz*. 1. Europäischer Katastrophenschutzkongress; 2005: Früherkennung, Alarmierung, Koordination; Bad Godesberg; (IDS Scheer AG). 2005 (siehe S. 18).
- [CFG+06] Luis CASTILLO, Juan FDEZ-OLIVARES, Óscar GARCÍA-PÉREZ u. a. „Efficiently handling temporal knowledge in an HTN planner“. In: *In 16th International Conference on Automated Planning and Abstract Reasoning for Planning and Coordination Scheduling (ICAPS-06)*. AAAI, 2006, S. 63–72 (siehe S. 25).
- [CG05] Ulrich CIMOLINO und Arvid GRAEGER. *Standardeinsatzregel*. Übersicht der veröffentlichten Broschüren www.standardeinsatzregel.org (abgerufen 01.12.2013). 2005 (siehe S. 14, 15, 37, 51).
- [Che76] P. Pin-Shan CHEN. „The entity-relationship model toward a unified view of data“. In: *ACM Trans. Database Syst.* 1.1 (März 1976), S. 9–36 (siehe S. 102).
- [CP07] Udo B. CRESPIN und Hanno PETER, Hrsg. *Handbuch für den Organisatorischen Leiter*. Stumpf + Kossendey Verlagsgesellschaft mbH, 2007 (siehe S. 14).
- [DAML] *DAML: The DARPA Agent Markup Language Homepage*. <http://daml.org> (abgerufen 20.12.2013) (siehe S. 83).
- [DHS06] DHS. *Writing Guide for Standard Operating Procedures*. U.S. Department of Homeland Security (DHS). 2006 (siehe S. 51).
- [DIN13050] DIN 13050. *Rettungswesen – Begriffe (Emergency services – Terms and definitions)*. DIN Deutsches Institut für Normung e. V., Beuth-Verlag. Ersatz für: DIN 13050:1996-06. 2009 (siehe S. 10).
- [DLCN] Evgeny ZOLIN. *The Description Logic Complexity Navigator: Complexity of reasoning in Description Logics*. <http://www.cs.man.ac.uk/~ezolin/dl/> (abgerufen am 20.12.2013) (siehe S. 84).
- [DW90] Asaf DEGANI und Earl L. WIENER. *Human Factors of Flight-Deck Checklists: The Normal Checklist*. Techn. Ber. NASA contractor report; NASA CR-177549, National Aeronautics & Space Administration, Ames Research Center, 1990 (siehe S. 2, 34, 35, 37).
- [DW91] Thomas L. DEAN und Michael P. WELLMAN. *Planning and control*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1991 (siehe S. 33).
- [DW93] A. DEGANI und E. L. WIENER. „Cockpit Checklists: Concepts, Design, and Use.“ In: *Human Factors* 35 (1993), S. 345–359 (siehe S. 2, 34, 35, 37).
- [Dör75] D. DÖRNER. *Problemlösen als Informationsverarbeitung*. Stuttgart: Kohlhammer, 1975 (siehe S. 1).
- [EHN94a] Kutluhan EROL, James A. HENDLER und Dana S. NAU. „HTN Planning: Complexity and Expressivity“. Englisch. In: *AAAI*. 1994, S. 1123–1128 (siehe S. 21).
- [EHN94b] Kutluhan EROL, James A. HENDLER und Dana S. NAU. „UMCP: A Sound and Complete Procedure for Hierarchical Task-Network Planning.“ In: *Proceedings of the 2nd International Conference on Artificial Intelligence Planning Systems (AIPS-94)*. 1994, S. 249–254 (siehe S. 21).
- [Ell11] Klaus ELLINGER. *Kursbuch Notfallmedizin: Orientiert am bundeseinheitlichen Curriculum Zusatzbezeichnung Notfallmedizin*. Deutscher Ärzte-Verlag, 2011 (siehe S. 15).
- [End00] M. R. ENDSLEY. „Theoretical underpinnings of situation awareness: a critical review“. In: *Situation Awareness Analysis and Measurement*. Hrsg. von M. R. ENDSLEY und D. J. GARLAND. Mahwah, NJ, USA: Lawrence Erlbaum Associates, 2000 (siehe S. 111).
- [Ero96] Kutluhan EROL. „Hierarchical Task Network planning: formalization, analysis, and implementation“. Advisors: Nau, D. and Hendler, J. Diss. University of Maryland, 1996 (siehe S. 22).

- [FaCT++] *FaCT++: OWL-DL Reasoner*. Department of Computer Science, University of Manchester, <http://owl.man.ac.uk/factplusplus/> (abgerufen am 20.12.2013) (siehe S. 85, 120).
- [FCG+06] Juan FDEZ-OLIVARES, Luis CASTILLO, Oscar GARCÍA-PÉREZ u. a. „Bringing Users and Planning Technology Together. Experiences in SIADEX“. In: *Proc. 16th International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI Press, 2006, S. 11–20 (siehe S. 25).
- [FEM99] FEMA. *Developing Effective Standard Operating Procedures For Fire and EMS Departments*. Hrsg. von Federal Emergency Management Agency’s United States Fire Administration (FEMA). United States Fire Administration, 1999 (siehe S. 51).
- [FG97] Dieter FREY und Siegfried GREIF. *Sozialpsychologie: Ein Handbuch in Schlüsselbegriffen*. Bd. 4. BeltzPVU, 1997 (siehe S. 66–69).
- [Fox97] Maria FOX. „Natural Hierarchical Planning Using Operator Decomposition“. In: *Proceedings of the 4th European Conference on Planning: Recent Advances in AI Planning*. ECP ’97. London, UK: Springer-Verlag, 1997, S. 195–207 (siehe S. 21).
- [FwDV100] *Feuerwehr-Dienstvorschrift 100*. Projektgruppe Feuerwehr-Dienstvorschriften des Ausschusses für Feuerwehrangelegenheiten, Katastrophenschutz und zivile Verteidigung (AFKzV), 1999, (siehe S. 6, 10, 13, 111).
- [Gaw10] Atul GAWANDE. *The Checklist Manifesto: How to Get Things Right*. Profile Books, 2010 (siehe S. 2, 34, 35, 37, 40, 44).
- [Gei95] K. GEIHS. *Client-Server-Systeme*. Thomson’s aktuelle Tutorien. Internat. Thomson Publ., 1995 (siehe S. 77).
- [GF95] Michael GRÜNINGER und Mark S. FOX. „Methodology for the Design and Evaluation of Ontologies“. In: 1995 (siehe S. 95, 97).
- [GFC04] Asunción GÓMEZ-PÉREZ, Mariano FERNÁNDEZ-LÓPEZ und O. CORCHO-GARCIA. *Ontological Engineering: With examples from the areas of Knowledge Management, e-Commerce and the Semantic Web (Advanced Information and Knowledge Processing)*. Springer, Berlin, 2004 (siehe S. 95).
- [GHA+05] J. GANCET, G. HATTENBERGER, R. ALAMI u. a. „Task Planning and Control for a multi-UAV system: architecture and algorithms“. In: *IEEE International Conference on Intelligent Robots and Systems, Edmonton (Canada)*. 2005 (siehe S. 21).
- [GIL98] Melinda T. GERVASIO, Wayne IBA und Pat LANGLEY. „Case-Based Seeding for an Interactive Crisis Response Assistant“. In: *In D.W. Aha & J.J. Daniels (Eds.) Case-Based Reasoning Integrations: Papers from the 1998 Workshop (Technical Report WS-98-15)*. Menlo. AAAI Press, 1998 (siehe S. 22).
- [GKB10] Michael R. GRÜNINGER, Markus KOHLER und Giancarlo BUONO. „Too much plane for one man to fly – Checklists“. In: *Bart International* 125 (2010), S. 78–79 (siehe S. 32).
- [GKM+11] A. GABDULKHAKOVA, B. KÖNIG-RIES, M. MÄHLER u. a. „Identifying and Supporting Information Needs in Mass Casualty Incidents – an Interdisciplinary Approach“. In: *Proceedings of the 8th International ISCRAM Conference*. Lisbon, Portugal. 2011 (siehe S. 50).
- [GLB86] E. GROCHLA, H. LIPPOLD und J. BREITHARDT. *Prüflisten zur Schwachstellenermittlung in Büro und Verwaltung*. Maschinenbau-Verlag, 1986 (siehe S. 32).
- [GNT04] Malik GHALLAB, Dana NAU und Paolo TRAVERSO. *Automated Planning: Theory and Practice*. Morgan Kaufmann Publishers, 2004 (siehe S. 21–23, 64, 65).
- [Gro04] Gudela GROTE. „Uncertainty management at the core of system design“. In: *Annual Reviews in Control* 28.2 (2004), S. 267–274 (siehe S. 39).
- [Gru03] David GRUSENMEYER. „Developing Effective Standard Operating Procedures“. In: (2003). Sr. Extension Associate PRO-DAIRY, Cornell Universit (siehe S. 51, 105).

