

Lehrstuhl für Informationstechnik im Maschinenwesen
der Technischen Universität München

Verteilte kooperative Geräteentwicklung in der Mechatronik

Florian Kuttig

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. rer. nat. Heiner Bubb

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Klaus Bender
2. Univ.-Prof. Dr.-Ing. Jörg Eberspächer

Die Dissertation wurde am 30.06.2005 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 31.10.2005 angenommen.

Verteilte kooperative Geräteentwicklung in der Mechatronik

Mechatronische Produkte besitzen heutzutage in vielen Wirtschaftszweigen einen hohen Stellenwert. Die erfolgreiche Integration der drei Disziplinen Mechanik, Elektronik und Softwaretechnik stellt jedoch für den Produktentwicklungsprozess eine besondere Herausforderung dar. Besonders in den integrativen Phasen, wo oftmals eine geografische Trennung des mechatronischen Entwicklungsgegenstands von verschiedensten beteiligten Spezialisten vorherrscht. Um die Geräteentwicklung dennoch effizient durchzuführen, sind Methoden und Technologien für die verteilte Kooperation unverzichtbar. Jedoch stellt gerade dies eine besondere Herausforderung dar, da unmittelbare Tätigkeiten am Entwicklungsgegenstand von entfernten Standorten auszuführen sind. Die vorliegende Arbeit entwickelt ein Framework für die effiziente verteilte Kooperation in den integrativen Phasen des mechatronischen Entwicklungsprozesses unter Berücksichtigung der dort vorherrschenden besonderen Herausforderungen. Das Framework beinhaltet Lösungen für die interpersonelle- und die Inter-Tool-Kommunikation sowie eine Modell für die sichere und effiziente Abwicklung entfernter Interaktionen.

Distributed Cooperative Development of Mechatronic Devices

Mechatronic devices are nowadays of high significance in various economic branches. The successful integration of the three disciplines mechanics, electronics and software-technique is a permanent challenge for product development processes, especially during the integrative phases, where the product and the involved specialists are often geographically separated. However, to achieve efficient product development, there is a special need for methods and technologies to establish distributed cooperation. Yet, this results in a special challenge as direct interaction with the device under development is executed from remote locations. This thesis develops a framework for efficient cooperation in the integrative phases of mechatronic product development processes in consideration of the existing special boundary conditions. The framework contains solutions for interpersonal- and inter-tool-communication as well as a model for efficient and safe handling of remote interactions.

Vorwort

Die vorliegende Arbeit entstand während meiner Assistententätigkeit am Lehrstuhl für Informationstechnik im Maschinenwesen der Technischen Universität München. Im Rahmen der Mitwirkung und Leitung umfangreicher industrienaher Forschungsprojekte in Zusammenarbeit mit vielen Partnerfirmen gewann ich die fachliche Basis auf dem Gebiet der Mechatronik und der industriellen Informationstechnik als wichtige Voraussetzung für die erfolgreiche Anfertigung dieser Arbeit.

Mein besonderer Dank gebührt meinem Doktorvater, Herrn Professor Dr. Klaus Bender für die hervorragende Betreuung meiner Dissertation und der fortwährenden Förderung meiner Arbeit. Der von ihm gewährte Freiraum zur Entwicklung und Realisierung eigener Ideen sowie die stetige Unterstützung für meine Tätigkeiten haben ganz wesentlich zur Erreichung meines Ziels beigetragen.

Gleichfalls bedanke ich mich ganz herzlich bei Herrn Professor Dr. Jörg Eberspächer für die freundliche Übernahme des Koreferats und das damit verbundene Interesse an meiner Arbeit. Herrn Prof. Dr. Heiner Bubb danke ich für die Übernahme des Vorsitzes der Prüfungskommission.

Ebenfalls besonderer Dank gebührt all denjenigen, auch ehemaligen Kollegen, die mir bei der Durchführung meiner Forschungsarbeiten hilfreich zur Seite standen und beim Schreiben der Dissertation in zahlreichen Fachdiskussionen wertvolle Anregungen gegeben haben. Insbesondere danke ich auch unserem Sekretariat für die Unterstützung in organisatorischen Angelegenheiten. Allen Mitarbeitern des Instituts danke ich für die hervorragende Zusammenarbeit über den gesamten Zeitraum meiner Tätigkeit.

Sämtliche der Forschungs- und Projektarbeiten wären nicht ohne die Unterstützung durch eine Vielzahl von Studenten möglich gewesen, die im Rahmen von Studienarbeiten und als wissenschaftliche Hilfskräfte einen wesentlichen Beitrag zur Erreichung der Ziele geleistet haben. Für das gezeigte Interesse und den Arbeitseinsatz möchte ich mich bei allen Beteiligten bedanken. Insbesondere danke ich Herrn Christian Schimmel für sein besonderes Engagement bei der Verwirklichung meiner Ideen und Konzepte im Rahmen umfangreicher prototypischer Implementierungen.

Nicht zuletzt danke ich meinen Eltern für das stets in mich gesetzte Vertrauen und die Unterstützung auf meinem gesamten Weg. Meiner geliebten Lebensgefährtin danke ich für die Unterstützung in den heißen Phasen der Arbeit und dafür, dass sie viele Entbehrungen geduldig in Kauf genommen hat.

München, im September 2005

Florian Kuttig

Inhaltsverzeichnis

1	Einführung in die Thematik	1
1.1	Motivation und Problemstellung.....	1
1.2	Ziel der Arbeit	2
1.3	Aufbau der Arbeit.....	3
2	Kooperation in der mechatronischen Produktentwicklung.....	5
2.1	Umfeld verteilter Produktentwicklung.....	5
2.1.1	Mechatronische Systeme.....	5
2.1.2	Entwicklungsprozess mechatronischer Produkte.....	7
2.1.3	Ausprägungen des Zusammenwirkens.....	8
2.1.4	Synchrone und Asynchrone Kooperation	11
2.2	Merkmale der Entwicklung mechatronischer Produkte.....	12
2.2.1	Wirtschaftliche Bedeutung mechatronischer Produkte.....	12
2.2.2	Charakteristik des Entwicklungsprozesses in der Mechatronik.....	13
2.2.3	Werkzeuge im Entwicklungsprozess	16
2.2.4	Integration multidisziplinärer Wissensträger	18
2.2.5	Integration externer Expertise in den Entwicklungsprozess	18
2.3	Herausforderungen des mechatronischen Entwicklungsprozesses	20
2.3.1	Allgemeine Situation am Markt	20
2.3.2	Problemstellungen der mechatronischen Produktentwicklung.....	21
2.3.3	Gegenwärtige Situation in der Mechatronik	23
2.3.4	Defizite im mechatronischen Entwicklungsprozess.....	24
2.4	Zusammenfassung und Aufgabenstellung	27
3	Methoden und Werkzeuge für die verteilte Kooperation.....	29
3.1	Verteilte Zusammenarbeit bei der Entwicklung mechatronischer Geräte	29
3.1.1	Konzentration von Expertise mittels Remote-Technologien.....	29
3.1.2	Anwendungsfelder	30
3.1.3	Anforderungen an verteilte Koaktionen im Umfeld der Mechatronik.....	31
3.1.4	Taxonomie der Kooperationsmittel.....	34
3.2	Bereiche mit etablierter verteilter Kooperation.....	35
3.2.1	Fernzugriff auf automatisierungstechnische Maschinen und Anlagen	35
3.2.2	Telechirurgie und Telerobotik.....	37
3.2.3	E-Learning und Distance-Learning.....	37
3.2.4	Spezifizierende und konstruktive Phasen.....	38
3.3	Modelle und Konzepte für die verteilte Kooperation.....	39
3.3.1	Allgemeines Phasenmodell der Ko-Aktion.....	40
3.3.2	KOMFORCE-Referenzmodell für die Mechatronikentwicklung.....	40
3.3.3	Modular Advanced Collaboration System	41
3.3.4	Multimedia Grundlagentechnologien.....	43

3.4	Technologien und Werkzeuge für verteilte Kooperation.....	44
3.4.1	Integrierte Tools und Conferencing Tools	45
3.4.2	Desktop-Sharing.....	46
3.4.3	Fernwirk-Tools.....	47
3.4.4	Einsatz von Remote-Tools unter dem Aspekt der IT-Sicherheit	48
3.5	Zusammenfassung und Bewertung	49
4	Rahmenkonzept für die verteilte Kollaboration.....	53
4.1	Analyse der Kommunikationsbeziehungen.....	54
4.1.1	Darstellung des Kommunikationsszenarios	55
4.1.2	Das kollaborative Setting	57
4.1.3	Bestimmung der Randbedingungen für die Kollaboration	59
4.2	Elemente des Lösungsansatzes.....	60
4.2.1	Architektur eines universellen Remote-Frameworks.....	61
4.2.2	Rollenmodell für die verteilte Kollaboration in der Mechatronik.....	63
4.2.3	Audiovisuelle-Kommunikation.....	67
4.2.4	Inter-Tool-Kommunikation.....	69
4.3	Beschreibung der Lösungselemente.....	70
4.3.1	Zugriffskontrollmodul.....	70
4.3.2	GUI-Modul.....	71
4.3.3	Kommunikationsmodul.....	72
4.3.4	Multimediateilkomponenten	72
4.4	Integration von Werkzeugen in das Rahmenkonzept.....	73
4.4.1	Modulares GUI-Konzept.....	73
4.4.2	Integration des Remote-Frameworks mit Werkzeugapplikationen.....	74
4.4.3	IT-Sicherheitsstrategien	76
4.4.4	Migrationsstrategie für die Remote-Erweiterung.....	78
4.5	Zusammenfassung.....	79
5	Universelles Remote-Framework.....	81
5.1	Gestuftes Zugriffskonzept für entfernte Interaktionen.....	81
5.1.1	Klassifikationsansatz nach IEC 61508.....	82
5.1.2	Definition von Zugriffsklassen.....	83
5.1.3	Zuordnungssystematik	86
5.1.4	Modellierung des Klassifikationskonzepts.....	89
5.1.5	Tracing-Konzept.....	92
5.2	Integration der Multimediafunktionalität.....	93
5.2.1	Auswahl der Multimediafunktionalität	94
5.2.2	Schnittstelle zum Multimediasystem	96
5.2.3	Einheitliche Benutzerschnittstelle.....	97
5.2.4	Analytische Betrachtung des Multimediateilkanals.....	97
5.3	Architektur des Remote-Frameworks	104
5.3.1	Statisches Modell	104
5.3.2	Dynamisches Modell.....	107
5.3.3	Generisches Fehlerbehandlungskonzept	111
5.3.4	Basistechnologien für die Remoting-Funktionalität	112
5.4	IT-Infrastrukturen und IT-Security	114
5.4.1	Identifikation des Bedrohungspotentials.....	114
5.4.2	Gegenmaßnahmen.....	115
5.4.3	Infrastrukturelle Ausprägungen.....	115
5.4.4	Integration des Remote-Frameworks in IT-Sicherheitsstrukturen.....	118
5.5	Zusammenfassung.....	121

6	Evaluierung im Rahmen einer industriellen Anwendung	123
6.1	Entwicklung von Geräten mit Feldbusschnittstelle.....	123
6.1.1	Einführung in das Anwendungsgebiet	124
6.1.2	Rollenverteilung beim Zertifizierungstest.....	125
6.1.3	Das Prüfwerkzeug MoTest.....	127
6.1.4	Verteilte Kollaboration beim Zertifizierungstest	129
6.2	Realisierung des verteilten Prüfwerkzeugs	130
6.2.1	Schnittstellen des Remote-Frameworks.....	130
6.2.2	Softwarearchitektur des Prüfsystems	131
6.2.3	Realisierung der interpersonellen Schnittstelle	133
6.2.4	Umsetzung des Rollenmodells im Prüfwerkzeug	134
6.3	Bewertung des Remote-Frameworks	136
6.3.1	Szenario der industriellen Anwendung	136
6.3.2	IT-Infrastruktur.....	138
6.3.3	Analyse der Reaktionszeiten	139
6.3.4	Qualitätsbetrachtungen des Multimediakanals.....	141
6.4	Zusammenfassung.....	143
7	Abschließende Betrachtung.....	145
7.1	Zusammenfassung.....	145
7.2	Bewertung	147
7.3	Ausblick	148
	Literaturverzeichnis.....	149
	Abbildungsverzeichnis	165
	Tabellenverzeichnis.....	166
	Stichwortverzeichnis	167
	Anhang A Anwendungsszenarien.....	169
	Anhang B Rollendefinitionen.....	171
	Anhang C Zugriffsklassen.....	173
	Anhang D Multimediale Kommunikation	181
	Anhang E Spezifikation des Remote-Frameworks	183
	Anhang F Normen und Richtlinien IT Sicherheit.....	201

Glossar

Asynchrone Kooperation

→Kooperation erfolgt wechselseitig zu unterschiedlichen, nicht aufeinander abgestimmten Zeiten, die Einhaltung einer Zeitschranke für den Informationsaustausch ist dabei nicht garantiert.

Entwicklungsprozess

Die Summe aller Vorgänge die stattfinden, damit ein Produkt von der Idee bis zum verkaufsfertigen Erzeugnis gelangt.

http

Hyper Text Transfer Protocol. Zustandsloses Datenaustauschprotokoll, vorzugsweise zur Übertragung textbasierter Nachrichten. Hat sich insbesondere im World Wide Web als Standardprotokoll etabliert. Setzt in der Regel auf dem →TCP-Protokoll auf.

Integrative Phasen

Diejenigen Phasen im mechatronischen →Entwicklungsprozess, die sämtliche Integrationsprozesse des realen Produkts bzw. dessen Teilkomponenten beinhalten.

IP

Internet Protocol. Basisprotokoll für die logische Adressierung von Rechnern im Internet bzw. Intranet.

IP-basierte Netze

Rechnernetzwerke, die das →IP-Protokoll zur Abstrahierung der logischen Adressierung der eingebundenen Rechner von der unterlagerten Netzwerktechnologie einsetzen. Eine Besonderheit IP-basierter Netze ist die Tatsache, dass Datenpakete auf dem Weg vom Sender zum Empfänger unterschiedliche Wege durch das Netzwerk nehmen können.

KMU

Abkürzung für *Kleine Mittelständische Unternehmen*.

Kollaboration

Direkte und sich wechselseitig beeinflussende, tätige Auseinandersetzung zur Lösung oder Bewältigung einer Aufgabe.

Kooperation

Der Begriff besitzt doppelte Bedeutung: Erstens ist Kooperation die Bereitschaft, miteinander zu handeln, Zweitens beschreibt der Begriff die arbeitsteilige Bearbeitung von Aufgaben durch mehrere Individuen.

Koordination

Wechselseitiges Abstimmen von Handlungen und Tätigkeiten durch die organisatorische und zeitliche Festlegung von Handlungs- und Tätigkeitsabläufen.

Mechatronisches Gerät

Wirkung verursachender Gegenstand, der Technologien aus den drei Domänen Mechanik, Elektronik und Softwaretechnik zur Erbringung seiner Gesamtfunktionalität vereint.

Multimediatechnologien

Technologien zur Darstellung bzw. Übermittlung multimedialer Inhalte, d.h. Sprache, Video, Bilder, textuelle Informationen etc.

Remote-Technologien

Der Begriff Remote-Technologien bzw. →Teletechnologien umfasst allgemein alle Mittel um digitale Informationen elektronisch zwischen geografisch entfernten Orten zu übertragen.

SOAP

Ursprünglich die Abkürzung für *Simple Object Access Protocol*. →XML-basiertes, standardisiertes Framework für den flexiblen Transport applikationsspezifischer Informationen.

Spezifizierende Phasen

Diejenigen Phasen im mechatronischen →Entwicklungsprozess in denen das Design sowie die Spezifikation des Produktes bzw. dessen Teilkomponenten erfolgt.

Synchrone Kooperation

Gleichzeitiges Zusammenarbeiten der Kooperationspartner, der Austausch der Informationen findet dabei gleichlaufend statt, d. h. die Beteiligten sind zu jeder Zeit kommunikationsbereit.

TCP

Transmission Control Protocol. Verbindungsorientiertes Protokoll, welches einen virtuellen Kanal für den bidirektionalen Datenaustausch zwischen zwei Rechnern bereitstellt. setzt in der Regel auf dem →IP-Protokoll auf.

Teletechnologien

Siehe →Remote-Technologien

Umsetzende Phasen

Diejenigen Phasen im mechatronischen →Entwicklungsprozess in denen die Umsetzung der Spezifikation des Produkts bzw. dessen Teilkomponenten in reale Produkte bzw. Teilkomponenten, u. U. in Form von Prototypen, erfolgt.

V-Modell

→Vorgehensmodell, welches ursprünglich für die Softwaretechnik entwickelt wurde, in erweiterter Form aber auch in der Mechatronikentwicklung Verbreitung gefunden hat. Stellt →spezifizierende und →integrative Phasen gegenüber und setzt diese zueinander in Bezug.

Vorgehensmodell

Ein Vorgehensmodell stellt eine standardisierte Ablaufmethodik zur Verfügung. Aufgabe eines Vorgehensmodells ist es, die allgemein in einem Entwicklungsprozess auftretenden Aufgabenstellungen und Aktivitäten in ihrer logischen Ordnung darzustellen.

Werkzeugapplikation

Werkzeug in Form einer Softwareapplikation welches auf einer Rechnerplattform installiert wird. Im hier behandelten Kontext handelt es sich um Werkzeugapplikationen zur Unterstützung des →Entwicklungsprozesses in den →Integrativen Phasen.

Werkzeuginstanz

→Werkzeugapplikation, die sich auf einer Rechnerplattform gerade in der Ausführung befindet.

XML

Extensible Markup Language. Im Jahr 1998 vom *World Wide Web Consortium* spezifizierte Auszeichnungssprache zur Beschreibung, Strukturierung und Transformation beliebiger Daten.

XML-Schema

Legt die Struktur von →XML-Dokumenten fest.

1 Einführung in die Thematik

1.1 Motivation und Problemstellung

Mechatronische Produkte besitzen heutzutage in vielen Wirtschaftszweigen einen hohen Stellenwert [ISERMANN 02]. Nicht nur in Branchen wie dem Automobilbau, dem Flugzeugbau, der Investitionsgüterindustrie und der Konsumgüterindustrie stellen komplexe mechatronische Geräte und Systeme Schlüsselkomponenten dar. Die erfolgreiche, intensive Integration von Mechanik, Elektronik und Software wird somit zum bestimmenden Faktor der modernen Produktentwicklung. Angesichts der mittlerweile weitgehend ausgeschöpften Rationalisierungspotentiale im Fertigungsbereich gilt es, die in der Produktentwicklung vorhandenen Potentiale zu nutzen [ZWICKER 99].

Vor dem Hintergrund hoher Komplexität moderner Produkte muss die Integration der drei komplementären Kompetenzfelder Mechanik, Elektronik und Softwaretechnik erfolgreich und effizient durchgeführt werden [VDI 03]. Die moderne Geräteentwicklung ist gekennzeichnet durch eine Vielzahl von beteiligten Spezialisten. Der Trend der weiter steigenden Gesamtkomplexität führt dabei zu einer immer weiteren Spezialisierung der einzelnen Teildisziplinen. Letztendlich kann die Leistungsfähigkeit des Gesamtprodukts jedoch nur durch das enge Zusammenwirken aller Disziplinen gesteigert werden.

In allen Phasen der mechatronischen Produktentwicklung stellt die Erfüllung dieser Aufgabe eine besondere Herausforderung dar. Die Integration verschiedenster Spezialisten aus ganz unterschiedlichen Fachgebieten ist deshalb kompliziert, da jeder mit seinem Fachbereich eine eigene Begriffswelt und Vorgehensweise verbindet. Hinzu kommt, dass die Beteiligten nicht alle am gleichen Ort zur Verfügung stehen, sondern bisweilen an unterschiedlichen geografischen Standorten wirken. Gerade kleine mittelständische Unternehmen stehen vor dem Problem, aus Kostengründen nicht sämtliche benötigte Expertise vorhalten zu können und müssen folglich entweder auf Outsourcing-Lösungen ausweichen oder auf externe Spezialisten zurückgreifen. Dies steht im Widerspruch zu dem vorhanden hohen Integrations- und Abstimmungsbedarf der mechatronischen Produktentwicklung.

Auch wenn nach wie vor viele Herausforderungen zu bewältigen sind, so finden sich, insbesondere in den konstruktiven Phasen der Produktentwicklung, Ansätze um der Problemstellung gerecht zu werden. Moderne Mittel zur Unterstützung der Kommunikation, virtuelle Marktplätze und Mechanismen zum schnellen Datenaustausch stehen zur Verfügung und können den Entwicklungsprozess

wirkungsvoll unterstützen [GAUSEMEIER+ 00]. In den integrativen Phasen der Produktentwicklung ist die Problemstellung jedoch weitaus schwieriger. Im Gegensatz zu virtuellen Modellen stehen dort durchgängig reale Geräte bzw. Teilkomponenten im Mittelpunkt der Entwicklungsaktivitäten. Beteiligte Spezialisten müssen sich also vor Ort einfinden, was Zeitverzug und nicht selten hohe Kosten verursacht [RÖMER 03]. Aufgrund der besonderen Randbedingungen bei der Arbeit mit realen Geräten, stehen Mittel zur Integration der Tätigkeiten verteilter Spezialisten an einem örtlich gebundenen Entwicklungsgegenstand bisher nicht zur Verfügung. Folglich verläuft die mechatronische Produktentwicklung in den integrativen Phasen oftmals ineffizient.

1.2 Ziel der Arbeit

Die beschriebene Problemstellung ist nur dann lösbar, wenn geeignete Mittel für die Zusammenarbeit geografisch verteilter Experten, unter Einbeziehung des realen Entwicklungsgegenstands, zur Verfügung stehen. Dabei müssen die Besonderheiten dieses Szenarios und der darin verwendeten Werkzeuge beachtet und die besonderen Randbedingungen erfüllt werden. Bisherige Mittel für die verteilte Zusammenarbeit entsprechen diesen Anforderungen nicht im nötigen Umfang. Es ist daher eine Lösung zu entwickeln, die eine verteilte Zusammenarbeit in den integrativen Phasen der mechatronischen Produktentwicklung ermöglicht. Von besonderer Bedeutung ist dabei die einfache Nutzbarkeit der Lösung für das verteilte Entwicklungsteam. Diverse Studien, wie beispielsweise [MUNKVOLD 03], belegen immer wieder das Scheitern von Technologien für die verteilte Kollaboration aufgrund mangelnder Akzeptanz seitens der Anwender. Die im Rahmen dieser Arbeit verfolgten Ziele gliedern sich in zwei Kategorien:

Zum Ersten soll die zu entwickelnde Lösung einfache Kopplungsmechanismen für verteilte Entwicklungswerkzeuge in den integrativen Phasen zur Verfügung stellen. Dabei müssen sämtliche der dort gegebenen Besonderheiten berücksichtigt werden. Hierzu zählen Schutzmechanismen für entfernte Interaktionen mit realen mechatronischen Systemen, die auch Gefährdungspotential für Mensch und Umwelt beherbergen können. Weiterhin ein granulares, abgestuftes Zugriffskonzept auf die Funktionen und den Datenhaushalt der eingesetzten Entwicklungswerkzeuge. Ebenso bedeutsam ist ein Rollenkonzept für die Anpassung der verteilt genutzten Werkzeuge an die spezifischen Aufgabenbereiche der beteiligten Spezialisten. Die Kopplung muss zudem die Anforderungen an IT-Sicherheit berücksichtigen und universell eine breite Palette von Entwicklungswerkzeugen unterstützen.

Zum Zweiten soll die Lösung auch die interpersonelle Kommunikation zwischen verteilten Experten unterstützen. Hierbei kommen Multimediatechnologien zum Einsatz, die an die speziellen Anforderungen des vorliegenden Szenarios angepasst werden. Im Vordergrund steht auch hier wiederum die einfache Benutzbarkeit der Technologien: Die Multimedialösungen werden so in die

Entwicklungswerkzeuge integriert, dass der Anwender keinen Aufwand für die Installation und Konfiguration aufbringen muss. Im Sinne der einheitlichen Bedienung unterschiedlichster Multimediaapplikationen wird ein entsprechendes Benutzer-Interface spezifiziert.

Die Lösung resultiert in einem einheitlichen Framework für die verteilte Kooperation in den integrativen Phasen der Mechatronikentwicklung unter Einbeziehung von entfernten Interaktionen mit realen Geräten und speziellen, rechnergestützten Entwicklungswerkzeugen. Das Framework entbindet den Anwender zudem von zusätzlichem Aufwand bei der Nutzung von verteilter Funktionalität. Es ist so ausgelegt, dass es auch mit eingeschränkten Kommunikationsressourcen wirkungsvoll eingesetzt werden kann.

1.3 Aufbau der Arbeit

Die vorliegende Arbeit ist gemäß dem in Abbildung 1-1 dargestellten Schema strukturiert. Kapitel 2 nimmt zunächst eine Definition der für die Thematik relevanten Begriffe vor. Anschließend erfolgt eine detaillierte Analyse der gegenwärtigen Situation in den integrativen Phasen der mechatronischen Produktentwicklung. Anhand des Ergebnisses dieser Analyse wird die exakte Problemstellung identifiziert sowie die Aufgabenstellung formuliert.

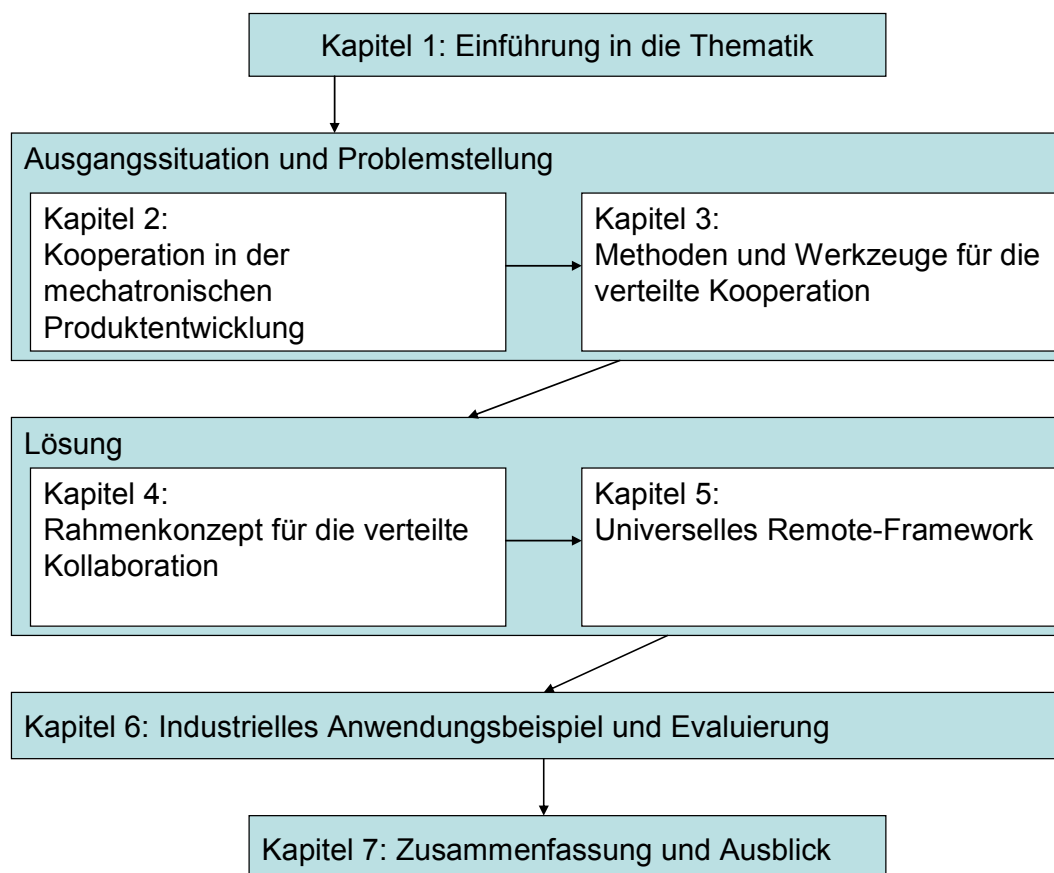


Abbildung 1-1: Struktur der vorliegenden Arbeit

Das nachfolgende Kapitel 3 analysiert existierende Lösungen aus dem Bereich der verteilten Kooperation und kommt zu dem Schluss, dass keine der bisher existenten Lösungen den Anforderungen des vorliegenden Anwendungsfalls genügt. Dabei werden nicht nur die Technologien für sich, sondern auch benachbarte Bereiche untersucht in denen verteilte Kooperation Anwendung findet.

Kapitel 4 erarbeitet ein Konzept für ein universelles Remote-Framework, welches auch in den integrativen Produktentwicklungsphasen verteilte Kooperation ermöglicht. Hierzu wird zunächst das zugrunde liegende Kommunikationsszenario exakt beschrieben. Die Beschreibung dient als Basis für die Erarbeitung des Konzepts sowohl für das Gesamtsystem als auch für die einzelnen Komponenten der Lösung.

Die Bestandteile des Gesamtkonzepts werden anschließend in Kapitel 5 erarbeitet und spezifiziert. Hierfür werden detaillierte Lösungen auf Basis geeigneter Grundlagentechnologien erarbeitet. Ebenfalls Bestandteil der Lösung ist eine Vorgehensweise für die Integration des geschaffenen Frameworks. Dies betrifft sowohl die Eingliederung in existierende IT-Landschaften und Security-Systeme als auch die Koppelung mit bestehenden Rechnerwerkzeugen.

In Kapitel 6 wird eine prototypische Anwendung der entwickelten Lösung umgesetzt und vorgestellt. Das Remote-Framework wird hierfür in Verbindung mit einem Rechnerwerkzeug zur Prüfung der Kommunikationsschnittstelle von mechatronischen Geräten der Investitionsgüterindustrie eingesetzt. Die Ergebnisse werden im Rahmen einer industriellen Anwendung evaluiert.

Abschließend erfolgt in Kapitel 7 eine Zusammenfassung der Ergebnisse sowie eine Bewertung der erarbeiteten Lösung hinsichtlich der in der Aufgabenstellung festgelegten Anforderungen. Zuletzt wird ein Ausblick auf weitere Anwendungsmöglichkeiten der entwickelten Lösung sowie weiterführende Forschungsaktivitäten gegeben.

2 Kooperation in der mechatronischen Produktentwicklung

Analog zur Komplexität moderner mechatronischer Produkte selbst ist auch der Produktentwicklungsprozess durch eine hohe Komplexität gekennzeichnet [VDI 03]. Ausschlaggebend für ein erfolgreiches Entwicklungsprojekt ist dabei das Zusammenspiel der unterschiedlichen beteiligten Disziplinen Mechanik, Elektronik und Informationstechnik auch unter gesamtwirtschaftlichen Gesichtspunkten.

In diesem Kapitel werden, nach erfolgter Begriffsbildung zum behandelten Thema, die Charakteristika des Entwicklungsprozesses für mechatronische Produkte dargestellt. Die dabei identifizierten Defizite werden analysiert und die Problemstellung präzisiert. Anschließend erfolgt die Ableitung der Aufgabenstellung.

2.1 Umfeld verteilter Produktentwicklung

Das Thema Kooperation in der Produktentwicklung umfasst eine breite Palette von Begrifflichkeiten, die je nach Anwenderkreis mit unscharfen, bis hin zu völlig unterschiedlichen Interpretationen belegt sind. Zum Zweck der einheitlichen Begriffsbildung erfolgt zunächst die Definition der wichtigsten Schlüsselbegriffe für die in der Arbeit behandelten Teilgebiete. Gegebenenfalls erfolgt eine Einordnung der Begriffe in den bearbeiteten Kontext bzw. eine Einschränkung auf das betrachtete Teilgebiet.

2.1.1 Mechatronische Systeme

Der Begriff *Mechatronik* wurde 1969 durch die japanische Firma *Yaskawa Electric Cooperation* geprägt und setzt sich aus den Wortbestandteilen *Mechanik* und *Elektronik* zusammen. Sehr allgemein wird in der Literatur unter dem Begriff die enge Integration der drei Ingenieursdisziplinen Mechanik, Elektronik und Informationstechnik verstanden¹. Jedoch existiert ein weiter Spielraum bezüglich einer präzisen Definition. Dies betrifft vor allem die Art der Integration der einzelnen Teilkomponenten: funktional oder räumlich². Daneben haben

¹ Beispielsweise siehe [HEIMANN+ 01], [SCHERNIKAU 01], [GAUSEMEIER+ 00].

² Nach SCHERNIKAU bezeichnet *funktionale Integration* die Realisierung von Produktfunktionen mittels der Softwaretechnik. *Räumliche Integration* die Realisierung der Lösungsprinzipien in einer Baueinheit.

sich weitere abgeleitete Begriffe wie *mechatronisches Gerät*, *mechatronisches Produkt* und *mechatronisches System* verbreitet.

Gemäß einschlägiger Literatur kennzeichnet ein mechatronisches System die Aufnahme, Verarbeitung und Ausgabe von Signalen unter enger Verknüpfung der drei Disziplinen Mechanik, Elektronik und Informationstechnik [HEIMANN+ 01]. Nach GAUSEMEIER stellt auch die Vernetzung mehrerer mechatronischer Teilkomponenten ein mechatronisches System dar [GAUSEMEIER+ 00]. In Anlehnung an [GAUSEMEIER+ 00] und [HEIMANN+ 01] wird die folgende Definition für den Begriff mechatronisches System abgeleitet:

Definition 2-1: Mechatronisches System

Ein mechatronisches System interagiert mit seiner physikalischen (z. B. technischer Prozess) und seiner informationstechnischen Umgebung (z. B. Benutzerschnittstelle). Es nimmt Signale auf, verarbeitet diese elektronisch und gibt Signale aus, die es z. B. in Kräfte und Bewegungen umsetzt. Dies erfolgt unter enger Verknüpfung der Disziplinen Mechanik, Elektronik und Informationsverarbeitung mit dem Ziel, die Funktionalität des Systems zu verbessern und zu erhöhen. Ein mechatronisches System stellt dabei die Gesamtheit aller Teilkomponenten dar, die zur Erfüllung einer Aufgabenstellung miteinander interagieren.

Die prinzipiellen Grundstrukturen eines mechatronischen Systems lassen sich in Abbildung 2-1 identifizieren: Das Grundsystem (meist mechanisch), Aktoren/Sensoren und Prozessoren die informationsverarbeitende Aufgaben übernehmen.

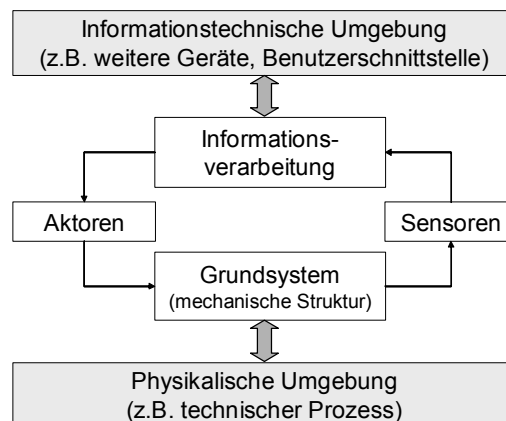


Abbildung 2-1: Struktur mechatronischer Systeme (in Anlehnung an [HEIMANN+ 01])

Nach [ISERMANN 99] werden ganz allgemein integrierte mechanisch-elektronische Systeme als *mechatronische Systeme* bezeichnet, wobei die Grenze zu anderen mit der Elektronik interagierenden Systemen fließend ist. Diese Feststellung in Verbindung mit Definition 2-1 verleiht dem Begriff mechatronisches System makroskopischen Charakter: Der Begriff umfasst alle an einer Funktionserfüllung beteiligter Komponenten. Vor dem Hintergrund der

mechatronischen Produktentwicklung bedeutet dies, dass auch sämtliche entwicklungsunterstützenden Einrichtungen, an die das Produkt direkt angekoppelt wird (z. B. Prüfstände, Simulationsumgebungen), zu einem mechatronischen System gezählt werden.

Der Begriff *Gerät* bezeichnet im allgemeinen Sprachgebrauch einen konkreten Gegenstand, der eine Wirkung verursacht [DUDEN 02]. Hieraus erfolgt die Ableitung einer Definition für den Begriff *mechatronisches Gerät*:

Definition 2-2: Mechatronisches Gerät

Lassen sich in einem mechatronischen System räumlich abgeschlossene Teilsysteme mit definierten Schnittstellen isolieren, so werden diese Teilsysteme als *mechatronisches Gerät* bezeichnet. Ein *mechatronisches Gerät* kann wiederum für sich als ein räumlich eng begrenztes bzw. hoch integriertes mechatronisches System, bestehend aus mehreren Teilkomponenten, betrachtet werden.

Eine Schlüsselcharakteristik mechatronischer Geräte ist die Heterogenität der Teilkomponenten und die hohe Komplexität im Vergleich zu klassischen Produkten [KALLENBACH 01]. Ebenfalls kennzeichnend ist der hohe Anteil an Software in mechatronischen Produkten, der zur bestimmenden Komponente bei der Funktionserbringung geworden ist [BMBF 00], [KOÇ+ 03], [SCHÖN 00].

Im allgemeinen Sprachgebrauch sowie in einschlägiger Literatur taucht oftmals auch der Begriff *mechatronisches Produkt* auf. Deshalb erfolgt zusätzlich eine Einordnung dieses Begriffs in Bezug zu den gegebenen Definitionen: Mechatronische Geräte, die eine Marktreife erlangt haben, werden im behandelten Kontext als *mechatronische Produkte* bezeichnet.

2.1.2 Entwicklungsprozess mechatronischer Produkte

Mit dem Ausdruck Produktentwicklung wird der Vorgang bezeichnet, den eine Ware von der Idee bis zum verkaufsfähigen (serienreifen) Erzeugnis nimmt [GAUSEMEIER+ 99], [WIKIPEDIA 04]. LINDEMANN bezeichnet die Produktentwicklung als Organisationseinheit, Tätigkeit und Unternehmensprozess [LINDEMANN 00A]. Die Summe aller Vorgänge, die stattfinden, damit ein Produkt bis zum verkaufsfertigen Erzeugnis gelangt, wird unter der Bezeichnung Entwicklungsprozess zusammengefasst. Im Folgenden wird diese sehr allgemeine Aussage weiter präzisiert. Ein Ansatz dazu ist das Phasenkonzept wie es HABERFELLNER für die Systementwicklung definiert (vgl. Abbildung 2-2).

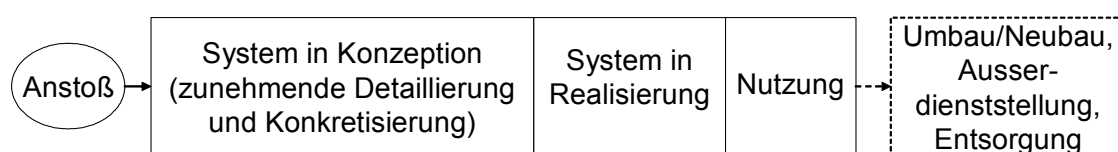


Abbildung 2-2: Phasenkonzept für Produktentwicklungsprozesse [HABERFELLNER+ 02]

Detaillierte Ansätze wie beispielsweise in [ANDERL 02], [EHRENSPIEL 95] und [LINDEMANN 00A] beschreiben den vollständigen Lebenszyklus eines Produkts. Aufgeschlüsselt werden hier auch die über den eigentlichen Entwicklungsprozess hinausgehenden Phasen Produktvertrieb, Produktnutzung und gegebenenfalls Recycling bzw. Entsorgung. Für die vorliegende Arbeit erfolgt eine eingeschränkte Interpretation des Begriffs *Entwicklung mechatronischer Systeme*:

Definition 2-3: Entwicklung mechatronischer Systeme

Der Umfang des Begriffs Entwicklung mechatronischer Geräte konzentriert sich in dieser Arbeit auf Tätigkeiten der Phase *System in Realisierung* des Phasenmodells in Abbildung 2-2. Additiv sind Tätigkeiten in der Nutzungsphase des Product-Lifecycles eingeschlossen, die über die üblichen Wartungs- und Service-Funktionen³ hinausgehen und der Verbesserung des Produkts dienen.

Nicht nur das behandelte Themengebiet, sondern viele Bereiche der modernen Wissenschaft sind heutzutage durch hohe Komplexität und Interdisziplinarität gekennzeichnet. Die Domäne der Mechatronik muss diesem Kreis zugeordnet werden. Nach HEIMANN stellt die Integration der drei Teildisziplinen im Entwicklungsprozess mechatronischer Systeme eine wesentliche Besonderheit dar [HEIMANN+ 01]. Dies gilt für alle Phasen des Entwicklungsprozesses wie außerdem aus dem, im Rahmen des Forschungsprojekts *EQUAL*⁴ entwickelten 3-Ebenen-Modell⁵ für den Produktentwicklungsprozess in der Mechatronik hervor geht.

2.1.3 Ausprägungen des Zusammenwirkens

Die Zusammenarbeit mehrerer Beteiligter im Team ist Kennzeichen der modernen Produktentwicklung. Vor diesem Hintergrund werden häufig die Begriffe Kooperation und Kollaboration verwendet. In unserem heutigen Umfeld werden diese Begriffe oft in verschiedenem Kontext verwendet, wenn es um die Beschreibung gemeinschaftlichen Wirkens geht. Die genaue Bedeutung, d.h. die Art und Weise dieses Wirkens, wird im Zusammenhang mit der Verwendung der Begriffe meist nicht festgelegt. Dabei ist es von enormer Wichtigkeit, die optimale Form des Zusammenwirkens präzise in Abhängigkeit der jeweiligen Anforderungen zu definieren, um letztendlich einen Emergenzeffekt⁶ zu erzielen.

³ Eine Übersicht gibt beispielsweise [BENDER+ 02B].

⁴ Verbundprojekt EQUAL – Methoden zur Unterstützung der entwicklungsbegleitenden Qualitätssicherung von eingebetteter Software; Förderkennzeichen 02PP1060.

⁵ Das sog. 3-Ebenen-Modell basiert auf dem klassischen V-Modell für die Softwareentwicklung [V-MODELL 97] und wurde um die Bedürfnisse der mechatronischen Entwicklung erweitert [BENDER 04B] [KOÇ+ 01].

⁶ Erreichen eines besseren Ergebnisses aufgrund des Zusammenwirkens mehrerer Personen [STOLLER-SCHAI 03].

Nachfolgender Abschnitt nimmt eine Präzisierung der so genannten *Ko*-Begriffe vor, die im modernen Sprachgebrauch verbreitet sind. Gemäß [DUDEN 02] und [DROSDOWSKI+ 1963] bedeutet die Vorsilbe *Ko*: *zusammen mit, gemeinsam*. STOLLER-SCHAI identifiziert drei maßgebliche *Ko*-Begriffe in Verbindung mit der Zusammenarbeit beteiligter Personen: *Koordination*, *Kooperation* und *Kollaboration* [STOLLER-SCHAI 03]. Diese sind wie folgt definiert:

Definition 2-4: Koordination

"Koordination ist das wechselseitige Abstimmen von Handlungen und Tätigkeiten durch die organisatorische und zeitliche Festlegung von Handlungs- und Tätigkeitsabläufen (Prozesse) durch implizite oder explizite Regeln sowie durch Zuordnung von Teilhandlungen an verschiedene Rollenakteure" [STOLLER-SCHAI 03].