- [Gru93] Thomas R. GRUBER. „A translation approach to portable ontology specifications“. In: *Knowledge Acquisition 5* (1993), S. 199–220 (siehe S. 81).
- [Hac06] Achim HACKSTEIN. *Einsatztaktik – Rettungsdienst kompakt Band 2*. Hrsg. von Frank FLAKE und Klaus RUNGGALDIER. Stumpf + Kossendey Verlagsgesellschaft mbH, 2006 (siehe S. 11).
- [Hac97] Winfried HACKER. *Allgemeine Arbeitspsychologie. Psychische Regulation von Arbeitstätigkeiten*. Huber, Bern, 1997 (siehe S. 67).
- [HB11] Matthew HORRIDGE und Sean BECHHOFFER. „The OWL API: A Java API for OWL ontologies“. In: *Semantic Web 2.1* (Jan. 2011), S. 11–21 (siehe S. 85).
- [HBS13] M. HOFMANN, H. BETKE und S. SACKMANN. „A Novel Architecture for Disaster Response Workflow Management Systems“. In: *Proceedings of the Information Systems for Crisis Response and Management conference (ISCRAM)*. 2013, S. 338–343 (siehe S. 21).
- [Hei09] Rudi HEIMANN. „Entscheidungsfindung in polizeilichen Einsatzlagen – Softwareunterstütztes Informations- und Kommunikationsmanagement“. In: *Lecture Notes in Informatics, INFORMATIK 2009 – Im Focus das Leben*. Hrsg. von Rüdiger Reischuk (ed.) STEFAN FISCHER (ED.) Erik Maehle (ed.) Gesellschaft für Informatik, Bonn. 2009, S. 1378–1392 (siehe S. 16).
- [HJW94] C. W. HOLSAPPLE, V. S. JACOB und Andrew B. WHINSTON. *Operations Research and Artificial Intelligence*. Ablex Publishing Corporation, 1994 (siehe S. 60).
- [HMW01] Volker HAARSLEV, Ralf MÖLLER und Michael WESSEL. „The Description Logic ALCNHR+ extended with Concrete Domains: A Practically Motivated Approach“. In: *In Proceedings of International Joint Conference on Automated Reasoning, IJ-CAR'2001*. Springer, 2001, S. 29–44 (siehe S. 84).
- [HP06] B. M. HALES und P. J. PRONOVOST. „The checklist – a tool for error management and performance improvement“. In: *Journal of Critical Care* 21 (2006), S. 231–235 (siehe S. 35, 40).
- [HPK+12] A. Von der HEYDE, P. PRESTING, A. KLUGE u. a. „Social norms and their impact on safety-related rule violations in process control: Does it make a difference if operators are aware that residents will be injured?“ In: *Proceedings of the 56th Annual Meeting of the Human Factors and Ergonomics Society (HFES)*. 2012 (siehe S. 15).
- [HPM+07] Ian HORROCKS, Peter F. PATEL-SCHNEIDER, Deborah L. MCGUINNESS u. a. „OWL: a Description Logic Based Ontology Language for the Semantic Web“. In: *The Description Logic Handbook: Theory, Implementation, and Applications (2nd Edition)*. Hrsg. von Franz BAADER, Diego CALVANESE, Deborah MCGUINNESS u. a. Cambridge University Press, 2007. Kap. 14 (siehe S. 84).
- [HS98] A. R. HALE und P. SWUSTE. „Safety rules: procedural freedom or action constraint?“ In: *Safety Science* 29.3 (Aug. 1998), S. 163–177 (siehe S. 39).
- [Jab95] Stefan JABLONSKI. *Workflow-Management-Systeme: Modellierung und Architektur*. International Thomson Publishing, 1995 (siehe S. 18, 76).
- [Jac98] Peter JACKSON. *Introduction to Expert Systems*. 3rd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998 (siehe S. 20).
- [KG12] Annette KLUGE und Britta GRAUEL. „Checklisten und Arbeitshilfen („Job Aids“) – ist die Gestaltung wichtig?“ In: *Wirtschafts- & Organisationspsychologie, Komplexität und Lernen – Newsletter für Organisationales Lernen, Simulation und Training*. 24. Fachbereich Wirtschafts- und Organisationspsychologie, Fakultät für Ingenieurwissenschaften, Fachbereich Informatik und Angewandte Kognitionswissenschaft & OPSY – HSG, 2012 (siehe S. 31).

- [KGB12] A. KLUGE, B. GRAUEL und D. BURKOLTER. „Combining principles of Cognitive Load Theory and diagnostic error analysis for designing job aids: Effects on motivation and diagnostic performance in a process control task“. In: *Applied Ergonomics (2012)*. Elsevier, 2012 (siehe S. 31).
- [KGGK+10] U. KRÜGER, A. GABDULKHAKOVA, B. KÖNIG-RIES u. a. „Semantic Service Infrastructure for Intelligent Checklist Support Systems“. In: *International Workshop on Emergency Management through Service Oriented Architectures*. Ghent, 2010 (siehe S. 3, 20, 45, 56, 75).
- [KHK+06] K. G. KANZ, P. HORNBURGER, M. V. KAY u. a. „mSTaRT-Algorithmus für Sichtung, Behandlung und Transport bei einem Massenanfall von Verletzten“. In: *Notfall + Rettungsmedizin 3.3* (2006). Springer Medizin Verlag, S. 264–270 (siehe S. 1).
- [Kla01] Joachim KLAUSNER. „Planen und intelligentes Workflow-Management“. Diss. Friedrich-Schiller-Universität Jena, 2001 (siehe S. 18, 76).
- [KNS92] G. KELLER, M. NÜTTGENS und A. W. SCHEER. „Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK)“. In: *Wirtschaftsinformatik 89* (1992), S. 1579–1587 (siehe S. 18, 106).
- [KPC06] Stasinos KONSTANTOPOULOS, Georgios PALIOURAS und Symeon CHATZINOTAS. „SHARE-ODS: An Ontology Data Service for Search and Rescue Operations“. Englisch. In: *SETN*. 2006, S. 525–528 (siehe S. 20).
- [KPS+08] S. KONSTANTOPOULOS, J. POTTEBAUM, J. SCHON u. a. „Ontology-Based Rescue Operation Management“. In: *Mobile Response*. 2008, S. 112–121 (siehe S. 99).
- [Kry72] K. D. KRYTER. „Speech communication“. In: *Human engineering guide to equipment design*, H. Van Cott and R. G. Kinkade (1972). Washington, DC: U.S. Government Printing Office. (siehe S. 37).
- [KWB12] U. KRÜGER, F. WUCHOLT und C. BECKSTEIN. „Electronic Checklist Support for Disaster Response“. In: *9th Internat. Conference on Information Systems for Crisis Response and Management, Human Experiences in the Design of Crisis Response and Management Services and Systems*. Vancouver, Canada, 2012 (siehe S. 3, 17, 45, 78).
- [Lan11] Laura LANDRO. „The Secret to Fighting Infections“. In: *The Wall Street Journal (online)* (2011). Dow Jones and Company, Inc. (siehe S. 35).
- [LI09] Vita LANFRANCHI und Neil IRESON. „User requirements for a collective intelligence emergency response system“. In: *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*. BCS-HCI '09. Cambridge, UK: British Computer Society, 2009, S. 198–203 (siehe S. 44).
- [Lip09] Roland LIPP. *Lehrbuch für präklinische Notfallmedizin (LPN) Bd.4 Berufskunde und Einsatztaktik*. Stumpf + Kossendey Verlagsgesellschaft mbH, 2009 (siehe S. 5).
- [LPK09] Christian LINDEMANN, Stephan PRÖDEL und Rainer KOCH. *Modellierung von Prozessen in der Feuerwehrdomäne zur Identifikation von Informationsbedarfen*. Workshop: IT-Unterstützung von Rettungskräften, Jenaer Schriften zur Mathematik und Informatik Math/Inf/03/09. 2009 (siehe S. 18, 20).
- [LSB13] Shuangyan LIU, Duncan SHAW und Christopher BREWSTER. „Ontologies for Crisis Management: A Review of State of the Art in Ontology Design and Usability“. In: *Proceedings of the Information Systems for Crisis Response and Management conference (ISCRAM)*. 2013, S. 349–359 (siehe S. 98).
- [Lut03] Carsten LUTZ. „Description Logics with Concrete Domains – A Survey“. In: *Advances in Modal Logics Volume 4*. King’s College Publications, 2003 (siehe S. 84, 102).
- [Lüd07] Sascha Rolf LÜDER. *Recht und Praxis der nichtpolizeilichen Gefahrenabwehr*. BWV Berliner-Wissenschaft (August 31, 2007), 2007 (siehe S. 9).

- [MAB+99] Héctor MUÑOZ-AVILA, David W. AHA, Len BRESLOW u. a. *HICAP: An Interactive Case-Based Planning Architecture and its Application to Noncombatant Evacuation Operations*. 1999 (siehe S. 22, 23).
- [Man09] Tanja MANSER. „Komplexität handhaben – Handeln vereinheitlichen – Organisationen sicher gestalten“. In: *Human Factors*. Hrsg. von Petra BADKE-SCHAUB, Gesine HOFINGER, u. a. Springer Berlin Heidelberg, 2009. Kap. 17, S. 273–288 (siehe S. 1, 30).
- [MBM+07] David MARTIN, Mark BURSTEIN, Drew MCDERMOTT u. a. „Bringing Semantics to Web Services with OWL-S“. In: *World Wide Web* 10.3 (2007), S. 243–277 (siehe S. 98).
- [Min75] Marvin MINSKY. „A Framework for Representing Knowledge“. In: *The Psychology of Computer Vision*. Hrsg. von P. WINSTON. McGraw-Hill, New York, 1975, S. 211–277 (siehe S. 83).
- [Mit97] Steven W. MITCHELL. „A hybrid architecture for real-time mixed-initiative planning and control“. In: *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence*. AAAI’97/IAAI’97. Providence, Rhode Island: AAAI Press, 1997, S. 1032–1037 (siehe S. 22).
- [MSH09] Boris MOTIK, Rob SHEARER und Ian HORROCKS. „Hypertableau Reasoning for Description Logics“. In: *Journal of Artificial Intelligence Research* 36 (2009), S. 165–228 (siehe S. 87).
- [NAI+03] D. S. NAU, T.-C. AU, O. ILGHAMI u. a. „SHOP2: An HTN Planning System“. In: *Journal of Artificial Intelligence Research* 20 (2003), S. 379–404 (siehe S. 25).
- [NK95] Bernhard NEBEL und Jana KOEHLER. „Plan Reuse Versus Plan Generation: A Theoretical and Empirical Analysis“. In: *Artificial Intelligence* 76.1-2 (1995), S. 427–454 (siehe S. 68).
- [NKW08] Ulrich NATKE, Tobias KRAMER und Kirsten WOLFF. „STABOS@IG NRW – Ein Meldewesen unterstützt die Krisenstabsarbeit bei der nichtpolizeilichen Gefahrenabwehr und der Polizei in Nordrhein-Westfalen“. In: 2 (2008). Schriftenreihe der Landesdatenverarbeitungszentrale (LDVZ) Nordrhein-Westfalen, S. 17–18 (siehe S. 16).
- [NM01] Natalya F. NOY und Deborah L. MCGUINNESS. *Ontology Development 101: A Guide to Creating Your First Ontology*. Techn. Ber. Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880. Stanford Knowledge Systems Laboratory, 2001 (siehe S. 83, 95, 97, 98).
- [NMAC+01] Dana NAU, Héctor MUNOZ-AVILA, Yue CAO u. a. „Total-Order Planning with Partially Ordered Subtasks“. In: *Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, August 2001*. 2001 (siehe S. 69).
- [Nor88] Donald A. NORMAN. *The Psychology Of Everyday Things*. Basic Books, Juni 1988 (siehe S. 29).
- [OWL-API] W3C. *OWL API: Java API for OWL Ontologies*. W3C Member Submission. <http://www.w3.org/Submission/OWL-S> (letzter Abruf: Aug. 2013). 2004 (siehe S. 120).
- [OWL-S] W3C. *OWL-S: Semantic Markup for Web Services*. W3C Member Submission. <http://www.w3.org/Submission/OWL-S> (abgerufen am 20.12.2013). 2004 (siehe S. 24).
- [OWL2] *OWL 2 Web Ontology Language, W3C Recommendation 11 December 2012*. W3C. <http://www.w3.org/TR/owl2-overview/> (abgerufen am 20.12.2013) (siehe S. 84).

- [Par10] C. PARK. *The Case for Electronic Checklists in Hospitals*. The Hospital Management.net. www.hospitalmanagement.net/features/feature85551 (abgerufen am 20.12.2013). 2010 (siehe S. 39).
- [PD91] E. PALMER und A. DEGANI. „Electronic checklists: Evaluation of two levels of automation.“ In: *Proceedings of the Sixth International Aviation Psychology Symposium*. Columbus, Ohio: The Ohio State University, 1991, pp. 178–183 (siehe S. 44).
- [PDV100] *PDV100 – Polizeidienstvorschrift: Führung und Einsatz der Polizei (VS-NfD)*. Verlag Deutsche Polizeiliteratur (VDP) GmbH. 1999 (siehe S. 7).
- [Pellet] *Pellet: OWL 2 Reasoner for Java*. Clark & Parsia, LLC. <http://clarkparsia.com/pellet> (abgerufen am 20.12.2013) (siehe S. 85).
- [Pet62] Carl Adam PETRI. „Kommunikation mit Automaten“. ger. Diss. Universität Hamburg, 1962 (siehe S. 18).
- [PM01] Hanno PETER und Klaus MAURER. *Die Leitstelle beim MANV*. Hrsg. von Hanno PETER und Klaus MAURER. Stumpf + Kossendey Verlagsgesellschaft mbH, 2001 (siehe S. 5, 6, 8, 58, 61).
- [PMU01] Hanno PETER, Thomas MITSCHKE und Theodor UHR. *Notarzt und Rettungsassistent beim MANV – SEGmente 3*. Hrsg. von Klaus Maurer und HANNO PETER. Stumpf + Kossendey Verlagsgesellschaft mbH, 2001 (siehe S. 8, 13).
- [PR09] Gertraud PEINEL und Thomas ROSE. *Prozessmodellierung für das Notfallmanagement*. Kurzbeiträge zum Workshop: IT-Unterstützung von Rettungskräften. Lübeck. Fraunhofer FIT, Schloss Birlinghoven. 2009 (siehe S. 18, 19, 28, 106).
- [Protégé] *Protégé – a free, open source ontology editor and knowledgebase framework*. Stanford Center for Biomedical Informatics Research. <http://protege.stanford.edu/> (abgerufen am 20.12.2013) (siehe S. 85).
- [PRW12a] G. PEINEL, T. ROSE und A. WOLLERT. „The Myth of Business Process Modelling for Emergency Management Planning“. In: *9th International Conference on Information Systems for Crisis Response and Management*. Track Research Methods. Vancouver, Canada, 2012 (siehe S. 19, 44).
- [PRW12b] Gertraud PEINEL, Thomas ROSE und Alexander WOLLERT. „Cross-Organizational Preplanning in Emergency Management with IT-Supported Smart Checklists“. In: *7th Security Research Conference, September 4th - 6th, 2012*. 2012 (siehe S. 19, 21, 28, 41, 45, 78, 88).
- [Racer] Racer Systems GmbH & Co. KG. *Renamed ABox and Concept Expression Reasoner*. <http://www.racer-systems.com/> (abgerufen am 20.12.2013) (siehe S. 85).
- [Ras82] Jens RASMUSSEN. „Human errors. A taxonomy for describing human malfunction in industrial installations“. In: *Journal of Occupational Accidents* 4.2-4 (1982), S. 311–333 (siehe S. 29).
- [Rea90] James T. REASON. *Human Error*. Cambridge University Press, 1990 (siehe S. 1, 29, 31, 107).
- [Rea94] James T. REASON. *Menschliches Versagen: Psychologische Risikofaktoren und moderne Technologien*. Spektrum Psychologie. Spektrum Akademischer Verlag, 1994 (siehe S. 1).
- [Rei07] Ralph REINWARTH. „Standard Operating Procedures als Entscheidungsgrundlage in der Luftfahrt.“ In: Hrsg. von Stefan STROHSCHNEIDER. Bd. 2. Menschen in kritischen Situationen – Schriftenreihe der Plattform Menschen in komplexen Arbeitswelten e.V. Verlag für Polizei-Wissenschaft, 2007. Kap. 2, S. 14–23 (siehe S. 43).
- [RG91] A. ROSSETT und J. GAUTIER-DOWNES. *A handbook of job aids*. Pfeiffer, 1991 (siehe S. 31).