Koordination stellt somit die Basis jedes gemeinschaftlichen Handelns dar. Von besonderer Relevanz sind die Begriffe Kooperation und Kollaboration. Sowohl im Sprachgebrauch als auch in der Literatur wie beispielsweise bei [ALLESPOCH 99] erfolgt oftmals eine Vermischung dieser beiden Begriffe. Vor dem Hintergrund dieser Arbeit wird eine klare Trennung bevorzugt, wie sie beispielsweise von STOLLER-SCHAI, LUBICH und TODESCO vorgenommen wird [LUBICH 95], [TODESCO 01]. Der Begriff Kooperation besitzt dabei eine Doppeldeutigkeit:

Definition 2-5: Kooperation

"Kooperation ist die Manifestation der Bereitschaft zweier oder mehrerer Individuen miteinander zu handeln um ein gemeinsames Ziel zu erreichen (= kooperatives Setting)."

"Kooperation ist die arbeitsteilige Bearbeitung von Teilaufgaben, die zu einer Gesamtaufgabe gehören und die innerhalb eines gemeinsam gestalteten und ausgehandelten Kontextes (kooperatives Setting) und unter Verwendung gemeinsamer Ressourcen individuell abgearbeitet werden."

[STOLLER-SCHAI 03]

Definition 2-6: Kollaboration

"Kollaboration ist die von zwei oder mehreren Personen an gemeinsamen Zielen ausgerichtete, direkte und sich wechselseitig beeinflussende tätige Auseinandersetzung zur Lösung oder Bewältigung einer Aufgabe oder Problemstellung. Dies geschieht innerhalb eines gemeinsam gestalteten und ausgehandelten Kontextes (gemeinsamer Bedeutungsraum, kooperatives Setting) in physischer Ko-Präsenz und unter Verwendung gemeinsamer Ressourcen (= kollaboratives Handeln)."

[STOLLER-SCHAI 03]

Anhand dieser Definitionen wird zunächst folgende Interpretation des Begriffes Kooperation vorgenommen: Erstens ist Kooperation, im Sinne von Kooperationsbereitschaft, eine Voraussetzung für Kollaboration. Zweitens bedeutet Kooperation die individuelle Abarbeitung von Teilaufgaben in einem gemeinsamen Projekt. Kollaboration hingegen repräsentiert die gemeinsame wechselseitige Bearbeitung solcher Teilaufgaben. Von besonderer Bedeutung für die vorliegende Arbeit ist eine Erweiterung des Begriffes Kollaboration: Es handelt sich um die so genannte *E-Kollaboration* (bzw. engl. *E-Collaboration*). Aufgrund unterschiedlichster Interpretationen sei folgende Festlegung getroffen:

Definition 2-7: E-Collaboration

"E-Collaboration ist die von zwei oder mehreren Personen an gemeinsamen Zielen ausgerichtete, direkte und sich wechselseitig beeinflussende tätige Auseinandersetzung zur Lösung oder Bewältigung einer Aufgabe oder Problemstellung. Dies geschieht innerhalb eines gemeinsam gestalteten und ausgehandelten Kontextes (gemeinsamer Bedeutungsraum, kooperatives Setting) und unter Verwendung gemeinsamer Ressourcen basierend auf einer rechnergestützten Infrastruktur." [STOLLER-SCHAI 03]

Die beschriebenen Begriffe *Koordination*, *Kooperation*, *Kollaboration* und *E-Collaboration* werden auch unter dem Begriff der *Koaktion* zusammengefasst, dessen Bedeutung ganz allgemein das gemeinschaftliche Handeln, bzw. agieren ist. Jegliche Koaktion findet vor dem Hintergrund eines Handlungsfeldes statt. Dieses bildet gewissermaßen die Rahmenbedingungen⁷ der gemeinschaftlichen Aktivitäten und beinhaltet den bearbeiteten Aufgabenkomplex, der wiederum Auswirkungen auf die kooperativen Prozesse hat [STOLLER-SCHAI 03]. Das hier betrachtete Handlungsfeld ist die in Abschnitt 2.1.2 eingeführte Entwicklung mechatronischer Geräte.

Betrachtet man den Fall, dass sämtliche an einer Koaktion beteiligten Personen und Ressourcen am Ort des Handelns physisch präsent sind, so ist leicht nachvollziehbar, dass von den Beteiligten umgehend eine Reaktion auf Ereignisse erfolgen kann. Im Fall der räumlichen Trennung der Beteiligten muss der Zusammenschluss zuerst mittels geeigneter Medien erfolgen. Je nach Art des Mediums entsteht dabei ein Zeitverzug sowohl für die Übertragung von Informationen zu den einzelnen Teilnehmern als auch für die Übermittlung der Antworten. Der Zeitverzug kann sich dabei in der Spanne von sehr gering (z. B. Telefongespräch) bis hin zu unbestimmt (z. B. Briefverkehr) erstrecken. Wird das Ziel verfolgt, eine verteilte Koaktion möglichst so durchzuführen als ob die Beteiligten vor Ort wären, so müssen, aus ergonomischen Gründen,

⁷ Beispiele für Rahmenbedingungen sind: beteiligte Personen, verfügbare Mittel, strategische Orientierung etc.

Zeitschranken hinsichtlich der maximalen Verzögerung der Reaktion auf Ereignisse⁸ eingehalten werden [VDMA 04].

2.1.4 Synchroner und Asynchroner Kooperation

Systeme zur Unterstützung der Kooperation zwischen Beteiligten existieren in unterschiedlichsten Ausprägungen, so dass eine grundlegende Ordnung notwendig ist. In der Literatur ist die Basisklassifizierung in so genannte *synchrone* und *asynchrone* kooperative Anwendungen weit verbreitet. Diese Klassifizierung lässt sich auf verschiedenen Ebenen des ISO/OSI-Referenzmodells⁹ vornehmen [LITTLE+ 90]. Die hier gegebenen Definitionen beziehen sich auf die höheren Anwendungsschichten des Modells. In Anlehnung an [LUBICH 95] und [STURZEBECKER 00] definieren sich die beiden Klassen wie folgt:

Definition 2-8: Synchrone Systeme

Synchrone Systeme sind durch gleichzeitiges Zusammenarbeiten der Kommunikationspartner mit einem hohen Interaktionsgrad gekennzeichnet. Der Austausch von Informationen zwischen den Kommunikationspartnern findet gleichlaufend statt, wobei alle Beteiligten kommunikationsbereit sind und der Informationsaustausch innerhalb einer maximalen Zeitspanne garantiert ist.

Synchrone Systeme beinhalten oftmals Multimediaunterstützung. Beispiele sind das Telefon bzw. Audio/Video-Konferenzsysteme.

Definition 2-9: Asynchrone Systeme

Asynchrone Systeme verlangen keine gleichzeitige Anwesenheit der Kommunikationsteilnehmer sondern unterstützen eine inkrementelle Arbeitsweise. Das Zusammenarbeiten der Teilnehmer erfolgt zu unterschiedlichen, nicht aufeinander abgestimmten Zeiten wobei eine maximale Zeitspanne für den Informationsaustausch nicht garantiert ist.

Bekanntere Beispiele für Asynchrone Systeme sind Email oder Internet-Foren. Im Zusammenhang mit kooperativen Tätigkeiten tauchen oftmals die Begriffe *Online* und *Offline* auf.

⁸ Aus Gründen der Vereinfachung wird im Folgenden auch die Ausgabe einer Information als Ereignis betrachtet.

⁹ OSI- (*Open Systems Interconnection*) Referenzmodell der ISO (International Standards Organisation) zur Schaffung und Verbindung offener Kommunikationsmodelle, entwickelt in den 80er Jahren [BENDER 04A]. (Aktuelle Spezifikation siehe [ISO 94])

Definition 2-10: Online

Ist ein Teilnehmer eines kooperativen Verbundes Online, so ist dieser Teilnehmer über die Kooperationsanwendung aktiv und permanent in das kooperative System eingebunden.

Definition 2-11: Offline

Ist ein Teilnehmer eines kooperativen Verbundes Offline, so erfolgt die Ausführung einer Teilaufgabe unabhängig und ohne Verbindung zum kooperativen System.

Die Bedeutung leitet sich, ebenfalls auf der Ebene der höheren Anwendungsschichten des ISO/OSI-Modells, aus der Klassifizierung in synchrone und asynchrone Systeme ab: Der Begriff *Online* steht im engen Zusammenhang mit der Klasse der synchronen Systeme. Aus den Definitionen ergibt sich zudem der Zusammenhang des Begriffs *Offline* mit der Klasse der asynchronen Systeme.

2.2 Merkmale der Entwicklung mechatronischer Produkte

2.2.1 Wirtschaftliche Bedeutung mechatronischer Produkte

Viele technologische Bereiche sind heutzutage verstärkt durch den Einsatz mechatronischer Lösungen gekennzeichnet. Einige Beispiele hierfür sind: Automobil (z. B. ABS, Motorsteuergeräte [GÖBEL 04], [SCHÖNER 02]), Flugzeug (z. B. Hochauftriebssystem [SCHÄFER 04]), Werkzeugmaschinen [SIEGLER 98], Automatisierungstechnik (z. B. feldbusfähige Aktoren und Sensoren, Mehrachshandhabungsgeräte [KOÇ+ 03], [MEYER 04], [RÖMER 03]), Präzisionsmaschinen sowie die Konsumgüterindustrie (z. B. Videogeräte, Photokameras [GAUSEMEIER+ 00]).

Diese Beispielliste veranschaulicht eine der wichtigen derzeit stattfindenden Entwicklungen: Die Durchdringung des Maschinenbaus, der Fahrzeugtechnik, der feinmechanischen und automatisierungstechnischen Geräte etc. mit Mechatronik [ISERMANN 02]. Durch modulare Strukturen (kleine, kostengünstige und kompatible Einheiten werden intelligent kombiniert) ermöglicht die Mechatronik Kostenersparnis bei zugleich geringerer Anfälligkeit der Produkte [LINDEMANN 00B]. Die enorme Bedeutung der Mechatronik belegt auch GAUSEMEIER: "Mechatronik stellt für die Schlüsselbranchen Maschinen- und Anlagenbau, Automobilindustrie und Elektrotechnik ein weiteres Erfolgspotenzial dar" [GAUSEMEIER+ 00].

Eine solche Bedeutung zwingt jedoch auch dazu, mechatronische Produkte effektiv und innovativ zu entwickeln. Hierfür sind entsprechende

Entwicklungsmethoden und Vorgehensweisen notwendig. Die Besonderheiten des Entwicklungsprozesses mechatronischer Geräte werden nachfolgend betrachtet.

2.2.2 Charakteristik des Entwicklungsprozesses in der Mechatronik

Ziel dieses Kapitels ist es, die Charakteristika des Entwicklungsprozesses mechatronischer Systeme darzustellen. Aufbauend auf die eingangs gegebene Definition wird deshalb zunächst eine detailliertere Antwort auf die Frage: *Was beinhaltet ein mechatronisches System?* gegeben. Eine im Vergleich zu Kapitel 2.1.1 weitaus granularere Darstellung eines typischen mechatronischen Systems gibt Abbildung 2-3. Auffällig ist die Vielzahl an unterschiedlichen Komponenten aus unterschiedlichen Domänen:

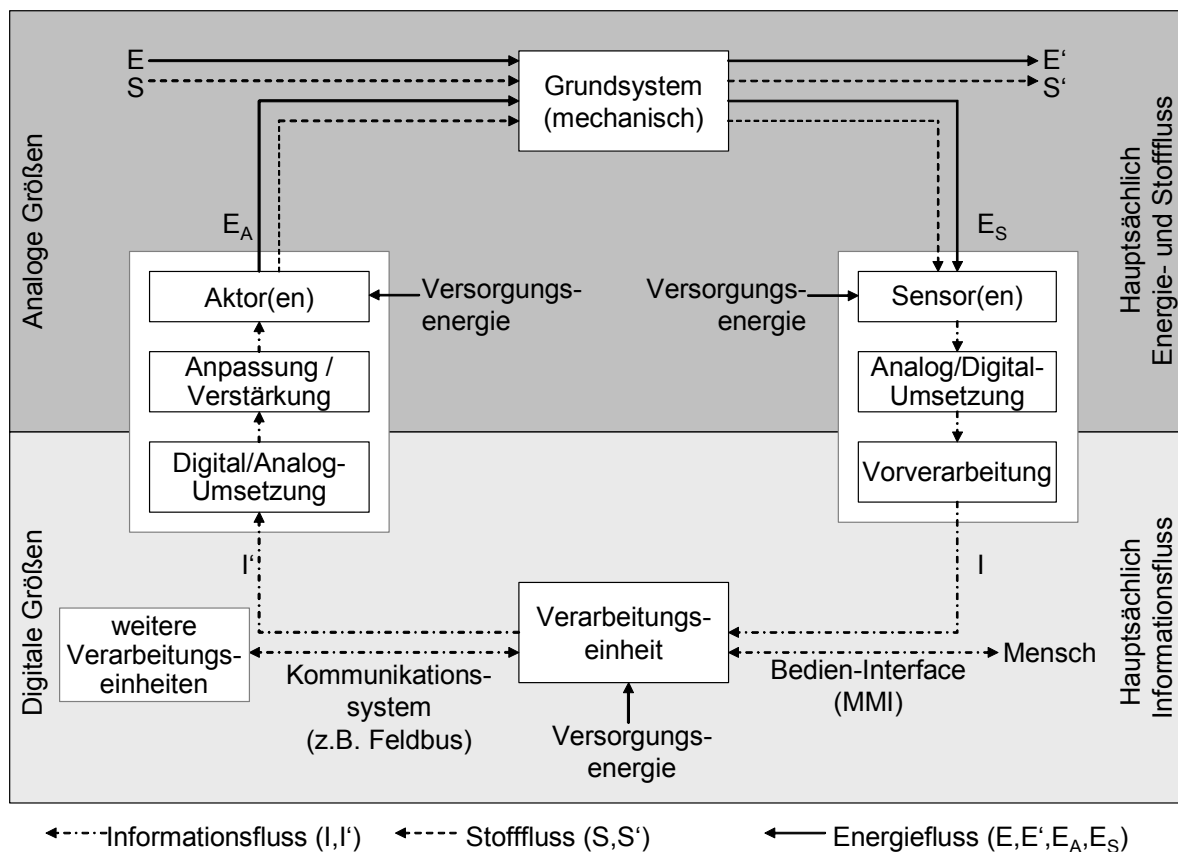


Abbildung 2-3: Das mechatronische Basissystem [KALLMEYER 98], [LIPPOLD 01]

Die im dargestellten mechatronischen Basissystem enthaltenen Komponenten verdeutlichen die Interaktion der drei Disziplinen Mechanik (Technische Mechanik, Maschinenbau, Feinwerktechnik), Elektrotechnik/Elektronik (Mikroelektronik, Leistungselektronik, Messtechnik, Aktoren), Softwaretechnik und Regelungstechnik¹⁰ [HEIMANN+ 01], [GAUSEMEIER+ 00]. Wie bereits

¹⁰ In diverser Literatur wie beispielsweise [HEIMANN+ 01]] erfolgt eine Zusammenfassung der Disziplinen Regelungstechnik und Softwaretechnik zum übergreifenden Gebiet der Infor-

angedeutet, weisen mechatronische Systeme neben dem hohen Integrationsgrad auch ein hohes Maß an Komplexität auf [DOHMEN 02], [VDI 03]. Die "rasche Zunahme der funktionellen Komplexität, heterogener Teilsysteme" registriert schon früh KALLENBACH [KALLENBACH 94]. KALLMEYER sieht die Hauptfunktionen mechatronischer Systeme charakteristischerweise durch mehrere Domänen bestimmt und das Potential solcher Systeme in der integrativen, domänenübergreifenden und interdisziplinären Zusammenarbeit [KALLMEYER 98].

Dies impliziert zwangsläufig hohe Anforderungen an den Entwicklungsprozess für mechatronische Produkte [VDI 03]. Die Erfüllung der Anforderungen ist umso schwieriger, als dass die Teildisziplinen durchwegs ihre eigenen, domänenspezifischen Vorgehensmodelle und Entwurfssysteme für die Produktentwicklung besitzen [KALLENBACH+ 97]. Diese sind jeweils alleine betrachtet für den domänenübergreifenden mechatronischen Entwicklungsprozess unzureichend. Zu diesem Schluss kommt STÜTZEL nach einer eingehenden Untersuchung der fachspezifischen Entwicklungsprozesse und dem Versuch, diese auf die Mechatronik zu übertragen [STÜTZEL 02].

Für die Entwicklung mechatronischer Systeme in unterschiedlichsten Anwendungsbereichen hat sich auf der Makroebene mittlerweile das so genannte V-Modell als Vorgehensmodell [V-MODELL 97] etabliert¹¹ [VDI 03]. Dieses Modell besitzt seinen Ursprung in der Softwaretechnik und ist als Referenzmodell für die Produktentwicklung in der Mechatronik anwendbar [GAUSEMEIER+ 00]. Im Rahmen des Forschungsprojektes EQUAL wurde das V-Modell hinsichtlich der Anforderungen der Mechatronik weiterentwickelt. Als Resultat entstand das so genannte 3-Ebenen-Modell (vgl. Abbildung 2-4) welches eine Verfeinerung des prinzipiellen Phasenkonzepts für Produktentwicklungsprozesse aus Abbildung 2-2 darstellt.

Das 3-Ebenen-Modell nimmt eine Einteilung in System-, Modul- und Komponentenebene vor. Die geforderte gleichberechtigte Einbindung aller beteiligten Disziplinen ist hierbei grundlegend berücksichtigt [BENDER 03]. Anhand dieser Einteilung lassen sich drei verschiedene Tätigkeitsbereiche im Entwicklungsprozess definieren:

1. Im linken Ast finden sich auf der Systemebene das Design und auf der Subsystemebene die spezifizierenden Tätigkeiten. Dieser Bereich wird im Folgenden als der spezifizierende Bereich zusammengefasst.
2. Auf der Komponenten-Ebene in der Mitte der Grafik erfolgen die umzusetzenden Tätigkeiten für die Teilkomponenten des Systems. Diese können im Falle einer genauen Spezifikation weitgehend losgelöst von einander ausgeführt werden.

mationsverarbeitung mit den Schwerpunkten Systemtheorie, Prozessdatenverarbeitung und künstliche Intelligenz.

¹¹ Belege hierfür finden sich in einer Vielzahl aktueller Publikationen wie [BOTELLÉ+ 04], [KOÇ+ 01], [NITSCHKE 04], [OTTERBACH 04], [SCHÄFER 04].

3. Der rechte Ast, der sowohl auf der Subsystem- als auch auf der Systemebene sämtliche Integrationsprozesse beinhaltet, ist durch stetige und intensive Beteiligung des realen Produkts bzw. dessen Teilkomponenten¹² aus unterschiedlichen Bereichen gekennzeichnet [HEIMANN+ 98] (integrative Phasen).

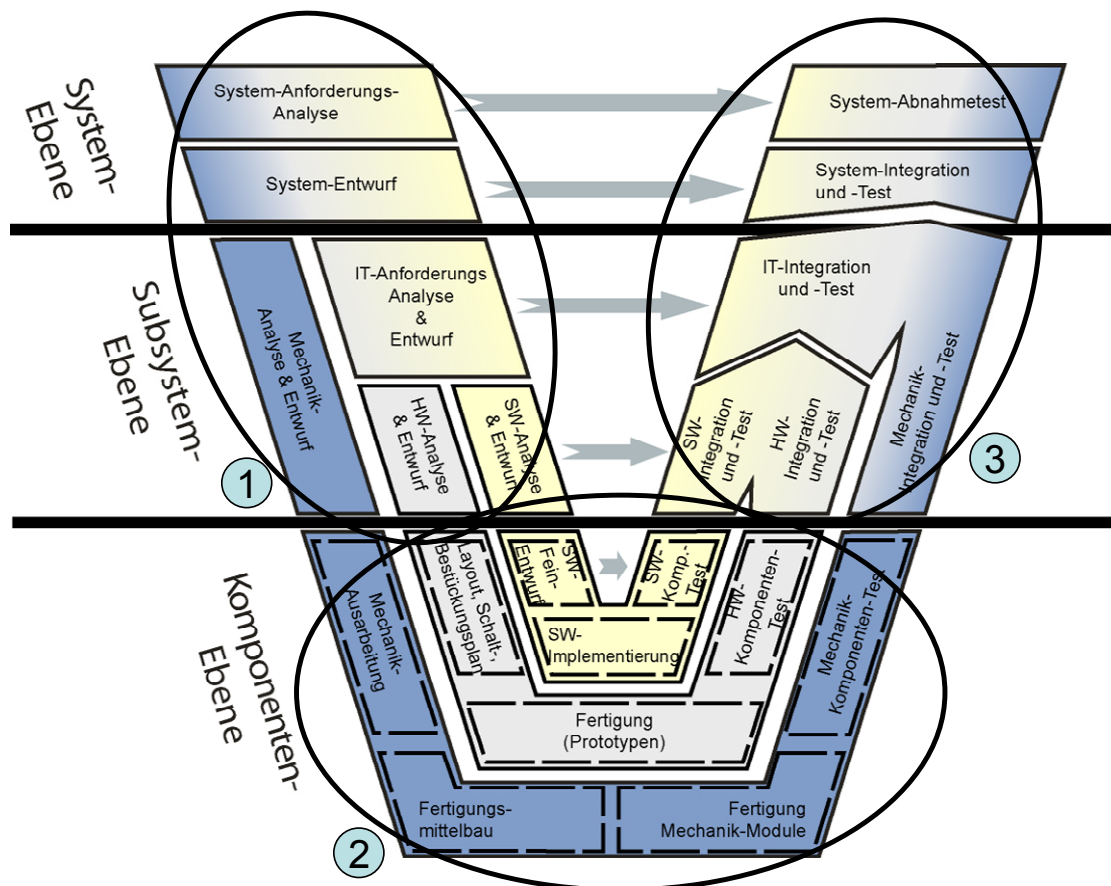


Abbildung 2-4: 3-Ebenen-Vorgehensmodell der Mechatronik [BENDER 04B], [RUSS 04]

Die interdisziplinäre Zusammenarbeit der beteiligten Fachbereiche am realen Entwicklungsgegenstand erfordert naturgemäß ein hohes Maß an Koordination. Aufgaben müssen in diesen Phasen durch eine enge Zusammenarbeit der Beteiligten gelöst werden, da Tätigkeiten am Gerät nicht mehr losgelöst, sondern im mechatronischen Gesamtzusammenhang erfolgen müssen. Die Betrachtungen konzentrieren sich im weiteren Verlauf dieser Arbeit auf die integrativen Phasen des mechatronischen Entwicklungsprozesses.

¹² Der Begriff *Komponente* ist in verschiedenen Disziplinen jeweils mit eigenen Bedeutungen belegt. Hier bedeutet der Begriff ein beliebiges Teil des zu entwickelnden mechatronischen Produktes entweder in Form von Mechanik, Elektronik oder Software.

2.2.3 Werkzeuge im Entwicklungsprozess

In der mechatronischen Produktentwicklung stellt sich die Aufgabe sowohl die komplexen Teilgebiete zu beherrschen als auch diese effizient zu integrieren. Diese Herausforderung kann heutzutage nicht ohne den Einsatz geeigneter Rechnerwerkzeuge effizient und wirtschaftlich gelöst werden [DOHMEN 02]. Der Einsatz von rechnergestützten entwicklungsunterstützenden Werkzeugen wird nicht nur explizit von Experten gefordert [GAUSEMEIER+ 00], [HEIMANN+ 98], [ISERMANN 99] sondern auch in vielen Erfahrungsberichten immer wieder belegt, beispielsweise in [VDI 97], [VDI 04]. Betroffen sind hiervon alle drei der in 2.2.2 beschriebenen Phasen.

In den hier schwerpunktmäßig betrachteten integrativen Phasen stehen konkret Tätigkeiten wie Prototypenbau, Integration von Elektronik-Komponenten, Software und Mechanik-Komponenten, Test und Inbetriebnahme, Erprobung und Optimierung im Mittelpunkt. Am Ende steht die experimentelle Verifikation des mechatronischen Gesamtsystems, oftmals auch eine Zertifizierung des Produkts durch unabhängige Institutionen. Diese Art von Tätigkeiten wird vor allem durch rechnergestützte Prüfstände, Prüf- und Test-Tools sowie Analyse-Tools unterstützt [AHREND+ 04a], [AHREND+ 04b], [LANDWEHR 01]. Eine besondere Rolle in der mechatronischen Geräteentwicklung spielt, sowohl in den umsetzenden als auch in den integrativen Phasen, der Einsatz so genannter *HIL*-Techniken¹³. Dabei handelt es sich nach [VDI 03] um die "Integration von realen Komponenten (Bauteilen) und Systemmodellen in eine gemeinsame Simulationsumgebung". "Die *HIL*-Nachbildung (Simulation) dynamischer Systeme durch physikalische und mathematische Modelle muss dabei in Echtzeit¹⁴ und unter Nachbildung der physikalischen Lasten erfolgen". *HIL*-Umgebungen stellen bisweilen selbst hochkomplexe Systeme aus Aktoren und Sensoren, Echtzeitrechnern, Simulationsanwendungen und Ablaufumgebungen dar [SCHMITZ 97]. Für GAUSEMEIER sind *HIL*-Techniken fester Bestandteil des mechatronischen Produktentwicklungsprozesses wie in Abbildung 2-5 gezeigt wird.

Analog zu den unterschiedlichen, an der Entwicklung eines mechatronischen Produktes beteiligten Disziplinen kennzeichnet eine breite Palette und hohe Spezialisierung den Bereich der Entwicklungswerkzeuge. Die im Gesamtsystem zu integrierenden Teilkomponenten stammen oftmals von unterschiedlichen Zulieferern¹⁵ [ISERMANN 99]. Folglich weisen Entwicklungswerkzeuge deshalb

¹³ *Hardware-In-The-Loop*

¹⁴ Rechnersysteme, die mit externen technischen Systemen in Wechselwirkung stehen und bei denen die Datenverarbeitung zeitlich mit den in den externen Systemen ablaufenden Vorgängen Schritt hält, werden als *Echtzeitsysteme* bezeichnet [LAUBER+ 99].

¹⁵ Als wichtiger Bestandteil heutiger mechatronischer Produkte sind Bussysteme zu nennen [ISERMANN 99]. Insbesondere diese offenen Systeme ermöglichen und fördern die Integration von Komponenten unterschiedlichster Hersteller.

zusätzlich herstellereigenschaftliche Eigenschaften auf¹⁶. Parallel zur steigenden Produktkomplexität [RUSS+ 04] besitzen auch die entwicklungsunterstützenden Werkzeuge selbst mittlerweile eine hohe Komplexität. Erfahrungen in Forschungsprojekten [BENDER+ 02A], [RÖMER 03] und bei Eigenentwicklungen zeigen zudem, dass manche Werkzeuge speziell in den integrativen Phasen nur in großen Zeitabständen benötigt werden¹⁷. Diese gelegentliche Nutzung bewirkt, dass sich keine stetig vorhandene Kompetenz im Umgang mit den Werkzeugen innerhalb des Unternehmens ausbildet. Treten komplexe Sonderfälle bei Arbeiten mit bestimmten Rechnerwerkzeugen auf, so verursacht die Lösung einer solchen Situation Zeitverzug. Zusätzlich besteht die Gefahr der ineffizienten oder gar falschen Nutzung von Werkzeugen, die eigentlich eine Effizienzsteigerung bewirken sollten. Der effektive und wirtschaftliche Einsatz entsprechender Werkzeuge erfordert also viel Spezialwissen seitens der Entwickler. In Folge dieser festgestellten Spezialisierung ist es nur schwer oder auch gar nicht möglich, die gesamte Expertise in einem kleinen Personenkreis zu vereinen, was den Einsatz verschiedener Spezialisten bedingt.

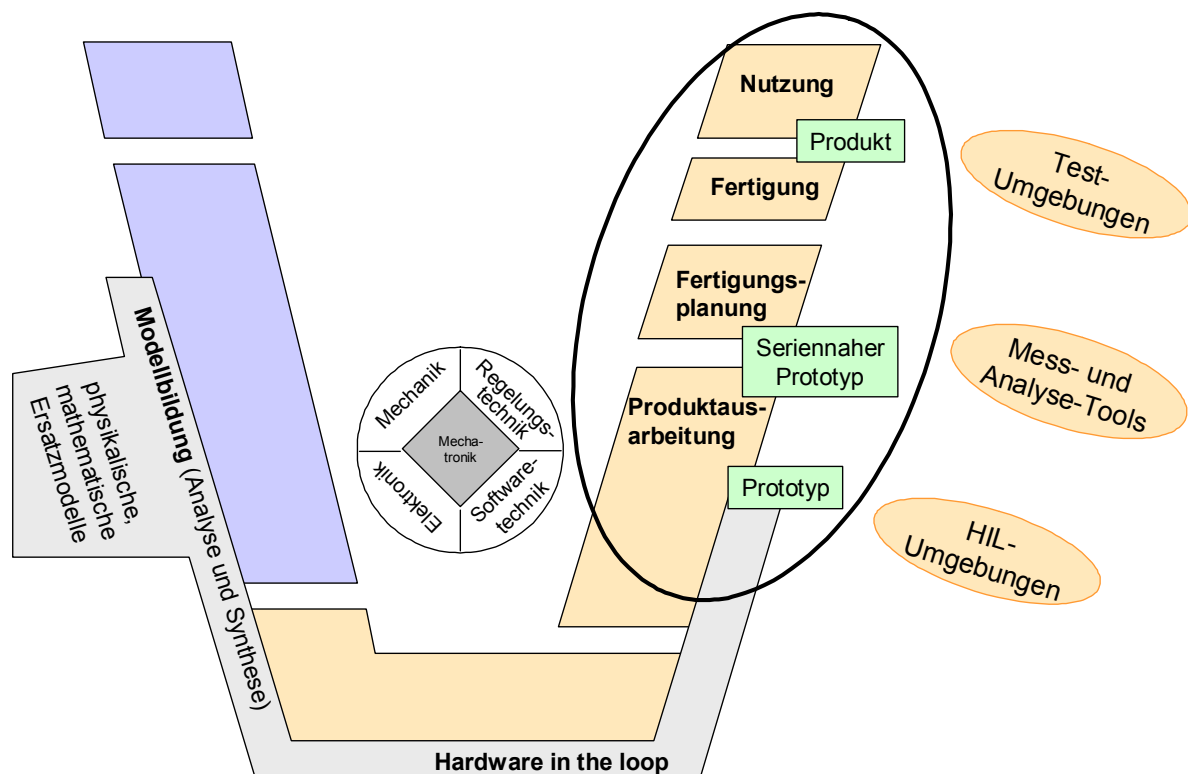


Abbildung 2-5: Werkzeuge in den integrativen Phasen (basierend auf [Gausemeier+ 00])

¹⁶ Siehe hierzu auch [BENDER+ 02A]: Das Verbundprojekt *Dezentrale kooperierende offene Mikrosysteme*, gefördert durch das Bundesministerium für Bildung und Forschung (BMBF), Förderkennzeichen 16SV894/1

¹⁷ Beispiel: Inbetriebnahme-Tool für intelligente Antriebe bei der Inbetriebnahme eines Hochregallagers, siehe [MEYER 04].

2.2.4 Integration multidisziplinärer Wissensträger

In Kapitel 2.2.2 wurde die Multidisziplinarität der Mechatronik und folglich die hohe Spezialisierung von Teilaufgaben bei der mechatronischen Geräteentwicklung identifiziert. Kapitel 2.2.3 kommt zu dem Schluss, dass der Entwicklungsprozess nicht ohne den Einsatz rechnergestützter Werkzeuge stattfinden kann und die Verwendung solcher Werkzeuge ebenfalls zu spezialisierten Aufgaben führt. Dabei steht der Entwicklungsprozess vor dem Hintergrund einer hohen Gesamtkomplexität der zu entwickelnden Produkte, die immer wieder in einschlägiger Literatur wie beispielsweise [BENDER+ 02A], [DOHMEN 02], [MILBERG 96] belegt wird.

Dies führt zu dem Ergebnis, dass auch die integrativen Tätigkeiten nicht mehr von einer kleinen Personengruppe effizient durchgeführt werden können. Vielmehr erfolgt die Entwicklung durch möglichst effektive Zusammenarbeit von Wissensträgern aus den einzelnen Aufgabengebieten in einem Netzwerk von Spezialisten. SCHÖN geht sogar noch einen Schritt weiter und definiert einen Teilnehmer an einer mechatronischen Produktentwicklung als einen Experten in seinem Fachgebiet, der wenig tief greifende Kenntnisse in anderen Disziplinen besitzt und keinesfalls auf allen Gebieten gleichermaßen bewandert ist [SCHÖN 00]. In den integrativen Phasen stehen zudem die physikalischen Komponenten des Systems im Mittelpunkt der Entwicklungstätigkeiten (Prototypen, Prüfstände, Messgeräte, reale Umgebung). Dies bewirkt eine geografische Bindung der mit den Komponenten interagierenden Beteiligten an den jeweiligen Ort der Entwicklung. Beispiele hierfür sind: Optimierungen an der Hardware, Platzieren von Messeinrichtungen, Aufzeichnung des Verhaltens zur Laufzeit etc.

2.2.5 Integration externer Expertise in den Entwicklungsprozess

Aus den genannten Gründen folgt, dass es einen immensen Aufwand darstellt, die gesamte, für die Integration benötigte Expertise und die technische Ausrüstung permanent am Ort der Entwicklung vor zu halten. Dies wäre für viele Entwicklungsvorhaben unwirtschaftlich: In großen Unternehmen sind Fachabteilungen oft über mehrere Standorte verteilt, viele KMU verfügen gar nicht über entsprechende Ressourcen. Betroffen hiervon sind umso mehr diejenigen Firmen, die über ein geringeres Produktspektrum mit langen Produktzyklen verfügen und deshalb Neuentwicklungen in größeren Zeitabständen durchführen. Ist die Verteilung von Aufgaben an externe Ressourcen in den Einzeldisziplinen ohnehin oft anzutreffen, so ergibt sich durch die gesteigerte Komplexität der mechatronischen Entwicklung erst recht ein solcher Bedarf. Die VDI-Richtlinie 2206 [VDI 03] empfiehlt generell, mechatronische Entwicklungsteams auf einen Kern zu reduzieren und Spezialisten aus weiteren Bereichen bei Bedarf hinzuzuziehen. Als Konsequenz entsteht die Notwendigkeit, Teilaufgaben der Entwicklung nicht im Unternehmen zu bearbeiten, sondern auf externe Entwicklungspartner zurückzugreifen. Hierfür existieren grundsätzlich zwei Möglichkeiten:

1. Externe Fachleute werden für die Dauer der Aufgabe vor Ort in den Entwicklungsprozess in einem Unternehmen integriert.
2. Teilaufgaben werden an andere Firmen oder Fachabteilungen vergeben (Outsourcing).

Outsourcing kann dann effizient angewendet werden wenn die folgenden Bedingungen erfüllt sind:

- Es handelt sich um abgeschlossene Teilaufgaben.
- Es erfolgt eine genaue Spezifikation der Teilaufgaben.
- Die Teilaufgaben sind ausreichend umfangreich.

Ein Beispiel ist der Trend in der Automobilindustrie, Hardware und Software für Steuergeräte von unterschiedlichen Quellen zu beziehen [GORONCY 04]. Vor allem in größeren Systemen ist es üblich, dass Prozess (Physik bzw. Mechanik) und die Automatisierung (Hardware, Software, Regelungstechnik) von verschiedenen Herstellern stammen [ISERMANN 99]¹⁸. Abbildung 2-6 veranschaulicht den Sachverhalt, wobei verschiedene Schattierungen der Kreisobjekte unterschiedliche Dienstleister bzw. Zulieferer repräsentieren.

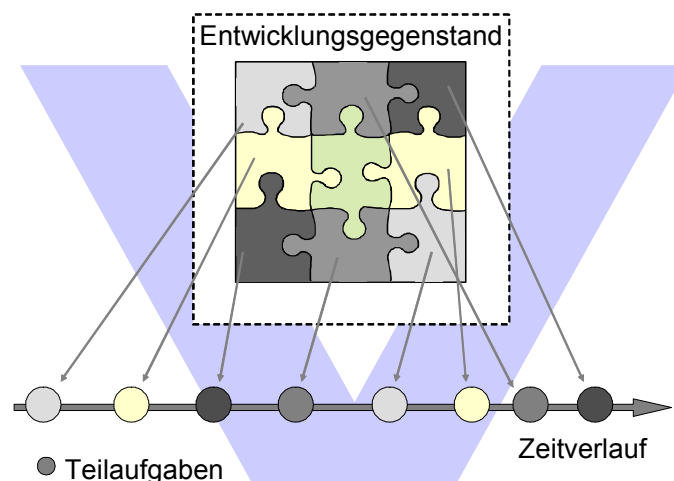


Abbildung 2-6: Teilaufgaben im Verlauf des Entwicklungsprozesses

Der geschilderte Sachverhalt zeigt den Bedarf an Outsourcing-Lösungen auf. In der Literatur wird ein solcher Bedarf beispielsweise schon früh durch [BULLINGER+ 96], [EVERSHEIM+ 96] und [RING 95] belegt. [GAUSEMEIER+ 00], [HEIMANN+ 01], [ISERMANN 99] und [VDI 03] fordern jedoch übereinstimmend die Integration aller beteiligten Disziplinen für die erfolgreiche Entwicklung mechatronischer Systeme. Wie gezeigt wurde, stellt dies gerade für die

¹⁸ Allgemein wird dieser Trend mit dem Begriff *Partitionierung* umschrieben. Der Begriff beschreibt die "Aufteilung eines Systems auf mehrere Einheiten/Module, z. B. Funktionserfüllung durch Komponenten verschiedener Hersteller oder Fachdisziplinen; Aufteilen eines Systems in einen Teil, der in Hardware und einen Teil, der in Software implementiert wird" [VDI 03].

integrativen Phasen eine besondere Herausforderung dar, da hier immer der physikalische Entwicklungsgegenstand im Mittelpunkt der Zusammenarbeit steht. Sämtliche extern realisierten Teilkomponenten müssen an einem Ort zusammengeführt und aufeinander abgestimmt werden. Folglich ist es notwendig, in diesem Prozess auch die korrespondierende Expertise zusammenzuführen und in den Entwicklungsprozess zu integrieren. Ebenfalls eine Grenze des Outsourcing-Modells nennt STEUSLOFF in [VDI 05]: „Eine Auslagerung von Services ist nur soweit möglich, wie das eigene Wissen über die Produktion bzw. Produkte mit Dienstleistern kommuniziert werden kann oder darf“.

2.3 Herausforderungen des mechatronischen Entwicklungsprozesses

Nachdem in den vorangegangenen Abschnitten die Charakteristika des mechatronischen Entwicklungsprozesses herausgearbeitet wurden, ist es Ziel dieses Kapitels zu überprüfen, inwiefern in den integrativen Phasen die Anforderungen an eine wirtschaftliche Produktentwicklung erfüllt werden. Einleitend werden zunächst die allgemeinen Anforderungen, wie sie heute am Markt vorhanden sind, zusammengefasst. Weiterhin wird analysiert, welche Anforderungen sich daraus für die Entwicklungsprozesse mechatronischer Produkte ableiten. Anschließend erfolgt eine Überprüfung, inwieweit die aktuelle Situation den vorherrschenden Anforderungen entspricht. Vorhandene Defizite werden abschließend zusammengefasst.

2.3.1 Allgemeine Situation am Markt

Die Entwicklung moderner technischer Produkte sieht sich heutzutage mit Herausforderungen konfrontiert, die vornehmlich unter den Überbegriffen „steigende Komplexität“ und „steigende Dynamik des Wandels“ der Unternehmenswelt zusammengefasst werden können [MILBERG 96]. Die wichtigsten Punkte für die an dieser Stelle durchgeführte Betrachtung sind im Folgenden dargestellt:

- **Verschärfte Dynamik in der Entwicklung und Produktion**

Dies ist zurückzuführen auf immer kürzere Produktzyklen in Folge immer kürzer werdender Innovationszyklen¹⁹ [DOHMEN 02], [EHRENSPIEL 95].

- **Zeit- und Termindruck**

Nach LEDER stehen Entwicklungs- und Konstruktionsabteilungen unter großem Realisierungsdruck. Verzögerungen im Entwicklungsprozess können schnell dazu führen, dass ein Konkurrenzprodukt vorzeitig Marktanteile gewinnt und dadurch Verluste entstehen [EHRENSPIEL 95], [LEDER 99].

¹⁹ Als Innovationszyklus wird die Zeitspanne betrachtet, in der zirka 50 % des Wissens über eine Technologie veraltet ist und durch neues Wissen ersetzt wird [EHLERS 00].

- **Forderung nach Sicherheit, Qualität und Zuverlässigkeit dominierend**

Der Wettbewerbsdruck am Markt und Vorschriften wie beispielsweise Produkthaftungsgesetze oder DIN ISO 9000 veranlassen diese Forderungen. Als Konsequenz steigt der Aufwand für Qualitätssicherungsnachweise und den zugehörigen Dokumentationen.

- **Qualität und Zuverlässigkeit ohne Mehrkosten**

Nach BAUER herrscht speziell in der Investitionsgüterindustrie, die am Standort Deutschland um so mehr unter einem hohen Kostendruck steht, die Forderung, Produkte mit hoher Qualität und Zuverlässigkeit zum gleichen Preis anzubieten [BAUER 98]. Die Einhaltung teilweise sehr komplexer Sicherheits- und Qualitätsstandards eines Gerätes wird vom Kunden als selbstverständlich angesehen. Beispielsweise rechtfertigt heute die Konformität eines Feldgerätes zu einer SIL-Klasse²⁰ trotz des erhöhten Entwicklungsaufwandes keinen höheren Verkaufspreis mehr.

Es sei bemerkt, dass es sich bei den aufgezählten Punkten um Anforderungen handelt, die unabhängig von der betrachteten Domäne allgemeine Anforderung an die heutige Produktentwicklung darstellen. Spezialfälle werden in dieser Arbeit nicht behandelt. Die sich aus diesen Anforderungen ergebenden Konsequenzen für die Produktentwicklung werden aus Sicht der Mechatronik nachfolgend beschrieben.

2.3.2 Problemstellungen der mechatronischen Produktentwicklung

Die Reaktionen auf die Anforderungen an moderne Produkte lassen sich neben den bekannten Anforderungen an wirtschaftliche Fertigung und Produktion unter der recht allgemeinen Forderung nach Verbesserung der Entwicklungsprozesse zusammenfassen. Diese Forderung wird in der einschlägigen Literatur immer wieder hervorgehoben. LEDER sieht das Potential zur Erhöhung der Wettbewerbsfähigkeit in der Optimierung von Entwicklungszeit und Entwicklungskosten [LEDER 99]. In Übereinstimmung dazu steht die Forderung von GAUSEMEIER nach einer Reduzierung der Produktentwicklungszeiten [GAUSEMEIER+ 04]. Insbesondere leiten sich demnach für die Entwicklung mechatronischer Produkte die nachfolgend aufgezählten Forderungen ab.

- Wie auch in anderen komplexen Wissenschaften wird in der Mechatronik, deren Hauptschwierigkeit die Integration verschiedener Teildisziplinen mit eigenen Vorgehensweisen, eigenen Werkzeugen und einer jeweils speziellen Begriffswelt ist, die Forderung nach einem methodischen Vorgehen erhoben [VDI 00].

²⁰ *Safety Integrity Level* nach IEC 61508 [SMITH+ 01]

- Um die Entwicklungszeit zu reduzieren, ist es notwendig, Teilprozesse während der Entwicklung zu parallelisieren. Damit dies effizient geschieht ist wiederum ein methodisches Vorgehen unabdingbar.
- Verstärkter Einsatz von Rechnerwerkzeugen um die hohe Komplexität mechatronischer Produkte zu beherrschen [DOHMEN 02], [ISERMANN+ 00].
- Umfassende Qualitätssicherung
- Damit die Anforderung nach neuen Innovationen erfüllt werden kann, ist es gerade für viele KMU notwendig, sich auf das Kerngeschäft zu konzentrieren [BULLINGER+ 96], [EVERSHEIM+ 96] (siehe Abbildung 2-7).

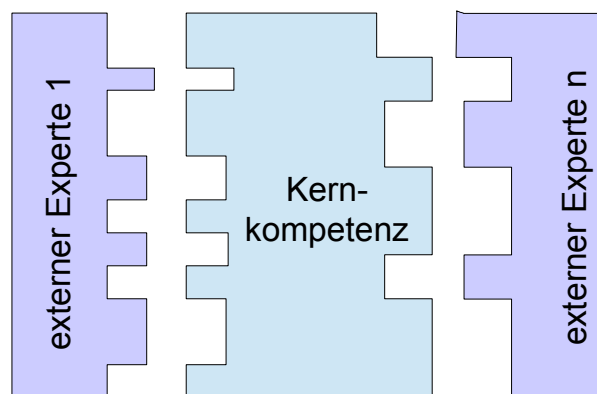


Abbildung 2-7: Ergänzung der Kernkompetenz durch externe Ressourcen

Die Erfüllung dieser Forderungen stellt auch in den integrativen Phasen der mechatronischen Produktentwicklung eine schwierige Aufgabe dar. Hier herrschen die unter Kapitel 2.2 dargelegten Charakteristiken vor. Um den Anforderungen gerecht zu werden, müssen spezielle Problemstellungen gelöst werden. Die im Folgenden genannten Problemstellungen leiten sich aus den allgemeinen Anforderungen an die Produktentwicklung und den in Kapitel 2.2 beschriebenen Merkmalen ab:

- Enge Integration aller Beteiligten:
Dies erfordert eine enge Kooperation zwischen unterschiedlichen Experten aus verschiedenen Fachdisziplinen die am Entwicklungsprojekt beteiligt sind.
- Integration von Experten und Integration von Werkzeugen:
Um Entwicklungsprozesse parallel ablaufen zu lassen ist nicht nur die Kooperation zwischen den beteiligten Spezialisten notwendig, sondern auch die Integration der verwendeten Werkzeuge muss vollzogen werden.
- Effiziente, oftmals spontane und temporäre Integration von Expertise am physikalischen Entwicklungsgegenstand.
- Umfassende Qualitätssicherungsmaßnahmen während des Integrationsprozesses, insbesondere auch für die in mechatronischen Systemen eingebettete Software [BENDER 04B], [BENDER+ 01], [VA 00].

- Schnelle Reaktion auf unvorhergesehene Probleme um Verzögerungen zu vermeiden.

Im Folgenden wird geprüft, inwiefern diese Problemstellungen in der gegenwärtigen Situation bewältigt werden.

2.3.3 Gegenwärtige Situation in der Mechatronik

Als Ergebnis der bisher durchgeführten Betrachtungen lässt sich zusammenfassend die heutige Situation bei der Entwicklung und Herstellung mechatronischer Produkte festhalten. Diese ist durch eine Reihe bestimmender Faktoren gekennzeichnet:

- **Softwaredominanz in mechatronischen Systemen**

In komplexen mechatronischen Systemen nimmt eingebettete Software die funktionsbestimmende Rolle ein. Die hierdurch gewonnene Flexibilität und Funktionalität verleitet jedoch auch zu einer Varianten- und Versionsvielfalt. In jedem Fall müssen sämtliche Produktvarianten für sich den Qualitätsanforderungen genügen, was die Aufwendungen für die Qualitätssicherung weiter erhöht. Auch wenn mittlerweile ein ausgeprägtes Bewusstsein vorherrscht, so ist die Umsetzung der Maßnahmen bisweilen schwierig, komplex²¹ und kostenintensiv.

- **Stückzahlen**

Die Spannbreite der Stückzahlen von mechatronischen Produkten reicht von Spezialanfertigungen bzw. Kleinserien (z. B. Investitionsgüterindustrie) bis hin zu Großserien (z. B. Automobilindustrie). Insbesondere bei geringen Stückzahlen wirken sich unoptimierte Entwicklungsprozesse negativ auf die Wirtschaftlichkeit aus.

- **Werkzeugkomplexität**

Nicht nur die Komplexität mechatronischer Produkte, sondern auch die Komplexität entwicklungsunterstützender Werkzeuge bewirkt eine Verteilung der Tool-Kompetenz auf verschiedene Know-how-Träger.

- **Systemintegration**

Die Integrationsdichte mechatronischer Produkte stellt einerseits einen bedeutenden Vorteil dar, bringt aber auch neue Herausforderungen mit sich. Einflüsse von Kommunikationssystemen (z. B. Feldbusse) und Energieübertragungssystemen (z. B. Kabel, Steckverbinder) müssen berücksichtigt werden. Bei der räumlichen Integration treten Wechselwirkungen zwischen den Komponenten auf (z. B. Elektromagnetische Störungen, Temperatur, Schwingungen) [VDI 03]. Teilkomponenten müssen deshalb innerhalb des Gesamtsystems betrachtet werden, um sämtliche Verflechtungen zu

²¹ Beispiel siehe [SMITH+ 01]

berücksichtigen. Viele Teilaufgaben sind somit spätestens in der Integrationsphase nicht mehr aus dem Entwicklungsprozess herauslösbar sondern werden vor Ort vollzogen. Aufgrund der weiter steigenden Produktkomplexität festigt sich dieser Trend.

▪ **Verteilung der Expertise**

Die meisten KMU können es sich aus wirtschaftlichen Gründen nicht leisten, die gesamte für die Produktentwicklung benötigte Expertise ständig vor zu halten. Es muss deshalb auf externe Experten zurückgegriffen werden. In großen Firmen sind unterschiedliche Spezialabteilungen oftmals über mehrere Standorte verteilt. Lösungselemente werden von verschiedenen Zulieferern bezogen [EVERSHEIM+ 96], [GORONCY 04]. Offene Systeme und Feldbuslösungen fördern diesen Trend [BAUER 98]. Die Konsequenz ist die Streuung von Spezialwissen auf verschiedene Personenkreise (siehe auch Abbildung 2-8).

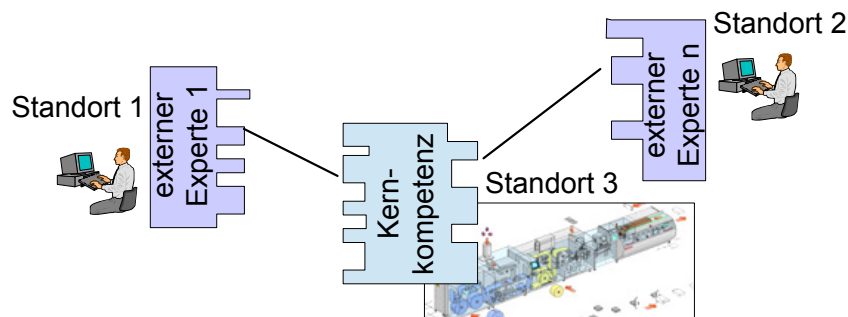


Abbildung 2-8: Geografische Verteilung externer Ressourcen

Die Gesamtsituation lässt sich wie folgt zusammenfassen: Die mechatronische Geräteentwicklung ist von nicht herauslösbaren Teilaufgaben geprägt, die von einem breit gefächerten und geografisch verteilten Personenkreis bearbeitet werden. Insbesondere in den integrativen Phasen steht der Bedarf nach externen Ressourcen im Widerspruch zur Notwendigkeit der Expertenintegration vor Ort. Dies führt zu dem Schluss, dass die in Kapitel 2.3.2 aufgezählten Anforderungen an den Produktentwicklungsprozess vor allem in den integrativen Phasen nicht oder nur unter hohem Aufwand erreicht werden können. Die Defizite aus heutiger Sicht sind im nachfolgenden Kapitel detailliert zusammengestellt.

2.3.4 Defizite im mechatronischen Entwicklungsprozess

Das Dilemma in den integrativen Phasen des mechatronischen Entwicklungsprozesses ist einerseits der oftmals nur temporäre und spontane Bedarf nach externer Expertise²² und andererseits die Ortsgebundenheit sowie die enge Verflechtung der mechatronischen Teilkomponenten. Die besonders für die

²² Fehlendes interdisziplinäres Wissen bei Experten bestätigt auch EICHELMANN: "Gerade Spezialisten fällt es nicht leicht den Gesamtüberblick zu halten, also auch gleichzeitig Generalist zu sein" [EICHELMANN 95].

Mechatronik geforderte integrierte Produktentwicklung [GAUSEMEIER+ 00] wird in diesen Phasen folglich nur mangelhaft umgesetzt. In vielen Fällen führt diese Tatsache dazu, dass notwendige Maßnahmen in den integrativen Phasen nur ineffizient bzw. gar nicht durchgeführt werden, da es an benötigter Unterstützung und Spezial-Know-how mangelt. Insbesondere bestehen konkret die folgenden Defizite:

- Externe Spezialisten müssen zum Ort der Produktentwicklung anreisen, dies verursacht hohe Kosten und Zeitverzug. Lange Zyklen ergeben sich beispielsweise durch Terminnot und den Koordinationsbedarf mit externen Stellen. Jede Dienstleisterfirma ist zumeist an weiteren Projekten beteiligt, was das Koordinationsproblem weiter verschärft [EHLERS 00].
- Der Entwicklungsgegenstand bzw. Teilkomponenten davon werden zwischen den Entwicklern und externen Dienstleistern verschickt, dies führt ebenfalls zu Zeitverlusten [KUTTIG+ 03], [RÖMER 03].
- Viele Teilaufgaben stellen Arbeitspakete mit geringem Umfang dar, fallen dafür jedoch spontan an. Wirtschaftliches Outsourcing kann hier nicht durchgeführt werden.
- Verteilte Firmenstandorte in großen Unternehmen führen dazu, dass Fachabteilungen redundant unterhalten werden müssen, um benötigte Expertise im Integrationsprozess vor Ort zu haben.
- Frühe entwicklungsbegleitende Tests an realen Produktkomponenten werden, aufgrund der räumlichen Trennung der Fachabteilungen und mangelndem Test-Know-how vor Ort, nicht oder nur unzureichend durchgeführt [RÖMER 03], [VA 99].
- Nutzbringende rechnergestützte Werkzeuge werden, aus Gründen der Komplexität und mangelnder Tool-Kompetenz vor Ort nur sehr ineffizient bzw. gar nicht eingesetzt [KÖNIG 95]. Das mögliche Optimierungspotential durch Werkzeugunterstützung wird nicht ausgeschöpft.

Die Konsequenz dieser Defizite ist die Verzögerungen des Entwicklungsprojekts und die Verlängerung von Durchlaufzeiten in der Produktentwicklung. Diese Verzögerung verursacht bisweilen hohe Verluste. LEDER gibt hierzu ein Beispiel für Produkte mit einem Lebenszyklus von 5 Jahren wonach die Verzögerung der Entwicklungszeit um sechs Monate Gewinneinbußen von bis zu 30% zur Folge haben kann [LEDER 99]. Die beschriebenen Defizite werden nachfolgend anhand eines konkreten Szenarios der mechatronischen Produktentwicklung veranschaulicht:

Durchflussregler für den Einsatz in verfahrenstechnische Anlagen

Bei dem zu entwickelnden Gerät handelt es sich um eine Kombination aus Durchflussmesser und Regelventil, welches in Rohrleitungen eingebaut wird. Eine von außen vorgegebene Durchflussmenge für ein Fluid wird durch das Gerät mit Hilfe eines Regelalgorithmus eingestellt (vgl. Abbildung 2-9).

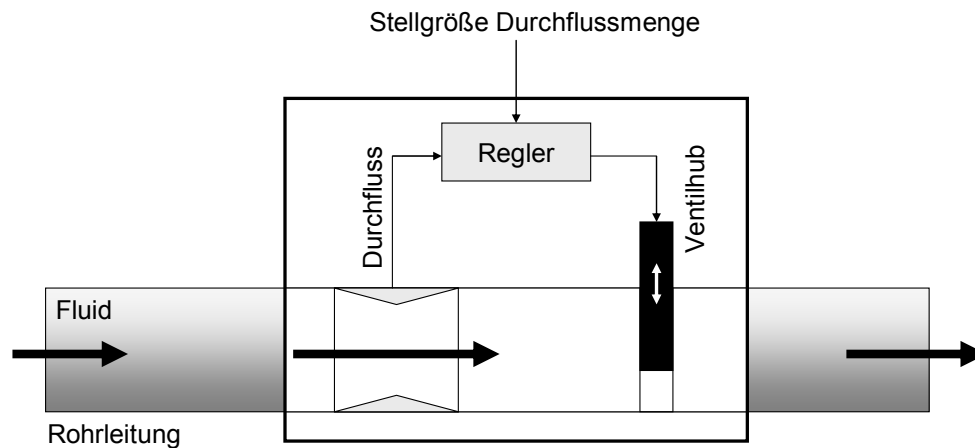


Abbildung 2-9: Durchflussregler

Für das zu entwickelnde Gerät gelten Randbedingungen: Anschaltung für die Feldbussysteme *PROFIBUS*, *FOUNDATION fieldbus* und *INTERBUS* sowie Konformität zu SIL-Klasse 3 nach IEC 61508. Eine Zusammenstellung wichtiger Entwicklungsschritte veranschaulicht die Problematik:

- Beim Prototypenbau erfolgt zunächst der Test der Messeinrichtung. Durchflussmesser beruhen auf zum Teil komplexen physikalischen Wirkprinzipien, was eine spezielle Laborausstattung vor Ort und Fachwissen erfordert.
- Die gesamte Steuerungselektronik wird anschließend in einer HIL-Umgebung eingebaut, um den Regelalgorithmus zu überprüfen und zu optimieren. Hierfür muss ein Prüfstandsexperte anwesend sein.
- In einem weiteren Schritt wird das komplette Gerät in eine realen Versuchsvorrichtung integriert, um die korrekte Funktionsweise des Gesamtsystems mit verschiedenen Fluiden zu überprüfen und zu optimieren. Für solche Tests existiert beim Hersteller ein spezielles Labor. Um das Verhalten online zu beobachten und gegebenenfalls umgehende Änderungen, beispielsweise an der Firmware durchzuführen, werden die verantwortlichen Entwickler vor Ort hinzugezogen.
- Um IEC 61508-Konformität zu erreichen, muss der Gerätehersteller den entsprechenden Nachweis erbringen. Da es sich um eine komplexe Norm handelt, wird ein externer Spezialist auf diesem Gebiet für einige Tage im Haus beschäftigt.
- Nachdem die Feldbusanschlüsse realisiert sind, muss auf verschiedenen Ebenen des ISO/OSI-Modells der Konformitätsnachweis erbracht werden. Hierzu zählen Physical-Layer-Test, Conformance-Test, Interoperabilitätstest, eventuell ein Profiltest sowie der Test der heute üblichen elektronischen Gerätebeschreibung. Diese Tests, welche nicht selten in Zertifikaten durch unabhängige Institutionen enden, werden von einer geringen Zahl externer Prüflabore durchgeführt. Hierfür muss das Gerät versandt werden. Ergeben

sich Fehler während des Tests, geht das Gerät zurück an den Hersteller und der Testzyklus muss nach erfolgter Nachbesserung von neuem beginnen. Nicht selten sorgen Terminprobleme für Verzögerungen.

Das Beispiel veranschaulicht viele Stellen im Produktentwicklungsprozess, an denen die zuvor beschriebenen Defizite auftreten: Die Anforderungen an spontane, ortsunabhängige Integration von Expertise unter Einbeziehung des realen Gerätes ist oftmals mit hohem Aufwand verbunden. Zusammenfassend lässt sich die identifizierte Problemstellung für die mechatronische Produktentwicklung wie folgt darstellen:

Die in den integrativen Phasen des mechatronischen Entwicklungsprozesses vor Ort, oftmals spontan und nur temporär benötigte externe Kompetenz ist schwierig in das Entwicklungsvorhaben zu integrieren. Der daraus resultierende Zeitverzug im Entwicklungsprozess führt zu Ineffizienz und folglich Gewinneinbußen.

Zur Bewältigung dieser Problemstellung wird die Forderung nach einer verbesserten Zusammenarbeit räumlich verteilter Experten in den integrativen Phasen erhoben. Hieraus leitet sich die Aufgabenstellung dieser Arbeit ab.

2.4 Zusammenfassung und Aufgabenstellung

Mechatronische Produkte nehmen heutzutage eine Schlüsselposition in einem weiten Branchenspektrum ein. Dabei handelt es sich um komplexe Systeme, in denen die drei Disziplinen Maschinenbau, Elektrotechnik und Informatik eng miteinander verschmolzen sind, und deren Funktionalität in hohem Maße durch Software bestimmt ist. Folglich ist der Entwicklungsprozess mechatronischer Produkte selbst durch Komplexität und Multidisziplinarität gekennzeichnet. Erfolgreiche Produktentwicklung basiert in der Mechatronik auf geeigneten Vorgehensmodellen und auf der Unterstützung durch oftmals komplexe, jedoch unverzichtbare Rechnerwerkzeuge. Als Vorgehensmodell hat sich das so genannte V-Modell in vielen Bereichen mechatronischer Produkte etabliert. Anhand dieses Modells lassen sich im mechatronischen Entwicklungsprozess neben den spezifizierenden und den umsetzenden Phasen die so genannten integrativen Phasen, in denen die Realisierung und Integration realer Prototypen bzw. Produktkomponenten stattfindet, identifizieren.

Aufgrund der Multidisziplinarität sind die durchzuführenden Aufgaben auf mehrere Spezialisten aus unterschiedlichsten Domänen verteilt. Bei mechatronischen Entwicklungen werden folglich mehrere Experten benötigt, um innerhalb einer gegebenen Zeit und mit vertretbarem Aufwand ein Problem zu lösen. Da Expertise speziell in den integrativen Phasen oftmals nur temporär, dafür aber spontan benötigt wird, ist es unwirtschaftlich, Spezialisten permanent vor Ort in

das Entwicklungsprojekt zu integrieren. Deshalb und dadurch dass insbesondere viele KMU gezwungen sind, sich auf das Kerngeschäft zu konzentrieren, entsteht der Bedarf nach Einsatz externer Know-how-Träger. Diese sind jedoch im Normalfall nicht am Entwicklungsstandort konzentriert sondern geografisch verteilt.

Verschärfte Dynamik in der Produktentwicklung, dominierende Qualitätsanforderungen und der daraus resultierende Zeit- und Termindruck lassen die mechatronische Entwicklung zu einer besonderen Herausforderung werden. Die essentielle Aufgabe ist es hier, Entwicklungsverbünde zwischen verschiedensten, verteilt lokalisierten Spezialisten zu schließen. Immer dann, wenn das physikalische System im Mittelpunkt der Aktivitäten steht, also in den integrativen Phasen, treten an dieser Stelle Schwierigkeiten auf. Der Entwicklungsverbund muss am Ort des Systems geschlossen werden, was entweder Reisen von Experten oder den Versand des Systems selbst erfordert. Das Problem ist der hieraus entstehende Zeitverzug des Entwicklungsprozesses sowie die anfallenden hohen Kosten. Beides wirkt sich gewinnmindernd aus.

Ziel der vorliegenden Arbeit ist es, die verteilte Zusammenarbeit von Experten unter Einbeziehung des physikalischen mechatronischen Systems in den integrativen Phasen des Entwicklungsprozesses zu verbessern. Dabei wird insbesondere Wert auf den Aspekt des Mehrwerts und der Effizienzsteigerung gelegt. Dieses Ziel steht vor dem Hintergrund der im Verlauf dieses Kapitels identifizierten Anforderungen:

- Expertise muss ortsunabhängig zur Verfügung stehen.
- Expertise muss temporär und spontan verfügbar sein.
- Der entfernte Zugriff auf physikalische Systeme muss auf sichere Art und Weise möglich sein.

Daraus ergibt sich die folgende Aufgabenstellung: Es ist eine Schnittstelle zu schaffen, durch welche die geografische Lücke zwischen dem mechatronischen System und entfernten Spezialisten geschlossen wird. Die Technologien für die Kommunikation zwischen Rechnerwerkzeugen, Übermittlung eines realitätsnahen Prozessbildes und Kommunikation zwischen den beteiligten Personen sind so zu verschmelzen, dass eine universell einsetzbare Schnittstelle zur Verbindung des entfernten Experten und dem Ort der Produktentwicklung entsteht. Da der entfernte Zugriff auf reale physikalische Systeme immer ein hohes Gefahrenpotential birgt, ist die Integration geeigneter Sicherheitstechnologien von essentieller Bedeutung.

Das nachfolgende Kapitel untersucht, welche der heute existenten Technologien und Konzepte der verteilten Zusammenarbeit zur Lösung der vorgestellten Problematik Beitrag leisten können.

3 Methoden und Werkzeuge für die verteilte Kooperation

Nachdem in Kapitel 2 die Problemstellung in den integrativen Phasen der heutigen Mechatronikentwicklung identifiziert wurde, diskutiert dieses Kapitel zunächst einen Ansatz zur Lösung der Problematik. Die Anwendungsfelder eines solchen Ansatzes werden vorgestellt und daraus die Anforderungen an eine neue Lösung abgeleitet. Im Weiteren wird geprüft, inwieweit existierende Konzepte und Technologien diesen Anforderungen gerecht werden.

3.1 Verteilte Zusammenarbeit bei der Entwicklung mechatronischer Geräte

3.1.1 Konzentration von Expertise mittels Remote-Technologien

Ein vielversprechender Ausweg aus der existierenden Problematik ist, die geografische Lücke zwischen dem Entwicklungsort und dem Standort verschiedenartigster Expertise kommunikationstechnisch zu schließen. Hierfür müssen der geografisch entfernte Experte, der lokale Entwickler und der physikalische Entwicklungsgegenstand so gekoppelt werden, dass sämtliche Aufgaben wahrgenommen werden können, ohne einen Ortswechsel vorzunehmen. Den am Entwicklungsprojekt beteiligten Experten müssen folglich sämtliche benötigten Informationen (Systemgrößen und Prozessgrößen) über das physikalische mechatronische System ortsunabhängig verfügbar gemacht werden. Umgekehrt muss es dem geografisch entfernten Experten ermöglicht werden, mit sämtlichen weiteren Beteiligten einschließlich des physikalischen Systems zu interagieren. Ein solches Szenario erfordert den Einsatz umfangreicher und leistungsstarker Informations- und Telekommunikationstechnologien²³ bzw. Remote-Technologien²⁴. Vor dem Hintergrund des in Kapitel 2.1.3 eingeführten Begriffs

²³ Allgemein als *IuK-Technologien* (Informations- und Kommunikationstechnologien) bezeichnet.

²⁴ Der Begriff *Remote-Technologien* bzw. *TeleTechnologien* umfasst allgemein alle Mittel um digitale Informationen elektronisch zwischen geografisch entfernten Orten zu übertragen [DFAM 04]. Einen großen Stellenwert besitzen die so genannten Internet-Technologien. Unter diesen Begriff fallen gemäß den Definitionen in einschlägiger Fachliteratur [BUSCH 02], [POHLMANN 00], [WEISS 02] sämtliche Technologien, die im heutigen Internet weite Verbreitung erlangt haben und einen Standard bzw. Quasi-Standard darstellen.

E-Collaboration wird der Einsatz solcher Methoden und Technologien auch als Computer Supported Cooperative Work (CSCW²⁵) bezeichnet.

Aus Sicht der Produktentwicklung liegen die Vorteile des Einsatzes von Remote-Technologien auf der Hand:

- Die Notwendigkeit des "Vor Ort Seins" von Experten wird reduziert bzw. entfällt ganz.
- Das spontane und befristete Hinzuziehen externer Expertise wird vereinfacht.
- Kosten und Zeitverzug werden reduziert.

Obgleich dieser Vorteile erfolgt die Verbreitung von CSCW in den integrativen Phasen des mechatronischen Entwicklungsprozesses bislang schleppend. Der Grund dafür liegt in der erweiterten Komplexität und der Notwendigkeit eines speziellen Setups für die eingesetzten Remote-Techniken. Dies resultiert aus der besonderen Problemstellung der direkten Interaktion von Experten mit dem realen physikalischen System von geografisch entfernten Standorten aus.

3.1.2 Anwendungsfelder

Die Anwendungsfelder von CSCW in den integrativen Phasen der Mechatronikentwicklung werden im Folgenden weiter konkretisiert. Dies erfolgt anhand der in diesen Phasen notwendigen Tätigkeiten:

- **Qualitätssicherung:** Zertifizierung, entwicklungsbegleitende Tests, Dokumentation, Abnahmetests.
- **Hardware-In-The-Loop-Techniken:** Prüfstandsteuerung, Modifikationen am System und direkte Auswerten der Reaktionen.
- **Optimierung des Produkts:** Veränderung des Geräte-Setups, Zeitnahe Beobachtung des Verhaltens.
- **Fehlersuche:** Debugging, Modifikationen der eingebetteten Software, Anschluss und Konfiguration von Messsystemen, zeitnahe und direkte Auswertung von Messungen.
- **Komponentenintegration:** Beobachtung des Verhaltens, Modifikationen der eingebetteten Software, Modifikationen der Hardware.

Aus diesen Anwendungsfeldern leiten sich vier Klassen von Tätigkeiten ab, die von Entwicklern durchgeführt werden:

²⁵ Unter dem Begriff *CSCW* werden Methoden und Technologien zusammengefasst, die durch den Einsatz von Hard- und Software die Zusammenarbeit von örtlich verteilten Gruppen, Personen und Teams unterstützen [ZWICKER 98]. Auch der, jedoch sehr allgemeine und unpräzise Begriff Telearbeit findet in diesem Zusammenhang Verbreitung.

1. Direkter Eingriff in das mechatronische System, bestehend aus physikalischem Entwicklungsgegenstand und Entwicklungsumgebung, über Softwarebasierte Funktionalität.
2. Direkte Beobachtung von Ereignissen und sämtlichen Prozessgrößen des Systems.
3. Richtiges Setup und Modifikation des Entwicklungsgegenstands und der Entwicklungsumgebung durch Eingriffe in die Hardware.
4. Kommunikation der an der Entwicklung beteiligten Experten untereinander.

Diese vier Klassen lassen sich zu zwei verschiedenen Arten von Kommunikationskanälen zusammenfassen, die für die verteilte Zusammenarbeit in den integrativen Phasen benötigt werden:

- Der Kanal für die Interpersonelle Kommunikation
- Der Kanal für die Kommunikation zwischen den Werkzeugen (Maschine-Maschine-Kommunikation oder Inter-Tool-Kommunikation)

Zur Veranschaulichung werden in Anhang A einige Beispiele für die aufgezählten Tätigkeiten gegeben.

3.1.3 Anforderungen an verteilte Koaktionen im Umfeld der Mechatronik

Anhand der bisherigen Ausführungen können drei Klassen von Anforderungen an den Einsatz von Remote-Technologien in den integrativen Phasen der mechatronischen Geräteentwicklung identifiziert werden. Hierzu zählen erstens: grundlegende Anforderungen, die ganz allgemein in der Produktentwicklung aufgeworfen werden und zweitens: spezielle Anforderungen, die aus den Anwendungsfeldern in den integrativen Phasen entstehen. Drittens müssen auch allgemeingültige Anforderungen an CSCW-Technologien berücksichtigt werden, die sich hinsichtlich eines effizienzsteigernden Einsatzes stellen.

Klasse 1: Anforderungen der mechatronischen Produktentwicklung

Anforderung A 1.1: Gemeinsame Nutzung von Rechnerwerkzeugen

Einen Schwerpunkt der verteilten Zusammenarbeit in der mechatronischen Entwicklung bildet die gemeinsame Nutzung von Rechnerwerkzeugen. Verschiedenste Werkzeuge sind mit geringem Aufwand zu verbinden.

Anforderung A 1.2: Gleichzeitige Daten- und Multimediaverbindung

Zeitgleich mit der Arbeit an rechnerbasierten Werkzeugen besteht ein hoher Kommunikationsbedarf zwischen den an der Entwicklung beteiligten Experten. Die Kommunikation zwischen den Werkzeugen und zwischen den Experten muss so vereint sein, dass die Kooperationslösung die Tool-Komplexität nicht weiter erhöht.

Anforderung A 1.3: Rollenorientierte Benutzerschnittstellen

Aufgrund des multidisziplinären Charakters der mechatronischen Entwicklung repräsentieren die verschiedenen beteiligten Experten unterschiedliche Rollen, die den jeweiligen Themengebieten und Zuständigkeiten entsprechen. Die Benutzerschnittstellen kooperativer Werkzeuge müssen an diese Rollen angepasst sein (Komplementäres Kompetenzfeld).

Klasse 2: Anforderungen in den integrativen Phasen**Anforderung A 2.1: Rückverfolgbarkeit der Interaktionshistorie**

Sämtliche Interaktionen eines entfernten Benutzers müssen zu jeder Zeit nachvollziehbar und rückverfolgbar sein.

Anforderung A 2.2: Gestufter Zugriff auf Werkzeugfunktionalität

Rechnerwerkzeuge der integrativen Phasen sind in ihrem Wirkungsbereich nicht auf die Rechnerumgebung beschränkt und Interaktionen mit den Tools betreffen auch die physikalische Umgebung. Kritische Funktionen bedürfen deshalb eines gestuften Zugriffskonzepts. Die Stufung betrifft ganz allgemein die Ausführung von unterschiedlich kritischen Funktionen über Remote-Mechanismen und ist unabhängig von Benutzerrollen.

Anforderung A 2.3: IT-Sicherheit

Hohe IT-Sicherheit, da Interaktionen mit physikalischen Systemen immer auch Auswirkungen auf deren Umgebung haben. Zugriffe Unberechtigter bedeuten im Extremfall Gefahren für Mensch und Umwelt.

Anforderung A 2.4: Einfache Peer-To-Peer-Kopplung

Vordergründig ist die direkte Interaktion eines entfernten Experten mit dem mechatronischen System zu gewährleisten. Der Schwerpunkt liegt deshalb auf der direkten Peer-To-Peer-Werkzeugkopplung.²⁶

²⁶ Die Interaktion eines entfernten Experten mit dem System hat nicht den Charakter einer verteilten Konferenz.

Klasse 3: Allgemeine Anforderungen an den Einsatz von CSCW-Tools

Anforderung A 3.1: Angemessene Reaktionszeiten

Die Integration realer Geräte muss unter zeitlichen Aspekten gemäß Kapitel 2.1.4 erfolgen. Insbesondere stellt dieser Aspekt auch Anforderungen an die Kodierung der Daten und der Umfang der zu übertragenden Datenmenge. Schnelle Reaktionszeiten müssen auch bei schmalbandigen Übertragungsmedien²⁷ gewährleistet sein [KRAUSE 00].

Anforderung A 3.2: Anpassung an den Anwendungsfall

Das System zur E-Collaboration muss dem jeweiligen Anwendungsfall angepasst und in den individuellen Arbeitsbereich des Anwenders integriert werden. Dem Anwender muss jederzeit bewusst sein welche Ressourcen öffentlich und welche privat zugänglich sind [MUNKVOLD 03], [SPRINGER 01].

Anforderung A 3.3: Einfache Nutzung der Kooperationsfunktionalität

Kooperationsfunktionen müssen einfach nutzbar sein und insbesondere mit Werkzeugen verknüpft werden können, die für individuelle Arbeitsaufgaben verwendet werden. Dabei ist die Stabilität der Kooperationswerkzeuge von essentieller Bedeutung, um Akzeptanz beim Anwender zu erreichen [SPRINGER 01].

Die identifizierten Anforderungen gelten im weiteren Verlauf dieses Kapitels als Ausgangsbasis für die Bewertung von existierenden Szenarien, Methoden und Technologien aus dem CSCW-Bereich.

²⁷ Zu den schmalbandigen Übertragungsmedien zählen Medien mit einer Bandbreite kleiner oder gleich der ISDN-Geschwindigkeit (64 kbit/s bzw. 128 kbit/s). Auch wenn heutzutage in der Regel Breitbandanschlüsse verfügbar sind, erfolgt vielerorts immer noch die Nutzung von Telefonverbindungen, insbesondere in Ländern mit einer weniger stark entwickelten Kommunikationsinfrastruktur.

3.1.4 Taxonomie der Kooperationsmittel

Um eine fundierte Analyse der existierenden CSCW-Landschaft durchführen zu können, erfolgt zunächst eine Strukturierung der computergestützten Mittel zur Realisierung von E-Collaboration. Als Basis dient Abbildung 3-1 in Anlehnung an [MUNKVOLD 03] und [SPRINGER 01].

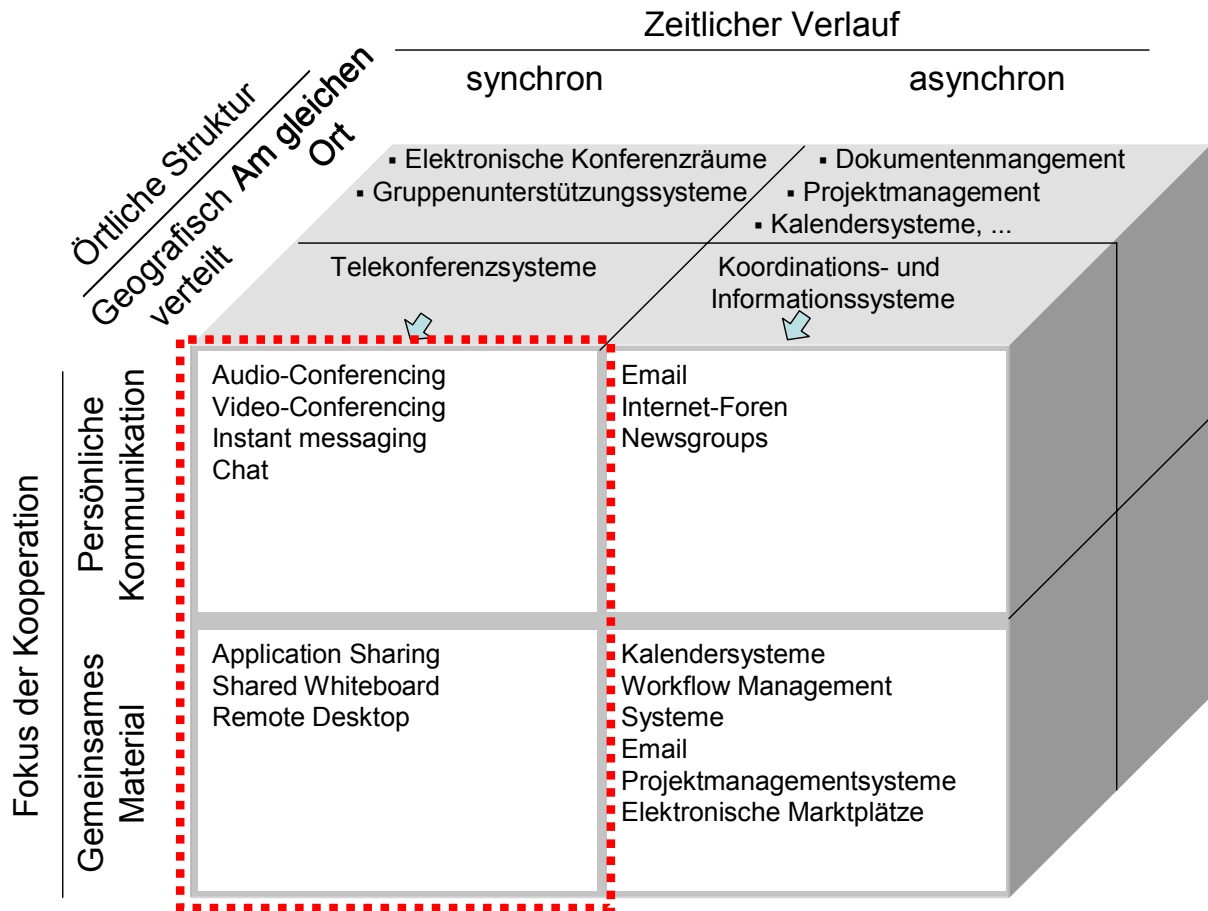


Abbildung 3-1: Strukturierung von Kommunikationsressourcen für die E-Collaboration

Die in der Abbildung dargestellten Mittel zur Unterstützung der Kooperation zwischen Personen am gleichen Ort spielen für die hier behandelte Thematik keine Rolle und werden deshalb nicht näher betrachtet. Die Mittel zur Kooperation zwischen geografisch verteilten Personen gliedern sich zunächst in synchrone und asynchrone Systeme (vgl. Kapitel 2.1.4). Aus den bislang erarbeiteten Merkmalen der integrativen Phasen der Mechatronikentwicklung ergibt sich, dass der Schwerpunkt dort auf synchroner E-Collaboration liegt (umrahmter Bereich). Die Mittel lassen sich weiterhin in Mittel für die persönliche Kommunikation zwischen den Beteiligten und Mittel zur gemeinsamen Nutzung von Daten und Applikationen unterteilen.

3.2 Bereiche mit etablierter verteilter Kooperation

In den nachfolgenden Unterkapiteln erfolgt die Untersuchung von Anwendungsfeldern von Remote-Technologien, die entweder vergleichbare Merkmale zum betrachteten Handlungsfeld aufweisen oder übertragbare Teilaspekte erwarten lassen. Dies ermöglicht eine bessere Einordnung der identifizierten Problemstellung. Das Feld der E-Collaboration stellt sich als sehr breit gefächert dar. Für diese Arbeit irrelevante Bereiche wie beispielsweise E-Commerce-Lösungen, elektronische Marktplätze oder Agententechnologien²⁸ werden im Folgenden nicht weiter behandelt.

3.2.1 Fernzugriff auf automatisierungstechnische Maschinen und Anlagen

Ein in den letzten Jahren stark angewachsenes Einsatzfeld für Remote-Technologien ist die Automatisierungstechnik. Auch dort erfolgt ein Zugriff auf reale Maschinen und Anlagen von entfernten Standpunkten aus. Das Szenario umfasst das Engineering von automatisierungstechnischen Maschinen und Anlagen, Fernleiten und Ferninstandhalten sowie das Asset-Management²⁹. Zum Engineering zählen Tätigkeiten wie Inbetriebnahme, Parametrierung und Konfiguration, siehe hierzu auch [BENDER+ 02B], [PRITSCHOW+ 02]. Der Begriff Fernleiten umfasst Aktionen wie Fernbedienen, Fernbeobachten, d. h. Prozessvisualisierung, Vorgabe von Stellgrößen, Sollwerten und Alarmgrenzen. Ferninstandhaltung umfasst die Oberbegriffe Ferndiagnose (Erfassung des Ist-Zustandes der Anlage), Fernwartung (Erhaltung des Sollzustandes) und Fernreparatur (Widerherstellung des Sollzustandes, z. B. Geräte-Reset, Software-Update) [LITZ+ 00].

Tabelle 3-1 gibt einen Überblick über die Nutzung von Remote-Funktionalität in der Automatisierungstechnik.

²⁸ Siehe hierzu Teilprojekt C4 des SFB 467 Wandlungsfähige Unternehmensstrukturen der Universität Stuttgart [SFB467 04]

²⁹ Unter *Asset-Management* werden Tätigkeiten und Maßnahmen verstanden, die dazu dienen, den Wert einer Anlage zu erhalten oder zu steigern. Dazu gehören die Betriebsführung, Prozessführung, Verfahrensoptimierung sowie die Wert erhaltende und möglichst Wert steigernde Instandhaltung, d.h. die Optimierung zwischen dem Aufwand für die Instandhaltung und Ertüchtigung der Anlage einerseits und der für die Produktion geforderten Verfügbarkeit der Anlage andererseits. [NAMUR 01]

Funktion	Zweck	Inhalte
Vermittlung allgemeiner Informationen	Gelegentliche Information von Kunden bzw. Interessenten	Aufbereitete Daten für Nicht-Fachleute mit Übersichtscharakter
Eingeschränkter Zugang zu Prozess- und Anlagendaten	<ul style="list-style-type: none"> ▪ Überwachung und Sicherung einer Anlage z. B. im operatorlosen Betrieb ▪ Gelegentlicher Zugriff 	<ul style="list-style-type: none"> ▪ Ausgewählte, wenige Online-Prozessdaten, einfachste Stellmöglichkeiten ▪ Ereignisgesteuerte Alarmierung
Umfassender Zugang zu Prozess- und Anlagendaten	<ul style="list-style-type: none"> ▪ Service ▪ Unterstützung bei der Inbetriebnahme oder bei Fehlern durch Spezialisten ▪ Vollständiger Remote-Betrieb der Anlage 	<ul style="list-style-type: none"> ▪ Vollständiger Zugang zu allen Online-Daten ▪ Ausgebaute evtl. grafische Bedienoberfläche

Tabelle 3-1: Remote-Funktionalität in der Automatisierungstechnik [BRAUNE+ 99]

Neben den etablierten Anwendungsfeldern wird der Aspekt des Fernzugriffs auf Maschinen und Anlagen in diversen Forschungsprojekten behandelt. Beispiele sind PROTEUS (Integrationsplattform für verschiedene Maintenance-Tools) [BANGEMANN+ 03] oder DeKOS [BENDER 01]. Weitere Bestrebungen verfolgen das Ziel, eine Austauschmöglichkeit zwischen verschiedenen Engineering-Werkzeugen zu ermöglichen, indem eine durchgängige Datenintegration angestrebt wird [STROBEL 03]. Hierbei spielt die deklarative Metasprache XML [W3C 00] eine wichtige Rolle. Die wichtigsten Anforderungen an Fernzugriffslösungen in der Automatisierungstechnik sind in [BETTENHAUSEN+ 02] zusammengefasst. Hinzu kommt als herausragender Punkt die Anforderung an die IT-Sicherheit [DFAM 04], [UHLMANN+ 00].

Das in diesem Unterkapitel beschriebene Szenario umfasst hauptsächlich den direkten Online-Zugriff auf Daten von Maschinen und Anlagen im laufenden Betrieb, vorzugsweise mit standardisierten Mitteln. Die Bemühungen gehen folglich in Richtung eines optimierten Zugriffs auf den Datenhaushalt. Die Zusammenarbeit zwischen entfernten Experten an der Anlage steht nicht im Vordergrund. Methoden zur synchronen Kopplung von Rechnerwerkzeugen unter Einbeziehung der Expertenkommunikation sind bis dato nicht existent. Für den in dieser Arbeit behandelten Anwendungsfall sind die Bemühungen um IT-Sicherheit sowie die Bestrebungen nach einheitlichen und durchgängigen Datenformaten relevant.

3.2.2 Telechirurgie und Telerobotik

Der Bereich Telechirurgie in der Medizin scheint auf den ersten Blick parallele Problemstellungen wie die integrativen Phasen der Mechatronikentwicklung zu beinhalten. Nach SIEWERT gliedert sich der Einsatz von Remote-Technologien in der chirurgischen Medizin in die vier Szenarien Telekonsultation, Telekonferenz, Telepräsenz und Telechirurgie [SIEWERT+ 99]. Während Telekonsultation von einem eher asynchronen Austausch relevanter Patientendaten geprägt ist, erfolgt die Telekonferenz in synchroner Form wobei Informationsquellen wie beispielsweise hochauflösende Bilder online zur Verfügung stehen müssen. Unter Telepräsenz wird das aktive Eingreifen eines entfernten Kommunikationspartners in ein Szenario vor Ort verstanden. Hierzu zählen etwa die Positionierung von Kameras oder das Leiten des Operators vor Ort beispielsweise mittels eines virtuellen Zeigerinstruments. Telechirurgie bezeichnet die aktive Durchführung von Operationsschritten durch einen entfernten Operateur durch ferngesteuerte Roboter. Gerade letzterer Bereich ist aus technologischer Sicht auf absehbare Zeit noch nicht existent. Entsprechende Aktorik und Sensorik befindet sich bisher erst im Experimentalstadium. Ein Beispiel für aktuelle Forschungsaktivitäten hierzu findet sich in [SFB453 04].