- [Ros04] Patrick ROSS. „Human Factors Issues of the Aircraft Checklist“. In: *JAAER* 13.2 (2004), S. 9–14 (siehe S. 37, 40).
- [RPA08] Thomas ROSE, Gertraud PEINEL und Emilija ARSENOVA. „Process Management Support for Emergency Management Procedures“. In: *Proceedings of Collaboration and the Knowledge Economy: Issues, Applications, Case Studies. Pt.2 : eChallenges e-2008 Conference*. Hrsg. von P. CUNNINGHAM. October. IOS Press, Amsterdam, NL, 2008, S. 1069–1076 (siehe S. 15, 18).
- [RS08] John ROBILLARD und Roger SAMBROOK. *USAF Emergency and Incident Management Systems: A Systematic Analysis of Functional Requirements*. en. University of Colorado, Colorado Springs (UCCS). 2008 (siehe S. 44).
- [RW07] Uwe RÜPPEL und Armin WAGENKNECHT. „Improving emergency management by formal dynamic process-modelling“. In: *Proceedings of the 24th Conference on Information Technology in Construction (W78), Maribor, Slovenia*. 2007, S. 559–564 (siehe S. 21).
- [Sac73] Earl D. SACERDOTI. „Planning in a hierarchy of abstraction spaces“. In: *Proceedings of the 3rd international joint conference on Artificial intelligence*. Stanford, USA: Morgan Kaufmann Publishers Inc., 1973, S. 412–422 (siehe S. 21).
- [Sac75] Earl David SACERDOTI. „A structure for plans and behavior.“ Diss. Stanford, CA, USA, 1975 (siehe S. 64).
- [SB02] Bernd SCHATTENBERG und Susanne BIUNDO. „On the Identification and Use of Hierarchical Resources in Planning and Scheduling“. In: *Proceedings of the 6th International Conference on AI Planning and Scheduling (AIPS'02)*. Hrsg. von Malik GHALLAB, Joachim HERTZBERG und Paolo TRAVERSO. Toulouse, France: AAAI Press, Menlo Park, California, 2002, S. 263–272 (siehe S. 26).
- [SBB08] Bernd SCHATTENBERG, Susanne BIUNDO und Julien BIDOT. „Werkzeugunterstützung für konsistente Domänenmodelle“. In: *Proceedings of the 22nd Workshop on Planen, Scheduling und Konfigurieren, Entwerfen (PuK 2008)*. 2008 (siehe S. 27).
- [SBG+09] Bernd SCHATTENBERG, Julien BIDOT, Sascha GESSLER u. a. „A Framework For Interactive Hybrid Planning“. In: *KI 2008: Advances in Artificial Intelligence, Proceedings of the 32nd Annual German Conference on AI*. Hrsg. von Bärbel MERTSCHING, Marcus HUND und Zaheer AZIZ. Bd. 5803. Lecture Notes in Artificial Intelligence. Paderborn, Germany: Springer, 2009, S. 17–24 (siehe S. 26).
- [SBI85] James G. SCHMOLZE, Bolt BERANEK und Newman INC. „An overview of the KL-ONE knowledge representation system“. In: *Cognitive Science* 9 (1985), S. 171–216 (siehe S. 83).
- [SBQ+09] M. SOBOLL, B. BINDER, C. QUIX u. a. „Vorgehensmodelle und Implementierungsfragen – Akquisition – Lokalisierung – soziale Maßnahmen – Werkzeuge: 16. Workshop der Fachgruppe WI-VM der Gesellschaft für Informatik e. V. (GI)“. In: Hrsg. von Reinhard HÖHN und Oliver LINSSEN. Shaker Verlag, 2009. Kap. Prozessmodellierung der mobilen Datenerfassung für den Rettungsdienst bei einer Großschadenslage, S. 97–108 (siehe S. 10, 18).
- [Sch06] H. SCHAUB. „Enzyklopädie der Psychologie C/II/8: Denken und Problemlösen“. In: Hrsg. von J. Funke (HRSG.) Göttingen: Hogrefe, 2006. Kap. Störungen und Fehler beim Denken und Problemlösen, S. 447–478 (siehe S. 1).
- [Sch07] Jörg SCHMIDT. *Einsatzkonzept MANV Überörtlich (ÜMANV)*. Rheinische Projektgruppe „MANV Überörtlich“, Köln. Einsatzkonzept zur Bewältigung von Großschadenslagen mit 500 - 1200 Patienten durch überörtliche Unterstützung, Zusammenarbeit mit der BF Köln. 2007 (siehe S. 19).
- [Sch08] Daniel SCHLIER. *Fehlerquelle Mensch: Ein industrieübergreifendes Problem: Human Factors und die "Dirty Dozen" in luftfahrtfremden Arbeitsprozessen*. Vdm Verlag Dr. Müller (April 2008), 2008 (siehe S. 26, 27).

- [Scr07] Michael SCRIVEN. *The Logic and Methodology of Checklists*. Western Michigan University. http://www.wmich.edu/evalctr/archive_checklists/papers/logic&methodology_dec07.pdf (abgerufen am 20.12.2013). 2007 (siehe S. 32).
- [Sim62] Herbert A. SIMON. „The architecture of complexity“. In: *Proceedings of the American Philosophical Society*. Bd. 106. 6. American Philosophical Society, Philadelphia, 1962, S. 467–482 (siehe S. 1).
- [SMS10] S. SCHUMANN, M. MÄHLER und S. STROHSCHNEIDER. „Die machen ihren eigenen Stiefel: Interorganisationale Zusammenarbeit von Behörden und Organisationen mit Sicherheitsaufgaben“. In: *Polizei & Wissenschaft* 3 (2010), S. 41–49 (siehe S. 114).
- [SS04] Steffen STAAB und Rudi STUDER, Hrsg. *Handbook on Ontologies*. International Handbooks on Information Systems. Springer, 2004 (siehe S. 83).
- [Sta10] Susanne STARKE. *Führungskultur in High Risk Environments: Eine empirische Untersuchung in den Arbeitsfeldern Polizei, Medizin, Business Continuity Management*. Verlag für Polizei-Wissenschaft, 2010 (siehe S. 115).
- [StadtVS06] STADT VILLINGEN-SCHWENNINGEN, BÜRGERAMT, FEUERWEHR UND KATASTROPHENSCHUTZ. *Standard-Einsatz-Regeln – Handlungsanweisungen für die Aus-, Fortbildung und den Einsatz*. www.feuerwehr-vs.de/standard_einsatz_regeln_ff_vs.pdf (abgerufen am 20.12.2013). 2006 (siehe S. 52, 138).
- [Sus75] Gerald Jay SUSSMAN. „A Computer Model of Skill Acquisition“. Diss. MIT, 1975 (siehe S. 69).
- [Tat00] Austin TATE. „Intelligible AI Planning – Generating Plans Represented as a Set of Constraints“. In: *proceedings of ES2000, The Twentieth British Computer Society Special Group on Expert Systems International Conference on Knowledge Based Systems and Applied Artificial Intelligence*. Springer, 2000, S. 3–16 (siehe S. 23).
- [Tat76] Austin TATE. *Project Planning Using a Hierarchic Non-linear Planner*. Techn. Ber. Department of Artificial Intelligence Research Report No 25, University of Edinburgh, Edinburgh, 1976 (siehe S. 21).
- [TDB+04] Austin TATE, Jeff DALTON, Jeffrey M. BRADSHAW u. a. *Coalition Search and Rescue – Task Support: Intelligent Task Achieving Agents on the Semantic Web*. Techn. Ber. Artificial Intelligence Applications Institute The University of Edinburgh & Florida Institute for Human & Machine Cognition (IHMC), 2004 (siehe S. 21, 23, 24).
- [TDD96] Austin TATE, Brian DRABBLE und Jeff DALTON. *O-Plan: a Knowledge-Based Planner and its Application to Logistics*. 1996 (siehe S. 23).
- [TDS02] A. TATE, J. DALTON und J. STADER. „I-P2 – Intelligent Process Panels to Support Coalition Operations“. In: *Proceedings of the Second International Conference on Knowledge Systems for Coalition Operations (KSCO-2002)*, Toulouse, France, 2002 (siehe S. 109).
- [Ten88] Josh D. TENENBERG. „Abstraction in planning“. Order No: GAX88-16885. Diss. Rochester, NY, USA, 1988 (siehe S. 21).
- [ThürRettG] *Thüringer Rettungsdienstgesetz (ThüRettG)*. Thüringer Staatsanzeiger Nr. 8. 2008 (siehe S. 6).
- [UBJ+03] A. USZOK, J. BRADSHAW, R. JEFFERS u. a. „KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction, and Enforcement“. In: *Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks. POLICY '03*. Washington, DC, USA: IEEE Computer Society, 2003 (siehe S. 23).
- [UBJ+04] Andrzej USZOK, Jeffrey M. BRADSHAW, Matthew JOHNSON u. a. „KAoS Policy Management for Semantic Web Services“. In: *IEEE Intelligent Systems* 19 (2004), S. 32–41 (siehe S. 23).

- [UG96] Mike USCHOLD und Michael GRÜNINGER. „Ontologies: Principles, Methods and Applications“. In: *Knowledge Engineering Review* 11 (1996), S. 93–136 (siehe S. 97, 101).
- [Vol83] Walter VOLPERT. „Das Modell der hierarchisch-sequentiellen Handlungsorganisation“. In: *Kognitive und motivationale Aspekte der Handlung*; hrsg. von W. HACKER, W. VOLPERT und M. von CRANACH. Bern: Huber, 1983, S. 38–58 (siehe S. 68).
- [Wag06] U. WAGNER. „Situation Awareness: Ein Konzept für den Rettungsdienst“. In: *Rettungsdienst* 29 (2006). Stumpf + Kossendey Verlag, S. 766–771 (siehe S. 111).
- [Wan06] Hai WANG. *Frames and OWL side by side*. The University of Manchester. Presentation to Protégé, http://protege.stanford.edu/conference/2006/submissions/slides/7.2wang_protege2006.pdf (abgerufen am 20.12.2013). 2006 (siehe S. 96).
- [WD92] David WILKINS und Roberto V. DESIMONE. „Applying an AI Planner to Military Operations Planning“. In: *Intelligent Scheduling*. Morgan Kaufmann, 1992, S. 685–709 (siehe S. 21).
- [Wei02] Robert WEIHMANN. *Kriminalistik, ein Grundriss für Studium und Praxis*. 6. Auflage. Verlag deutsche Polizeiliteratur, 2002 (siehe S. 7).
- [Wic84] Christopher D. WICKENS. *Engineering psychology and human performance*. Merrill, 1984 (siehe S. 37).
- [Wie94] G. WIENDIECK. *Arbeits- und Organisationspsychologie*. Berlin, München: Quintessenz, 1994 (siehe S. 67).
- [Wil10] Robert M. WILLIAMSON. „Checklists: The Often Overlooked Tool“. In: (2010). Strategic Work Systems, Inc. Columbus, NC (siehe S. 35).
- [WKK11] F. WUCHOLT, U. KRÜGER und S. KERN. *Mobiles Checklisten-Support-System im Einsatzszenario einer Großschadenslage*. Lecture Notes in Informatics, Band P192, Informatik 2011, Workshop IT-Unterstützung für Rettungskräfte, Berlin. 2011 (siehe S. 3, 12, 19, 45).
- [WM11] Alexander WOLLERT und Ganga MANNABETH. *Plan Authoring Applications for Emergency Management*. Workshop: IT-Unterstützung von Rettungskräften. 2011 (siehe S. 19).
- [WMY11] F. WUCHOLT, M. MÄHLER und Y. YILDIRIM-KRANNIG. „Framework conditions and fields of application of an IT-Rescue Management Support System (IT-RMSS) for authorities and organizations with safety responsibilities (BOS) in mass casualty incident (MCI)“. In: *Lecture Notes in Informatics (LNI)-Proceedings. Informatik 2011, Workshop IT-Unterstützung für Rettungskräfte, Berlin, 2011*. 2011 (siehe S. 44, 48, 114).
- [WP10] Gerhard WICKLER und Stephen POTTER. „Standard Operating Procedures: Collaborative Development and Distributed Use“. In: *7th International Conference on Information Systems for Crisis Response and Management, Seattle, USA, 2010* (siehe S. 24).
- [WTP06] Gerhard WICKLER, Austin TATE und Stephen POTTER. „Using the <I-N-C-A> Constraint Model as a Shared Representation of Intentions for Emergency Response“. In: *Proceedings of the First International Workshop on Agent Technology for Disaster Management (ATDM), at the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Future*. 2006, S. 8–12 (siehe S. 24, 28, 76).
- [WYM+11] F. WUCHOLT, Y. YILDIRIM-KRANNIG, M. MAHLER u. a. „Cultural Analysis and Formal Standardised Language – a Mass Casualty Incident Perspective“. In: *8th International Conference on Information Systems for Crisis Response and Management. User Centred Design Process for Emergency Management Information Systems, Lisbon, Portugal, 2011* (siehe S. 3, 17, 73).

- [YM94] R. Michael YOUNG und Johanna D. MOORE. „DPOCL: A Principled Approach to Discourse Planning“. In: *In Proceedings of the 7th International Workshop on Natural Language Generation*. 1994, S. 13–20 (siehe S. 21).
- [YP+94] R. Michael YOUNG, Martha E. POLLACK, u. a. „Decomposition and Causality in Partial-Order Planning“. In: *In 2nd Int. Conf. on AI Planning Systems: AIPS-94*. 1994, S. 188–193 (siehe S. 21).
- [Fer99] M. FERNÁNDEZ-LÓPEZ. „Overview of Methodologies for Building Ontologies“. In: *Proc. Workshop on Ontologies and Problem-Solving Methods*. 1999 (siehe S. 95).

Photonachweise

Abbildung 4.5: RODGER JOHNSON, Quelle: <http://www.united-virtual.com/forum/viewtopic.php?f=36&t=15885>, <http://i225.photobucket.com/albums/dd159/rodgerjohnson/DSCF2186.jpg> (abgerufen am 20.12.2013), 2008, (siehe S. 38).

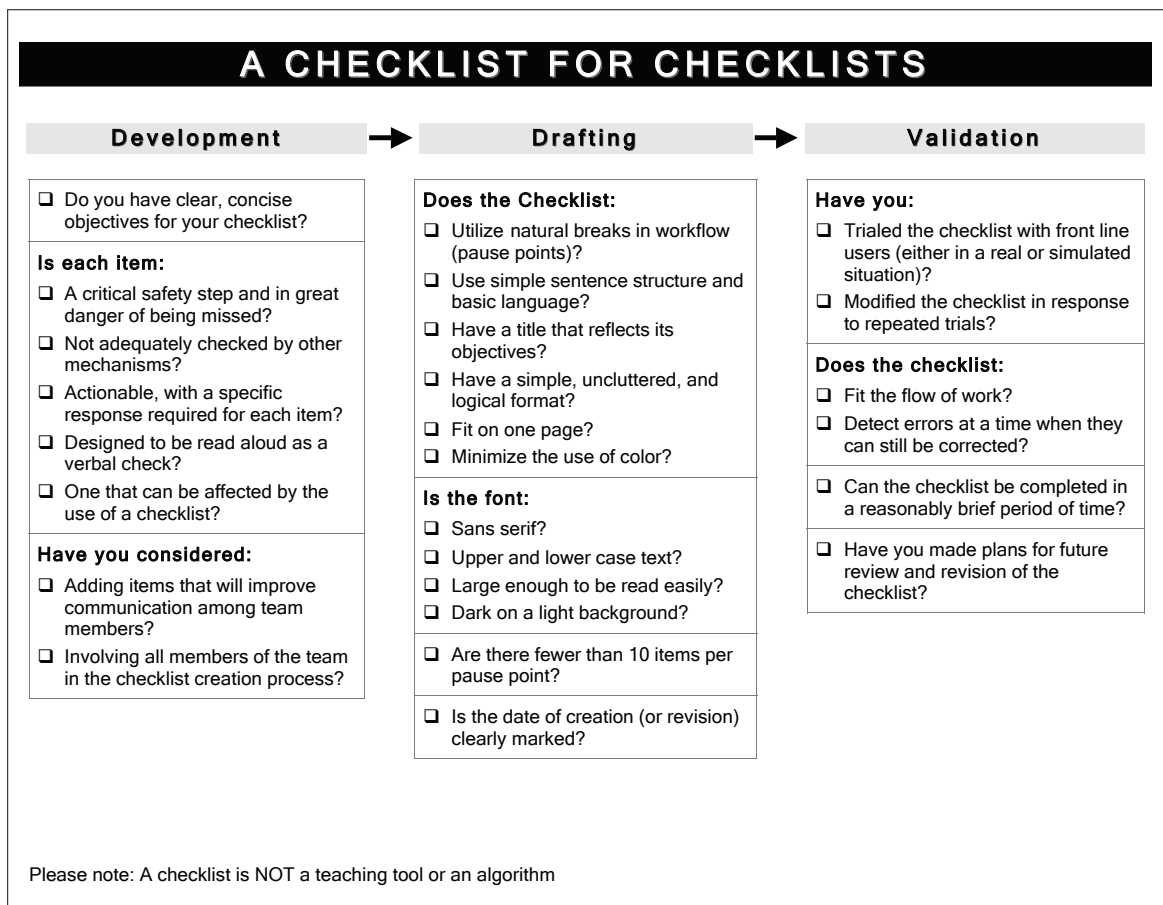
Abbildung 4.4: KENT WIEN, 2009, Lizenz: CC BY-NC 2.0, Quelle: http://farm3.static.flickr.com/2046/2186148441_8452d1c610_b.jpg (abgerufen am 20.12.2013), (siehe S. 38).