Auch im Bereich der Telemedizin werden kritische Vorgänge von geografisch entfernten Personen gesteuert, die Anforderungen an die Zuverlässigkeit der Remote-Techniken sind als mindestens gleich hoch zu bewerten. Im Fehlerfall sind die Auswirkungen für den Patienten bisweilen dramatisch. Dennoch kann die Telemedizin nicht einfach auf die hier behandelte Problemstellung übertragen werden. Telechirurgie bzw. Telerobotik besitzt als Kernproblem die realitätsnahe Übertragung haptischer und taktiler Informationen, zwischen Operateur und Remote-System. Dies beinhaltet die Übermittlung von Aktor- und Sensordaten unter harten Echtzeitbedingungen über weite Strecken und nicht echtzeitfähige Kommunikationsnetze. Die anderen aufgezählten Szenarien besitzen überwiegend Konferenzcharakter ohne gleichzeitige Kopplung von verteilten Applikationen. Relevant ist lediglich der Einsatz geeigneter Audio/Video-Übertragungsverfahren.

3.2.3 E-Learning und Distance-Learning

Unter E-Learning wird der Einsatz von Informations- und Kommunikationstechnologien im Kontext des Lernens und Lehrens verstanden [WIKIPEDIA 04]. E-Learning-Projekte setzen Computer und Multimediatechnologien zur Durchführung von Lernangeboten ein. Durch Remote-Technologien wird es zudem möglich, die räumliche Abhängigkeit zwischen Lehrern und Lernenden aufzuheben³⁰. Tabelle 3-2 nimmt eine Einteilung der unterschiedlichen Ausprägungen von E-Learning vor.

³⁰ Auch als *Distance-Learning* bezeichnet.

	Gleicher Ort	Teilweise entfernt	Entfernt
Asynchron			Eigenständige Kurse (z.B. auf CD-ROM)
Teilweise Asynchron		Distance-Learning in Universitäten	Virtuelle Universitäten
Synchron	Traditionelle Klassenzimmer		Videoübertragung von Kursen

Tabelle 3-2: Einordnung von E-Learning-Systemen nach [Xu 00]

Eine wesentliche Rolle im Distance-Learning spielen Werkzeuge zur multimedialen Interkommunikation zwischen Lehrern und Lernenden. Aufgabe solcher Werkzeuge ist nicht nur die Verteilung von Lerninhalten. Vielmehr müssen Funktionen zur Regelung der Dynamik einer Lerngemeinschaft zur Verfügung stehen, beispielsweise die Handhabung von Rückfragen oder Diskussionsbeiträgen. Eine im Rahmen der behandelten Thematik zu beachtende Ausprägung von Distance-Learning sind so genannte Online-Experimente. Hier ist es den Lernenden möglich, mittels Remote-Technologien bestimmte Funktionen an realen Versuchsaufbauten auszuführen und die Auswirkungen multimedial zu beobachten, siehe hierzu beispielsweise [HOYER+ 03], [FALTIN+ 03]. Eine Kopplung hochfunktioneller Tools findet dabei jedoch nicht statt.

Beim Anwendungsszenario Distance-Learning wird, mit Ausnahme von Online-Experimenten, denen kein Gefährdungspotential zugeschrieben wird, nicht auf reale Geräte zugegriffen. Vielmehr steht die Informationsverbreitung mit geeigneten Rechnerwerkzeugen im Vordergrund. Die dort verwendeten Multimediatechnologien und Tools sind ein Orientierungspunkt für multimediale Kommunikationslösungen zwischen beteiligten Entwicklungsexperten.

3.2.4 Spezifizierende und konstruktive Phasen

In den spezifizierenden und konstruktiven Phasen nach Abbildung 2-4 haben CSCW-, bzw. Remote-Technologien bislang verstärkt Verbreitung erfahren. Zu diesen Phasen zählen die Kategorien: Workflow-Management, Projektmanagement [LUBICH 95], gemeinsame Datenräume (EDM³¹) [ZWICKER 98], Produktdatenaustausch und verteiltes Konstruieren [GAUSEMEIER+ 00], [KRAUSE+ 00]. In diesen Kategorien finden sowohl synchrone als auch asynchrone Kommunikationsformen Anwendung. Einen Überblick gibt Tabelle 3-3.

Für die betrachteten integrativen Phasen des mechatronischen Entwicklungsprozesses liegt der Schwerpunkt auf synchronen Kooperationsformen. Die Kategorien mit asynchronen Ausprägungen werden deshalb nicht näher betrachtet.

³¹ Engineering Data Management

	Synchron	Asynchron
Workflow-Management		X
Projektmanagement	X	X
Gemeinsame Datenräume		X
Produktdatenaustausch		X
Verteiltes Konstruieren	X	X
Virtuelle Marktplätze		X

Tabelle 3-3: Synchrone und Asynchrone Kommunikation in den konstruktiven Phasen

Insbesondere beim verteilten Konstruieren findet verstärkt synchrone Kommunikation zwischen Entwicklern in Form von verteilten Konferenzen unterstützt durch Multimedia statt [KIESEWETTER 97], [KRAUSE+ 00]. Aber auch die Kopplung von CAD-Werkzeugen wird vollzogen, um das gemeinsame Arbeiten von verteilten Konstrukteuren an einem CAD-Modell zu ermöglichen. Forschungsprojekte wie beispielsweise SFB Improve [MARQUART+ 03] versuchen die Integration von domänenspezifischen Werkzeugen der Verfahrenstechnik in den Planungsphasen voranzutreiben und setzen dabei ebenfalls auf multimedial unterstützte, verteilte Kooperation.

Der wesentliche Unterschied zu den integrativen Phasen liegt in der Bearbeitung digitaler Produktmodelle im Gegensatz zu realen Systemen. Die vorherrschenden existierenden Anforderungen werden von Lösungen für die konstruktiven Phasen nicht erfüllt. Das Thema Werkzeugkopplung bezieht sich in den konstruktiven Phasen schwerpunktmäßig auf CAD-Systeme. Ein hierfür existierendes Referenzmodell wird im weiteren Verlauf dieses Kapitels analysiert.

3.3 Modelle und Konzepte für die verteilte Kooperation

Das nachfolgende Kapitel prüft, inwieweit bereits Modelle für die verteilte Kooperation existieren, die für die betrachtete Problemstellung zur Lösungsfindung beitragen. Hierfür wird zunächst ein allgemeingültiges Modell für die Durchführung einer Ko-Aktion vorgestellt. Anschließend wird ein Referenzmodell für die Werkzeugkopplung in den Konstruktiven Phasen der Mechatronikentwicklung sowie eine universelle Architektur für die Gestaltung von CSCW-Systemen analysiert. Den Abschluss bildet ein Einblick in die Technologien mit denen solche Modelle umgesetzt werden können.

3.3.1 Allgemeines Phasenmodell der Ko-Aktion

Das Phasenmodell der Ko-Aktion nach STOLLER-SCHAI in Abbildung 3-2 stellt die Grundlage jeden gemeinschaftlichen Handelns dar. Ein vorherrschendes kooperatives Setting (vgl. Kapitel 2.1.3) bildet das Rahmenwerk für alle durchlaufenen Handlungsphasen. Diverse Phasen können rekursiv durchlaufen bzw. übersprungen werden. Allgemein entsteht ein Ergebnis aus dem Wechselspiel zwischen kollaborativen und kooperativen Phasen.

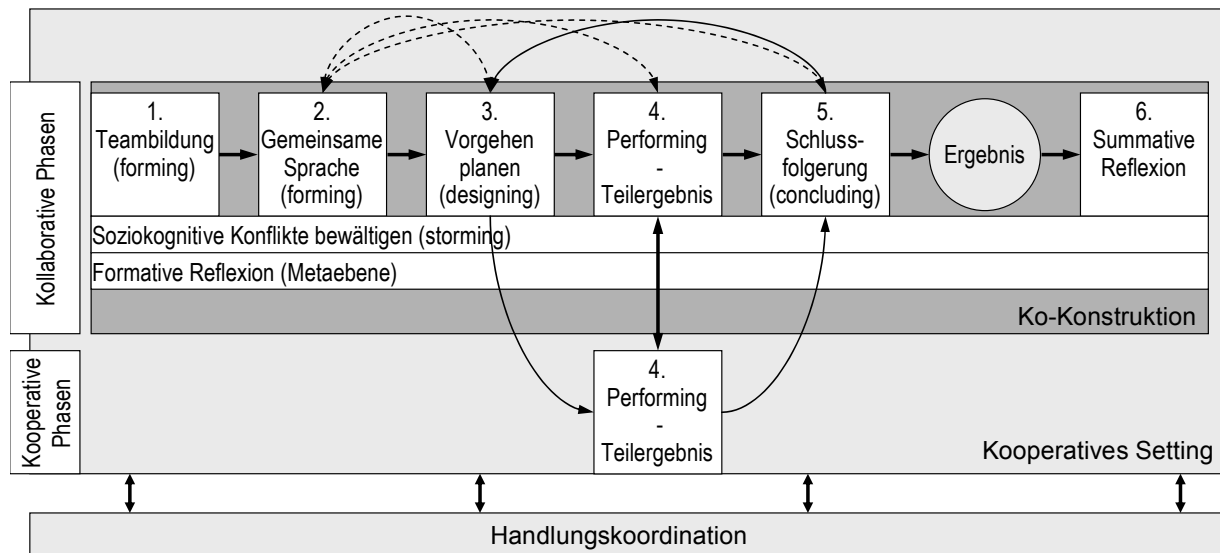


Abbildung 3-2: Phasenmodell einer idealtypischen Ko-Aktion [STOLLER-SCHAI 03]

Das Phasenmodell gibt eine allgemeingültige Vorgehensweise für die Durchführung von Ko-Aktionen zwischen beteiligten Personen an die Hand. Die konkrete Realisierung von verteilter Zusammenarbeit in einem bestimmten Handlungsfeld ist in dem Modell noch nicht berücksichtigt. Bei verteilten Ko-Aktionen finden einige Phasen im virtuellen Raum, d. h. unterstützt durch CSCW-Werkzeuge statt. Eine Unterstützung beim Einsatz solcher Werkzeuge für den konkreten Anwendungsfall wird nicht an die Hand gegeben. Generell existieren Untersuchungen und Empfehlungen zur Gestaltung von Prozessen zur Einführung von Telekooperationssystemen in bestehende Arbeitsumgebungen, beispielsweise in [FINGER 99], [MUNKVOLD 03]. Diese Empfehlungen und Vorgehensweisen sind jedoch in der Ebene der Unternehmensprozesse und -strukturen angeordnet. Die Detailspezifikation spezieller Handlungsfelder wird unzureichend berücksichtigt. Konkrete Lösungen für die Umsetzung von verteilter Kooperation zur Lösung der gegebenen Problemstellung existieren nicht.

3.3.2 KOMFORCE-Referenzmodell für die Mechatronikentwicklung

In Kapitel 2 wurde die besondere Wichtigkeit der Integration in der Mechatronikentwicklung herausgestellt. Integration kann dabei auf unterschiedlichen

semantischen Ebenen erfolgen. Das KOMFORCE³²-Referenzmodell unterscheidet verschiedene Ebenen der Integration von entwicklungsunterstützenden Rechnerwerkzeugen (CAE-Verfahren).

<p>Verfahrenstechnische Integration Konfiguration der Methoden. Bestimmung der einzusetzenden CAE-Werkzeuge</p>
<p>Prozesstechnische Integration Aufgaben- und datenflussorientiertes Prozess- und Projektmanagement</p>
<p>Modelltechnische Integration Nutzung eines gemeinsamen Produktmodells, Abbildung zwischen spezifischen Produktmodellen</p>
<p>Systemtechnische Integration Kopplung der IT-Werkzeuge auf Basis von CORBA³³, DCOM³⁴ und Internet-Technologien</p>

Tabelle 3-4: Die vier Ebenen des KOMFORCE-Referenzmodells [GAUSEMEIER+ 00]

Das Modell wurde entwickelt um trotz der Vielfalt an Entwicklungswerkzeugen integrierte Entwicklungsumgebungen zu schaffen. Die beiden untersten Ebenen stellen die technischen Randbedingungen dar unter denen verschiedene Werkzeuge zusammenarbeiten können. Die beiden oberen Ebenen legen die organisatorischen Rahmenbedingungen fest [GAUSEMEIER+ 00].

Das KOMFORCE-Referenzmodell zielt auf die Integration von CAE-Werkzeugen und somit auf die konstruktiven Phasen des Entwicklungsprozesses ab. Die Problematik der Kopplung von verteilten Werkzeugen in den integrativen Phasen wird nicht adressiert. Die definierten semantischen Ebenen stellen jedoch eine generelle Orientierungshilfe dar, um eine Koppelungsebene für IT-Werkzeuge zu definieren.

3.3.3 Modular Advanced Collaboration System

Das Modular Advanced Collaboration System (MACS) stellt ein modulares und flexibles Framework zur Realisierung von CSCW-Systemen dar [STURZEBECKER 00]. Die Architektur von MACS ist so ausgelegt, dass dem CSCW-System durch den einfachen Austausch von Modulen verschiedene Kommunikationstechnologien (z. B. ITU³⁵-Standards, Mbone³⁶) unterlagert

³² Kommunikations- und Forschungskreis für Integrationstechnologien in Computer Aided Design und Engineering

³³ Common Object Request Broker Architecture [MEIER 02]

³⁴ Distributed Component Object Model [MICROSOFT 95]

³⁵ International Telecommunication Union [ITU 04]

³⁶ *Multicast BackBONE*: Internet-basiertes Netz für Multicast-Kommunikation [Fahner+ 01]

werden können. Ebenso ist auf der oberen Anwendungsschicht die Einbindung verschiedener Applikationsmodule (z. B. Videokonferenzsystem, Shared-Whiteboard) in das System möglich. Hierzu existieren spezifizierte, einheitliche Schnittstellen. Die Steuerung und Koordination der einzelnen Module in MACS wird von einem zentralen Kontroll-Modul übernommen, welches zudem Basisdienste wie beispielsweise Session-Kontrolle und Benutzermanagement zur Verfügung stellt [STURZEBECHER 00]. Abbildung 3-3 veranschaulicht die grundlegende Architektur von MACS.

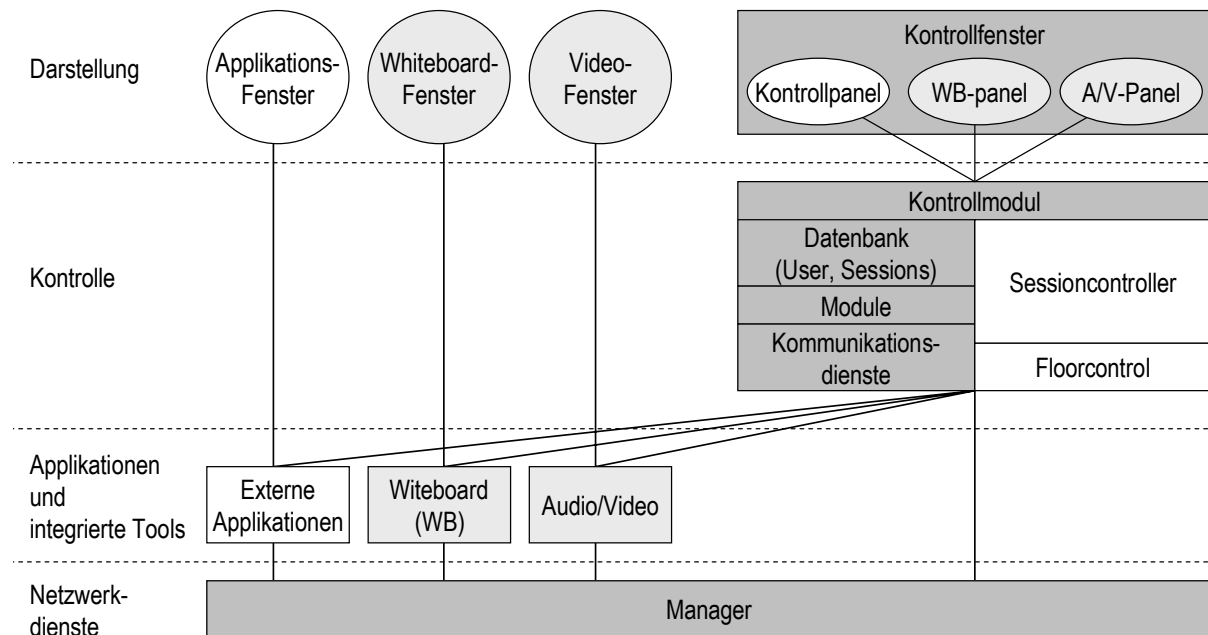


Abbildung 3-3: Die Grundlegende MACS-Architektur [STURZEBECHER 00]

MACS stellt ein umfangreiches System mit vielen Funktionen zur Initiierung und Durchführung von verteilten Konferenzen dar. Im Fokus steht die Abwicklung von m:n Multipoint-Kommunikation. Aufgaben wie Benutzerverwaltung, Session-Kontrolle, Anbindung an das Kommunikationssystem werden von MACS ausgeführt, so dass Eigenimplementierungen unnötig sind. Basis-Tools wie Audio/Video-Übertragung oder Shared-Whiteboard stehen ebenfalls zur Verfügung [STURZEBECHER 00]. STURZEBECHER gibt zwar ein Anwendungsbeispiel von MACS in Verbindung mit dem Software-Engineering-Tool Tel-SEE, Verbreitung hat das System jedoch überwiegend im Distance-Learning-Bereich erfahren. Anwendungsbeispiele finden sich in [ZITTERBART 02].

Für die in dieser Arbeit bearbeitete Problemstellung stellt sich MACS als unzureichende Lösung dar. Die spezifizierten Interfaces berücksichtigen nicht den benötigten, feingranularen Zugriff auf die Funktionalität der zu koppelnden Werkzeuge. Das Framework ist für Multipoint-Kommunikation ausgelegt, die dadurch entstehende Komplexität (Installation, Handling, Bedienung) ist für den vorliegenden Anwendungsfall mit Peer-to-Peer-Charakter unnötig, der

entstehende Overhead ein unnötiger Performance-Verlust. Eine Schnittstelle für die Werkzeugkopplung ist in MACS nicht vorhanden. Hinzu kommt, dass das Framework auf JAVA³⁷-Basis beruht, was erfahrungsgemäß höhere Hardwareressourcen bedingt um eine ausreichende Ausführungsgeschwindigkeit sicher zu stellen. In Anbetracht dieser Tatsachen lässt MACS für den vorliegenden Anwendungsfall keine Vorteile im Vergleich zu etablierten Standard-Technologien und Konzepten erkennen.

3.3.4 Multimedia Grundlagentechnologien

Im Bereich der E-Collaboration haben sich mittlerweile eine Reihe von Standard-Technologien etabliert, welche die Basis für CSCW-Anwendungen bilden. Hierzu zählen vorwiegend Methoden zur Kodierung und Übertragung von Audio- und Videodaten sowie zur Kontrolle und Verwaltung von verteilten Konferenzen. Die wohl wichtigsten Vertreter sind die von der ITU spezifizierten Standards der H.323-Serie (Paketvermittelte Multimediakommunikation) und der T.120-Serie (Datenprotokolle für Multimedia Konferenzen). Parallel dazu haben sich Standards der IETF³⁸, wie beispielsweise das SIP³⁹-Protokoll, etabliert. Diese Standards wurden ursprünglich für die Verwendung in leitungsvermittelten Netzen, z. B. ISDN entwickelt, mittlerweile erfolgte die Adaption auf IP-basierte Netze. Die ITU-Standards werden von der überwiegenden Anzahl der heute verfügbaren CSCW-Tools als Grundlage genutzt. Abbildung 3-4 ordnet die wichtigsten der derzeit etablierten Standards gemäß dem ISO-OSI-Referenzmodell ein:

CSCW-Applikationen			
Kommunikationsverwaltung	Audio	Video	Daten
SIP (IETF) H.323 (ITU)	G.711 G.722 G.729, etc.	H.261 H.263	T.120
	RTP		
TCP	UDP		
IP			
LAN, WAN			

SIP ... Session Initiation Protocol

RTP ... Real Time Protocol

Abbildung 3-4: Einordnung von Multimediastandards nach [ELKO 04], [WINKLER 98]

³⁷ Plattformunabhängige, objektorientierte Programmiersprache. Der Quellcode wird zunächst in einen Bytecode kompiliert, der zur Laufzeit interpretiert wird [SUN 04].

³⁸ Internet Engineering Taskforce

³⁹ Session Initiation Protocol [IETF 05]

Ebenfalls erwähnenswert ist das MBONE-Netz: Es handelt sich dabei um eine Internet-gestützte, multicasting-fähige Kommunikationsinfrastruktur. Neben diesen Kommunikationsstandards existieren eine Reihe weiterer Standards zur Kodierung und Komprimierung der zu übermittelnden Audio- und Videodaten. Der wohl bekannteste ist das MPEG⁴⁰-Verfahren oder der bei VoIP⁴¹ eingesetzte G729A Standard der ITU.

Zur Lösung elementarer, multimedialer Kommunikationsaufgaben steht eine Vielzahl etablierter und standardisierter Technologien zur Verfügung, so dass diese Grundbausteine nicht neu entwickelt werden müssen. Die Entwicklung von Technologien zur multimedialen Kommunikation steht in dieser Arbeit deshalb nicht im Fokus der Betrachtungen. Vielmehr wird im Rahmen der Lösungsfindung das Bestreben verfolgt, geeignete Basistechnologien auszuwählen und für die Gesamtlösung so anzupassen, dass die Rahmenbedingungen hinsichtlich maximaler Reaktionszeiten erfüllt sind.

3.4 Technologien und Werkzeuge für verteilte Kooperation

Bevor im Rahmen dieses Unterkapitels eine Betrachtung der derzeit verbreiteten Werkzeuge zur Unterstützung von verteilter Kooperation erfolgt, werden diese anhand Tabelle 3-5 in Hauptkategorien eingeordnet:

Hauptkategorien	Ausprägungen von Werkzeugen
Interpersonelle Kommunikationstechnologien	<ul style="list-style-type: none"> ▪ Email ▪ Instant Messaging ▪ Audio/Video-Conferencing
Verteilte Informations-Ressourcen/-Räume	<ul style="list-style-type: none"> ▪ Document Management Systems ▪ Web-basierte Team/Projekt-Räume ▪ Data Conferencing/Sharing, Application-Sharing, Desktop-Sharing ▪ Elektronisches schwarzes Brett
Meeting-Support-Technologien	<ul style="list-style-type: none"> ▪ Electronic Meeting Systems
Koordinationsstechnologien	<ul style="list-style-type: none"> ▪ Workflow Management Systems ▪ Kalender- und Planungssysteme ▪ Web-Portale
Integrierte Produkte: Whiteboard, Application Sharing, Instant Messaging, File Transfer, Audio/Video	<ul style="list-style-type: none"> ▪ Integrierte Team-Support Lösungen, E-Learning-Technologien ▪ Integrierte Conferencing Lösungen

Tabelle 3-5: Werkzeuge zur Unterstützung von verteilter Kooperation [MUNKVOLD 03]

⁴⁰ Motion Pictures Expert Group [FRAUNHOFER 05]

⁴¹ Voice over IP: Übermittlung von Sprache in IP-basierten Netzwerken

Vor dem Hintergrund der gegebenen Problemstellung werden folgende Klassen detaillierter untersucht:

- **Integrierte Produkte:** Diese stellen die Basis für heutige verteilte Konferenzen dar
- **Verteilte Informations-Ressourcen:** Hier ist insbesondere das Desktop-Sharing von Bedeutung

Als eigene relevante Klasse werden applikationsspezifische “Remote“-Erweiterungen von Werkzeugen für Fernwirkung und Fernvisualisierung in der Automatisierungstechnik untersucht.

3.4.1 Integrierte Tools und Conferencing Tools

Moderne Tools zur Unterstützung von verteilten Konferenzen besitzen durchwegs einen großen Funktionsumfang, so dass diese Werkzeuge fast durchgängig der Kategorie der integrierten Tools zuzuordnen sind. Zu den Funktionen zählen nach aktuellem Stand der Technik: Audio/Videoübertragung, Shared-Whiteboard (gemeinsame virtuelle Zeichentafel), Instant Messaging (Übermittlung von Textnachrichten), Dateitransfer und oftmals auch Application Sharing (gemeinsamer Zugriff auf eine Anwendung). Ein sehr verbreiteter Vertreter solcher CSCW-Werkzeuge ist *NetMeeting* der Firma Microsoft, da dieses Produkt ohne Zusatzkosten zur Verfügung steht [MUNKVOLD 03]. Weitere Beispiele sind MBone-basierte Tools: das Java-basierte *Habanero* und kommerzielle Produkte wie *ASAP*⁴². Einige der Tools besitzen von Grund auf eher Peer-to-Peer-Charakter, während andere umfangreiche Funktionalität zur Benutzerverwaltung und Teilnehmerlokalisierung enthalten. Viele Werkzeuge lassen sich auch in Verbindung mit einem Konferenz-Server einsetzen, der die Teilnehmerverwaltung zentral regelt (vgl. Abbildung 3-5).

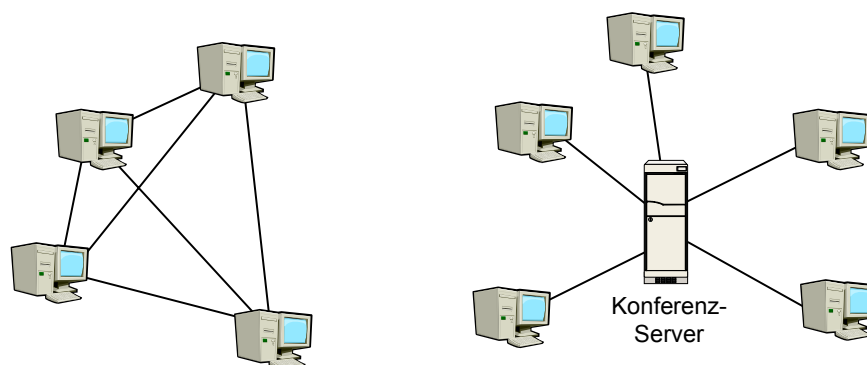


Abbildung 3-5: Einsatz eines Konferenz-Servers

⁴² Eine gute Übersicht von kommerziellen Conferencing-Tools findet sich beispielsweise unter [COLLABORATION 04] oder [THINKOFIT 04].

Da es sich bei den Features Audio/Video-Übertragung, Shared-Whiteboard, Messaging und Dateitransfer um etablierte Standardfunktionalität handelt, ist deren Verwendung für die hier behandelte Problemstellung in Betracht zu ziehen. Der Kanal zur Kopplung hochfunktioneller Werkzeuge der Mechatronikentwicklung kann durch integrierte CSCW-Werkzeuge nicht realisiert werden. Das hierfür auf den ersten Blick in Frage kommende Application-Sharing ist aus verschiedensten Gründen als kritisch zu bewerten:

- Mit Hilfe von Application-Sharing-Funktionalität ist kein gestufter Zugriff auf die Funktionalität einer Applikation realisierbar. Der Aktionsradius eines Remote-Teilnehmers bleibt dem lokalen Anwender verborgen⁴³. Es reicht nicht aus, die verteilt genutzten Applikationen auszuwählen, sondern es müssen die Funktionen innerhalb der Applikationen granular selektierbar sein.
- Ausreichende Logging- und Tracking-Funktionalität ist nicht vorhanden.
- Nicht angepasste Frontends machen die Zusammenarbeit in einem kompletären Kompetenzfeld ineffizient.
- Die Stabilität von existierenden Arbeitsplatzsystemen wird oftmals als problematisch angesehen, was zu Akzeptanzhürden führt [SPRINGER 01].
- Application-Sharing verursacht hohes Datenaufkommen und ist deshalb in schmalbandigen Netzen als problematisch anzusehen [SPRINGER 01].

Application-Sharing findet oftmals im CAD-Bereich für die gleichzeitige, verteilte Bearbeitung von CAD-Modellen Anwendung. Ein Beispiel hierfür ist das Application-Sharing Programm *jointX*, siehe auch [KRAUSE+ 00].

3.4.2 Desktop-Sharing

Eine gesteigerte Form des Application-Sharing stellen so genannte Desktop-Sharing-Werkzeuge dar. Dabei wird nicht nur ein einzelnes Applikationsfenster für die Bearbeitung durch einen entfernten Teilnehmer freigegeben, sondern es wird die vollständige Benutzeroberfläche des Arbeitsplatzrechners übertragen. Somit ist es dem entfernten Teilnehmer möglich, sämtliche Funktionalität des Host-Rechners zu nutzen. Dies bedeutet, der Teilnehmer besitzt die gleiche Kontrolle über den Rechner, als ob er lokal interagieren würde.

In der Windows-Welt finden die von Microsoft mitgelieferten Tools *Terminal-Services* bzw. *Remote-Desktop* breite Anwendung [MICROSOFT 04]. In der Unix-Welt hat sich *X-Windows* etabliert [X.ORG 04]. Daneben existieren Applikationen wie beispielsweise *PC-Anywhere* (Abbildung 3-6) oder *VNC* [VNC 04].

⁴³ Siehe hierzu auch [STURZEBECKER 00]

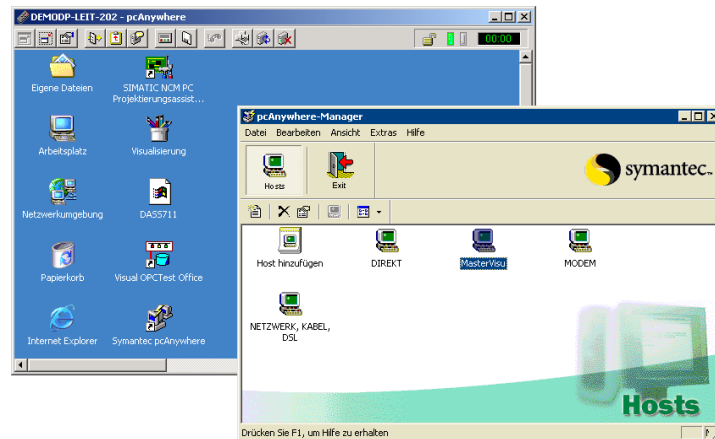


Abbildung 3-6: Beispiel für ein Desktop-Sharing-Werkzeug

Desktop-Sharing findet vor allem im administrativen Bereich Anwendung. Ein weiteres Szenario ist die Übertragung der grafischen Oberfläche eines Host-Rechners auf ein weiteres Ausgabegerät lediglich zur Beobachtung, beispielsweise zu Schulungszwecken. Desktop-Sharing-Werkzeuge sind ähnlich wie Application-Sharing-Tools zu bewerten. Der Aktionsradius eines entfernten Teilnehmers auf einem Host-PC ist jedoch noch weitaus größer. Der Remote-Nutzer verfügt prinzipiell über uneingeschränkte Kontrolle, seine Aktionen sind schwierig zu überwachen bzw. nachzuvollziehen. Der Schutz kritischer Bereiche auf dem Host-Rechner ist, wenn überhaupt, nur aufwändig zu realisieren, was zu niedriger Akzeptanz solcher Lösungen hinsichtlich der gegebenen Problemstellung führt. Dies kommt insbesondere dann zum tragen, wenn externe firmenfremde Beteiligte auf Rechner innerhalb des Unternehmens zugreifen oder die Beteiligten über unterschiedliche Rechteniveaus verfügen.

3.4.3 Fernwirk-Tools

Auch Werkzeuge zur Realisierung von Fernwirktechniken können im weiteren Sinne zu den kooperativen Werkzeugen gezählt werden. Sie ermöglichen es, Prozessdaten von Maschinen und Anlagen an geografisch entfernte Orte zu übertragen und dort darzustellen (Fernvisualisierung). In manchen Anwendungsszenarien werden Prozessdaten von entfernten Teilnehmern auch verändert und somit der Prozessablauf beeinflusst. Nach heutigem Stand der Technik kommen für solche Einsatzszenarien verstärkt Internet-Technologien wie beispielsweise das http-Protokoll oder XML zum Einsatz.

Damit entfernte Teilnehmer auf Prozessdaten zugreifen können, muss zunächst eine Verbindung der Anlage mit der Außenwelt erfolgen. Dies geschieht in vielen Fällen über ein Leit- oder SCADA⁴⁴-System, auf dem sich eine lokale

⁴⁴ Supervisory Control and Data Acquisition. SCADA-Systeme dienen zum Überwachen und Steuern von technischen Prozessen.

Visualisierungs-Software befindet, die beispielsweise über OPC⁴⁵ oder proprietäre Techniken mit dem Prozess gekoppelt ist. Viele moderne Visualisierungssysteme bieten weiterführend entsprechende Schnittstellen oder Softwarekomponenten für den Fernzugriff an. Ein Beispiel ist das System *SuiteVoyager* der Firma Wonderware als Erweiterung der Visualisierungssoftware *InTouch* [WONDERWARE 04].

Kommerzielle Fernwirksysteme sind darauf spezialisiert, den direkten Zugriff auf Anlagendaten über geografische Entfernungen anzubieten. Entsprechende Mechanismen zur Koppelung verteilter Anwendungen unter den Kapitel 3.1.3 in genannten Anforderungen stehen nicht im Fokus dieses Anwendungsszenarios. Fernwirksysteme spielen für die hier behandelte Problemstellung eine untergeordnete Rolle. Jedoch sind die in diesem Bereich eingesetzten Technologien zur Daten- und Funktionskodierung wie beispielsweise XML oder Webservices⁴⁶ auch als Basistechnologien für eine der Problemstellung entsprechenden Lösung in Betracht zu ziehen.

3.4.4 Einsatz von Remote-Tools unter dem Aspekt der IT-Sicherheit

Werden im Rahmen von verteilter Kooperation Daten über öffentliche Netze ausgetauscht, so stellen sich stets Anforderungen an die Sicherheit der Datenübertragung [NAEDELE 03]. Dies sind mindestens die Forderungen nach Integrität, Vertraulichkeit und Authentizität.

- Integrität beschreibt die Unversehrtheit der Daten, also die Garantie, dass die übermittelten Daten nicht durch unautorisierte Dritte verändert wurden sondern unverfälscht den Empfänger erreichen.
- Vertraulichkeit beinhaltet die Unlesbarkeit der übermittelten Daten für unberechtigte Dritte. Sollte es Unberechtigten gelingen, die Kommunikation mitzuhören, so darf daraus kein Informationsgewinn über die Inhalte der Nachricht möglich sein.
- Authentizität beschreibt die Echtheit der Daten. Dies bedeutet, dass eine Nachricht tatsächlich vom Absender stammt und nicht von einem Dritten, der sich unter Täuschungsabsichten als der Absender ausgibt.

Werden Werkzeuge zur verteilten Kooperation eingesetzt, so steht bei der überwiegenden Anzahl existierender Applikationen die Funktionalität im Vordergrund, während die Aspekte der Datensicherheit nicht behandelt werden. Einige kommerzielle Programme bieten zwar grundsätzlich die Möglichkeit der verschlüsselten Datenübertragung an, die meisten lassen den Aspekt der

⁴⁵ *Open Process Control*: Offener Standard zum Austausch von Prozessdaten zwischen automatisierungstechnischen, PC-basierten Applikationen. Spezifiziert durch die OPC-Foundation [OPCFoundation 04].

⁴⁶ Mit einer URI eindeutig identifizierte Softwareanwendung, deren Schnittstellen als XML-Artefakte definiert, beschrieben und gefunden werden können.

Datensicherheit jedoch außer Acht. Unabhängig davon bleibt die Integration von Applikationen zur verteilten Kooperation in das Gesamtsicherheitskonzept eines Unternehmens für den Anwender eine wichtige Aufgabe [DFAM 04].

3.5 Zusammenfassung und Bewertung

Als Lösungsmöglichkeit für die vorherrschende Problematik wurde Eingang dieses Kapitels der Einsatz von CSCW-Technologien identifiziert. Die Anwendungsfelder für den Einsatz solcher Technologien im Rahmen der integrativen Phasen der mechatronischen Produktentwicklung wurden anschließend konkretisiert und daraus spezielle Anforderungen an eine Lösung abgeleitet. Vor dem Hintergrund dieser Anforderungen wurden zunächst benachbarte Gebiete untersucht, in denen CSCW-Technologien bereits erfolgreich eingesetzt werden. Anschließend erfolgte eine Untersuchung von existierenden Konzepten für die verteilte Zusammenarbeit, die mit dem hier behandelten Thema in Verbindung stehen. Zuletzt wurden Werkzeuge zur verteilten Kooperation analysiert, die ebenfalls für die vorherrschende Problemstellung in Betracht zu ziehen sind.

Zunächst erfolgt die Beurteilung der Relevanz benachbarter Gebiete. Es wird hierfür eine qualitative Bewertung bevorzugt, da pauschale Aussagen über eine Eignung nicht sinnvoll erscheinen, sondern vielmehr versucht wird, geeignete Ansätze nutzbringend zu transferieren.

- Fernzugriff in der Automatisierungstechnik fokussiert den Zugriff auf Online-Prozessdaten bzw. Engineering-Daten. Kollaboration zwischen Experten steht nicht im Vordergrund, folglich existieren hierzu keine Lösungen. Verwendete Technologien zur Datenstrukturierung und Übertragung wie beispielsweise XML sind für Teillösungen von Relevanz.
- Telechirurgie und Telerobotik haben die harte Echtzeitübertragung von Steuerungsdaten über weite Strecken sowie die realitätsnahe Übertragung taktiler Informationen zum Ziel. Beides bezieht sich nicht auf die hier behandelte Problemstellung. Verwendete Technologien zur Audio/Video-Übertragung sind für Teillösungen von Bedeutung.
- Bei Distance-Learning steht die multimediale Übertragung von Lerninhalten an eine verteilte Gruppe Lernender im Vordergrund. Physikalische Systeme sind, wenn überhaupt, nur als unkritische Labormodelle involviert. Auch hier sind die verwendeten Audio/Video-Technologien von Relevanz.
- In den konstruktiven Phasen der Mechatronikentwicklung beschränkt sich die Kollaboration auf digitale Modelle. Die Zusammenarbeit besitzt in vielen Fällen Konferenzcharakter. Auch die dort verwendeten Multimedialösungen für die interpersonelle Kommunikation müssen beachtet werden.

Die zusammenfassende Beurteilung der untersuchten Modelle und Konzepte zur verteilten Zusammenarbeit stellt sich wie folgt dar:

- Phasenmodell der Ko-Aktion: Es handelt sich um ein Rahmenmodell, welches eine grundlegende Vorgehensweise für die Organisation und den Ablauf kooperativer Handlungen an die Hand gibt. Konkrete Lösungen für die betrachtete Problemstellung werden nicht gegeben.
- KOMFORCE-Referenzmodell: Grundlegendes Modell zur Kopplung mechatronischer Entwicklungswerkzeuge in den konstruktiven Phasen. Lösungen zur Realisierung von verteilter Kooperation werden nicht gegeben.
- MACS: Implementiertes, modulares Konzept für den Aufbau von CSCW-Werkzeugen. Der Anwendungsschwerpunkt liegt im Konferenz- und E-Learning Bereich. Die dortigen modularen Ansätze können zur Orientierung dienen.

Die Ergebnisse der Untersuchung von Technologien und Werkzeugen sind in Tabelle 3-6 zusammengefasst und gegenüber den Anforderungen gemäß Kapitel 3.1.3 angetragen:

	Integrierte Werkzeuge	Desktop-Sharing	Fernwirk-Tools	CSCW-Basistechnologien
A 1.1	+	+	--	++
A 1.2	+	+	-	+
A 1.3	--	--	+	--
A 2.1	--	--	+	--
A 2.2	--	--	+	-
A 2.3	-	-	-	-
A 2.4	+	+	-	+
A 3.1	-	--	+	+
A 3.2	-	-	--	+
A 3.3	+	+	--	--
Legende: ++ erfüllt, + teilweise erfüllt, - größtenteils nicht erfüllt, -- nicht erfüllt				

Tabelle 3-6: Bewertung existierender CSCW-Lösungen

Aus Tabelle 3-6 ist ersichtlich, dass derzeit keine Lösung existiert, die der Problemstellung unter den vorherrschenden Anforderungen umfassend gerecht wird. Verfügbare integrierte Werkzeuge und Desktop-Sharing Tools scheinen zwar bei grober Betrachtung die benötigte Basisfunktionalität zu besitzen,

weisen aber hinsichtlich der problemspezifischen Anforderungen entscheidende Defizite auf (z. B. fehlende Rückverfolgbarkeit, fehlende Rollenorientierung, fehlende granulare Zugriffsabstufung). Die in der Literatur (z.B. [MUNKVOLD 03], [SPRINGER 01]) geforderte Anpassbarkeit der CSCW-Werkzeuge an die Benutzer sowie das Einsatzgebiet ist nicht gegeben, da der Schwerpunkt auf dem Einsatz in verteilten Konferenzen liegt. Erfahrungsberichte [FINGER 99], [MUNKVOLD 03], [REICHWALD 97], [SPRINGER 01], wonach der Einsatz solcher Systeme in Unternehmen oftmals nur sehr schleppend erfolgt, bestätigen die Tatsache dass viele aktuelle Integrierte Werkzeuge auch allgemeine Anforderungen an CSCW-Systeme (A 3.1 – A 3.3) nicht oder nur unzureichend erfüllen. Verfügbare Audio/Video-Lösungen können jedoch als etablierte Standards betrachtet und genutzt werden.

Fernwirk-Tools entsprechen in Ansätzen den hier existenten Anforderungen für den Fernzugriff auf reale Systeme, sind jedoch zu unflexibel für den vorliegenden Anwendungsfall und bisweilen kompliziert zu handhaben. Gemeinsame Nutzung, Multimediakommunikation und Werkzeugkopplung wird hinsichtlich der vorliegenden Problemstellung unzureichend unterstützt. Die verwendeten Technologien zur Daten- und Funktionskodierung können in Ansätzen einer Gesamtlösung dienen.

Als gesonderter Punkt sind die multimedialen Basistechnologien zu betrachten. Keine der Technologien stellt für sich eine umfassende Lösung der vorliegenden Problemstellung dar. Es handelt sich jedoch in vielen Fällen um etablierte Standards, die als Bausteine für eine angepasste Gesamtlösung dienen können.

Zusammenfassend ergibt die Analyse, dass keine der vorhandenen Methoden und Technologien die Anforderungen an verteilte Kooperation in den integrativen Phasen der Mechatronikentwicklung vollständig erfüllt. Dem im betrachteten Handlungsfeld vorherrschenden Sicherheitsbedürfnis wird von existierenden Systemen entweder gar nicht, oder nur ansatzweise Rechnung getragen. Es fehlt eine dem Handlungsfeld angepasste Sicherheitsstrategie bzw. Integrationskonzepte für bestehende IT-Infrastrukturen.

Das nachfolgende Kapitel stellt das Konzept einer Gesamtlösung für die vorherrschende Problemstellung vor. Zunächst wird das Szenario für die Zusammenarbeit weiter präzisiert und daraus, unter Einbeziehung der in Kapitel 3.1.3 dargestellten Anforderungen, die Spezifikation der Lösung erarbeitet.

4 Rahmenkonzept für die verteilte Kollaboration

Im Verlauf der bisherigen Ausführungen wurde gezeigt, dass in den integrativen Phasen der mechatronischen Produktentwicklung Defizite hinsichtlich der verteilten Zusammenarbeit zwischen beteiligten Spezialisten existieren. Untersuchungen auf dem Gebiet der verteilten Kooperation und der CSCW-Technologien haben ergeben, dass bislang keine Mittel existieren welche die vorhandenen Probleme hinsichtlich der gegebenen Anforderungen lösen.

Im Verlauf dieses Kapitels wird ein Rahmenkonzept für ein universelles Remote-Framework entwickelt das es ermöglicht, verteilte Koaktionen auch in den integrativen Phasen der Mechatronikentwicklung effizient durchzuführen. Dabei wird die in Abbildung 4-1 dargestellte Vorgehensweise verfolgt.

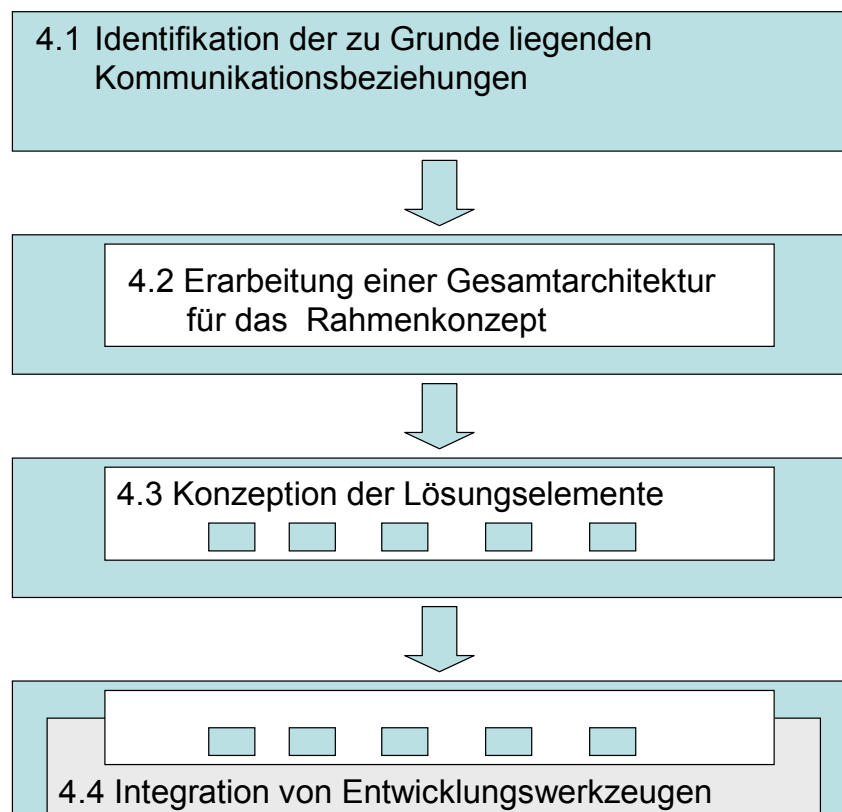


Abbildung 4-1: Entwicklung des Konzepts für ein universelles Remote-Framework

Kapitel 4.1 identifiziert zunächst die in der vorhandenen Problemstellung existierenden Kommunikationsbeziehungen und leitet daraus Konsequenzen für das

Lösungskonzept ab. Aufbauend darauf stellt Kapitel 4.2 die Gesamtarchitektur des Rahmenkonzepts vor, welche in Kapitel 4.3 weiter verfeinert wird. Kapitel 4.4 beschäftigt sich mit der Integration von Entwicklungswerkzeugen in das Remote-Framework.

4.1 Analyse der Kommunikationsbeziehungen

Um CSCW- und andere Remote-Technologien in einem Unternehmen gewinnbringend einzusetzen, ist es Voraussetzung, diese Technologien den vorgegebenen Einsatzszenarien anzupassen (vgl. Kapitel 3.4.1). In der Realität stellt die Bewältigung dieser Anforderung in den überwiegenden Fällen eine komplexe Aufgabe dar. Remote-Kooperationstechnologien gehören typischerweise nicht zur Kernkompetenz eines Greäteentwicklers. Dieser findet sich folglich in einem Spannungsfeld aus drei Faktoren: die Vielzahl der zur Verfügung stehenden Technologien, die Aufgabe der Anpassung an das vorliegende Szenario sowie die Erfüllung existierender Anforderungen. Erst die umfassende Berücksichtigung dieser drei Faktoren führt zu einer wirtschaftlichen, effizienten und bei den Anwendern akzeptierten Lösung (vgl. Abbildung 4-2).

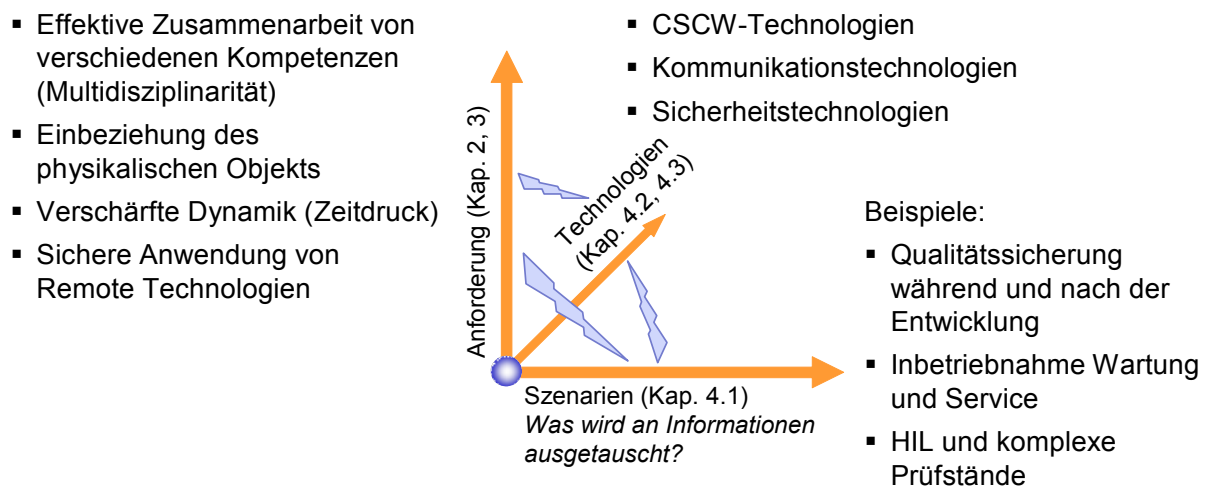


Abbildung 4-2: Grundlegende Faktoren einer angepassten Remote-Lösung

Gerade in KMU, wo das Vorhandensein entsprechender Kompetenzen nicht vorausgesetzt werden kann, stehen die Firmen und Anwender vor Schwierigkeiten bezüglich der Auswahl passender Remote-Technologien und Kooperationskonzepten. Hinzu kommt als weitere Herausforderung die Sicherheitsproblematik in Verbindung mit der Nutzung von Remote-Lösungen.

In diesem Unterabschnitt erfolgt eine genaue Identifikation und Definition des im Bereich der integrativen Phasen der Mechatronikentwicklung vorherrschenden Kommunikationsszenarios. Zunächst werden die Fragestellungen

Wer kommuniziert mit wem? und Was wird an Informationen ausgetauscht? präzisiert. Ein solches Vorgehen ist essentiell, um passende CSCW-Mittel für das Lösungskonzept zu identifizieren, nur so ist eine effiziente Nutzung seitens der Anwender gewährleistet. Das Kapitel endet mit der Vorstellung eines Grobkonzepts für eine den Anforderungen entsprechenden Gesamtlösung.

4.1.1 Darstellung des Kommunikationsszenarios

Eingangs ist zu klären, welche und wie viele Personen in den integrativen Phasen an der gleichzeitigen Bearbeitung einer gemeinsamen Problemstellung unter Einbeziehung des physikalischen Entwicklungsgegenstands beteiligt sind. Als Basis für die Aufstellung des Szenarios dienen untersuchte Beispielanwendungen (Anhang A) sowie die Ergebnisse des Forschungsprojekts *TeleTest*⁴⁷.

Bereits in Kapitel 3 wurde angedeutet, dass es sich bei dem zu Grunde liegenden Szenario im Wesentlichen um eine Eins-zu-Eins-Beziehung zwischen kooperierenden Experten handelt. Diese Einschränkung liegt darin begründet, dass es sicherheitstechnisch als äußerst kritisch zu bewerten ist, wenn von verschiedenen Stellen aus gleichzeitig Interaktionen mit einem physikalischen Gerät erfolgen. Die Überwachung und Koordination der Zugriffe sowie die Abgrenzung der Teilnehmeraktionen gegeneinander stellt eine komplexe und fehleranfällige Aufgabe dar, was dem Ziel einer möglichst einfachen verteilten Kooperation entgegenwirkt. Zumal im hier behandelten Anwendungsgebiet der Bedarf für ein solches Szenario typischerweise nicht besteht, befindet sich diese Problematik nicht im Fokus der vorliegenden Arbeit. Kapitel 2 verdeutlicht, dass sich vielmehr viele verschiedene Kompetenzen auf jeweils spezialisierte Personen verteilen, als dass eine große Personenzahl eines Kompetenzbereichs gleichzeitig verteilte Koaktionen durchführt.

Als Folgerung aus Kapitel 2 ergibt sich, dass verstärkt verschiedene Experten aus unterschiedlichen Spezialgebieten in kurzen Zeitabständen zum Einsatz kommen. Dabei werden verschiedenste Komponenten eines mechatronischen Systems von unterschiedlichen Personen bearbeitet. Ebenfalls kommen dabei verschiedenste Rechnerwerkzeuge zum Einsatz. Für diese Einsatzfälle wird jeweils eine 1:1-Beziehung zwischen dem Experten vor Ort am lokalen Frontend des Rechnerwerkzeugs und dem korrespondierende Experten am entfernten Frontend zu Grunde gelegt (vgl. Abbildung 4-3).

⁴⁷ *Verteiltes Kooperatives Testen mit Hilfe der Internet-Technologien*, gefördert durch die Deutsch Forschungsgesellschaft für die Anwendung der Mikroelektronik (DFAM), AIF-Forschungsvorhaben 13665/N1

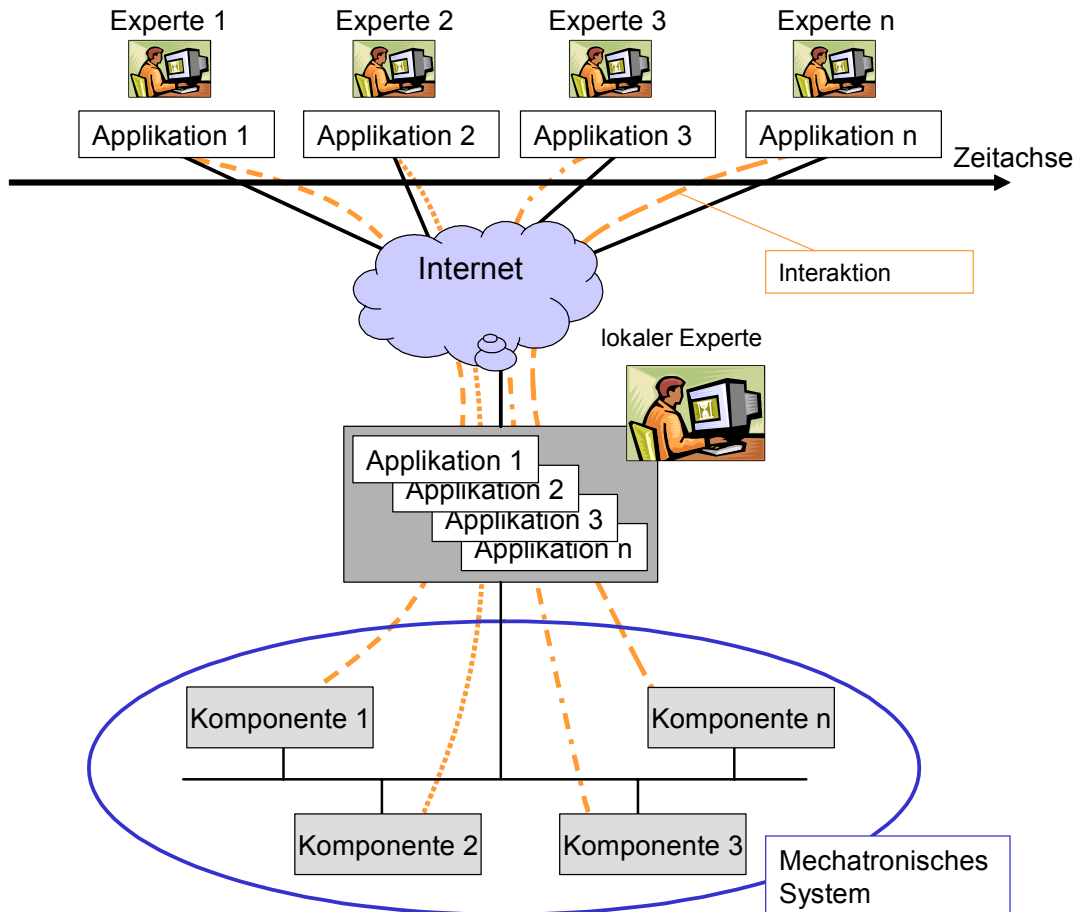


Abbildung 4-3: Zugriff verschiedener Experten auf ein mechatronisches System

Das Kommunikationsszenario für die vorliegende Problemstellung wird wie folgt zusammengefasst:

Definition 4-1: Kommunikationsszenario für die integrativen Phasen der Mechatronikentwicklung

Ein lokaler Entwickler arbeitet mit verschiedenen geografisch entfernten Experten zusammen, die mit unterschiedlichen mechatronischen Komponenten interagieren. Dies kann zeitgleich oder zeitversetzt erfolgen. Die Interaktion erfolgt dabei jeweils über eigene Rechnerwerkzeuge zwischen deren verteilten Frontends eine 1:1-Beziehung besteht. Für die bearbeitete Problemstellung ist dies gleichzusetzen mit einer 1:1-Kommunikationsbeziehung zwischen dem lokalen Entwickler und einem entfernten Experten.

Das Ziel eines Frameworks für die verteilte Zusammenarbeit ist es, dem entfernten Experten die Mitarbeit möglichst so zu ermöglichen, als wäre er am Ort des Geschehens anwesend. Weiterhin geht aus der Aufgabenstellung hervor, dass der entfernte Experte ein Problem nicht in Form eines Outsourcing-Modells selbständig und unabhängig, sondern im Verbund mit dem lokalen Experten bearbeitet. Folglich ist auf beiden Seiten eine Erwartungshaltung an die

maximale Reaktionszeit des jeweiligen Partners vorhanden. Voraussetzung hierfür ist eine Online-Verbindung zwischen den beiden Beteiligten, die Kommunikationsbeziehung für das hier behandelte Szenario besitzt folglich Online-Charakter (vgl. Definition 2-10). Gleichfalls muss das Szenario in die Klasse der Kollaboration (vgl. Definition 2-6) eingeordnet werden.

Assoziation:		1 : 1	1 : n	N : m
Charakter	Informationsfluss			
Synchron	Unidirektional	Instruktion	Vorlesung	
	Bidirektional	Dialog		Konferenz
Asynchron	Unidirektional	Benachrichtigung	Rundschreiben	
	Bidirektional	Korrespondenz		Schwarzes Brett

Tabelle 4-1: Klassifikation der Ausprägungen von Interaktion, mit Beispielen [LUBICH 95]

Tabelle 4-1 identifiziert das zu Grunde liegende Kommunikationsszenario innerhalb der möglichen Ausprägungen von Interaktion. Die von LUBICH identifizierten Klassen stellen ein allgemeingültiges Ordnungsschema unabhängig von Kommunikationstechnologien dar und besitzen deshalb auch nach aktuellem Stand der Technik Gültigkeit.

4.1.2 Das kollaborative Setting

Für das in Kapitel 4.1.1 identifizierte Kommunikationsszenario ist im weiteren Verlauf das passende kollaborative Setting, das heißt die strukturellen und technischen Rahmenbedingungen für das gemeinschaftliche Handeln zu bestimmen. Diese richten sich neben den speziellen Anforderungen der bearbeiteten Problemstellung auch nach universellen Kriterien. Diese Kriterien lassen sich durch Festlegung derjenigen Phasen im Phasenmodell der Koaktion bestimmen, in denen die Zusammenarbeit schwerpunktmäßig stattfindet.

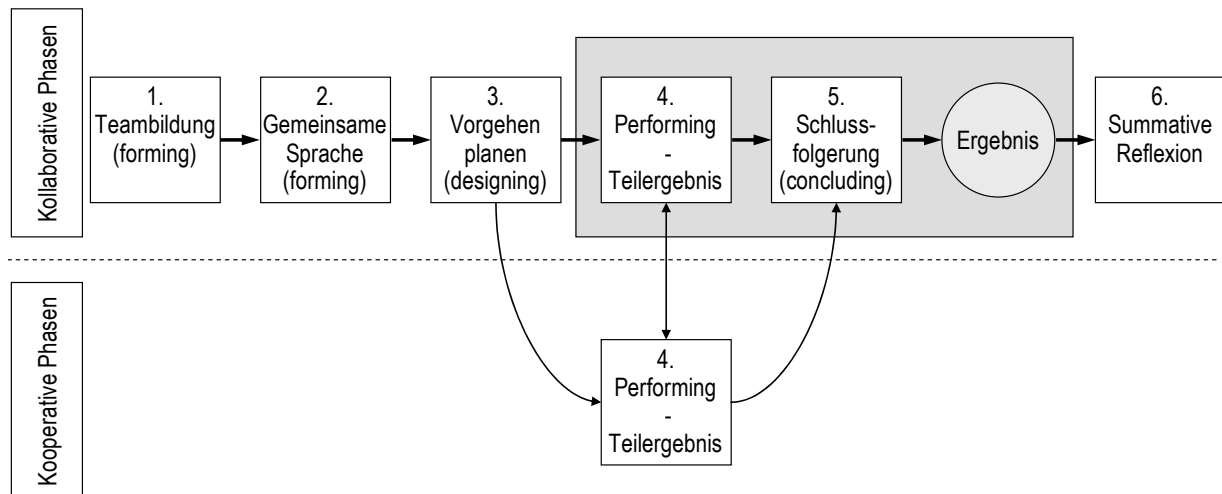


Abbildung 4-4: Einordnung des Kommunikationsszenarios

Abbildung 4-4 kennzeichnet die relevanten Phasen im Modell. Die Einschränkung auf den grau hinterlegten Bereich begründet sich wie folgt:

- Als Folge der Charakteristik einer 1:1-Beziehung kann die Phase der Teambildung vernachlässigt werden.
- Die Zusammenarbeit beschränkt sich auf einen abgegrenzten Bereich im Gesamtprojekt. Es wird deshalb nur in Ausnahmefällen eine ausgeprägte Phase der gemeinsamen Begriffsbildung unter vielen Beteiligten notwendig sein. Aufgrund des begrenzten Umfangs kann diese direkt während des Performings erfolgen.
- Die Planung des Vorgehens ist im Vorfeld ebenfalls aufgrund des begrenzten Handlungsfeldes und des Ad-Hoc-Charakters der Zusammenarbeit von geringer Relevanz. Vielmehr muss das detaillierte gemeinschaftliche Vorgehen (Interaktionen mit dem realen System) unmittelbar vor einer jeweiligen Aktion immer wieder abgesprochen werden.
- Der entfernte Experte führt über die Remote-Anbindung keine eigenständigen abgeschlossenen Handlungen ohne Beteiligung des lokalen Entwicklers aus. Gemäß Definition 2-6 ist die Zusammenarbeit in den Bereich der Kollaboration einzuordnen.
- Die Bewertung der erzielten Ergebnisse des gemeinschaftlichen Handelns erfolgt online während der Zusammenarbeit, schon alleine deshalb, um weitere Schritte zu bestimmen. Eine ausgeprägte Phase der summativen Reflexion erfolgt hingegen selten und steht folglich nicht im Mittelpunkt.

4.1.3 Bestimmung der Randbedingungen für die Kollaboration

Die Eingrenzung auf die Phasen vier und fünf im zuvor dargestellten Phasenmodell ermöglicht in Verbindung mit dem Ergebnis aus Kapitel 4.1.1, eine Beschränkung auf die für die Zusammenarbeit wichtigen Kooperationsmittel. So sind Mittel zur Koordination vieler Beteiligter, wie etwa ein asynchrones verteiltes Terminkalendersystem, nicht notwendig. Da es sich im vorliegenden Kommunikationsszenario um eine 1:1-Beziehung handelt, kann auf Infrastrukturen wie beispielsweise Conferencing-Server⁴⁸ ebenfalls verzichtet werden. Die notwendigen Mittel werden lediglich durch die Phasen der konkreten Kollaboration bestimmt.

Im Gegensatz zu verteilten Konferenzen als weit verbreiteten Anwendungsfall von CSCW-Technologien steht bei der vorliegenden Problemstellung die interpersonelle Kommunikation nicht im Vordergrund. An diese Stelle tritt die gemeinsame Nutzung eines Rechnerwerkzeuges durch zwei geografisch getrennte Teilnehmer. Folglich ist es nicht notwendig ein möglichst realitätsnahes Gefühl der Anwesenheit des jeweiligen Kommunikationspartners zu erzeugen (beispielsweise um gemeinsame Diskussionsrunden zu führen). Informationen über die implizite Kommunikation⁴⁹ müssen also nur in geringem Maße übermittelt werden. Stattdessen ist es wichtig, dass die Kommunikationspartner durch die zu Grunde liegende Infrastruktur jederzeit in der Lage sind, schnell und einfach sachliche Informationen auszutauschen. Es wird deshalb die Sprache als das dominierende Mittel zur interpersonellen Kommunikation für das betrachtete kollaborative Setting identifiziert. Kommunikation mittels Videoübertragung wird als unterstützendes und nicht als dominierendes Mittel erachtet. Hinzu kommen weiter unterstützende Mechanismen für spezielle Anwendungsfälle wie beispielsweise die schnelle Übermittlung von einigen Zeilen Programmcode, Parametersätzen, Protokolldateien oder die Notwendigkeit einen Gedankengang schnell zu skizzieren. Die Summe dieser Mittel wird im weiteren Verlauf der vorliegenden Arbeit bestimmt.

Eine wesentliche Eigenschaft des zu Grunde liegenden kollaborativen Settings muss die Übermittlung von Interaktionen innerhalb eines abgeschlossenen Zeitraums sein. Dieser Zeitraum ist so bemessen, dass ergonomisches Zusammenarbeiten sichergestellt ist. Richtlinien hierzu finden sich unter anderem in [VDMA 04]. Tabelle 4-2 gibt einen Überblick über konkrete Zeitbedingungen wie sie beispielsweise für kooperative Multimediasysteme empfohlen werden.

⁴⁸ Zentraler Server über den sämtliche Teilnehmer einer verteilten Konferenz verbunden sind. Der Conferencing-Server übernimmt grundlegende Aufgaben wie beispielsweise Sessionverwaltung, Benutzerverwaltung und Verteilung der Datenströme an die Teilnehmer.

⁴⁹ Implizite Kommunikation umfasst die unbewusste Wahrnehmung und Übermittlung von Kommunikationsregeln zwischen den Kommunikationspartnern, beispielsweise durch Körpersprache oder Gesichtsausdruck.

Übertragungszeit	Qualitätscharakterisierung
0 – 150 ms	"akzeptabel für die meisten Anwendungen"
150 – 400 ms	"kann einige Anwendungen beeinträchtigen"
Über 400 ms	"unakzeptabel für allgemeine Netzwerkanwendungen"

Tabelle 4-2: Zeitbedingungen in kooperativen Multimediasystemen [HERMANN 97]

Ist die Ausführung einer Interaktion mit einer Überschreitung dieser Zeitschranken verbunden, so ist der entfernte Anwender darüber entsprechend zu informieren. Diese Feststellung führt zu:

Definition 4-2: Echtzeitfähige verteilte Entwicklung

Werden bei einer verteilten Koaktion in der Produktentwicklung bei der Reaktion auf Ereignisse von allen Beteiligten definierte Zeitgrenzen eingehalten so wird dies als echtzeitfähiges Entwickeln bezeichnet. Die Zeitgrenzen sind für den jeweiligen Anwendungsfall dann ausreichend wenn ergonomisches Zusammenarbeiten sichergestellt ist.

Der Begriff Echtzeit ist dabei in vielen Anwendungsfeldern mit zum Teil sehr unterschiedlichen Anforderungen verknüpft. In [GAUSEMEIER+ 00] wird zwischen so genannter harter und weicher Echtzeit unterschieden. Die Einteilung erfolgt aufgrund der Ausprägungen der Konsequenzen bei überschreiten der Zeitbedingungen. Ausdrücklich wird im vorgestellten Kontext nicht die harte Echtzeitbedingung wie sie beispielsweise in der Automatisierungstechnik vorherrscht als Maßstab zu Grunde gelegt. Die hier betrachtete Echtzeit begrenzt den zeitlichen Rahmen insofern, als dass effektives Zusammenarbeiten der Beteiligten gewährleistet ist, was in einem wesentlichen Punkt den gestellten Anforderungen (vgl. Kapitel 3.1.3) entspricht.

4.2 Elemente des Lösungsansatzes

Die vorliegende Arbeit konzipiert und spezifiziert ein Remote-Framework welches es ermöglicht, verschiedene Werkzeuge, die der Spezifikation entsprechen, mit Remote-Kollaborations-Funktionalität zu erweitern.

Im weiteren Verlauf erfolgt die Entwicklung des Grobkonzepts sowie der Basisarchitektur für das Framework als Grundlage für die Umsetzung des Lösungskonzepts. Der dabei verfolgte Ansatz geht davon aus, dass nicht allein die Anwender sondern auch die Hersteller mechatronischer Entwicklungswerkzeuge Beitrag zu einer Remote-Erweiterung ihrer Produkte leisten. Hierfür werden die in der vorliegenden Arbeit entwickelten universellen Schnittstellen zur

Verfügung gestellt und dadurch der zu leistende Aufwand so weit wie möglich reduziert. Solche Aktivitäten der Hersteller lassen sich über entsprechenden Kundendruck initiieren. Weitere Gründe für die Wahl dieses Ansatzes sind:

- Um allen Anforderungen zu genügen, sind Veränderungen an der Applikation unumgänglich. Hierfür fehlen dem Anwender sowohl die Möglichkeiten als auch der Zugang zu den Quellen und Interna der Werkzeug-Applikationen.
- Dem Anwender fehlt das Spezialwissen um seine Werkzeuge selbst mit Remote-Funktionalität auszustatten bzw. diese in das Werkzeug einzubinden. Dies würde zu bisherigen, unzureichenden Lösungen wie Remote-Desktop-Verbindungen bzw. zu unüberwindbaren Akzeptanzhürden seitens der Anwender führen. Dem Anwender muss ein fertiges, einfach zu nutzendes System an die Hand gegeben werden. Nur so lassen sich Probleme, die schon bei einfachen CSCW-Werkzeugen existieren [MUNKVOLD 03], bei der verteilten Nutzung komplexer Werkzeuge vermeiden.

4.2.1 Architektur eines universellen Remote-Frameworks

Die Architektur einer universellen Schnittstelle zur Realisierung der verteilten Zusammenarbeit zwischen Experten unter Einbeziehung des physikalischen Systems sieht grundlegend zwei logische Kommunikationskanäle vor: Es handelt sich um einen Kanal für die Abwicklung der Kommunikation zwischen den eingesetzten Rechnerwerkzeugen und einen Kanal zur Abwicklung der interpersonellen Kommunikation zwischen den Beteiligten. Die beiden Kanäle werden für den Anwender so gekapselt, dass dieser nur eine Kommunikationsverbindung handhaben muss. Die Administration der Verbindungen erfolgt über ein zentrales Frontend.

Abbildung 4-5 zeigt den grundlegenden Aufbau des Frameworks für die verteilte Kollaboration unter Einbeziehung des physikalischen Systems. Die besonderen Anforderungen, die sich aus dem entfernten Zugriff auf reale mechatronische Systeme ergeben, verbieten es, die Funktionalität des verwendeten Entwicklungstools einfach nach außen hin zu veröffentlichen. Aus diesem Grund ist ein Wrapper⁵⁰ vorgesehen, dessen Aufgabe es ist, sämtliche Werkzeugfunktionalität zu kapseln. Der Wrapper beinhaltet die Schnittstelle zum eingesetzten Werkzeug und stellt die Verbindung zu denjenigen Komponenten dar, welche die Remote-Funktionalität implementieren. Eine tragende Rolle in dem Konzept spielt dabei die Komponente zur Kontrolle und Überwachung des Zugriffs von außen. Sämtliche Aktionen des entfernten Benutzers werden durch

⁵⁰ Übersetzung aus dem Englischen: Schutzhülle. Allgemein ein Programm, das als Interface zwischen dem aufrufenden und dem umschlossenen Programmcode agiert. Dies kann aus Kompatibilitätsgründen eingesetzt werden, wenn z.B. der umschlossene Code eine andere Programmiersprache verwendet, zur Sicherheit, um z.B. den Zugriff einzuschränken bzw. zu erweitern oder um etwas zu emulieren [WIKIPEDIA 05].

die Zugriffskontrolle überprüft und nur im positiven Fall an den Wrapper und anschließend zum Werkzeug weitergeleitet. Die Abwicklung des gesamten Datenverkehrs erfolgt über ein Kommunikationsmodul. Dieses sorgt für die Integration der beiden Kommunikationskanäle und stellt letztendlich die Verbindung zwischen der lokalen und der entfernten Werkzeuginstanz dar. Eine wichtige Aufgabe der Kommunikationskomponente ist die Bereitstellung von Sicherheitsfunktionalität für die Datenübertragung.

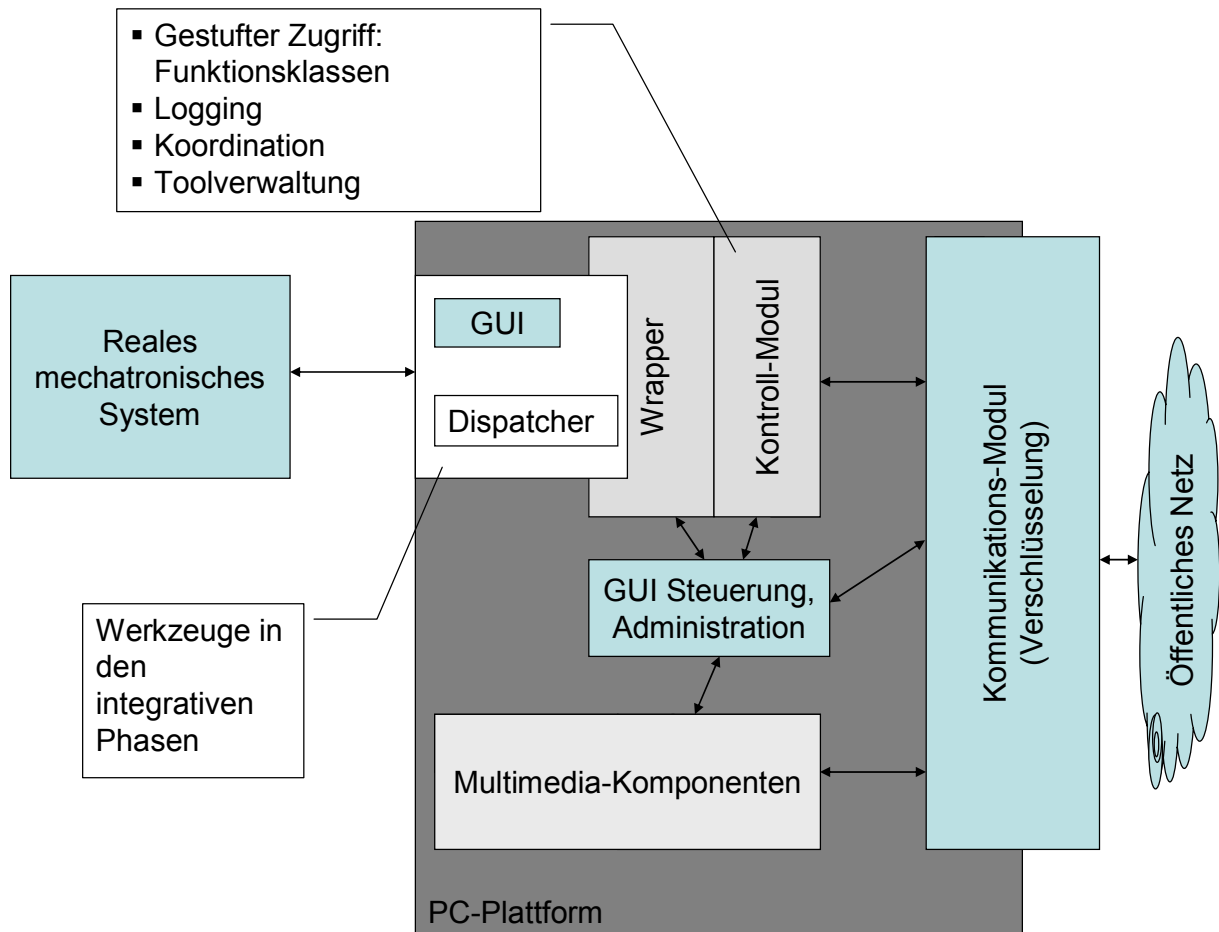


Abbildung 4-5: Gesamtarchitektur der Lösung

Die Multimediakomponenten sind außerhalb der Zugriffskontrollkomponente angeordnet, diesen Komponenten wird kein Zugriff auf die Werkzeugfunktionalität gestattet. Jedoch erfolgt die Übertragung der Multimediadaten ebenfalls über das zentrale Kommunikationsmodul um auch hier mittels Verschlüsselung die Integrität und Vertraulichkeit der Daten sicherzustellen, falls dies vom Anwender gewünscht wird.

Um alle Anforderungen zu erfüllen, sieht die Architektur weiterhin zwei Komponenten vor, die innerhalb des Rechnerwerkzeuges integriert sind, bzw. die Anforderungen an den Aufbau der Applikation darstellen. Es handelt sich um

die Benutzeroberfläche (GUI⁵¹) sowie um eine Dispatcher-Komponente⁵². Um der Anforderung nach einer rollenorientierten Benutzeroberfläche zu genügen, sieht das GUI einen modularen Aufbau der grafischen Komponenten vor. Diese können, einem Baukastenprinzip entsprechend, den Anforderungen der jeweiligen Benutzerrolle zusammengestellt werden.

Der Dispatcher dient als Mittel, um den verteilten Einsatz des Rechnerwerkzeuges einfach zu realisieren. Hierfür wird ein Konzept für einheitliche interne Schnittstellen entwickelt, welches es ermöglicht, die Applikation ohne Implementierungs- und Konfigurationsaufwand sowohl Client-seitig, als auch Server-seitig, mit Verbindung zum realen mechatronischen System einzusetzen. Dieses Konzept trägt insbesondere der Anforderung nach einfacher Benutzbarkeit Rechnung, reduziert aber auch den Aufwand für die Implementierung der Remote-Funktionalität seitens des Tool-Herstellers.

4.2.2 Rollenmodell für die verteilte Kollaboration in der Mechatronik

Nicht nur innerhalb eines mechatronischen Entwicklungsprojekts ist es erforderlich, Rollen für die Mitarbeiter des Projekt-Teams festzulegen [BMI 05]. Anhand dieser Rollen sind die Aufgaben und Kompetenzbereiche eines Mitarbeiters definiert, der die entsprechende Rolle einnimmt, aber auch die notwendigen Fähigkeiten charakterisiert. Verschiedene Rollen können gleichfalls mit der Verwendung unterschiedlicher Werkzeuge, Spezialisierung auf Fachgebiete und unterschiedlichen Abstraktionsniveaus innerhalb des Entwicklungsprojekts verbunden sein. Auch wenn in der Praxis Überschneidungen nicht vermeidbar sind, ist eine Abgrenzung der Rollen gegeneinander der essentielle Sinn eines Rollenmodells. Es ist leicht nachvollziehbar, dass mit jeder Rolle auch eine bestimmte Sichtweise auf das Entwicklungsprojekt bzw. ein bestimmtes Umfeld verbunden ist. Innerhalb dieses Rahmens nimmt der Rollenträger seine Aufgaben im Entwicklungsprojekt wahr und führt entsprechende Tätigkeiten aus.

Der Forderung nach Anpassung einer verteilten Applikation an die jeweilige Rolle des Benutzers muss ein Konzept für verteilte Werkzeuge genüge leisten, um ergonomisches und effizientes Zusammenarbeiten der Beteiligten zu gewährleisten. Zusätzlich sollen Basisrollen als Ausgangsbasis für individuelle Konfigurationen an die Hand gegeben werden um den Anwender den Einstieg in ein verteiltes Werkzeug zu erleichtern. Die Benutzerschnittstelle der Applikation darf deshalb nicht starr, sondern muss in Abhängigkeit von den Rollen der beteiligten Spezialisten flexibel anpassbar sein. Hierfür ist zunächst festzulegen, auf welcher Abstraktionsebene die Anpassung stattfindet, d. h. wie granular die Rollendefinition erfolgt. Es lassen sich zwei Extreme identifizieren:

⁵¹ Graphic User Interface

⁵² Übersetzt: Verteilerkomponente. Sorgt hier für das Routing von Nachrichten in einer verteilten Softwarearchitektur.

- Spezifische Rollen für einen speziellen Anwendungsfall bzw. ein spezielles Anwendungsgebiet
- Universelle Rollen für die Mechatronikentwicklung

Zwischen diesen beiden Extremen definieren unterschiedlichste Organisationen Rollenmodelle. Die Ergebnisse einer Analyse der wichtigsten Vertreter werden im Folgenden kurz vorgestellt:

- VDI/VDE 2187 – Einheitliche Anzeige- und Bedienoberflächen für digitale Feldgeräte [VDI2187 02]:

Das Rollenmodell ist spezialisiert für digitale Feldgeräte und dient vorwiegend der Zugriffsbeschränkung auf die Gerätefunktionalität und weniger als Basis für angepassten Werkzeug-Schnittstellen.

- FDT⁵³ [PNO 01]:

Auch bei FDT dienen die definierten Rollen weniger der Ergonomie als der Zugriffsbeschränkung. Die Rollen sind den Sicherheitsstufen und nicht den Spezialgebieten eines mechatronischen Entwicklungsprojekts angepasst.

- NAMUR⁵⁴-Empfehlung NE 91 [NAMUR 01]: Anforderung an Systeme für anlagennahes Management

Spezialisiert auf die Verfahrenstechnik. Gibt Hilfestellung zur Abschätzung des Nutzens von Asset-Management-Systemen, enthält eine Anforderungsliste bezüglich der Funktionalität solcher Systeme. Generell wird die Forderung nach Nutzerorientierten Oberflächen und unterschiedlichen Prozess-Sichten erhoben. Hierfür werden drei grundlegende Benutzerrollen entsprechend der Verfahrenstechnik definiert.

- DeKOS [BENDER+ 02A]:

Generische Bedienoberflächen, welche die Zuordnung von Funktionalität zur Feldgerätebedienung zu Benutzerrollen ermöglichen (in Anlehnung an FDT). Durch Auswahl einer Benutzerrolle erfolgt die Filterung der angebotenen Funktionen. Zusätzlich ist die Auswahl von bestimmter Sichten möglich, hier werden die von der Applikation angebotenen Funktionen in entsprechende Klassen eingeordnet und entsprechend dargestellt, z. B. FDT-Sicht oder NAMUR-Sicht. Auch die Definition eigener Sichten und Interfaces ist möglich.

- Carnegie Mellon Software Engineering Institute [BERGNER 99]:

Im Rahmen eines internationalen Kongresses wurde am Software Engineering Institut der Carnegie Mellon Universität ein Rollenmodell für die

⁵³ *Field Device Tool*: Framework für Tools zur Bedienung, Parametrierung, Konfiguration etc. von intelligenten Automatisierungsgeräten.

⁵⁴ Normenarbeitsgemeinschaft Mess- und Regelungstechnik der chemischen Industrie.

Softwareentwicklung vorgestellt. Dieses beinhaltet die Rollen Komponentenentwickler, Komponentenintegrator, Systemanalyst, Systemarchitekt und Projektkoordinator. Das Anwendungsgebiet bezieht sich zwar auf den Softwarebereich, dennoch sind die Rollen so abstrakt gehalten, dass sie auch für die Mechatronikentwicklung im Allgemeinen Relevanz besitzen.

- V-Modell XT [BMI 05]:

Die Erweiterung des für die Softwareentwicklung definierten V-Modells auf die Mechatronik wurde mit der Veröffentlichung der Spezifikation für das V-Modell XT vollzogen. Die in [BERGNER 99] definierten Rollen finden sich auch hier wieder, sind jedoch um ein Vielfaches verfeinert. Es existieren insgesamt 34 Rollen für beteiligte Akteure über den gesamten Zeitraum eines mechatronischen Entwicklungsprojekts. Die in den integrativen Phasen bedeutsamen Rollen lassen sich gut isolieren.

Die Analyse der vorgestellten etablierten Modelle verdeutlicht zunächst die Vielfalt der existierenden anwendungsbezogenen Benutzerrollen. Die dort definierten Modelle lassen sich nicht zu einem universellen Rollenmodell für die integrativen Phasen der Mechatronikentwicklung vereinen, da den individuellen Spezialgebieten zu sehr Rechnung getragen wird. Hinzu kommt, dass die Rollendefinitionen zu sehr auf die Vergabe von Benutzerrechten abzielen, was nicht dem Scope dieser Arbeit entspricht. Die untersuchten anwendungsbezogenen Rollenmodelle kommen also nicht für die Definition von Basisrollen in Frage.

Interessant erscheint hingegen die Festlegung von Rollen auf einem abstrakten Niveau gemäß BERGNER und V-Modell XT. Der Abstraktionsgrad ist hier weitaus größer, weshalb diese Modelle als Ausgangsbasis für die Entwicklung von Basisrollen geeignet erscheinen. Durch Anpassung der Modelle auf die Besonderheiten der integrativen Phasen erhält der Anwender die geforderte Unterstützung in Form von vorwählbaren Basiskonfigurationen für ein Frontend. Für die Entwicklung der Basisrollen werden im weiteren Verlauf relevante Spezifika der integrativen Phasen betrachtet:

Gemäß Definition 2-1 besteht ein mechatronisches System aus Teilkomponenten, die jeweils für sich wiederum mechatronische Systeme darstellen können. Zwischen den Teilkomponenten bestehen Kommunikationsverbindungen, über welche die Koordination und Zusammenarbeit der Komponenten erfolgt. Sowohl Teilkomponenten als auch Kommunikationskomponenten stammen in vielen Fällen von verschiedenen Zulieferern aus unterschiedlichen Fachgebieten. Dabei kann es sich entweder um standardisierte Lösungen handeln, für die öffentlich zugängliche Spezifikationen existieren oder um proprietäre Lösungen, die dem Spezialwissen des jeweiligen Herstellers entsprechen. Letztere sind in der Regel nicht offen gelegt.