Abbildungsverzeichnis

2.1	MANV-Stufen versus Regelbetrieb (aus Sicht des Rettungsdienstes).	9
2.2	Reihenfolge des Eintreffens bestimmter Einsatzkräfte und Einsatzphasen.	11
2.3	Klassisches Modell des Führungsvorgangs (nach [FwDV100]).	13
2.4	Gelehrter Führungsvorgang aus Sicht der Feuerwehr (Quelle: [CP07]).	14
3.1	Zwei Ansätze: IT-Assistenz für manuelle vs. automatische Handlungsplanung. .	17
3.2	Anzeige der Task-Hierarchien im HICAP-Benutzerinterface. (Quelle: [MAB+99])	23
3.3	Einzelne Komponenten des CoSAR-TS GUI. (Quelle: [TDB+04])	24
3.4	Der sog. <i>lifecycle</i> von SIADEx in einem Waldbrandszenario. (Quelle: [FCG+06])	25
3.5	Auszug einer Task-Hierarchie aus Sicht des THW. (Quelle: [BS01])	27
4.1	Grundlegende Strategien im Umgang mit Unsicherheit (Quelle: [Man09]).	30
4.2	Checklisten für unterschiedliche Situations-Klassen.	34
4.3	Grade der Sicherheitsanforderungen.	35
4.4	Teil einer MCL (<i>Take-Off</i>) in einer <i>Boeing 757</i>	38
4.5	Elektronische Checkliste im Cockpit einer <i>Boeing 777</i>	38
4.6	Zusammenhang zwischen „Leitlinien“, SOPs und Checklisten.	41
4.7	Arten der Korrespondenzen zwischen einer Checkliste und ihrer SOP.	41
5.1	Beispiel einer definierten Rollenstruktur (aus [BOS-O]).	50
5.2	Beispiel einer Konkretisierung eines abstrakten Items.	57
5.3	Schema einer Konkretisierung abstrakter Items.	58
5.4	Rollenbegrenzung in einer Checklistenhierarchie.	59
5.5	Entwicklung einer CLH als AND/OR-Graph.	60
5.6	Grundprinzip der Angabe von alternativen Sub-Checklisten.	61
5.7	Ausprägung einer Checklistenhierarchie (1/2).	62
5.8	Ausprägung einer Checklistenhierarchie (2/2).	63
5.9	Das Prinzip der Task-Dekomposition.	65
5.10	Beispielhafter Auszug einer HTN-Planungs-Domäne als AND/OR-Graph.	66
5.11	Beispiel einer hierarchischen Aufgabendekodierung nach HACKER [Hac97].	67
5.12	Die Zyklische Einheit nach VOLPERT 1982 [FG97].	68
5.13	Das hierarchisch-sequenzielle Modell der Handlungsregulation [FG97].	69

5.14 Zuordnung der Aufgaben/Tasks über die Beschreibungen zu den Checklisten.	70
5.15 Auszug eines MANV-Organisationsaufbaus.	71
5.16 Annotation des Konzeptes „Bereitstellungsraum“ (siehe [BOS-O]).	72
5.17 Kulturanalyse und Knowledge Engineering.	73
6.1 Meta-Architekturskizze der wichtigsten CASIE-Komponenten.	75
6.2 Architektur eines WfMS (nach KLAUSNER [Kla01]).	76
6.3 Mögliche Klassen von Endgeräten.	78
6.4 Eigene Studie eines möglichen GUI (vgl. [KWB12]).	78
6.5 Die Komponente des CL-Repository.	80
6.6 Die Komponente des CL-Logbooks.	81
6.7 Architektur eines <i>Knowledge Base Systems</i> (KBS) (nach BAADER [BCM+03]).	82
6.8 OWL-Konstruktoren vs. DL (Quelle: [BHS08]).	85
6.9 OWL-Axiome vs. DL (Quelle: [BHS08]).	85
6.10 TBox-Auszug der BOS-Definitionen aus [BOS-O].	86
6.11 Beispiel eines Reasoningergebnisses (Protégé).	87
6.12 Explanation Schlussfolgerungsergebnis (Protégé).	87
6.13 Diagramm der möglichen Zustandsübergänge einer ICL.	88
6.14 Diagramm der Zustandsübergänge eines Items.	89
6.15 Singel-User- vs. Multi-User-Modus.	91
6.16 Aktivitätsdiagramm des reaktiven Systemverhaltens von CASIE.	92
6.17 Schritte der Ontologieentwicklung nach MCGUINNESS et al. [NM01].	95
6.18 Das GUI von Protégé (4.*) mit geladener BOS-Ontologie [BOS-O].	96
6.19 Beispielhafter Ausschnitt aus der BOS-Ontologie [BOS-O].	100
6.20 Auszug einer Rollendefinition der BOS-Ontologie in <i>Protégé</i>	101
6.21 Von SOPs zu Checklisten und zurück.	103
6.22 Beispielschema eines Prozessskeletts.	106
7.1 Schema des Prozessmonitorings.	109
7.2 a) Führungsvorgang vs. b) Situation Awareness im Entscheidungsprozess.	111

Anhang



Checkliste für eine Checkliste. Quelle: www.projectcheck.org/checklist-for-checklists.html (abgerufen am 20.12.2013)

Checkliste Brandmeldeanlage

Gerätehaus

- GF entnimmt Feuerwehrplan aus dem Schrank

Anfahrt

- Angriffstrupp rüstet sich mit PA aus.
- Anfahrt bis zum festgelegten Bereitstellungsraum (falls im Feuerwehrplan vermerkt)

Eintreffen

- „Lagemeldung auf Sicht“ an Leitstelle
- GF, A-Trupp und W-Trupp gehen zur BMZ (alle mit Funkgerät ausgerüstet)
- MA Trupp bleibt mit eingeschaltetem 2 m Band Funkgerät im Fahrzeug!
- Der Objektschlüssel wird **immer** aus dem Schlüsseltresor (FSD) entnommen

Auswertung der BMA

- GF, A-Trupp und W-Trupp begeben sich immer zur BMZ
- Aufschließen des Feuerwehrbedienfeldes
- Bei Verdacht auf Fehlalarm kann die Evakuierungshupe abgeschaltet werden
- Auswerten welche Melder ausgelöst haben und die Nummern aufschreiben
- Die dazugehörigen Feuerwehrlaufkarten auswählen
- GF und A-Trupp begeben sich zu dem ausgelösten Melder.
Mitzunehmen sind die Laufkarte, Funkgerät, Beleuchtungsmittel und **immer** der Objektschlüssel.
- Der W-Trupp bleibt mit einem Funkgerät an der BMZ
- Der A-Trupp öffnet Türen zu dem gefährdeten Bereich erst nach einem Türcheck
- Der GF gibt nach Erreichen des Melders umgehend Rückmeldung

Abschließende Maßnahmen

- Betriebsbuch der BMA schreiben und Meldernummer eintragen
- BMA in dem Feuerwehrbedienfeld zurückstellen
- Rückfrage in der Feuerwehrleitstelle und bestätigen lassen, dass der Alarm gelöscht ist und die Anlage wieder scharf ist.
- Feuerwehrbedienfeld wieder abschließen
- Alle Türen die aufgeschlossen wurden wieder abschließen
- Objektschlüssel wieder in das Depot stecken und um 90 Grad drehen
- Prüfen ob das FSD verriegelt.