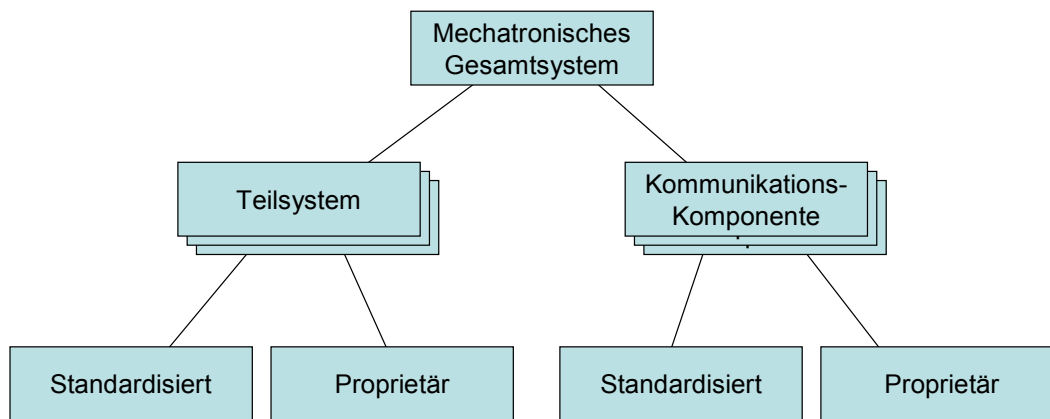


Abbildung 4-6: Fachgebietsunabhängige Klassifizierung mechatronischer Komponenten

Mit Hilfe der in Abbildung 4-6 dargestellten Klassifizierung lassen sich im weiteren Verlauf die potentiellen Nutzerkreise für verteilte Werkzeuge in den integrativen Phasen bestimmen. Die dargestellte Aufteilung ist unabhängig von fachspezifischen Gebieten. Anhand der Klassifizierung lassen sich Grundaufgaben identifizieren und den Nutzerkreisen von verteilten Werkzeugen zuordnen. Dies geschieht mittels der im Anhang B gegebenen Tabellen. Als zusätzliche Hilfestellung ist der Bezug der identifizierten Rollen zu den etablierten Modellen nach BERGNER und dem V-Modell XT angegeben. Die erarbeitete Rollendefinition verfeinert diese Modelle indem grundlegend zwischen standardisierten und eigen entwickelten Teilkomponenten unterschieden wird. Ebenfalls in den Tabellen enthalten sind die Ausprägungen, die das Frontend einer verteilten Applikation für die entsprechende Rolle aufweisen muss. Die vier definierten Basisrollen lassen sich wie folgt zusammenfassen:

Rollenbezeichnung:	Spezialist für Teilkomponente
V-Modell XT:	Softwarearchitekt, Softwareentwickler
Bergner:	Komponentenentwickler
Rollenbezeichnung:	Spezialist für offene (Kommunikations-) Standards
V-Modell XT:	Softwarearchitekt, Softwareentwickler
Bergner:	Komponentenentwickler
Rollenbezeichnung:	Spezialist für Gesamtsystem
V-Modell XT:	Systemarchitekt, Systemintegrator
Bergner:	Systemarchitekt
Rollenbezeichnung:	Prüfspezialist (Spezifikations-Spezialist)
V-Modell XT:	Prüfer, Anforderungsanalytiker (Auftraggeber und Auftragnehmer)
Bergner:	-

Tabelle 4-3: Rollenmodell für verteilte Werkzeuge in den integrativen Phasen

Die Definition der Rollen ist bewusst auf einem abstrakten Niveau gehalten, um für fachspezifische Anpassungen offen zu sein, die überschaubare Anzahl unterstützt die einfache Umsetzung in Werkzeugen. Weitere Verfeinerungen zur Optimierung der Ergonomie eines verteilten Werkzeugs sind ausdrücklich zugelassen. Die Durchführung der Optimierung ist Aufgabe des Anwenders, der Tool-Hersteller ist dafür verantwortlich, dass entsprechende Funktionalität in den Werkzeugen zur Verfügung steht. Die Zuordnung von Werkzeugfunktionen zu den Basisrollen für einen spezifischen Anwendungsfall ist soweit wie möglich durch den Hersteller vorzunehmen. Dies kann jedoch allenfalls als Vorschlag für den Anwender interpretiert werden, letztendlich muss dieser über individuelle Konfigurationsmöglichkeiten verfügen. Im weiteren Verlauf der Ausführungen wird hierfür eine modulare Benutzerschnittstelle in einer verteilten Werkzeugapplikation entwickelt.

4.2.3 Audiovisuelle-Kommunikation

Das hier entwickelte Gesamtkonzept für die verteilte Kollaboration in den integrativen Phasen der Mechatronikentwicklung sieht, neben dem Datenkanal für die Inter-Tool-Kommunikation, einen integrierten Multimediakanal vor. Dieser erfüllt zwei bedeutende Aufgaben innerhalb des entwickelten Lösungskonzepts:

1. Interpersonelle Kommunikation zwischen den beteiligten Personen
2. Vermittlung eines realitätsnahen Einblicks in die Situation vor Ort für den entfernten Experten

Die Charakteristik des ersten Punktes wurde in Kapitel 4.1.1 erarbeitet und definiert. Dort wird die Übermittlung von Sprache als das für die Problemstellung relevanteste Mittel zur interpersonellen Kommunikation identifiziert.

Die Vermittlung eines möglichst aussagekräftigen „vor Ort-Gefühls“ zur Unterstützung der Arbeit des entfernten Experten ist das Ziel des zweiten Punktes. Die bisherigen Ausführungen betonen, dass jegliche Aktion eines entfernten Experten auf aussagekräftigen, präzisen Daten basieren muss. Dennoch ist es in vielen Fällen von Vorteil, auch einen qualitativen Überblick der Situation vor Ort zu erhalten, beispielsweise um Prozessdaten einfacher zuordnen zu können und ein realitätsnahes Vorstellungsvermögen zu unterstützen. Anhand realer Bilder wird es dem lokalen Entwickler zudem erleichtert, einen Sachverhalt zu schildern und den entfernten Experten anhand realer Bilder anzuleiten.

Das hier vorgestellte Konzept sieht die Nutzung audiovisueller Komponenten wie folgt vor:

- Die interpersonelle Kommunikation wird vorwiegend über einen Audio-Kanal abgewickelt, der mindestens Telefonqualität aufweist.
- Im Gegensatz zu Konferenzsystemen steht die Übermittlung der Körpersprache nicht im Mittelpunkt. Die Videoübertragung wird deshalb als

optional angesehen, das Rahmenkonzept ist entsprechend ausgelegt. Dies erleichtert gleichfalls den Ad-Hoc-Einsatz von verteilten Werkzeugen, da Video-Systeme erfahrungsgemäß einen hohen Aufwand für das Setup mit sich ziehen [MUNKVOLD 03].

- Dem Videosystem wird die Rolle einer „mobilen Kamera“ zuteil. Diese kann gleichermaßen für die interpersonelle Kommunikation verwendet werden, ist aber ausgelegt, um dem entfernten Experten einen qualitativen Eindruck von der Situation vor Ort zu vermitteln. Eine qualitative Beurteilung verlangt im vorliegenden Anwendungsfall keine hochauflösenden Bilder und erfordert lediglich niedrige Frame-Raten⁵⁵. Der Anforderung nach geringem Bandbreitenbedarf wird folglich entsprochen. Das Konzept sieht eine bewegliche Kamera vor, die einfach und kurzfristig an verschiedenen Stellen des Entwicklungslabors platziert werden kann.

Das vorgestellte Konzept für die audiovisuelle Kommunikation wird anhand eines Beispiels veranschaulicht: Mechatronische Geräte, wie in Abbildung 2-9 dargestellt, sind in einer verfahrenstechnischen Anlage eingebaut. Zur Optimierung des Systems kann ein entfernter Spezialisten über ein verteiltes Werkzeug sämtlich Reglerparameter und Reglergrößen online bearbeiten. Durch ein Videobild kann er in Addition zu vorhandenen Messwerten die Auswirkung seiner Aktionen erfassen und vorab qualitativ interpretieren. Des Weiteren sind in der Anlage Geräte vorhanden, deren angezeigte Daten nicht über ein Bussystem ausgegeben werden und folglich auch nicht im Rechnerwerkzeug zur Verfügung stehen. Auch hier ist mittels der Videotechnik eine Kontrolle durch den entfernten Spezialisten möglich (vgl. Abbildung 4-7).



Abbildung 4-7: Screenshot des Live-Videobildes eines Stellventils

⁵⁵ Anzahl der Bilder pro Sekunde die in einem Video-Stream enthalten sind.

4.2.4 Inter-Tool-Kommunikation

Für die Abwicklung der Kommunikation zwischen den verteilten Rechnerwerkzeugen existiert ein Datenkanal, der aus Sicht des Anwenders mit dem Multimedialkanal gebündelt erscheint. Der Datenkanal überträgt sowohl die Funktionalität eines Tools zwischen der entfernten und der lokalen Instanz als auch umfangreiche Daten, die auf beiden Seiten anfallen können (beispielsweise Prüfprotokolle, Messreihen, Online Messwerte etc.).

Zudem ist es die Aufgabe der Werkzeugkopplung, die durch eine Remote-Erweiterung zusätzlich entstehenden Aufgaben zu übernehmen. Hierzu zählen beispielsweise die Handhabung der Verbindungskontrolle (Session-Management, Erkennung von Timeouts oder Verbindungsabbrüchen), Protokollierung und ein verteiltes Fehlermanagement. Neben diesen Funktionen besteht der Bedarf nach Einschränkung des Remote-Zugriffs in Abhängigkeit von dem damit verbundenen Gefährdungspotential. In diesem Zusammenhang sei erwähnt, dass die Einschränkung der Werkzeugfunktionalität in Abhängigkeit von Benutzeridentitäten nicht in diesen Aufgabenbereich fällt. Diese Aufgabe betrifft unabhängig von etwaigen Remote-Erweiterungen primär das Werkzeug an sich, also auch die lokale Installation. Die Realisierung eines dahin gehenden Zugriffskonzepts ist ohnehin durch das Werkzeug vorzunehmen. Der Kanal für die Werkzeugkopplung befasst sich hingegen mit der Zugriffsbeschränkung - unabhängig von Benutzern, jedoch abhängig von dem durch den Remote-Zugriff selbst entstehenden Gefährdungspotential. Er beinhaltet Mechanismen zur Beschränkung der Interaktionen seitens des entfernten Benutzers, also gewissermaßen Filterfunktionalität. Diese Filtermechanismen sind so in den Kanal integriert, dass für ein Werkzeug keine zusätzlichen Implementierungen vorzunehmen sind.

Für eine möglichst universelle Lösung ist die Realisierung des Datenaustauschs mit standardisierten Mitteln von essentieller Bedeutung. Nur so sind im Bedarfsfall Anpassungsarbeiten auf spezielle Anwendungsgebiete mit geringem Aufwand zu realisieren, was die Akzeptanz der Lösung sichert. Die Realisierung des Kanals erfolgt deshalb mittels verbreiteter und etablierter Standardlösungen der Informationstechnik. Eine besondere Rolle spielen etablierte Methoden aus dem Bereich des Remote-Zugriffs in der Automatisierungstechnik (vgl. Kapitel 3.2.1). Die Auswahl der Technologien erfolgt unter dem Gesichtspunkt, die zu übermittelnden Daten möglichst zu minimieren. Nicht zuletzt ist die Absicherung des Datenverkehrs gegenüber Unbefugten und Angreifern zu gewährleisten. Hierfür wird eine Komponente in den Kanal integriert welche Verschlüsselungs- und Absicherungsaufgaben übernimmt. Wie die späteren Ausführungen zeigen, ist die Integration der Werkzeugkopplung in bestehende IT-Infrastrukturen zu gewährleisten. Entsprechende Vorgehensweisen und Lösungen werden im weiteren Verlauf dieser Arbeit entwickelt.

4.3 Beschreibung der Lösungselemente

Das folgende Kapitel beschreibt die Umsetzung des bisher erarbeiteten Rahmenkonzepts in ein universelles Remote-Framework. Hierfür werden die benötigten Lösungselemente im Einzelnen vorgestellt.

4.3.1 Zugriffskontrollmodul

Wie in Abschnitt 4.2.4 dargestellt, ist die Kontrolle des Remote-Zugriffs auf die Funktionalität eines Werkzeugs eine essentielle Aufgabe des Kanals für die Werkzeugkopplung. Diese Aufgabe wird von einem Kontrollmodul übernommen. Das Kontrollmodul beinhaltet das Konzept eines gestuften Zugriffs auf die Werkzeugfunktionen in Abhängigkeit vom Gefährdungspotential bei entfernten Interaktionen. Anhand dieses Konzepts werden solche Funktionen entsprechend eingestuft. Die Filterungsmaßnahmen selbst werden anhand der definierten Einstufung durch das Kontrollmodul (Abbildung 4-5) des Remote-Frameworks vorgenommen, siehe Abbildung 4-8.

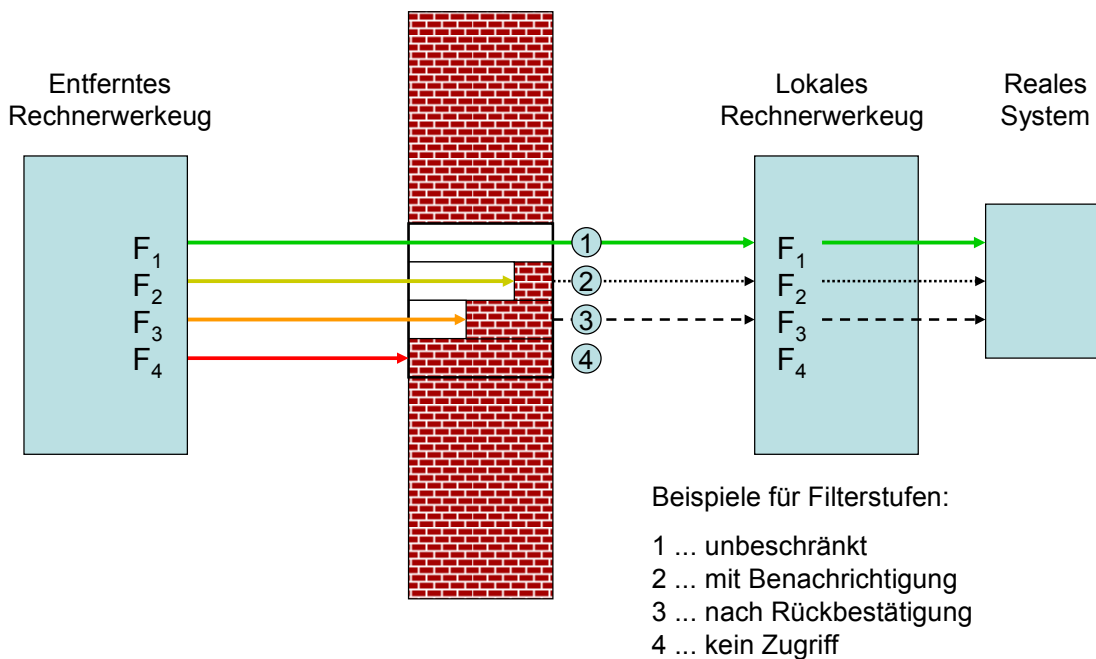


Abbildung 4-8: Filterkonzept für Remote-Interaktionen

Ein Anwender muss den Filter gemäß seiner Bedürfnisse konfigurieren können. Hierfür wird ein Konzept entwickelt, anhand dessen er die benötigten Funktionen in vorgegebene Filterstufen klassifizieren kann.

4.3.2 GUI-Modul

Wie in vielen anderen Technologiebereichen ist die Akzeptanz einer Lösung seitens der Anwender entscheidend von der Bedienergonomie abhängig. Um auch die Akzeptanz einer Lösung für die verteilte Zusammenarbeit unter den Anwendern zu sichern, ist eine ergonomische, intuitive und einfache Bedienung der Systems ein entscheidender Erfolgsfaktor. Die komplexe Anwendung sowie das aufwändige Setup vieler CSCW-Systemen führt in der Praxis immer wieder zur Akzeptanzhürden bei der Nutzung solcher Systeme (vgl. hierzu Kapitel 3.4). Die einfache Nutzung der Funktionalität für die verteilte Kollaboration ist deshalb ein essentielles Merkmal des entwickelten Remote-Frameworks.

Der Ansatzpunkt besteht darin, die umfangreiche Funktionalität einzelner Remote-Komponenten auf das unmittelbar Nötige zu beschränken, und dem Anwender an zentraler Stelle zur Verfügung zu stellen. Die aufwändige Handhabung vieler eigenständiger Softwarekomponenten und die damit verbundene Konfiguration der Applikationen entfällt.

Das Remote-Framework stellt deshalb ein zentrales Benutzer-Interface zur Verfügung, in welches die Funktionalität aller beteiligten Remote-Komponenten integriert ist. Eine Übersicht der Funktionalität gibt Tabelle 4-4: Es ist ersichtlich, dass diverse Funktionen in verschiedenen Komponenten redundant vorhanden sind. Diese werden durch die zentrale Bedienkomponente zusammengefasst.

Komponente	Funktionen
Wrapper	<ul style="list-style-type: none"> ▪ Verbindungskontrolle ▪ Fehlermeldungen
Kontroll-Modul	<ul style="list-style-type: none"> ▪ Parametrierung des Zugriffs ▪ Benutzerdialoge ▪ Koordination
Kommunikationsmodul	<ul style="list-style-type: none"> ▪ Einstellung der Verbindungsparameter ▪ Verbindungsaufbau ▪ Sessionkontrolle ▪ AAA⁵⁶-Funktionalität
Multimedia-komponenten	<ul style="list-style-type: none"> ▪ Einstellen der Verbingungsparameter ▪ Verbindungsaufbau ▪ Aktivierung der verschiedenen Multimediakomponenten

Tabelle 4-4: Funktionen der Remote-Komponenten

⁵⁶ Authentifizierung, Autorisierung, Accounting sind die grundsätzlichen Aufgaben die eine Zugangskontrolle zu einem IT-System leisten muss [BÖHMER 02].

Authentifizierung: Feststellung der Identität eines Benutzers

Autorisierung: Überprüfung der Zugriffsrechte eines Benutzers

Accounting: Protokollierung der Zugriffe eines Benutzers

4.3.3 Kommunikationsmodul

Das Remote-Framework für die verteilte Kollaboration in den integrativen Phasen der Mechatronikentwicklung bietet zwei logische Kommunikationskanäle an: Den Kanal für die Datenkommunikation zwischen den Werkzeuginstanzen und den Kanal für die Multimediakommunikation (Kapitel 4.2.1). Die Art und Weise der Realisierung dieser Kanäle kann jedoch für den praktischen Einsatz nicht fest vorgeschrieben werden. Aufgrund unterschiedlicher Anforderungen bezüglich der IT-Sicherheit kommen unterschiedliche Firewall-Systeme⁵⁷ und Firewall-Konfigurationen zum Einsatz. In Abhängigkeit davon müssen für die Kommunikationsverbindung unterschiedliche Mechanismen eingesetzt werden um Netzwerkgrenzen zu passieren. Solche Mechanismen stellt das Kommunikationsmodul variabel zur Verfügung. Weiterhin verfügen unterschiedliche Kommunikationsressourcen, beispielsweise Firmennetzwerk oder Telefonleitung, über unterschiedlich hohe Bandbreiten. Es treten deshalb verschiedenste Fälle auf:

- Die Kanäle können sowohl logisch als auch physikalisch, entweder vereint oder getrennt realisiert werden.
- Die Kommunikationsverbindung kann auf verschiedenen Ebenen des ISO/OSI-Referenzmodells erfolgen.
- Verschiedene Abläufe für den Verbindungsaufbau kommen zum Einsatz.

Die Kapselung der verschiedenen Ausführungen ist Aufgabe des Kommunikationsmoduls. Die Komponente stellt, unabhängig von der unterlagerten Ausführung, immer die beiden logischen Kanäle für Daten- und Multimediakommunikation integriert zur Verfügung. Dieses Konzept sichert die Flexibilität einer verteilten Lösung für unterschiedlichste Anwendungsfälle und Kommunikationsinfrastrukturen.

Für beide Kanäle übernimmt das Kommunikationsmodul zudem die Aufgaben Verbindungsaufbau, Session-Kontrolle sowie die notwendige AAA-Funktionalität. Im Sinne der Anpassung an bestehende IT-Infrastrukturen werden verschiedene Verbindungsmodi für den Datenkanal angeboten.

4.3.4 Multimediakomponenten

Eine der Hauptanforderungen an die Multimediakomponenten ist die einfache Nutzbarkeit, angepasst an den jeweiligen Anwendungsfall (vgl. Kapitel 3.4). Um diese Anforderung vollständig zu erfüllen, ist eine möglichst enge Integration der Multifunktionalität in das verteilte Rechnerwerkzeug anzustreben. Dies ist jedoch ein unrealistisches Szenario, da nicht erwartet werden kann, dass Hersteller von Entwicklungswerkzeugen ihre Applikationen durchgehend mit

⁵⁷ Eine Firewall ist eine Schwelle zwischen zwei Netzen, die überwunden werden muss, um Systeme im jeweils anderen Netz zu erreichen [ELLERMANN 02].

einer solchen Funktionalität ausstatten. Umgekehrt entspricht es ebenfalls nicht den Anforderungen, dass ein Anwender das Setup und die Konfiguration einer eigenständigen Multimediaapplikation in Addition zur eingesetzten Applikation auszuführen hat.

Das Konzept sieht deshalb den Einsatz externer Multimediakomponenten in Form einer Bibliothek vor, die vollständig über das zentrale Steuerungsmodul in das Framework eingebunden ist. Sämtliche Funktionen zur Bedienung der Multimediakomponenten werden über dieses Modul angeboten. Hierzu gehören sämtliche Basisfunktionen wie Adresslisten, Verbindungsauf- und Abbau sowie Einstellung der Verbindungsparameter. Dagegen erfolgt die Nutzung der eigentlichen Multifunktionalität über die grafischen Schnittstellen der eingebundenen Multimediastsoftware. Hierzu zählen beispielsweise das Fenster für die Ausgabe des Videobildes, Fenster für Instant-Messaging oder Dateitransfer.

Ein solches Konzept ermöglicht die Verwendung unterschiedlicher verfügbarer Multimediälösungen in verteilten Werkzeugen, beispielsweise wenn diese beim Anwender bereits vorinstalliert sind oder firmeninterne Richtlinien bestehen. Für die Integration in das Werkzeug ist lediglich ein passender Adapter auf die verwendete Bibliothek erforderlich. Auf diese Weise wird ein Kompromiss erreicht, der einerseits den Bedürfnissen der Anwender Rechnung trägt und andererseits den Aufwand für die Implementierung eines verteilt nutzbaren Werkzeuges begrenzt.

4.4 Integration von Werkzeugen in das Rahmenkonzept

Kapitel 4.2.1 kommt zu dem Schluss, dass zur Erfüllung aller Anforderungen an eine effiziente verteilte Kollaboration in den integrativen Phasen, Anpassungen an den Entwicklungswerkzeugen nicht vollends vermeidbar sind. Es ist deshalb ein Konzept für die Integration bestehender Tools mit dem Remote-Framework notwendig. Dieses Konzept stellt die Grundlage für die Anpassung von Werkzeugapplikationen dar und gibt entsprechende Hilfestellungen.

4.4.1 Modulares GUI-Konzept

Um die Forderung nach rollenbezogenen Benutzerschnittstellen für die verteilten Instanzen eines Entwicklungswerkzeuges zu erfüllen, ist ein möglichst einfach zu realisierendes Konzept für unterschiedliche Benutzerinterfaces zu erstellen. Als effektives Mittel zur Reduzierung des Migrationsaufwands für verteilte Anwendungen wurde in Kapitel 4.2.1 ein symmetrischer Ansatz gewählt. Dies bedeutet, dass ein Rechnerwerkzeug als lokale und als entfernte Instanz agieren kann und nicht zwei verschiedene Installationen benötigt werden. In Verbindung mit der Forderung nach rollenbezogenen Benutzerschnittstellen führt dies zu einem modularen GUI-Konzept. Demnach stellt die Applikation GUI-Elemente mit gruppierter Funktionalität in Form von GUI-Modulen

zur Verfügung. Diese besitzen einheitliche Programmier-Schnittstellen und können innerhalb eines Rahmen-GUIs beliebig aktiviert werden. Auf diese Weise ist es einfach möglich, GUI-Sets entsprechend der gewünschten Benutzerrolle zu verwenden. Weiterhin ist es dem Benutzer möglich, flexibel eigene Sets zu erstellen und bei Bedarf zu erweitern bzw. zu reduzieren (vgl. Abbildung 4-9).

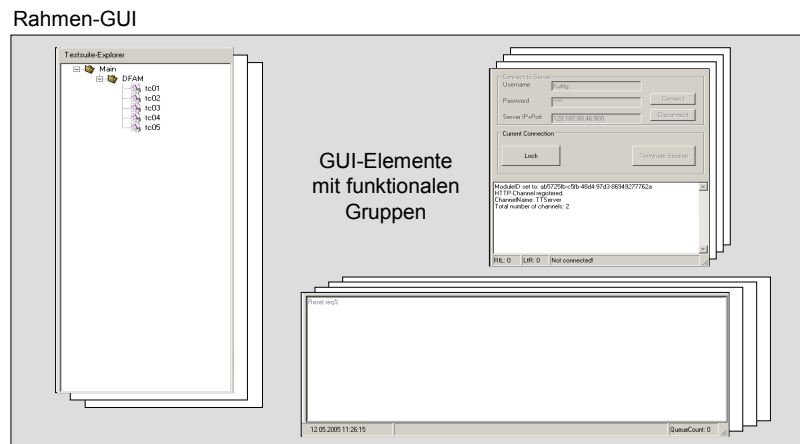


Abbildung 4-9: Modulare GUI-Elemente eines Rechnerwerkzeugs

Das modulare Konzept ermöglicht gleichfalls eine Erweiterung des Rollenumfangs, falls zukünftige Entwicklungsprojekte dies erfordern. Zudem wird dem Anwender eine weitere Granularisierung des abstrakten Rollenkonzepts auf einen konkreten Anwendungsbereich ermöglicht. Für den Tool-Hersteller reduziert sich der Aufwand indem nicht für jede benötigte Benutzerrolle ein vollständiges Benutzer-Interface zu erstellen ist. Die Spezifikation entsprechender Schnittstellen für GUI-Module erfolgt im weiteren Verlauf der Arbeit.

4.4.2 Integration des Remote-Frameworks mit Werkzeugapplikationen

Ein symmetrisches Konzept für die lokale und die entfernte Werkzeuginstanz erfordert Maßnahmen, um den Informationsfluss innerhalb der verteilten Applikation jeweils in die richtige Richtung zu leiten. Die Instanzen müssen zur Laufzeit zwischen den Modi lokaler Server und entfernter Client umschaltbar sein (siehe Abbildung 4-10). Das heißt der Informationsfluss erfolgt auf lokaler Seite direkt zwischen der Applikation und dem realen System und auf der Remote-Seite zwischen der entfernten und der lokalen Instanz.

Im Remote-Fall müssen Funktionsaufrufe zunächst über das Netzwerk an die lokale Instanz geleitet werden, erst nach Durchlaufen diverser Kontrollmechanismen erfolgt die Auslösung von Aktionen im realen System. Damit ein flexib-

les Routing möglich wird, existiert eine Proxy-Komponente⁵⁸ für die Funktionen des angeschlossenen mechatronischen Systems (System-Proxy).

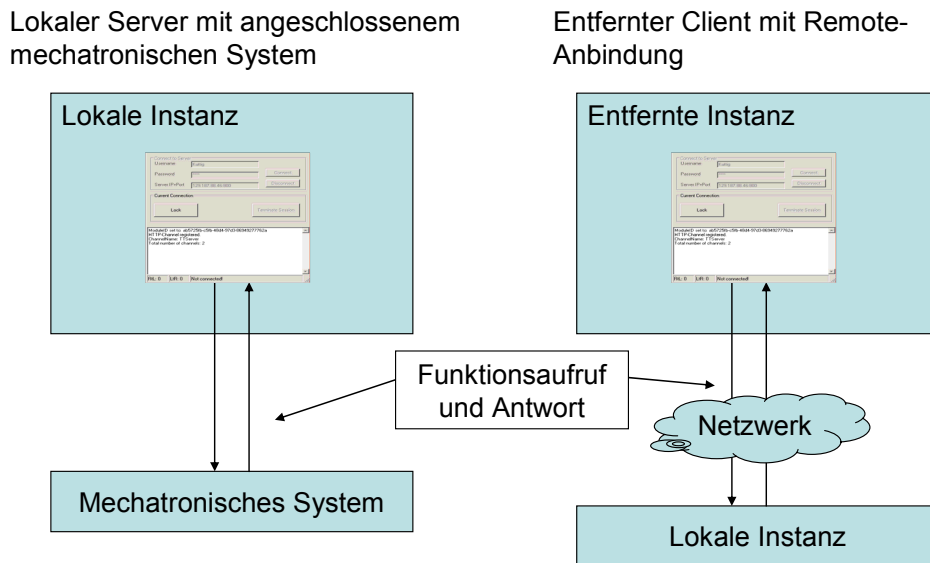


Abbildung 4-10: Die Betriebsmodi lokaler Server und entfernter Client

Diesem übergeordnet ist ein Dispatcher, der die Funktionsaufrufe der Benutzeroberfläche entgegen nimmt. Je nachdem ob es sich um die lokale- oder die Remote-Instanz handelt, werden Funktionsaufrufe entweder direkt an die Proxy-Komponente oder an das Wrapper-Modul der Remote-Komponenten weitergeleitet. Die Umschaltung zwischen Client- und Server-Modus erfolgt durch einfaches Verstellen eines Schnittstellenzeigers entweder auf den Systemproxy oder das Wrapper-Modul (siehe Abbildung 4-11).

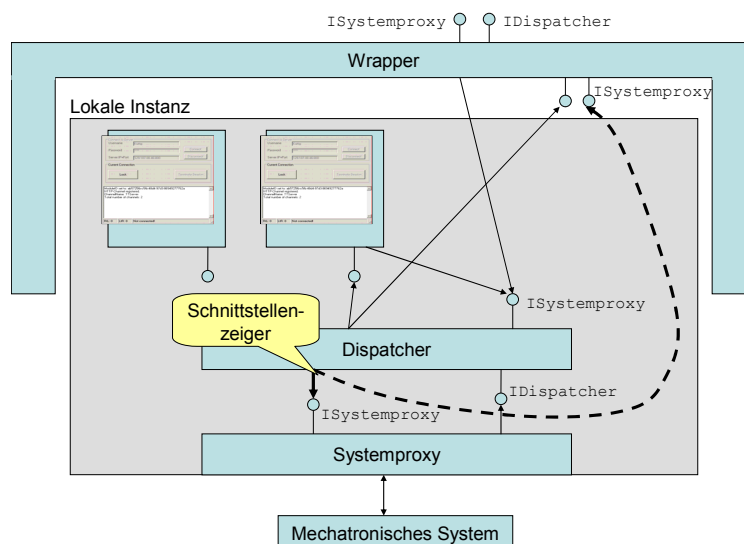


Abbildung 4-11: Konzept für flexibles Message-Routing auf lokaler- und Remote-Seite

⁵⁸ Stellvertreter-Komponente

In der Literatur wird ebenfalls zwischen symmetrischen und asymmetrischen Architekturen bezüglich der Datenhaltung in geografisch getrennten Applikationen unterschieden [HERRMANS 97]. Dabei geht es um die Fragestellung, ob alle Instanzen stets eine lokale Kopie des gesamten Datenhaushalts vorhalten und nur manipulierende Operationen austauschen oder ob die Datenhaltung zentral an einer Stelle erfolgt. Bedingt durch das zu Grunde liegende Szenario (vgl. Kapitel 4.1.1) scheint auf den ersten Blick eine asymmetrische Datenhaltung zwangsweise vorzuliegen: Die maßgeblichen Daten liegen innerhalb des physikalischen Systems und stehen den angeschlossenen Rechnerwerkzeugen zur Verfügung. Die Manipulation von Daten erfolgt letztendlich durch den Zugriff auf den Datenhaushalt des Systems.

Eine genauere Betrachtung der im Rahmen der verteilten Kollaboration bearbeiteten Daten offenbart jedoch zwei grundsätzliche Typen:

- Interne Daten des physikalischen Systems (z. B. Prozessgrößen), hier herrschen strikte Konsistenzbedingungen.
- Daten des Werkzeuges (z. B. Simulationsmodelle, Testfälle, Protokollaufzeichnungen), diese Daten können sich zwischen den verteilten Instanzen unterscheiden.

Für die Klasse der Werkzeugdaten ist im vorliegenden Konzept die Möglichkeit einer symmetrischen Datenhaltung vorgesehen. Im Gegensatz zu den allgemeingültigen Definitionen wird hier jedoch keine Konsistenzanforderung gestellt. Dies bringt folgende Vorteile mit sich:

- Nicht-globale Zustände können auf beiden Seiten gespeichert und bearbeitet werden.
- Offline-Arbeit ist möglich, d. h. eine Instanz kann als Stand-Alone-Werkzeug eingesetzt werden, beispielsweise um vorbereitende Aktionen durchzuführen.

Durch die Möglichkeit der nicht zwangsweise konsistenten symmetrischen Datenhaltung ergibt sich für die Nutzer die Möglichkeit, mit eigenen Datensätzen zu arbeiten, die jeweils für die andere Seite verborgen bleiben. Auf diese Weise wird Know-how-Schutz gewährleistet. Bei Bedarf können die Werkzeugdaten synchronisiert werden.

4.4.3 IT-Sicherheitsstrategien

Die sichere Übertragung der während der Kollaboration anfallenden Daten auch über öffentliche Netzwerke ist eine grundlegende Anforderung an ein verteiltes Werkzeug. Nicht nur, dass sensible Daten und Informationen Gefahr laufen in die Hände unberechtigter Dritter zu fallen. Beim Zugriff auf reale physikalische Systeme kann nicht geduldet werden, dass unberechtigte Personen die Kommunikation beeinflussen und so massive Schäden verursachen.

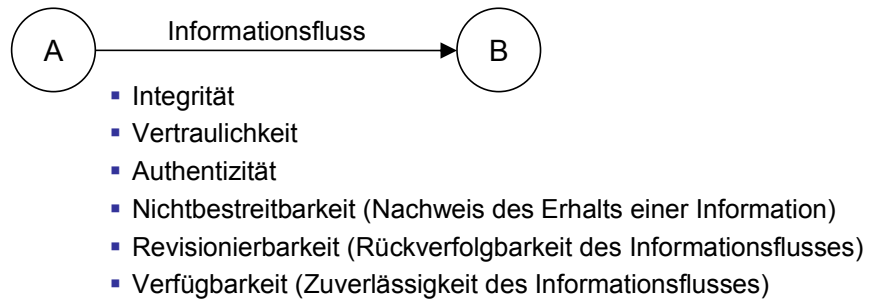


Abbildung 4-12: Anforderungen an sichere Kommunikation, nach [POHLMANN 00]

Abbildung 4-12 fasst die Anforderungen an die sichere Informationsübermittlung von einem Teilnehmer A zu einem Teilnehmer B zusammen. Diese Anforderungen müssen von einer Sicherheitskomponente erfüllt werden. Die Absicherung der Kommunikation zwischen verteilten Werkzeuginstanzen kann prinzipiell auf unterschiedlichen Ebenen des ISO/OSI-Referenzmodells erfolgen. Nach aktuellem Stand der Technik haben sich zwei Lösungsstrategien etabliert:

1. Sicherheitsfunktionen werden in das Rechnerwerkzeug integriert, ein- bzw. ausgehende Daten sind bereits abgesichert (Technologiebeispiel: SSL⁵⁹).
2. Die Absicherung der Daten erfolgt außerhalb des Rechnerwerkzeugs, entweder durch ein externes Softwaremodul oder eine externe Hardwarekomponente. In diesem Fall verarbeiten die Applikationen ungesicherte Daten, die Absicherung durch externe Komponenten erscheint transparent (Technologiebeispiel: IPsec⁶⁰).

Im hier vorgestellten Lösungskonzept wird das Prinzip einer externen Absicherung verfolgt. Die Gründe hierfür ergeben sich wie folgt und basieren auf Analysen einschlägiger Literatur wie [BÖHMER 02], [BUSCH 02], [POHLMANN 00]:

- Bei einer Vielzahl unterschiedlicher eingesetzter Applikationen ist die Flexibilität nur dann gewährleistet, wenn die Absicherung transparent erfolgt und nicht in jeder der Applikationen eigens implementiert werden muss.
- Steht bereits eine Sicherheitsinfrastruktur beispielsweise innerhalb des unternehmensweiten IT-Konzepts zur Verfügung, kann diese aufwandsarm genutzt werden.

⁵⁹ *Secure Socket Layer* (Absicherung auf Schicht vier des ISO/OSI-Modells, Sicherheitsfunktionen sind innerhalb des TCP/IP-Stacks realisiert) [NETSCAPE 98].

⁶⁰ *IP Secure* (Die Absicherung der Daten erfolgt auf Schicht 3 innerhalb des IP-Protokolls) [BÖHMER 02].

- Die Nutzung externer Sicherheitskomponenten in einer vorhandenen Infrastruktur entbindet den Anwender einer verteilten Applikation von der Handhabung einer applikationseigenen Sicherheitseinrichtung.
- Die Administration und Wartung der Sicherheitskomponente kann zentral vorgenommen werden, es muss nicht jede Werkzeuginstallation einzeln gepflegt werden.

4.4.4 Migrationsstrategie für die Remote-Erweiterung

Ein wichtiges Kriterium für das Rahmenkonzept ist die möglichst einfache Integration existierender Werkzeuge in das Remote-Framework. Es stellt sich die Frage: *Wie können existierende Werkzeuge schnell und unkompliziert mit dem Remote-Framework verbunden werden?*. Um allen Anforderungen gerecht zu werden, sind Modifikationen an bestehenden Tools unvermeidbar. Dieser Ansatz unterscheidet sich von bestehenden CSCW- und Kooperationslösungen (beispielsweise Desktop-Sharing). Ziel ist es jedoch, die Modifikationen möglichst gering zu halten, hierfür steht das Wrapper-Modul als zentraler Integrationspunkt zur Verfügung. Seitens der Applikation muss eine Verbindung mit dieser Komponente realisiert werden, dies ermöglicht das zuvor eingeführte Konzept eines zentralen Dispatchers. Bei der Ausstattung eines Tools mit Remote-Funktionalität gemäß der hier vorgestellten Lösung müssen sechs wesentliche Schritte durchlaufen werden. Eine Übersicht ist in Abbildung 4-13 dargestellt.

Der auf den ersten Blick umfangreich erscheinende Aufwand für den Tool-Hersteller reduziert sich angesichts der Tatsache, dass viele der in den Einzelschritten geforderten Voraussetzungen mit den Anforderungen an ein gutes Softwaredesign korrespondieren. Hierzu zählen beispielsweise modular strukturierte GUIs, objektorientierte Architekturen und eine saubere Trennung zwischen GUI und Ablaufsystem. Die elementaren Voraussetzungen für eine Remote-Erweiterung gemäß dem hier vorgestellten Konzept können demzufolge erfüllt werden, wenn bei der Neukonzeption eines Werkzeugs entsprechender Wert auf gutes Applikationsdesign gelegt wird. In diesem Fall ist für den Tool-Hersteller im Wesentlichen nur noch eine Anpassung der Schnittstellen an die für das Remote-Framework spezifizierten Interfaces vorzunehmen, um den Wunsch des Kunden nach einfacher Integration zu erfüllen.

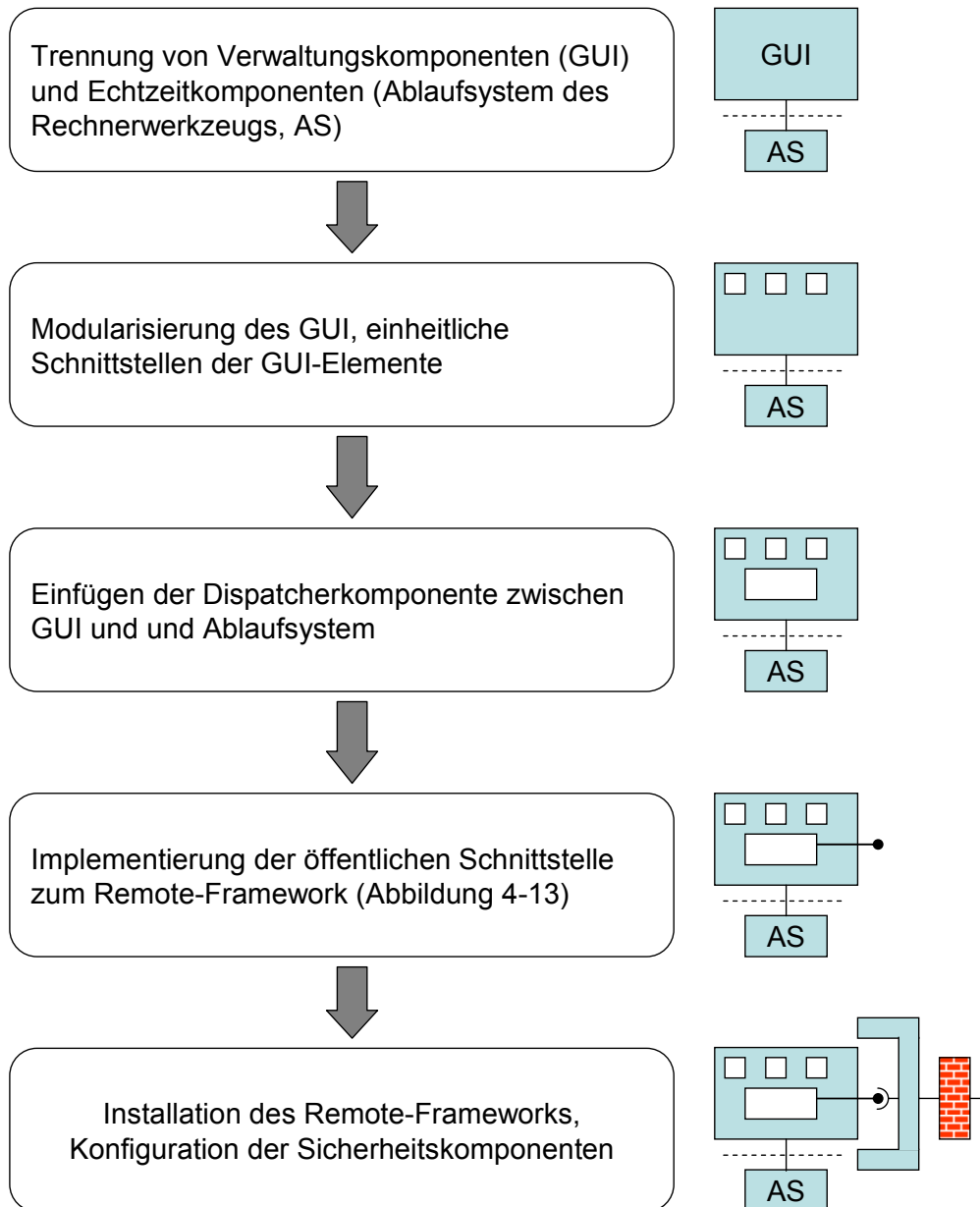


Abbildung 4-13: Vorgehensweise zur Realisierung eines verteilten Werkzeugs

4.5 Zusammenfassung

In diesem Kapitel wurde ein umfassendes Rahmenkonzept für die verteilte Kollaboration in den integrativen Phasen der Mechatronikentwicklung entworfen. Die Kollaboration umfasst dabei die Zusammenarbeit geografisch getrennter Experten unter Einbeziehung realer mechatronischer Systeme und unter Verwendung komplexer Entwicklungstools.