Bei Problemen beim Zurückstellen der Anlage

- Niemals selbständig Anlage abstellen oder Gruppen aus Betrieb nehmen!!
- Lässt die Anlage sich nicht zurückstellen oder schließt das FSD nicht, die Anlage nochmals manuell auslösen (Druckknopfmelder oder Freischaltelement)
- Anlage dem Betreiber zu weiteren Maßnahmen übergeben

Boeing 747 - 400 Normal Checklist

Pre-Flight	
Parking Brake.....	On
Battery Switch.....	On
Standby Power.....	Auto
EEC Switches.....	Norm
Bus Tie Switches.....	Auto
GEN CONT Switches.....	Pushed In
Hydraulic Demand Pumps.....	Off
Engine Hydraulic Pumps.....	On
External Power.....	On
<small>Turn On External Power #1 Turn On External Power #2 If Available</small>	
Utility Bus Switches (L&R).....	On
NAV Light.....	On
Emergency Light Switch.....	ARM
Alternate Flap Selector.....	Off
Flap Lever / Position Indicator.....	Up / Agree
Landing Gear.....	Down
IRS Mode Selectors.....	NAV
FMC Initialization.....	Enter Position
Fuel Crossfeeds.....	On
Fuel Pumps.....	Off
Nacelle Anti-Ice.....	Off
Wing Anti-Ice.....	Off
Exterior Lights.....	As Required
<small>NAV Lights And Logo Lights On All Other Lights Should Be Off</small>	
Window Heat.....	On
Windshield Wiper Switches.....	Off
Passenger Oxygen Switch.....	Norm
Yaw Damper Switches.....	On
APU Bleed Switch.....	Off
APU.....	Start
<small>When APU AVAIL lights illuminate - Press APU Generator Switches (1&2)</small>	
Outflow Valves.....	Open
Landing Altitude Switch.....	Auto
Outflow Valve Manual Switches.....	Auto
Cabin Altitude Auto Selector.....	Norm
Passenger Temp.....	Auto
Flight Deck Temp.....	Auto
Trim Air Switch.....	On
Recirc Fan Switches (UPR & LWR).....	On
AFT Cargo Heat Switch.....	Off
Equipment Cooling Selector.....	Norm
<small>If OAT Is Less Than 70° Set To NORM If OAT Is Above 70° Set To OVRD</small>	
High Flow Switch.....	Off
Gasper Switch.....	On
Pack Switches (1-2-3).....	Norm
ISLN Switches.....	Open
APU Bleed Switch.....	On
<small>Allow APU To Operate One Minute Before Activating APU Bleed Switch</small>	
Engine Bleed Switches.....	On
FMC.....	Program
Mode Control Panel (MCP).....	Set
Flight Director Switch.....	On
Auto Throttle.....	Off
IAS/MACH.....	V2 +10
LNAV/VNAV.....	Arm
HDG.....	Set
<small>Enter Runway Heading</small>	
ALT.....	Set
<small>Enter Your Cleared Altitude</small>	
Auto Pilot Disengage Bar.....	Up
Altimeter.....	Set
Lower CRT Selector.....	Norm
Inboard CRT Selector.....	Norm
Trim.....	Set
Speed Brake.....	Down
Throttles.....	Idle
Reverse Thrust Levers.....	Down
Fuel Control Switches.....	Cutoff
Stab Trim.....	Auto/Guarded
Communications Panel.....	Set
Audio Panel.....	Set
Auto Brake Selector.....	RTO
Passenger Signs.....	Auto
Flight Deck Door.....	Locked
Aileron / Rudder Trim.....	Set 0
TCAS/ATC.....	Test/Set
<small>Set TCAS To The Following Above During Takeoff N During Cruise Below During Descent</small>	
Ground Proximity Light.....	Extinguished
Ground Proximity Flap Override Switch.....	Off
Ground Proximity Config/Gear Override Switch.....	Off
Ground Proximity Terrain Override Switch.....	Off
Alternate Flaps Selector.....	Off
Alternate Flaps Arm Switch.....	Off
Alternate Gear Extend Switches.....	Off
Fuel Pumps.....	On
<small>Only Turn On Fuel Pumps for tanks that contain fuel</small>	
Fuel Crossfeeds.....	Set
<small>If Fuel Tank Quantity In Tank #2 Is More Than #1 And #3 Is More Than #4 Turn All Fuel Crossfeeds On</small>	
<small>If Fuel Tank Quantity In Tank #2 Is Less/Equal To #1 And #3 Is Less/Equal To #4 Turn Crossfeeds 1&4 off Turn Override Pumps 2&3 Off</small>	
Aircraft Doors.....	Closed
<small>Confirm All Doors Secure Prior To Pushback</small>	
Pushback	
Hydraulic Demand Pumps.....	Auto/Aux
<small>Set Hydraulic Demand Pump Selectors #1 - #2 - #3 To Auto Set Hydraulic Demand Pump Selector #4 to Aux</small>	
Red Anti-Collision Lights (Beacon).....	On
Pack Switches.....	Norm/Off
<small>Leave Pack Switch #1 On Norm - Turn Pack Switches #2 And #3 Off</small>	
EICAS (Recall Button).....	Push
<small>Confirm Only Appropriate Alert Messages Displayed On EICAS</small>	
EICAS (Cancel Button).....	Push
Manual Engine Start	
Autostart Selector.....	Off
Continuous Ignition.....	On
Engine Start Selector #4.....	Pull
Fuel Control Switch #4.....	Run
<small>When N2 Reaches Magenta Start Line - Set #4 Fuel Control Selector To Run</small>	
EICAS Engine Indicator.....	Monitor
<small>Verify Engine Start At 50% N2 - #4 Pull Start Selector Illumination Extinguished</small>	
Engine Start Selector #1.....	Pull
Fuel Control Switch #1.....	Run
<small>When N2 Reaches Magenta Start Line - Set #1 Fuel Control Selector To Run</small>	
EICAS Engine Indicator.....	Monitor
<small>Verify Engine Start At 50% N2 - #1 Pull Start Selector Illumination Extinguished</small>	
Engine Start Selector #3.....	Pull
Fuel Control Switch #3.....	Run
<small>When N2 Reaches Magenta Start Line - Set #3 Fuel Control Selector To Run</small>	
EICAS Engine Indicator.....	Monitor
<small>Verify Engine Start At 50% N2 - #3 Pull Start Selector Illumination Extinguished</small>	
Engine Start Selector #2.....	Pull
Fuel Control Switch #2.....	Run
<small>When N2 Reaches Magenta Start Line - Set #2 Fuel Control Selector To Run</small>	
EICAS Engine Indicator.....	Monitor
<small>Verify Engine Start At 50% N2 - #2 Pull Start Selector Illumination Extinguished</small>	
After Engine Start	
APU.....	Off
Hydraulic Demand Pump #4.....	Auto
Nacelle / Wing Anti-Ice.....	As Required
<small>Use If Temperature Is Below 10 Degrees Celsius & Visible Moisture Is Observed</small>	
AFT Cargo Heat.....	On
Isolation Switches.....	Open
Pack Switches.....	Auto
<small>Do Not Engage Packs Until The Engines Have Stabilized At Idle For 2 Minutes</small>	
EICAS Caution-Advisory-Status Messages.....	Check/Correct
<small>Ensure All Cautionary Messages Have Been Addressed Prior To Takeoff</small>	
Ground Equipment.....	Disconnect
Taxi	
Flaps.....	Set
Flight Controls.....	Check
Takeoff Performance.....	Confirm/Update
Cabin Crew Notification.....	Accomplish
Runway Turnoff Lights / Taxi Lights.....	On

Boeings 747-400 Checkliste für „normale“ Cockpit-Prozeduren. Quelle: <http://www.scribd.com/doc/111634018/Boeing-747-400-Checklist> (abgerufen am 20.12.2013)