Als Ausgangspunkt erfolgte zunächst die genaue Identifikation des Kommunikationsszenarios, welches durch die Anwendungsfälle in den integrativen Phasen bestimmt ist. Das Rahmenkonzept ist so ausgelegt, dass es maßgerecht auf

dieses Szenario zugeschnitten ist und zugleich den identifizierten Anforderungen entspricht. Hierfür wurde ein dem Szenario angepasstes kollaboratives Setting ausgewählt und Lösungsansätze vorgestellt, die den identifizierten Anforderungsklassen entsprechen.

Das Lösungskonzept basiert auf der in Kapitel 3 entwickelten Idee einer einheitlichen Schnittstelle, die mittels einer Wrapper-Komponente die Verbindung zwischen dem Entwicklungswerkzeug und einem Remote-Framework herstellt. Auf diese Weise muss das Werkzeug die Kollaborationsfunktionalität nicht selbst implementieren. Das Lösungskonzept beinhaltet sämtliche Funktionen, die für eine verteilte Kollaboration gemäß dem identifizierten Szenario benötigt werden. Hierzu gehören:

- Komponenten für die multimediale Kommunikation
- Mechanismen zur Zugriffsbeschränkung der Interaktionen von entfernten Benutzern
- Tracking- und Tracing-Funktionalität
- Abwicklung der Kommunikation mit entfernten Instanzen
- Verbindungsaufbau, Benutzerkontrolle und Session-Management
- IT-Sicherheitsmaßnahmen
- Eine zentrale Komponente zur Handhabung der Remote-Funktionalität
- Konzept zur Einstufung von Werkzeugfunktionalität in Funktionsklassen

Sämtliche Anforderungen an Effizienz und Wirtschaftlichkeit können nur dann erfüllt werden, wenn auch das eingesetzte Rechnerwerkzeug entsprechende Anpassungen beinhaltet. Dies betrifft zum einen das Benutzeroberfläche, um der Anforderung nach Rollenorientierung gerecht zu werden. Zum anderen muss die Schnittstelle vom Remote-Framework zur eigentlichen Toolfunktionalität so ausgelegt sein, dass verschiedene Werkzeuge flexibel und einfach zu integrieren sind. Ein solcher Ansatz führt zwar zu Aufwand auf Seiten des Toolsherstellers, beseitigt jedoch die Mängel existierender Remote- und CSCW-Lösungen (vgl. Kapitel 3). Um den Aufwand möglichst gering zu halten, beinhaltet das Konzept ein Lösungsmuster als Basis für die Gestaltung und Ausführung der notwendigen Modifikationen. Dieses besteht in Form einer symmetrischen Architektur interner Schnittstellen und der Definition angepasster Benutzerrollen, die mittels eines modularen GUI realisiert werden.

Im weiteren Verlauf der vorliegenden Arbeit erfolgt die Entwicklung des universellen Remote-Frameworks. Darin enthalten sind Maßnahmen und Lösungsmuster zur Realisierung verteilter Kollaboration auf Basis des vorliegenden Szenarios.

5 Universelles Remote-Framework

Nachdem in Kapitel 4 ein umfassendes Rahmenkonzept für die verteilte Kollaboration in den integrativen Phasen der Mechatronikentwicklung entworfen wurde, ist es Ziel dieses Kapitels, eine Spezifikation für das Remote-Framework zu erarbeiten. Es wurde bereits mehrfach betont, dass nur eine einfache Benutzbarkeit eines solchen Frameworks die Akzeptanz bei den Anwendern sichert. Gleichfalls muss für einen Tool-Hersteller der Aufwand für die Integration eines konkreten Rechnerwerkzeugs so gering wie möglich bleiben. Um diese Anforderungen zu erfüllen, werden neben der Spezifikation für das Framework auch Lösungsmuster bezüglich der einfachen Anwendbarkeit erarbeitet. Kernpunkt ist ein Klassifikationsansatz, mit dessen Hilfe der Anwender Werkzeugfunktionen entsprechend des Gefährdungspotentials in Zugriffsklassen einordnen kann. Für die Umsetzung des werkzeugseitigen Teils des Rahmenkonzepts stehen entsprechende Basisarchitekturen und Schnittstellenkonzepte zur Verfügung.

5.1 Gestuftes Zugriffskonzept für entfernte Interaktionen

Im Verlauf der bisherigen Ausführungen wurde bereits mehrmals die Bedeutung eines gestuften Zugriffs auf die Funktionen und den Datenhaushalt eines realen Systems von entfernten Standorten aus hervorgehoben. Um diese Anforderung zu erfüllen, beinhaltet das konzipierte Remote-Framework eine Komponente, die eine solche Zugriffskontrolle übernimmt und dadurch die Werkzeugapplikation von dieser Aufgabe entbindet. Da eine möglichst breite Abdeckung der Anwendungsgebiete angestrebt wird ist es nicht möglich, eine allgemeingültige Sicherheitsklassifizierung sämtlicher theoretisch möglicher Funktionen vorzunehmen und in die Zugriffskomponente einzuarbeiten. Zumal eine Sicherheitsklassifizierung in hohem Maße von den Randbedingungen abhängig ist, unter denen ein Fernzugriff auf ein mechatronisches System erfolgt (beispielsweise kann ein Prüfstand über besondere Absicherungsmaßnahmen verfügen). Es wird deshalb der Ansatz gewählt, die Einordnung von spezifischen Funktionen in Zugriffsklassen dem Anwender zu überlassen und ihm hierfür ein Klassifizierungsschema an die Hand zu geben. Die Entwicklung eines solchen Schemas erfolgt im Verlauf dieses Kapitels.

5.1.1 Klassifikationsansatz nach IEC 61508

Vor der Entwicklung eines eigenen Klassifizierungsschemas für sicherheitskritische Funktionen, ist eine Auseinandersetzung mit der international etablierten Sicherheitsnorm IEC 61508⁶¹ [IEC 05] unumgänglich. Die Norm dient weltweit als Grundnorm für die Spezifikation, den Entwurf und den Betrieb von elektrischen, elektronischen und programmierbaren elektronischen sicherheitstechnischen Systemen. Eine weitere Ausprägung findet sich in Form der Norm IEC 61511 für die Prozessindustrie [INGREY+ 04], [SMITH+ 01]. Kernpunkt der IEC 61508 ist die Abschätzung des Risikos beim Betrieb sicherheitsrelevanter Einrichtungen, hierbei fließen mehrere Risikoparameter wie beispielsweise das Schadensausmaß ein. Anhand der Risikoabschätzung wird der so genannte Sicherheits-Integritätslevel (Kurzbezeichnung SIL) für Komponenten und Systeme mit Sicherheitsaufgaben ermittelt, hierfür werden vier SIL-Klassen definiert. Erfüllen Komponenten die Anforderungen einer SIL-Klasse, besitzen sie eine entsprechend hohe Zuverlässigkeit für die Erfüllung der sicherheitsrelevanten Funktionen. Solche Geräte verfügen über Zertifikate, welche die Konformität zu einer SIL-Klasse bescheinigen.

Die vorgestellte IEC-Norm kann nicht als unmittelbare Grundlage für das benötigte Klassifizierungsschema verwendet werden. Der Fokus der SIL-Klassifizierung liegt auf der Sicherstellung von Zuverlässigkeit sicherheitsrelevanter Komponenten. Der vorliegende Anwendungsfall verlangt jedoch nach einer Einschränkung des Zugriffs auf Funktionalität in Abhängigkeit vom Gefahrenrisiko bei entfernten Interaktionen. Dennoch können vor allem die Verfahren zur Risikoermittlung als Grundlage herangezogen werden. Anstelle der Bestimmung von SIL-Klassen für risikobehaftete Anwendungen erfolgt die Bestimmung der Zugriffsklasse für risikobehaftete remote ausführbare Funktionen.

Die Norm 61508 unterscheidet grundsätzlich zwei Verfahren zur Bestimmung von Risikoklassen: quantitative und qualitative Verfahren [SMITH+ 01]. Die quantitative Vorgehensweise wird für die vorliegende Problemstellung als nicht adäquat erachtet. Da es sich in den integrativen Phasen des Entwicklungsprozesses nicht um marktreife mechatronische Geräte sondern um Systeme im Prototypen- bzw. Versuchsstadium handelt, ist eine Einordnung in Sicherheitsklassen auf quantitativer Basis für die Praxis untauglich. Vielmehr werden in diesen Phasen zwangsläufig Risiken für Fehlfunktionen in Kauf genommen, die beispielsweise durch Maßnahmen des Entwicklers vor Ort oder sonstige spezielle Absicherungsmaßnahmen abgefangen werden. Diese Randbedingungen lassen sich nur schwer in Zahlen fassen. In Abhängigkeit von der speziellen Situation vor Ort und der dort gültigen Richtlinien muss der lokale Entwickler im speziellen Fall abwägen, ob eine Freigabe kritischer Funktionen für den entfernten

⁶¹ In Europa ebenfalls verabschiedet als EN 61508, in Deutschland auch als VDE 0803 veröffentlicht [GALL 03].

Kollaborationspartner vertretbar ist, oder eingeschränkt werden muss. Für diese Aufgabe erscheinen qualitative Ansätze besser geeignet, auch wenn einzelnen Entscheidungen durchaus auch quantitative Aussagen zugrunde liegen können.

Zwei in der IEC-Norm vorgeschlagene und häufig verbreitete qualitative Verfahren sind nach SMITH die Risikomatrix-Analyse sowie die Risikograph-Analyse. Die Risikomatrix-Analyse beruht im Wesentlichen auf der Bestimmung einer Häufigkeit für das Auftreten von Fehlern. Wesentlich geeigneter erscheint deshalb die Risikograph-Analyse da hier der Einfluss der Fehlerhäufigkeit leicht zu eliminieren ist. Abbildung 5-1 zeigt die Bestimmung von SIL-Klassen mittels der Risikograph-Analyse:

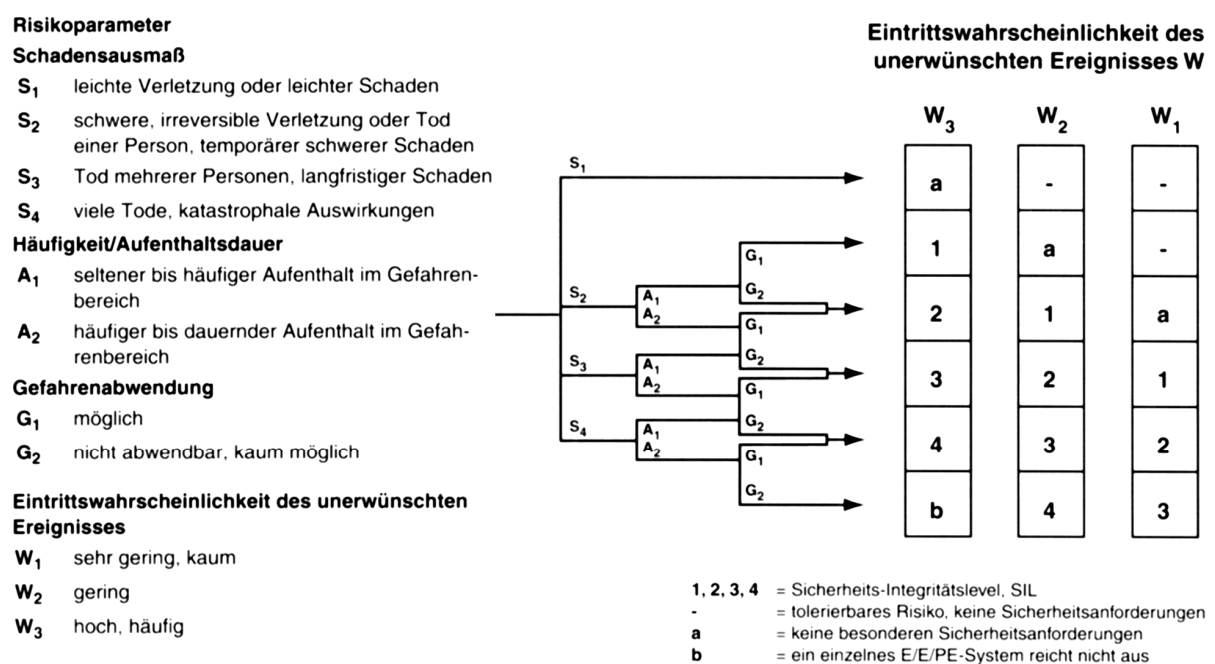


Abbildung 5-1: Risikograph-Analyse nach IEC 61508 [INGREY+ 04]

Die Grafik zeigt, dass der Einfluss der Fehlerhäufigkeit entfällt, wenn die Auswahlspalten im rechten Bereich auf eine Spalte reduziert werden. Diese Vorgehensweise wird zur Ermittlung der Funktionszuordnung in Zugriffsklassen für die vorliegende Problemstellung adaptiert.

5.1.2 Definition von Zugriffsklassen

In einem ersten Schritt erfolgt in diesem Unterabschnitt die Definition geeigneter Klassen als Ausgangsbasis für einen gestuften Zugriff auf die Funktionen und Daten von kollaborativen Werkzeugen in den integrativen Phasen. Die Ergebnisse der Analysen in Abschnitt 3 zeigen einen Mangel an solcher Funktionalität bei den nach heutigem Stand der Technik existenten Technologien und Tools für die verteilte Kooperation. Durch die in diesem Abschnitt entwickelte Vorgehensweise wird insbesondere der Anforderung A 2.2 gemäß Kapitel 3.1.3

Rechnung getragen. Abbildung 5-2 veranschaulicht das verfolgte Klassifizierungskonzept: Sämtliche Remote verfügbaren Funktionen werden vom Anwender in definierte Klassen eingeordnet und bei entfernten Interaktionen durch die Kontrollmechanismen des Remote-Frameworks entsprechend behandelt.

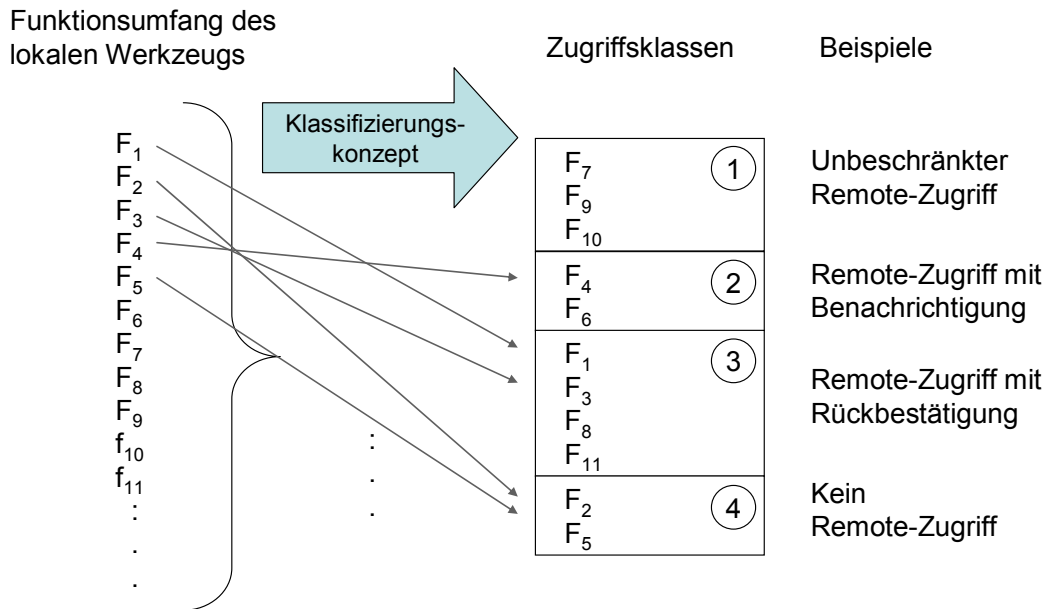


Abbildung 5-2: Einordnung der Werkzeugfunktionalität in Zugriffsklassen

In Zusammenhang mit der Definition von Funktionsklassen ist zunächst festzulegen, welche Sicherheitsziele verfolgt werden. Es wäre verfehlt, durch die Klassifizierung der Remote ausführbaren Funktionen ein Gesamtsystem zu erzielen, das den Anforderungen an ein fehlersicheres System entspricht. Allein aufgrund der Unabwägbarkeiten, die bei einer Datenübermittlung über öffentliche Netze vorhanden sind, ist ein solches Ziel mit angemessenem Aufwand nicht erreichbar. Der remote verfügbare Funktionsumfang müsste auf ein äußerst beschränktes Maß reduziert werden, was effiziente Kollaboration verhindert. Es wird deshalb von einem Werkzeug-System ausgegangen, welches der lokale Anwender entsprechend der Situation vor Ort abgesichert hat. Hierbei kommen individuelle Vorschriften und Richtlinien zur Anwendung. Die im Zusammenhang mit dem Remote-Zugriff definierten Klassen verfolgen das Ziel, lokal erreichte Sicherheitsstufen nicht durch die Remote-Erweiterung zu verletzen. Dies bedeutet vor allem: der lokale Anwender muss in letzter Instanz die Kontrolle über sämtliche kritischen Funktionen besitzen, die für die Remote-Ausführung zur Verfügung stehen. Die entwickelte Klassifizierung beantwortet also nicht die Frage, ob die Ausführung einer Funktion an sich als sicher oder unsicher zu bewerten ist, sondern ob die Remote-Verfügbarkeit der Funktion beschränkt oder ganz verhindert werden muss.

Unter diesen Gesichtspunkten werden die Zugriffsklassen gemäß Tabelle 5-1 definiert, Erklärungen zu den Klassen finden sich in Anhang C.

Bezeichner	Eigenschaft
1	Uneingeschränkter Remote-Zugriff
2a	Lesender Zugriff mit Benachrichtigung des lokalen Entwicklers
2b	Lesender Zugriff erst nach Freigabe durch den lokalen Entwickler
3a	Schreibender Zugriff mit Benachrichtigung des lokalen Entwicklers
3b	Schreibender Zugriff nach Freigabe durch den lokalen Entwickler
3c	Schreibender Zugriff nach Freigabe durch den lokalen Entwickler für jedem Einzelschritt
4a	Schreibender Zugriff nach zusätzlicher Freigabe durch den Bediener des Systems vor Ort
4b	Schreibender Zugriff nach zusätzlicher Freigabe durch den Bediener des Systems vor Ort für jedem Einzelschritt
5	Kein Remote-Zugriff

Tabelle 5-1: Funktionsklassen für die Beschränkung des Remote-Zugriffs

Neben der Möglichkeit zur Beschränkung des Remote-Zugriffs auf Funktionen ist in den Anforderungen gleichfalls die Möglichkeit der Beschränkung des Zugriffs auf den Datenhaushalt einer Werkzeuginstanz festgeschrieben. Demzufolge reicht die Klassifizierung der Funktionen allein nicht aus, sondern es muss ebenfalls die Klassifizierung kritischer Funktionsparameter möglich sein, um diese in der Zugriffskontrolle entsprechend zu behandeln. Diesem Anspruch trägt das Konzept ebenfalls Rechnung. Eine ähnliche Problemstellung beinhalten auch Funktionen zum Aufruf von Eingabedialogen auf Seiten des entfernten Benutzers. Auch hier ist es erforderlich, eine Zugriffskontrolle der Rückgabewerte vorzunehmen, bevor diese an das ausführende System weitergereicht werden.

Als ebenfalls bedeutender Punkt ist an dieser Stelle die Protokollierung (Logging) sämtlicher entfernter Benutzerinteraktionen zu nennen (Tracing-Funktionalität). Das Logging ist jedoch nicht zu verwechseln mit der expliziten Information des lokalen Benutzers über eine entfernte Interaktion. Ein Logging-Konzept wird unabhängig von den definierten Zugriffsklassen erarbeitet.

5.1.3 Zuordnungssystematik

Die Systematik zur Einordnung von remote ausführbaren Werkzeugfunktionen in Zugriffsklassen basiert auf der Vorgehensweise in IEC 61508 in Form eines Risikographen. Mit Hilfe dieser Systematik wird der Anwender bei der Einordnung der Funktionen unterstützt. Notwendige Einschränkungmaßnahmen leiten sich dann direkt aus den Zugriffsklassen ab und werden vom Remote-Framework durchgeführt.

Für die Definition einer Einordnungssystematik erfolgt zunächst die Betrachtung der Stationen, die ein Funktionsaufruf eines entfernten Experten passiert. Zu diesem Zweck ist eine abstrakte Darstellung nach Abbildung 5-3 gegeben.

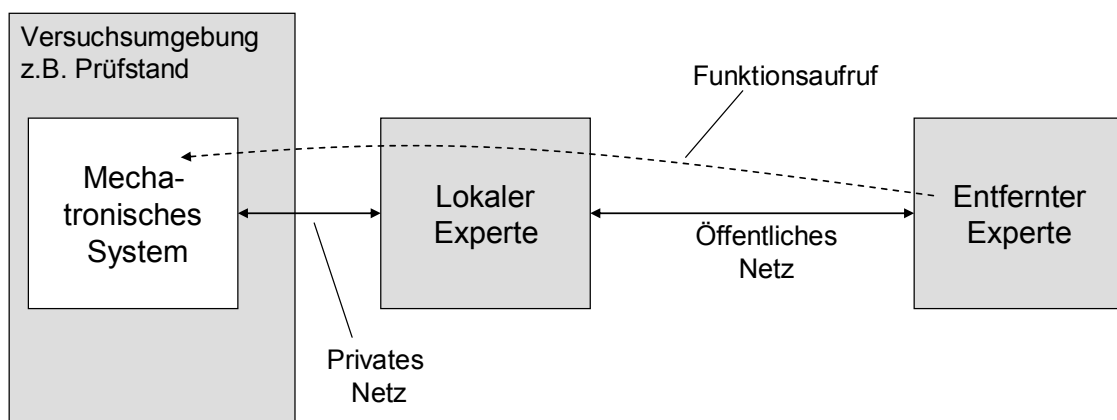


Abbildung 5-3: Stationen eines Funktionsaufrufs

Von der entfernten Werkzeuginstanz wird ein Funktionsaufruf in der Regel über ein öffentliches Netz an die lokale Instanz gesendet, die sich nicht zwingendermaßen zusammen mit dem realen System an einem gemeinsamen Ort befindet. Die lokale Instanz stößt die eigentliche Ausführung der Funktion im realen System an, dieses kann sich beispielsweise in einer abgeschlossenen Versuchsumgebung befinden, zu der ein lokaler Experte keine unmittelbare Verbindung hat. In diesem Fall können weitere Personen vor Ort sein, die das System betreuen, beobachtende Maßnahmen oder auch Absicherungsmaßnahmen durchführen. Der lokale Experte ist auf die Lagebeurteilung dieser Beteiligten angewiesen. Allgemeingültig wird also davon ausgegangen, dass neben den beiden kollaborierenden Experten noch ein oder mehrere Beteiligte in die Kollaboration einbezogen sind und Aufgaben übernehmen, deren Wahrnehmung dem lokalen Werkzeugbenutzer nicht möglich ist. Dieses, im Vergleich zu Abschnitt 4.1.1 verschärfte Szenario wird für die Herleitung der Einordnungssystematik zu Grunde gelegt. Entfallen diese besonderen Randbedingungen, führt die Systematik zu einem weniger restriktiven Ergebnis.

Im Folgenden werden diejenigen Risikoparameter definiert, die maßgeblich für die Einordnung von remote verfügbaren Funktionen in die definierten Zugriffsklassen sind:

- **Zugriffsart (Z):**

Die grundlegende Entscheidung besteht darin, ob eine Funktion schreibenden oder nur lesenden Zugriff auf das reale System vornimmt. Die Ausführung einer Nur-Lese-Funktion ist insofern kritisch, als dass vertrauliche Daten unberechtigt ausgelesen werden können. Bei der Ausführung schreibender Funktionen ist stets davon auszugehen, dass diese Aktionen im realen System auslösen.

Beispiele für die Beurteilung:

Z₁ Lesen eines Messprotokolls

Z₂ Ausführen eines Verfah-Befehls

- **Kritikalität (K):**

Dieser Parameter bezeichnet das Risiko, welches mit der Ausführung einer Funktion verbunden ist. Bei lesenden Funktionen betrifft dies die Intensität der Geheimhaltungsstufe, bei schreibenden Funktionen die Gefahren, die vom realen System ausgehen. Es wird empfohlen, für die Beurteilung dieses Parameters die Norm IEC 61508 zu Grunde zu legen.

Beispiele für die Beurteilung:

K₁ Es wird keine mechanische Aktion ausgelöst

K₂ Mechanische Aktionen können ausgelöst werden diese sind jedoch nicht mit Gefahren verbunden. Es ist ausreichend, die Auswirkungen durch den lokalen Experten zu beurteilen.

K₃ Gefährliche mechanische Aktionen werden ausgelöst. Vor Ort am System sind deshalb besondere Absicherungsmaßnahmen zu ergreifen.

K₄ Die Auswirkungen beherbergen ein solches Gefahrenpotential, dass ein Remote-Zugriff nicht zu verantworten ist.

- **Abschätzungsfähigkeit (A):**

Die Fähigkeit, die Auswirkungen eines entfernten Funktionsaufrufs abschätzen zu können, dient als weiterer Risikoparameter. Es wird beurteilt, inwieweit die Konsequenzen eines Funktionsablaufs überhaupt vorausgesagt werden können, oder ob dies nur für Einzelschritte möglich ist. In letzterem Fall darf die Funktion nur im Schrittbetrieb zur Ausführung gelangen. Ist die Konsequenz überhaupt nicht abschätzbar, erfolgt keine Freigabe für die Remote-Ausführung.

Beispiele für die Beurteilung:

- A₁ Zugriff auf offen gelegte Daten
- A₂ Zugriff auf geheime Daten, Setzen eines Parameters
- A₃ Änderung von Parametern während des Funktionsablaufs
- A₄ Unüberschaubare Verzweigungsvielfalt im Funktionsablauf

Die Einordnungssystematik ergibt sich anhand der definierten Risikoparameter und des in Abbildung 5-4 dargestellten Risikographen.

Risikoparameter bei Ausführung einer Funktion am realen System

Zugriffsart:

- Z₁ lesend
- Z₂ schreibend

Kritikalität:

- K₁ nicht kritisch, remote beurteilbar
- K₂ nicht kritisch, lokal beurteilbar
- K₃ kritisch, nur vor Ort am System beurteilbar
- K₄ unverhältnismäßig kritisch

Abschätzungsfähigkeit der Auswirkungen:

- A₁ Keine nennenswerten Auswirkungen
- A₂ Abschätzung der gesamten Auswirkungen einer Funktion möglich
- A₃ Abschätzung der Auswirkungen eines Einzelschrittes möglich
- A₄ Abschätzung der Auswirkungen nicht möglich

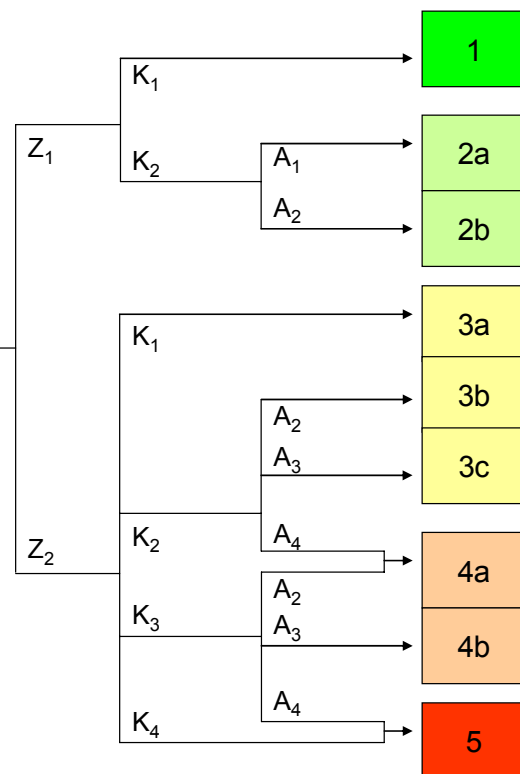


Abbildung 5-4: Risikograph für die Einordnung von Funktionen in Zugriffsklassen

Für die Durchführung der Klassifizierung sind folgende Anmerkungen von Bedeutung:

- Die Klassifizierung ist unabhängig vom Wissensstand bzw. der Rolle des Remote-Experten, da solche Gefahren beurteilt werden, die allein aus dem Szenario einer entfernt ausgelösten Interaktion entstehen. Diese Gefahren existieren unabhängig vom Wissensstand des entfernten Kollaborationspartners. Zudem ist davon auszugehen, dass dem entfernten Partner mit verhältnismäßigen Kollaborationsmitteln nie ein gleichwertiges Bild von der Situation vor Ort zur Verfügung steht wie einem lokalen Beteiligten.

- Die definierten Klassen sind als Mindestanforderung zu verstehen. Falls dies für spezielle Anwendungsfälle erachtet wird, kann stets eine restriktivere Zugriffsklasse gewählt werden.

5.1.4 Modellierung des Klassifikationskonzepts

Als Mittel zur Modellierung der Zuordnungssystematik innerhalb des Remote-Frameworks wird die deklarative Metasprache XML gewählt. XML hat insbesondere in Verbindung mit dem Einsatz von Internet-Technologien aber im speziellen auch in automatisierungstechnischen Anwendungen (vgl. Kapitel 3.2.1) weite Verbreitung erfahren. Aufgrund der vielen Vorteile wird die Sprache auch für den vorliegenden Anwendungsfall als das geeignete Mittel beurteilt. Im Folgenden erfolgt deshalb eine kurze Vorstellung von XML.

Die Sprache XML wurde im Jahr 1998 vom W3C⁶² als vereinfachte Untermenge der Auszeichnungssprache SGML⁶³ spezifiziert und dient zur Beschreibung, Strukturierung und Transformation beliebiger Daten. Der Vorteil von XML liegt dabei in der strikten Trennung von Datenstrukturen und Formatierungen. Die Struktur der Daten in einem XML-Dokument lässt sich mit Hilfe so genannter XML-Schemata oder DTD⁶⁴ präzise vorgeben, wobei sich in der Vergangenheit die Verwendung von Schemata aufgrund der erweiterten Möglichkeiten durchgesetzt hat. Für die Formatierung der Inhalte von XML-Dokumenten steht die XSL⁶⁵-Technologie zur Verfügung. Mit Hilfe so genannter Style-Sheets⁶⁶ kann festgelegt werden, in welcher Form die Ausgabe eines XML-Dokuments erfolgt. Ein weiterer Vorteil von XML ist die Vielzahl an oftmals frei verfügbaren Werkzeugen, die für die Arbeit mit Schemata, XML-Dokumenten und Style-Sheets zur Verfügung stehen. Aufgrund der mittlerweile weiten Verbreitung in vielen Anwendungsbereichen wird auf weitere Ausführungen über XML an dieser Stelle verzichtet und stattdessen auf weiterführende Literatur wie beispielsweise [LOBIN 01] oder [W3C 00] verwiesen.

Nachfolgend werden die wichtigsten Grundelemente des XML-Schemas zur formalen Definition einer Funktionsklassifizierung vorgestellt. Das vollständige Schema ist in Anhang E abgedruckt. Ebenso findet sich dort eine Beispieldatei für eine Funktionsklassifizierung.

⁶² World Wide Web Consortium

⁶³ Standardized Generalized Markup Language

⁶⁴ Document Type Definition

⁶⁵ eXtensible Style Sheet

⁶⁶ Übersetzt: Formatvorlagen

```

<!-- Declaration of Document Element -->
<xs:element name="CLASSIFICATIONS">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="FILEINFO" type="FileInfoType"/>
      <xs:element name="CLASSIFICATION" type="ClassificationType"
        minOccurs="1" maxOccurs="unbounded"/>
      <xs:element name="DESCRIPTION" type="DescriptionType"
        minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute name="NAME" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>

```

Abbildung 5-5: Definition des Wurzelements

Abbildung 5-5 zeigt die Definition des Wurzelements mit dem Namen CLASSIFICATIONS. Die Deklaration als `xs:complexType` ermöglicht das Vorhandensein mehrerer Unterelemente. Demzufolge kann das Element neben einer Beschreibung des Dateityps (FILEINFO) und einer textuellen Beschreibung (DESCRIPTION) mehrere Klassifizierungen vom Typ `ClassificationType` besitzen. Es können also für verschiedene Anwendungsfälle, beispielsweise verschiedene mechatronische Geräte, die jeweils zutreffenden Klassifizierungen verwaltet werden.

```

<!-- Declaration of ClassificationType -->
<xs:complexType name="ClassificationType">
  <xs:sequence>
    <xs:element name="FUNCTION" type="FunctionType" minOccurs="1"
      maxOccurs="unbounded"/>
    <xs:element Name="DESCRIPTION" type="DescriptionType"
      minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
  <xs:attribute name="NAME" type="xs:string" use="required"/>
</xs:complexType>

```

Abbildung 5-6: Definition des Grundelements für eine Klassifizierung

Das Element `ClassificationType` enthält alle Funktionen, die eine Werkzeugapplikation über das Remote-Framework für entfernte Interaktionen zur Verfügung stellt sowie ein Element für die textuelle Beschreibung. Diese sind in den FUNCTION-Elementen vom Typ `FunctionType` enthalten.

```
<!-- Declaration of FunctionType -->
<xs:complexType name="FunctionType">
  <xs:sequence>
    <xs:element name="PARAM" type="ParamType" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element name="DESCRIPTION" type="DescriptionType"
      minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="CLASS" type="ClassType" use="required"/>
  <xs:attribute name="NAME" type="xs:string" use="required"/>
</xs:complexType>
```

Abbildung 5-7: Definition des FunctionType-Elements

Ein `FunctionType`-Element beinhaltet als wichtigstes Attribut neben dem Funktionsnamen das Kennzeichen der zugehörigen Zugriffsklasse. Der Typ der Kennzeichnung ist im Schema mittels einer Enumeration definiert. Eine Funktion darf beliebig viele `PARAM`-Elemente vom Typ `ParamType` beinhalten, um auf diese Weise auch die Parameter von Funktionen der Zugriffskontrolle unterzuordnen.

```
<!-- Declaration of ParamType -->
<xs:complexType name="ParamType">
  <xs:sequence>
    <xs:element name="DESCRIPTION" type="DescriptionType"
      minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="CLASS" type="ClassType" use="optional"/>
  <xs:attribute name="NAME" type="xs:string" use="required"/>
  <xs:attribute name="RANGE" type="RangeType" use="optional"/>
</xs:complexType>
```

Abbildung 5-8: Handhabung von Funktionsparametern

Es fällt auf, dass auch die `ParamType`-Elemente die Möglichkeit der Einordnung in Zugriffsklassen beinhalten. Die Relation zur Klassifizierung der zugehörigen Funktion stellt sich wie folgt dar:

- Besitzen sowohl die Funktion als auch ein ihr zugehöriger Parameter eine Klassifizierung so gilt stets die höhere Sicherheitsstufe.
- Besitzt ein Parameter keine Klassifizierung so gilt pauschal die Sicherheitsstufe der Funktion.

Ebenfalls optional in Verbindung mit einer Parameterangabe ist die Angabe des `RANGE`-Attributs. Dieses kann eine Liste mit den zu filternden Parameterwerten oder Wertebereichen beinhalten. Ist das Attribut nicht enthalten, so erfolgt eine

Filterung durch die Zugriffskontrolle unabhängig von Parameterwerten. Das Format des RANGE-Attributs ist ebenfalls durch das Schema definiert.

Um dem Anwender die Erstellung einer Funktionsklassifizierung in Form einer XML-Beschreibung zu erleichtern, ist es sinnvoll, grafische Editoren zur Verfügung zu stellen. Diese ermöglichen beispielsweise assistentengestützt oder mittels Drag-and-Drop-Technik die Zuordnung von Funktionen und übernehmen die automatische Erstellung des schemakonformen XML-Dokuments. Ein Realisierungsbeispiel für eine ähnliche Aufgabenstellung findet sich im Verbundprojekt DeKOS [BENDER+ 02A]. Die Implementierung eines solchen Tools steht jedoch nicht im Fokus der vorliegenden Arbeit.

5.1.5 Tracing-Konzept

Als weitere Aufgabe der Zugriffskontrollkomponente wird in dieser Arbeit die Protokollierung sämtlicher, zwischen den verteilten Werkzeuginstanzen stattfindenden Interaktionen erachtet. Ganz allgemein wird in der einschlägigen Literatur (z.B. [BUSCH 02], [POHLMANN 00]) der Protokollierung relevanter Daten, nicht nur in remote zugänglichen IT-Systemen, ein hoher Stellenwert zugeschrieben. Dies geht auch aus den allgemeinen Anforderung an sichere Kommunikationssysteme nach Revisionierbarkeit (Rückverfolgbarkeit) und Nichtbestreitbarkeit hervor (vgl. Kapitel 4.4.3).

Um diese Anforderungen beim Einsatz eines verteilten Kollaborationswerkzeugs zu erfüllen, nimmt das Remote-Framework für beide Instanzen die Protokollierung sämtlicher ein- und ausgehender Daten vor. Die Protokollierung erfolgt unabhängig von der Einordnung einer Funktion in eine bestimmte Zugriffsklasse. Somit besteht auf beiden Seiten die Möglichkeit der Rückverfolgbarkeit bestimmter Interaktionen bzw. ist die Anforderung nach Nichtbestreitbarkeit erfüllt. Abbildung 5-9 und Abbildung 5-10 zeigen beispielhaft Ausschnitte von Protokolldateien (Logfiles), die durch das Remote-Framework während eines Protokolltests von felddbusfähigen Automatisierungsgeräten generiert wurden.

```

23.03.2005 16:00:48 Kuttig Session: 1a24778c-5eee-42c8-852b-59b16b737876
initiated. ClientIP:129.187.88.97:800 ServerIP:129.187.88.46:80
23.03.2005 16:00:53 Kuttig Outgoing CommRequest: Command: pw
23.03.2005 16:00:53 Kuttig Incomming CommRequest: StdOut:
D:/Programme/itm/MoTest/TXE/RTS/tcl
23.03.2005 16:03:08 Kuttig Incomming CommRequest: StdOut: 0 0 0 0 0 0 0 0
0x55 0x55 0x55 0x55
23.03.2005 16:03:08 Kuttig Incomming CommResponse: Command: 0 0 0 0 0 0 0
0 0x55 0x55 0x55 0x5

```

Abbildung 5-9: Ausschnitt aus dem Logfile einer entfernten Werkzeuginstanz

```
29.03.2005 08:52:58 Kuttig Session: b7663e73-5200-4c5c-a2c3-9d637afff7c6
initiated. ClientIP:129.187.88.46:800 ServerIP:129.187.88.97:800
29.03.2005 12:27:31 Kuttig Incomming CommRequest: Command: set DexData3
{0 0 0 0 0 0 0 0 0x55 0x55 0x55 0x55}
29.03.2005 12:27:32 Kuttig Outgoing CommResponse: Command: 0 0 0 0 0 0 0
0 0x55 0x55 0x55 0x55
```

Abbildung 5-10: Ausschnitt aus dem Logfile einer lokalen Werkzeuginstanz

Aufgezeichnet werden zunächst Informationen des Session-Managements über den Aufbau und Abbau der Verbindung. Jegliche Interaktion wird mit einem Zeitstempel sowie der Identität des entfernten Benutzers protokolliert. Dabei erfolgt die Kennzeichnung, ob es sich um ein- oder ausgehende Informationen handelt, die Aufzeichnung des Funktionsnamens sowie die Aufzeichnung der übermittelten Funktionsparameter. Da keine Interaktionen unter harten Echtzeitkriterien erfolgen, ist die Auflösung des Zeitstempels im Sekundenbereich ausreichend, um die Interaktion später bestimmten Ereignissen zuordnen zu können.

5.2 Integration der Multimediafunktionalität

Einen weiteren Schwerpunkt des Rahmenkonzepts bildet die möglichst anwenderfreundliche Nutzung von Multimediafunktionalität für die verteilte Kollaboration. Die Analyse in Kapitel 3 führte zu dem Ergebnis, dass bereits eine Vielzahl von Applikationen und Softwarebibliotheken für die Realisierung von Multimediafunktionalität existieren. Diese genügen für sich zwar nicht den speziellen Anforderungen, sind jedoch für die Realisierung der Grundfunktionalität nutzbar. Folglich sieht das Rahmenkonzept nicht die Entwicklung einer eigenen Multimedialösung vor, sondern verfolgt das Ziel der flexiblen Integration existierender Lösungen. Dies mindert einerseits den Aufwand für einen Anbieter des Remote-Frameworks und gibt andererseits dem Anwender die Wahlmöglichkeit bezüglich der genutzten Multimediaapplikation. Auf diese Weise ist zugleich die Möglichkeit der Anpassung an vorhandene interne Rahmenbedingungen sichergestellt.

Das Framework sieht eine entsprechende Schnittstelle zu den verwendeten Multimediakomponenten vor, über die der Anwender vom zentralen Steuerungsmodul aus agieren kann. Diese wird mittels eines Wrappers realisiert, der die proprietäre Funktionalität der Multimediaapplikationen über das Framework standardisiert zur Verfügung stellt. Die Funktionalität des Multimediakanals wird hierdurch gekapselt, ein aufwändiges Einarbeiten in die Spezifika unterschiedlicher Multimedia-Tools entfällt. Über verschiedene Wrapper-Ausführungen, vergleichbar mit Treiberbausteinen, ist die Integration nahezu beliebiger Multimediakomponenten in das Remote-Framework möglich.

5.2.1 Auswahl der Multimediafunktionalität

In Kapitel 4.2.3 wurden bereits die grundlegenden Mittel für die Kommunikation zwischen den beteiligten, räumlich getrennten Experten definiert: Es handelte sich um audio- und videobasierte, synchrone Mittel. Daneben existieren noch weitere Technologien, die gerade in Verbindung mit dem Begriff der *integrierten Anwendungen* zu dem Bereich der Multimediatechnologien zählen (vgl. Kapitel 3.4.1). Bevor der Umfang einer universellen Schnittstelle zwischen dem Remote-Framework und einem Multimedia-Wrapper definiert werden kann, ist es notwendig, die für eine verteilte Kollaboration sinnvollen Technologien zu identifizieren. Eine wertvolle Hilfestellung für die Auswahl gibt die Übersicht der Toolkategorien in Tabelle 5-2:

Kommunikationsmöglichkeiten					
Toolkategorie:	visuell	auditiv	schriftlich	synchron	asynchron
E-Mail, Built-in-E-Mail, Umfragetools, SMS ⁶⁷ -Kommunikation			x	(x)	x
VoIP-Kommunikation, Telefonkonferenz		x		x	
Newsgroups und webbasierte Diskussionsforen			x		x
Instant Messaging (IRC ⁶⁸ , webbasierter Chat)	(x)		x	x	x
Video- oder Telekonferenz, Webkonferenz, Videotelefon	x	x	(x)	x	(x)
Unified Messsaging ⁶⁹	x	x	x	x	x

Tabelle 5-2: Relevanz der existierenden Tool-Kategorien nach [STOLLER-SCHAI 03]

Die Zusammenstellung beinhaltet Kategorien, die den grundlegenden kollektiven Bedürfnissen der multimedialen Kommunikation Rechnung tragen. In den Spalten lässt sich die Ausprägung der Kommunikationsform identifizieren. Im vorliegenden Anwendungsfall handelt es sich um eine synchrone Ausprägung. Neben Audio/Videokommunikation wird hierfür der Einsatz von Instant-Messaging und Unified-Messaging empfohlen. Während die Einschätzung von Unified-Messaging als synchrone Technologie schon allein aufgrund der Definition des Begriffs nicht geteilt wird, ist Instant-Messaging als hilfreich für die synchrone Kollaboration zu bewerten. Im vorliegenden Anwendungsfall lässt

⁶⁷ Short Message Service

⁶⁸ Internet Relay Chat

⁶⁹ Aufbereiten von verschiedenen Nachrichten, z. B. Fax, Sprachnachricht in einer einheitlichen digitalen Form.

sich das Bedürfnis nach schnellem Austausch kurzer Textnachrichten, beispielsweise Parameter, Passwörter, Variablennamen etc. identifizieren, die durch Sprachkommunikation wesentlich aufwändiger zu übermitteln wären.