SURGICAL SAFETY CHECKLIST (FIRST EDITION)

Before induction of anaesthesia

Before skin incision

Before patient leaves operating room

SIGN IN	TIME OUT	SIGN OUT
<input type="checkbox"/> PATIENT HAS CONFIRMED <ul style="list-style-type: none"> • IDENTITY • SITE • PROCEDURE • CONSENT 	<input type="checkbox"/> CONFIRM ALL TEAM MEMBERS HAVE INTRODUCED THEMSELVES BY NAME AND ROLE	NURSE VERBALLY CONFIRMS WITH THE TEAM:
<input type="checkbox"/> SITE MARKED/NOT APPLICABLE	<input type="checkbox"/> SURGEON, ANAESTHESIA PROFESSIONAL AND NURSE VERBALLY CONFIRM <ul style="list-style-type: none"> • PATIENT • SITE • PROCEDURE 	<input type="checkbox"/> THE NAME OF THE PROCEDURE RECORDED
<input type="checkbox"/> ANAESTHESIA SAFETY CHECK COMPLETED	ANTICIPATED CRITICAL EVENTS	<input type="checkbox"/> THAT INSTRUMENT, SPONGE AND NEEDLE COUNTS ARE CORRECT (OR NOT APPLICABLE)
<input type="checkbox"/> PULSE OXIMETER ON PATIENT AND FUNCTIONING	<input type="checkbox"/> SURGEON REVIEWS: WHAT ARE THE CRITICAL OR UNEXPECTED STEPS, OPERATIVE DURATION, ANTICIPATED BLOOD LOSS?	<input type="checkbox"/> HOW THE SPECIMEN IS LABELLED (INCLUDING PATIENT NAME)
DOES PATIENT HAVE A:	<input type="checkbox"/> ANAESTHESIA TEAM REVIEWS: ARE THERE ANY PATIENT-SPECIFIC CONCERNS?	<input type="checkbox"/> WHETHER THERE ARE ANY EQUIPMENT PROBLEMS TO BE ADDRESSED
KNOWN ALLERGY?	<input type="checkbox"/> NURSING TEAM REVIEWS: HAS STERILITY (INCLUDING INDICATOR RESULTS) BEEN CONFIRMED? ARE THERE EQUIPMENT ISSUES OR ANY CONCERNS?	<input type="checkbox"/> SURGEON, ANAESTHESIA PROFESSIONAL AND NURSE REVIEW THE KEY CONCERNS FOR RECOVERY AND MANAGEMENT OF THIS PATIENT
NO	<input type="checkbox"/> HAS ANTIBIOTIC PROPHYLAXIS BEEN GIVEN WITHIN THE LAST 60 MINUTES?	
YES	YES <input type="checkbox"/> NOT APPLICABLE <input type="checkbox"/>	
DIFFICULT AIRWAY/ASPIRATION RISK?	<input type="checkbox"/> IS ESSENTIAL IMAGING DISPLAYED?	
NO	YES <input type="checkbox"/> NOT APPLICABLE <input type="checkbox"/>	
YES, AND EQUIPMENT/ASSISTANCE AVAILABLE		
RISK OF >500ML BLOOD LOSS (7ML/KG IN CHILDREN)?		
NO		
YES, AND ADEQUATE INTRAVENOUS ACCESS AND FLUIDS PLANNED		

THIS CHECKLIST IS NOT INTENDED TO BE COMPREHENSIVE. ADDITIONS AND MODIFICATIONS TO FIT LOCAL PRACTICE ARE ENCOURAGED.

WHO surgical safety checklist and implementation manual.

Einsatzstichworte für Brandeinsätze			
Alarmdurchsage		Erstalarmierung	
Einsatzstichwort	Meldebild	Einsatztaktische Parameter	Zusätzliche Einsatzmittel nach Lage
F GAS 1	Brand - einzelner Gasflaschen - einer Gasleitung	wie F 2 + Löschpulver + Wärmeschutzbekleidung + Ex-Warngerät	+ P 250 (FwA) + RW
F GAS 2	Brand - eines Gastanks - eines Gastankfahrzeugs - eines Gaskesselwagens	wie F 3 + Löschpulver+ Wärmeschutzbekleidung + Ex-Warngerät Rettungsdienst-Stichwort: R 2	+ P 250 (FwA) + RW
F LKW	Brand - eines LKW	wie F 2, aber 5.000 Liter Wasser + 120 Liter Schaummittel	+ weitere Tanklöschfahrzeuge bei Gefahrgut: + wie H Gefahr 2
F SCHIFF 1	Brand - eines Sportboots - einer Yacht - eines Segelboots	wie F 2 + 2 RTB / MZB + Tauchergruppe Rettungsdienst-Stichwort: R 1	+ weitere RTB / MZB + Gerätewagen-Gefahrgut, Ölsperre + GW-Wasserrettung

Ehrenwörtliche Erklärung

Hiermit erkläre ich,

- dass mir die geltende Promotionsordnung der Fakultät für Mathematik und Informatik der Friedrich-Schiller-Universität Jena bekannt ist,
- dass ich die Dissertation selbst angefertigt habe, keine Textabschnitte oder Ergebnisse eines Dritten oder eigener Prüfungsarbeiten ohne Kennzeichnung übernommen und alle von mir benutzten Hilfsmittel, persönliche Mitteilungen und Quellen in meiner Arbeit angegeben habe,
- dass die Hilfe eines Promotionsberaters nicht in Anspruch genommen wurde und dass Dritte weder unmittelbar noch mittelbar geldwerte Leistungen von mir für Arbeiten erhalten haben, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen,
- dass ich die Dissertation noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht habe, und
- dass ich die gleiche, eine in wesentlichen Teilen ähnliche oder eine andere Abhandlung nicht bei einer anderen Hochschule als Dissertation eingereicht habe.

Jena, den 21.12.2013

Veröffentlichungen

U. KRÜGER UND F. WUCHOLT: *Fehlervermeidung durch Checklisten-Assistenz in Großschadensereignissen – Potenzial und Vorteile elektronischer Checklisten*, Buchreihe, In: Plattform e.V. 2012/2013. Entscheiden in kritischen Situationen: Neue Perspektiven und Erkenntnisse. Herausgeber: Rudi Heimann, Stefan Strohschneider und Harald Schaub, Plattform Menschen in komplexen Arbeitswelten e. V., Verlag für Polizeiwissenschaft, Frankfurt, **2013**.

U. KRÜGER, F. WUCHOLT AND C. BECKSTEIN: *Electronic Checklist Support for Disaster Response*, In: Proceedings of the 9th International Conference on Information Systems for Crisis Response and Management, Human Experiences in the Design of Crisis Response and Management Services and Systems, Vancouver, Canada, April **2012**.

F. WUCHOLT, U. KRÜGER UND S. KERN: *Mobiles Checklisten-Support-System im Einsatzszenario einer Großschadenslage*, Informatik 2011, Workshop IT-Unterstützung für Rettungskräfte, Berlin, Oktober **2011**.

F. WUCHOLT, Y. YILDIRIM-KRANNIG, M. MAHLER, U. KRÜGER, AND C. BECKSTEIN: *Cultural Analysis and Formal Standardised Language — a Mass Casualty Incident Perspective*, In: Proceedings of the 8th International Conference on Information Systems for Crisis Response and Management (ISCRAM), User Centred Design Process for Emergency Management Information Systems, Lisbon, Portugal, May **2011**.

U. KRÜGER, A. GABDULKHAKOVA, V. SCHAU, AND C. BECKSTEIN: *Information and Management Support for Mass Casualty Incident Scenarios*, 2. Workshop IT-Unterstützung für Rettungskräfte im Rahmen der INFORMATIK 2010, Leipzig, September **2010**.

U. KRÜGER, A. GABDULKHAKOVA, B. KÖNIG-RIES, AND C. BECKSTEIN: *Semantic Service Infrastructure for Intelligent Checklist Support Systems*, In: Proceeding of the 2010 international Conference on Towards a service-based internet, International Workshop on Emergency Management through Service Oriented Architectures, ServiceWave'10, Ghent, December **2010**.

H. SACK, U. KRÜGER, AND M. DOM: *A Knowledge Base on NP-complete Decision Problems and its Application in Bibliographic Search*, In: Proceedings of Berliner XML-Tage 2006, Berlin, September **2006**.