Neben diesen grundlegenden Mitteln motivieren die in Abschnitt 4.1.2 festgelegten Randbedingungen den Bedarf nach weiteren unterstützenden Komponenten für die multimediale Kommunikation. Es wurde der Bedarf identifiziert, einen Sachverhalt schnell zu skizzieren und als Erklärungsunterstützung zu verwenden (beispielsweise das Skizzieren einer Verschaltung). Hierfür ist der Einsatz eines Shared-Whiteboards gerechtfertigt. Des Weiteren muss die einfache Möglichkeit des Dateitransfers zur Verfügung stehen. Auch wenn diese den asynchronen Techniken zugeordnet wird, kann auch während der Phasen synchroner Kollaboration das Bedürfnis nach einer solchen Funktion bestehen (beispielsweise die spontane Übermittlung einer Konfigurationsdatei). Damit für den Anwender nicht der Bedarf nach einer eigenständigen Applikation (z. B. FTP⁷⁰-Client) entsteht, muss diese Funktionalität von der integrierten Multimediaanwendung zur Verfügung gestellt werden.

Abbildung 5-11 veranschaulicht zusammenfassend diejenigen MultimediaKomponenten, die aufgrund der erarbeiteten Anforderungen für das Remote-Framework ausgewählt werden.

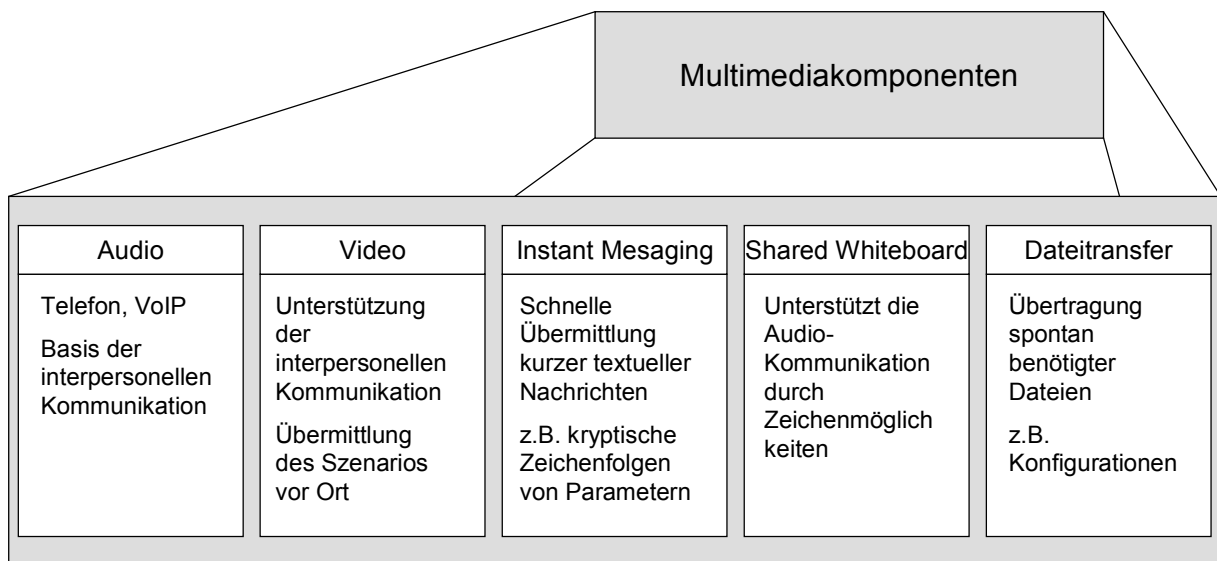


Abbildung 5-11: MultimediaKomponenten für das Remote-Framework

Auf Basis dieser Auswahl wird im weiteren Verlauf die Schnittstelle zwischen dem Remote-Framework und der Wrapper-Komponente zur Anbindung verschiedener Multimediaapplikationen definiert.

⁷⁰ File Transfer Protokoll

5.2.2 Schnittstelle zum Multimediasystem

Die Schnittstelle zwischen dem Remote-Framework und dem Multimedia-Wrapper muss sämtliche Funktionen umfassen, die erforderlich sind, um eine Multimediakomponente vom zentralen GUI aus zu verwenden. Der Umfang der Parametriermöglichkeiten beschränkt sich aufgrund der Vielfalt der existierenden Applikationen und der oftmals vorhandenen proprietären Funktionen auf den gemeinsamen Nenner grundlegender Funktionen. Diese können über das Framework ausgeführt werden, applikationsspezifische Anpassungen sind in den Applikationen selbst vorzunehmen. Die Schnittstelle ist in Abbildung 5-12 dargestellt und vollständig in Anhang E spezifiziert:

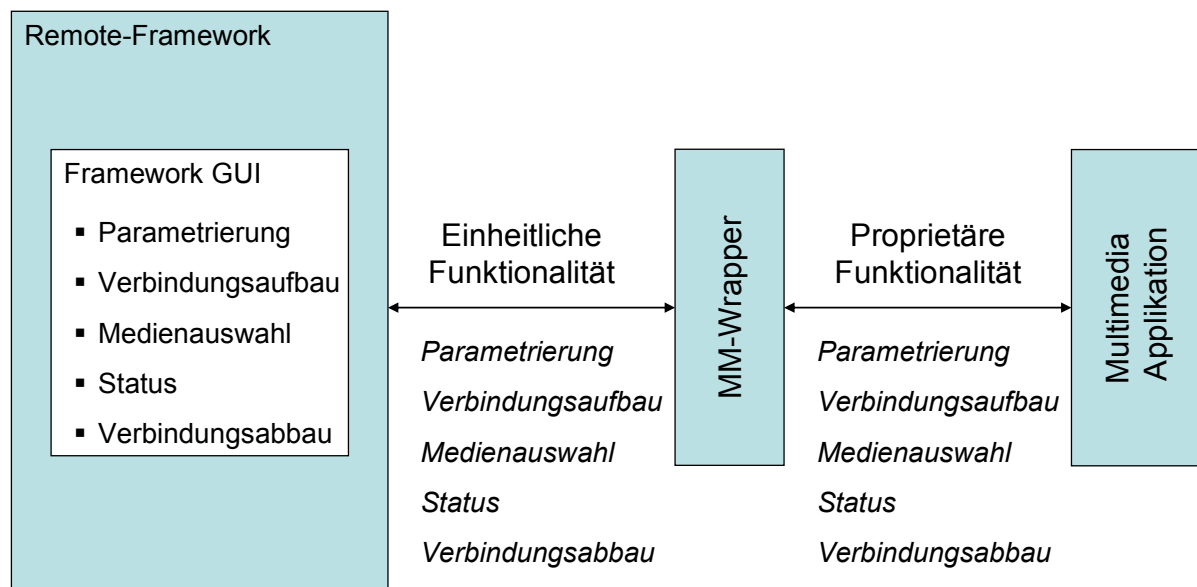


Abbildung 5-12: Schnittstelle zwischen Remote-Framework und Multimedia-Wrapper

Eines der Hauptkriterien für die Auswahl einer geeigneten Multimediaapplikation ist die Erfüllung der spezifizierten Grundfunktionalität sowie die Veröffentlichung dieser Funktionen nach außen. Weitere wichtige Entscheidungskriterien sind:

- Geringer Installationsaufwand für den Benutzer
- Gut dokumentierte Programmierschnittstelle (API)
- Niedrige Lizenzkosten
- Guter Support durch den Tool-Hersteller

5.2.3 Einheitliche Benutzerschnittstelle

Ziel dieses Kapitels ist nicht die Spezifikation eines detaillierten Styleguides, sondern die Festlegung der Funktionalität, die mindestens vom Remote-Framework zur Bedienung von Multimediakomponenten zur Verfügung gestellt wird. Das GUI für die Multifunktionalität stellt vier Gruppen von Funktionen zur Verfügung. Diese sind nachfolgend aufgezählt. Empfohlen wird die logische Gruppierung der Funktionen, beispielsweise auf eigenen Registerkarten. Die Möglichkeit, die für den schnellen Aufbau einer Multimediaverbindung von zentraler Stelle aus durchzuführen, stellt für den Anwender eine wesentliche Erleichterung dar: Der Aufwand für die Einarbeitung in verschiedene Multimediaapplikationen auf unterschiedlichen Systemen entfällt.

1. Verbindungsauf- und Abbau
2. Verwaltung der Multimediakomponenten
 - Dynamische Liste der verfügbaren Komponenten
 - Aktivierungsmöglichkeit für jede Komponente
 - Statusanzeige für jede Komponente sowohl durch detaillierte textuelle Informationen (Übertragungsqualität, Bit-Rate etc.) als auch durch eine schnell interpretierbare Symbolanzeige.
3. Parametriermöglichkeiten
 - Videoqualität (Videoformate, vgl. Tabelle 5-3)
 - Audioqualität (Kodierungsverfahren, Bit-Rate, Puffergröße)
 - Verbindungsparameter (Port-Nummern, Proxy-Einstellungen)
4. Adressierung des Kommunikationspartners
 - Eingabe oder Auswahl der Adresse (IP-Adresse bzw. URL)
 - Adressbuchfunktionalität

5.2.4 Analytische Betrachtung des Multimediakanals

Auch wenn die Abwicklung der Multimediakommunikation selbst den entsprechenden Tools obliegt, gibt es zwei entscheidende Kriterien bei der Übertragung von Multimediaströmen, denen sich auch der Anwender dieser Technologien bewusst sein muss. Dies ist zum einen die Berücksichtigung der zur Verfügung stehenden Netzwerkbandbreite und folglich die Adaption des Qualitätsanspruchs. Weiterhin die Auswahl passender Kodierungsverfahren⁷¹ für multimediale Daten. Ein zweiter Punkt ist die Berücksichtigung von QoS-Anforderungen⁷² an IP-basierte Netze, um Multimediadaten in zufrieden

⁷¹ Im allgemeinen Sprachgebrauch oftmals mit dem englischen Ausdruck *Codecs* bezeichnet.

⁷² Quality-of-Service-Anforderungen, übersetzt: Dienstgütemerkmale

stellender Qualität zu übermitteln. Damit der Anwender seinen Randbedingungen entsprechend die passenden Einstellungen vornehmen kann, ist es notwendig, sich mit den Basistechnologien auseinander zu setzen. Das folgende Kapitel beschreibt die relevanten Technologien in Kurzform und gibt Entscheidungshilfen sowie Berechnungsgrundlagen für die Parametrierung von Multimediaanwendungen.

Grundlegend basiert die Audio/Video⁷³-Kommunikation über IP-basierte Netze auf zwei Aufgaben:

1. Anrufsignalisierung:

Diese beinhaltet den Aufbau der Verbindung und umfasst die Abstimmung der verwendeten Protokolle und Codecs. Für die Anrufsignalisierung haben sich zwei Technologien etabliert: der H.323 Standard der ITU sowie das SIP-Protokoll der IETF. Beide Standards basieren auf dem TCP/IP-Protokollstack, eine Übersicht gibt Abbildung 5-13.

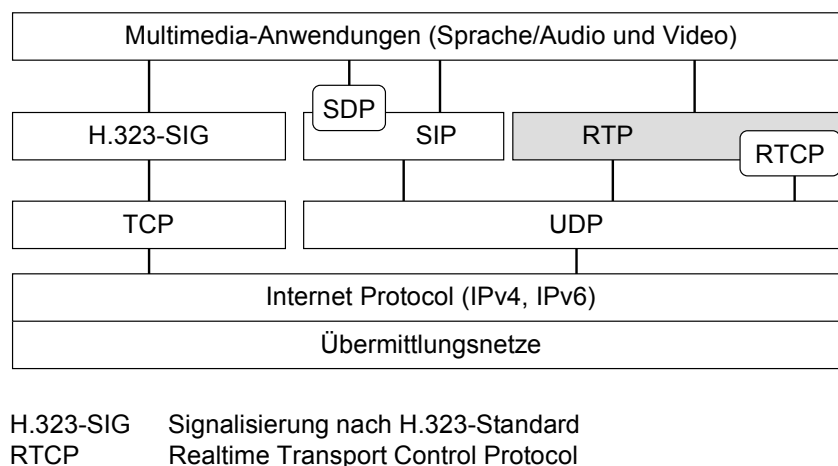


Abbildung 5-13: A/V-Übertragungsprotokolle im TCP/IP-Protokollstack

2. Übermittlung der A/V-Daten in Echtzeit:

Hierzu zählen die Kodierung, Übertragung und Dekodierung der multimedialen Informationen. Nach aktuellem Stand der Technik erfolgt die Übertragung von A/V-Daten in Echtzeit mittels des RTP-Protokolls. Dieses basiert auf dem verbindungslosen UDP-Protokoll (vgl.

Abbildung 5-13) und verfügt über Erweiterungen, die eine isochrone Kommunikation⁷⁴ über IP-basierte Netze ermöglichen (beispielsweise Zeitstempel). Innerhalb der RTP-Pakete befinden sich die kodierten

⁷³ Im Folgenden mit A/V abgekürzt

⁷⁴ Die isochrone Übermittlung von Datenpaketen zählt zu den wichtigsten Anforderungen multimedialer Kommunikation.

A/V-Daten. Für die Kodierung und Dekodierung steht eine Vielzahl an standardisierten Verfahren zur Verfügung.

Abbildung 5-14 veranschaulicht den prinzipiellen Ablauf sowie den Signalweg der multimedialen Datenübertragung durch ein IP-basiertes Netz. Zudem sind diverse Einflussfaktoren und beispielhaft einige Problemquellen wie *Jitter*⁷⁵-Effekte angegeben.

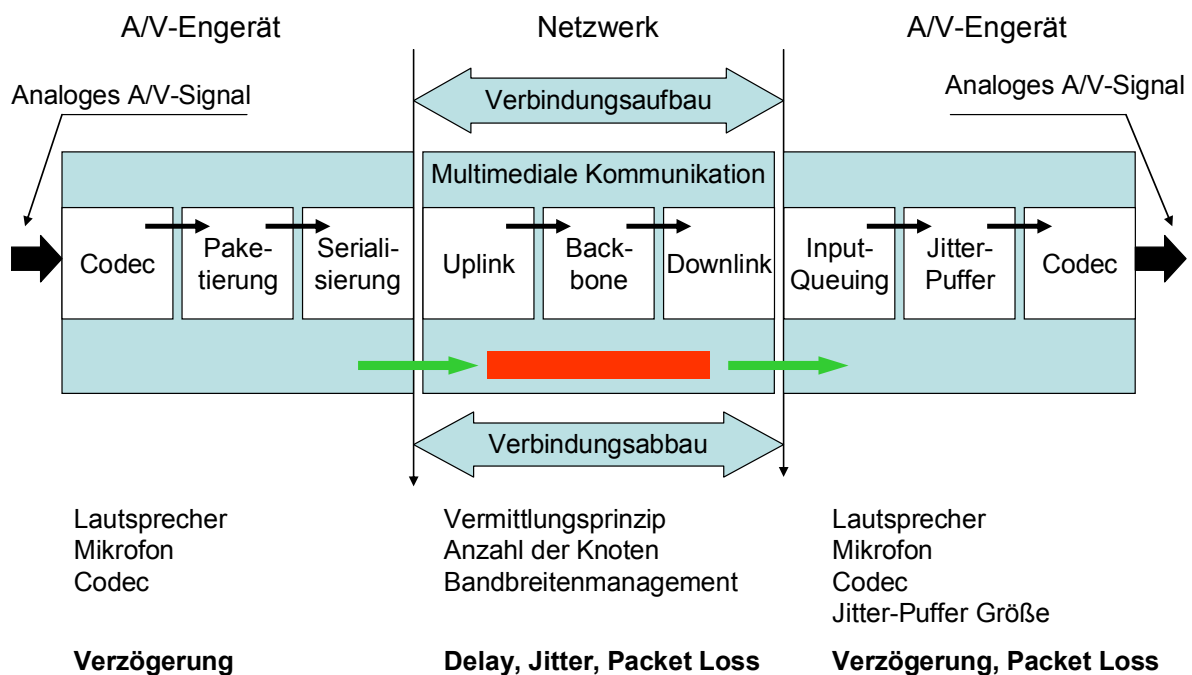


Abbildung 5-14: Funktionsweise der A/V-Kommunikation, nach [STOOS 04]

Gewöhnlich unterstützt eine Multimediaapplikation mehrere Kodierungsverfahren, einen Überblick der gängigsten Verfahren gibt Anhang D. Obgleich der geringeren Datenmengen stellt - im Gegensatz zur weit verbreiteten Meinung - die Audioübertragung weitaus höhere Anforderungen an die Qualität einer Netzwerkverbindung als die Übertragung von Videodaten. Die Verzögerung oder der Ausfall einiger Bilder bei der Videoübermittlung verursacht eine vergleichsweise geringe Beeinträchtigung der Informationen, da ersatzweise ein stehendes Bild zur Verfügung gestellt werden kann⁷⁶. Bei der Übermittlung von Audiodaten ist Vergleichbares nicht möglich. Schon der Verlust oder die Verzögerung weniger Audioinformationen führt hier zu Unverständlichkeit der

⁷⁵ Der Begriff *Jitter* bezeichnet Schwankungen in den Laufzeiten der zu übertragenden Datenpakete. Insbesondere IP-basierte Netze können aufgrund der unterschiedlichen Wege von Datenpaketen Jitter-Probleme verursachen. Ein hoher Jitter-Wert widerspricht der Forderung nach synchroner Datenübermittlung. Um Laufzeitschwankungen auszugleichen kommt beim Empfänger ein so genannter Jitter-Puffer oder auch Jitter-Buffer zum Einsatz.

⁷⁶ Die Behauptung bezieht sich nicht auf die Übertragung hochqualitativer Video-Streams wie beispielsweise bei digitalem Fernsehen.

übermittelten Sprache. Zumal Videoübertragung im entwickelten Konzept nur additiven Charakter besitzt (vgl. Kapitel 4.2.3), konzentrieren sich die Betrachtungen schwerpunktmäßig auf die Sprachübertragung. Dennoch sind einige Richtwerte für die Bandbreitenanforderung gängiger Videoformate in Tabelle 5-3 zusammengestellt. Angesichts der Forderung nach Nutzbarkeit des verteilten Systems über schmalbandige Medien werden Formate mit niedrigen Datenraten, wie das CIF bzw. QCIF-Format empfohlen.

Videoformat	Größe [Pixel]	Bildrate [Bilder/s]	Rohdaten [mb/s]	Datenrate komprimiert
HDTV über Antenne, Kabel, Satellit				
SMPTE 296M	1280x720	24 / 30 / 60	265 / 332 / 664	MPEG-2
SMPTE 295M	1920x1080	24 / 30 / 60I	597 / 746 / 746	20 – 45 mb/s
Videoproduktion				
ITU-R 601	720x480 720x576	60I 50I	249	MPEG-2 15 – 50 mb/s
High Quality video distribution (DVD)				
ITU-R 601	720x480 720x676	60I 50I	124	MPEG-2 4 – 8 mb/s
Intermediate quality video distribution (www)				
SIF	352x240 352x288	30 25	30	MPEG-1 1,5 mb/s
Video Konferenz über ISDN/Internet				
CIF	352x288	30	37	H.261/H.263 54 – 384 kb/s
Videotelefonie über drahtloses oder drahtgebundenes Modem				
QCIF	176x144	30	9.1	H.263 20 – 64 kb/s

I = Interlaced

Tabelle 5-3: Bandbreitenanforderungen der wichtigsten Videoformate [STEINBACH 04]

Für Sprachübertragungen dient die folgende Tabelle 5-4 zunächst als Grundlage für die Auswahl von Qualitätsstufen und Kodierungsverfahren. Zur Beurteilung der Sprachqualität eines Verfahrens ist der MOS⁷⁷-Wert angegeben. Die Interpretation der MOS-Skala findet sich in Anhang D.

⁷⁷ *Mean Opinion Score*: Objektives Kriterium zur Beurteilung der Sprachqualität auf einer Skala von 1 (= schlecht) bis 5 (= exzellent).

Codec	Benötigte Bandbreite [kbit/s]	MOS-Wert
PCM	64	4.3 - 4.5
ADPCM	16/24/32/40	3.4/3.6/3.9/4.2
CS-ACELP	8/6.4	4.0/3.8
LD-CELP	16	4.0 - 4.1
ACELP	5.3	3.5
MP-MLQ	6.3	3.7

Tabelle 5-4: Audio-Codex mit Bandbreitenanforderung und MOS-Wert [BADACH 04]

Wie bereits angesprochen, ist zur Erreichung hoher MOS-Werte die QoS-Fähigkeit des unterlagerten Kommunikationsnetzes ein wichtiges Kriterium. Im Vergleich zu leitungsvermittelten Netzen wie beispielsweise ISDN existieren bei IP-basierter Multimediakommunikation besondere Problemstellungen. Zu diesen zählen Verzögerungen bei der Auslieferung von Datenpaketen sowie Laufzeitschwankungen der Pakete, um die beiden Wichtigsten zu nennen [STOOS 04]. Eine Zusammenstellung der wichtigsten Performance-Parameter multimedialer Kommunikation gibt Anhang D. Besondere Auswirkungen, insbesondere auf die Qualität der Sprachübertragung, hat die so genannte Ende-zu-Ende-Verzögerung. Dieser Parameter ist insbesondere auch für den Anwender maßgeblich zur Optimierung des Jitter-Puffers einer Multimediaapplikation. Als Richtwert ist eine Ende-zu-Ende-Verzögerung von maximal 300 ms bis 400 ms angegeben, für eine hochqualitative Sprachübertragung stellen 150 ms die Obergrenze dar [HERMANN 97].

Anhand Abbildung 5-15 erfolgt die Berechnung der Ende-zu-Ende-Verzögerung beispielhaft für ein Netzwerk mit einer Übertragungskomponente (z. B. Switch oder Router). Das Beispiel ist für eine 1:1 Verbindung in kleineren Intranets repräsentativ und für den Einsatz mehrerer Netzwerkkomponenten skalierbar, berücksichtigt jedoch nicht die Netzauslastung. Sind die Netzwerkstruktur bzw. wichtige Netzwerkparameter nicht bekannt, so müssen ersatzweise Messungen durchgeführt werden.

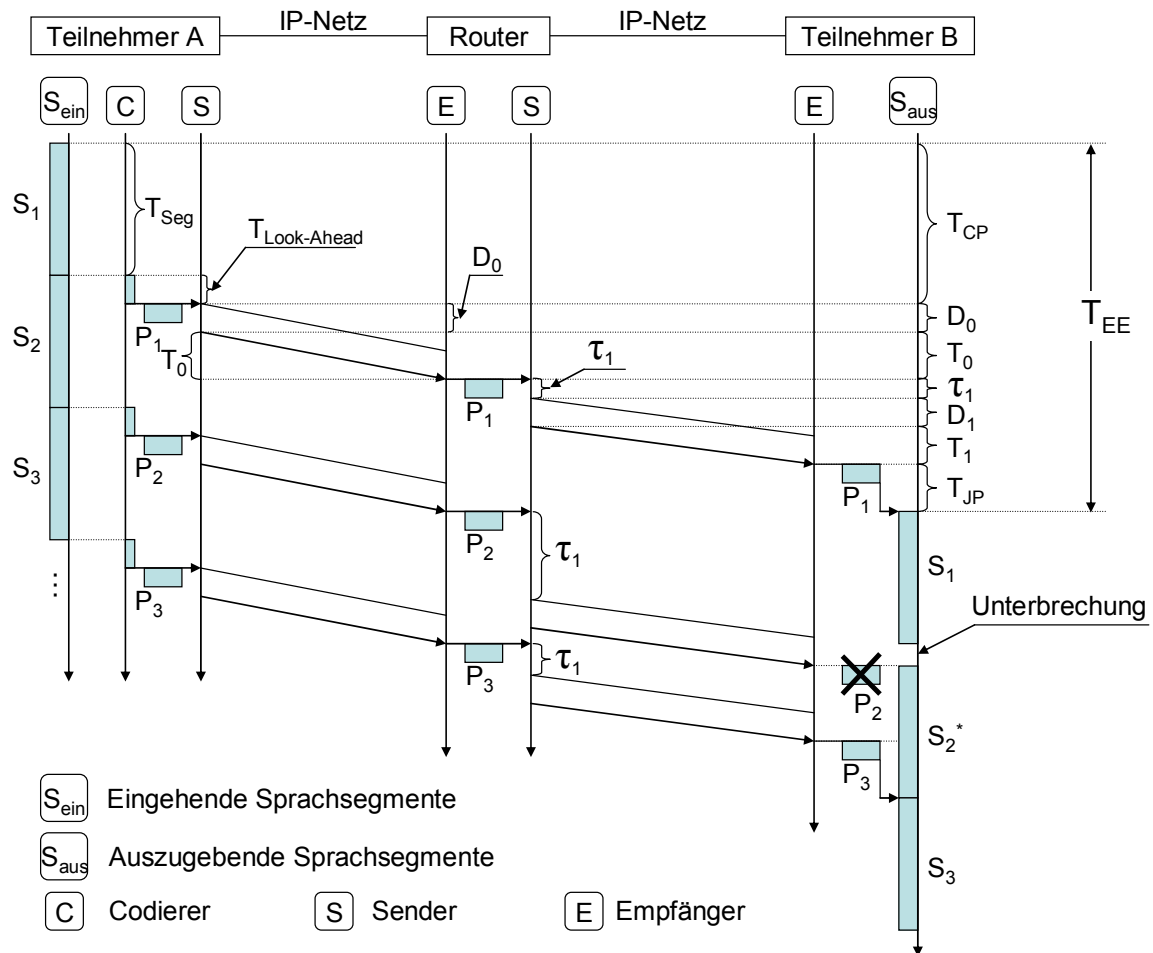


Abbildung 5-15: Einflussfaktoren auf die Ende-zu-Ende-Verzögerung [BADACH 04]

Mit Hilfe oben stehender Grafik lässt sich die Ende-zu-Ende-Verzögerung T_{EE} , basierend auf [BADACH 04] berechnen. Die Sprachsegmente werden bei Teilnehmer A erzeugt, über einen Router zu Teilnehmer B geschickt und dort ausgegeben (horizontale Achse). Vertikal ist der zeitliche Verlauf angetragen.

Ein Sprachsegment muss zunächst vollständig aufgezeichnet werden, bevor es an den Kodierer (C) übergeben werden kann. Diese Zeitdauer wird mit T_{Seg} bezeichnet. Anschließend erfolgt die Kodierung des Segments, wofür die Zeit $T_{Look-Ahead}$ benötigt wird. Die Zeit T_{CP} für die Erzeugung eines Sprachpakets ergibt sich also zu:

$$T_{CP} = T_{Seg} + T_{Look-Ahead}$$

T_{CP} wird auch als algorithmische Verzögerung bezeichnet. Sie kann üblicherweise bis zu 40 ms betragen (30 ms ist dabei eine gängige Segmentgröße). Da das Paket beim Aussenden serialisiert werden muss entsteht die Serialisierungsverzögerung D_0 . Diese berechnet sich wie folgt:

$$D_0 = \frac{\text{Paketgröße}[\text{Bit}]}{\text{Übertragungsrate}[\text{Bit} / \text{s}]}$$

Die Paketgröße setzt sich beispielsweise bei Verwendung von RTP über Ethernet zusammen aus:

Nutzlastlänge:	20 Byte
IP/UDP/RTP-Header:	> 40 Byte
MAC-Header:	12 Byte
Summe:	72 Byte

Das Versenden des Pakets zum Router nimmt die Zeit T_0 in Anspruch. Zur Ermittlung der Übertragungszeit T über eine Leitungsdistanz d wird in der Regel folgende Abschätzung vorgenommen:

$$T = \frac{1 \text{ ms}}{200 \text{ km}} * d [\text{km}]$$

Der Router, bzw. Switch verfügt über eine gewisse Latenzzeit, dazu addiert sich die Zeit, die das Paket evtl. in der Warteschlange verbringt bevor es vom Router auf das Netzwerkmedium weitergegeben werden kann. Die Summe dieser Verzögerungen wird als τ_1 bezeichnet. Jeder Router benötigt zudem wiederum Zeit für die Serialisierung (D_1). Das Versenden der Datenpakete zum nächsten Empfänger nimmt die Zeit T_1 in Anspruch.

Die Gesamtzeit für die Übertragung eines Paketes ergibt sich also zu:

$$t_{\bar{U}} = T_{CP} + D_0 + T_0 + \tau_1 + D_1 + T_1$$

Die Übermittlungszeit ist eine variable Größe, die insbesondere aufgrund der Parameter D_0 , τ_1 und D_1 variiert. Ist eine Anzahl k von Netzwerkknoten in den Signalweg eingeschaltet, so ergibt sich die Übertragungszeit vereinfacht zu:

$$t_{\bar{U}}(k) = T_{CP} + D_0 + T_0 + \sum_{n=1}^k (D_n + \tau_n + T_n)$$

Um den Jitter der Übertragungszeit auszugleichen wird das Paket im Extremfall für die Zeit T_{JP} im Jitter-Puffer gehalten. Die Ende-zu-Ende-Verzögerung T_{EE} ergibt sich also zu:

$$T_{EE} = t_{\bar{u}}(k) + T_{JP}$$

Als Richtwert für die Zwischenspeicherung gilt:

$$T_{JP} = 2 * T_{Seg}$$

Somit gilt letztendlich:

$$T_{EE} = t_{\ddot{U}}(k) + 2 * T_{Seg}$$

Im angegebenen Beispiel lässt sich auf Empfängerseite eine zeitliche Lücke zwischen dem ausgegebenen Sprachsegment S_1 und dem Folgesegment S_2 erkennen. In diesem Fall ist die Größe des Jitter-Puffers zu klein gewählt, sodass die Verzögerung bei der Übermittlung von P_2 nicht mehr ausgeglichen werden kann. Die richtige Anpassung des Jitter-Puffers an die Netzparameter und Randbedingungen ist also von enormer Wichtigkeit für eine qualitativ hochwertige A/V-Übertragung.

Anhand der Berechnungen und der Grafik in Abbildung 5-15 wird auch deutlich, dass die Einflussnahme des Anwenders auf die verantwortlichen Parameter begrenzt ist. Einige Parameter sind abhängig vom unterlagerten Netzwerk und den darin eingesetzten Komponenten. Optimierungsmaßnahmen in diesem Bereich obliegen dem Betreiber des Netzwerks. Der Katalog der oftmals komplexen Maßnahmen ist umfangreich. Es wird an dieser Stelle auf weiterführende Literatur wie beispielsweise [ARMITAGE 00], [BLACK 00] oder [JHA+ 02] verwiesen.

5.3 Architektur des Remote-Frameworks

Die Spezifikation der Architektur des Remote-Frameworks erfolgt auf formalem Weg unter Verwendung der Sprache UML⁷⁸. Es handelt sich dabei um eine Sprache zur Notation, Spezifikation, Konstruktion, Visualisierung und Dokumentation von Modellen für Softwaresysteme. UML berücksichtigt die gestiegenen Anforderungen bezüglich der Komplexität heutiger Systeme und deckt ein breites Spektrum von Anwendungsgebieten ab [OESTEREICH 04]. Nicht zuletzt deshalb hat sich die Sprache nicht nur in der Softwaretechnik als standardisiertes Mittel etabliert, vgl. [BALZERT 00].

5.3.1 Statisches Modell

Ausgehend von dem in Kapitel 4 erarbeiteten Konzept lässt sich das Objektmodell für die Anwendung des Remote-Frameworks in Verbindung mit verteilten Werkzeugen der integrativen Phasen aufstellen. Dieses zeigt Abbildung 5-16 in Form eines UML-Objektdiagramms:

⁷⁸ *Unified Modelling Language*, derzeit aktuelle Version: UML 2.0

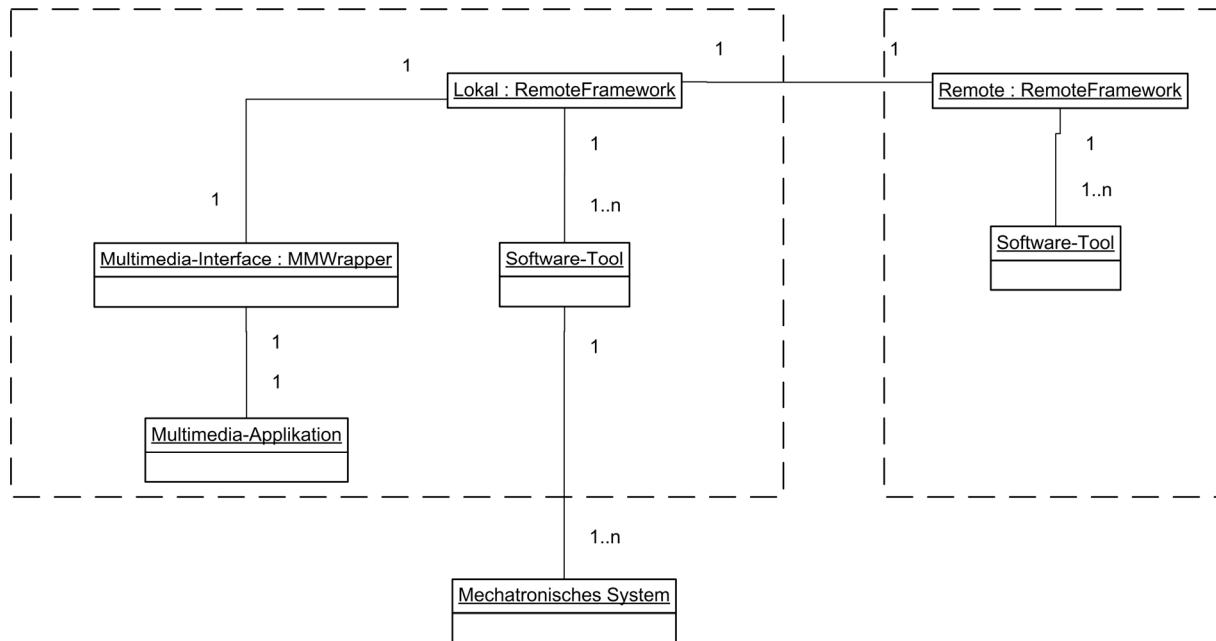


Abbildung 5-16: Objektmodell des Remote-Frameworks mit verteilten Werkzeugen

Für die Definition der Schnittstellen zwischen dem Remote-Framework und der Applikation ist insbesondere von Bedeutung, auf welcher funktionalen Ebene die Kopplung erfolgt. Als Auswahlkriterium hierfür dient das in Abschnitt 3.3.2 vorgestellte KOMFORCE-Referenz-Modell, welches für die technische Kopplung von verteilten Applikationen zwei Ebenen vorsieht:

1. Modelltechnische Integration

Die Koppelung erfolgt auf der Ebene der Produktmodelle. Dies bedeutet für das Remote-Framework, dass modellspezifische Funktionalität umgesetzt werden muss und somit eine Abhängigkeit der Schnittstellen von der eingesetzten Applikation besteht.

2. Systemtechnische Integration

Hier erfolgt die Kopplung auf technologischer Ebene unter Verwendung gleicher Basistechnologien auf beiden Seiten.

Letzterer Ansatz beinhaltet einerseits den Vorteil, die Schnittstelle flexibel für verschiedene Werkzeuge nutzen zu können, birgt jedoch andererseits beliebige semantische Freiheiten, die wiederum zu Inkompatibilitäten führen. Um dies zu vermeiden, wird zunächst eine schlanke Schnittstelle auf systemtechnischer Ebene gewählt. Auf modelltechnischer Ebene werden anschließend lediglich die Grundfunktionen definiert um Werkzeugfunktionalität zu verpacken und zu versenden. Funktionsaufrufe eines Werkzeugs werden dazu in ein spezifiziertes Aufrufobjekt verpackt, welches über die festgelegten Schnittstellenfunktionen an das Framework übergeben wird. Ein solches Objekt enthält, neben dem Funktionsnamen, sämtliche für die Ausführung notwendigen Parameter, Felder für Rückgabewerte und Felder für Fehlerobjekte. Die Spezifikation des vollstän-

digen Aufrufobjektes findet sich in Anhang E. Mit dieser Festlegung definieren sich die Schnittstellen der Komponenten wie in Abbildung 5-17 dargestellt. Zu beachten ist, dass sämtliche Programmierschnittstellen asynchron ausgeführt sind, um auch lange Bearbeitungszeiten in den ausführenden Komponenten einer Werkzeuginstanz zu ermöglichen.

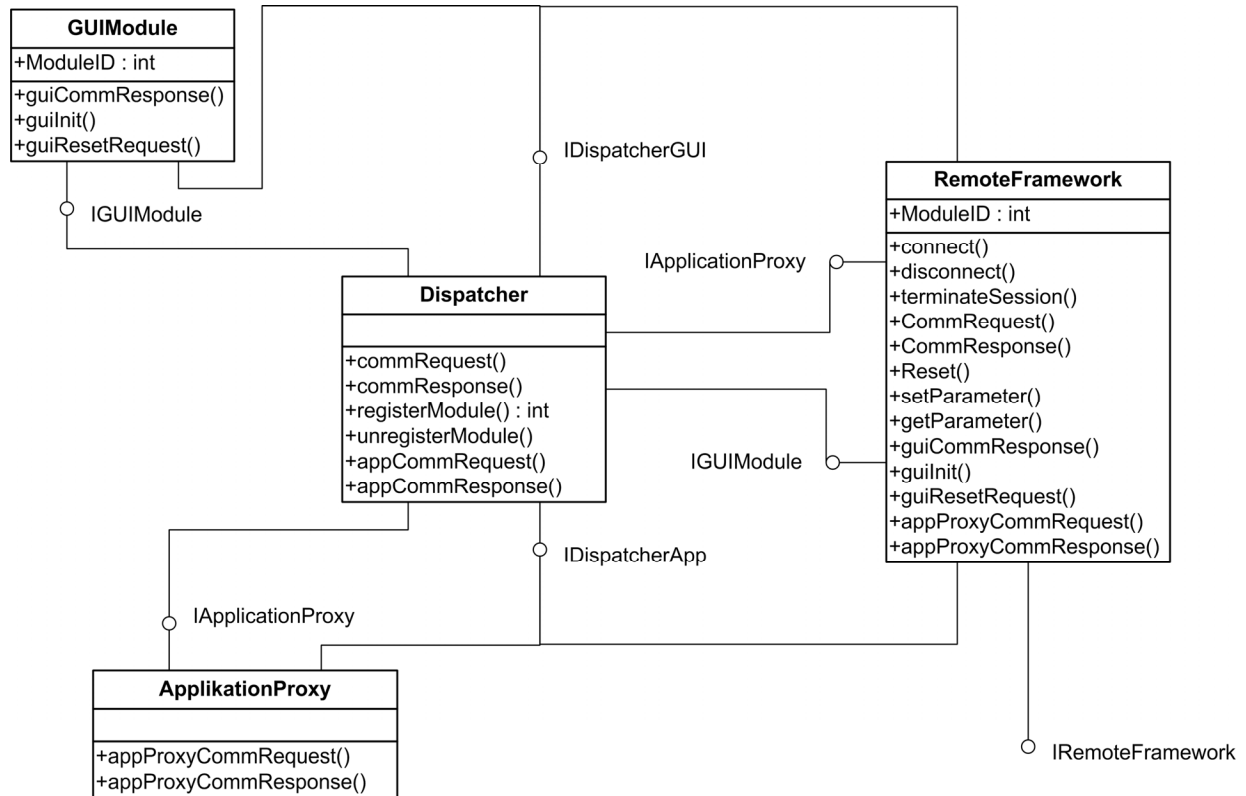


Abbildung 5-17: Schnittstelle zwischen Werkzeug und Remote-Framework

Der Dispatcher verfügt über zwei Schnittstellen:

- Die Schnittstelle `IDispatcherGUI` für ankommende Kommandos von GUI-Modulen oder dem Remote-Framework, diese werden über den so genannten `ApplicationProxy` an die ausführenden Komponenten der Applikation übergeben. Der `ApplicationProxy` übernimmt die Entpackung und Weiterverarbeitung der übermittelten standardisierten Aufrufobjekte.
- Die Schnittstelle `IDispatcherApp` für ankommende Kommandos oder Antworten von Seiten der ausführenden Komponenten oder des Remote-Frameworks.

Umgekehrt stellt das Remote-Framework, parallel zu den innerhalb der Applikation existierenden Komponenten, zwei Schnittstellen zur Verfügung:

- Die Schnittstelle `IGUIModule` für Nachrichten die vom Dispatcher an die angeschlossenen GUI-Module, bzw. an das Remote-Framework geleitet werden.
- Die Schnittstelle `IApplicationProxy` für Nachrichten an die ausführenden Komponenten bzw. im Remote-Fall an das Framework zur Weiterleitung an die lokale Instanz.

Auf diese Weise wird das symmetrische Schnittstellenkonzept für den verteilten Einsatz einer Werkzeuginstallation realisiert: Im Falle der Remote-Instanz muss der Zeiger des Dispatchers anstelle auf die ausführenden Komponenten des Werkzeugs auf das Remote-Framework umgebogen werden. Auf lokaler Seite leitet der Dispatcher Antworten von Remote-Aufrufen statt an ein lokales GUI-Modul gleichfalls an das Remote-Framework zur Versendung weiter. Das vollständige Klassendiagramm des Gesamtsystems inklusive der ausführlichen Schnittstellenspezifikation findet sich in Anhang E.

Der Dispatcher übernimmt weiterhin zwei wichtige Aufgaben im Zusammenhang mit dem verteilten Einsatz eines Werkzeuges:

- Koordination des Zugriffs mehrerer User auf das angeschlossene mechatronische System, also auf die ausführenden Komponenten des Werkzeugs. Während der Abarbeitung einer Funktion müssen weitere Aufrufe entweder geblockt oder zwischengespeichert werden.
- Realisierung des synchronisierten Modus für die Kollaboration. Hierzu zählen:
 - Abgleich von Datenbeständen, die auf beiden Seiten vorhanden sind und kollaborativ genutzt werden, bei Änderungen auf einer Seite.
 - Verteilung von Antworten auf Funktionsaufrufe sowohl an die betreffenden Module der lokalen Instanz, als auch an die Remote-Instanz.

Da diese beiden Aufgaben unabhängig vom Remote-Framework zu lösen sind, obliegt die Ausführung dem Hersteller des jeweiligen Software-Tools und ist in diesen Ausführungen nicht weiter spezifiziert. Die Abwicklung von lokalen sowie entfernten Funktionsaufrufen beschreibt detailliert das nachfolgende Kapitel.

5.3.2 Dynamisches Modell

Bei der Nutzung eines Rechnerwerkzeugs für die verteilte Kollaboration existieren zwei Szenarien:

1. Der lokale Entwickler führt eine Funktion über das Frontend der lokalen Werkzeuginstanz aus.
2. Der entfernte Experte führt eine Funktion in der lokalen Instanz über das Frontend der entfernten Werkzeuginstanz aus. Dabei laufen sämtliche Funktionsaufrufe über das Remote-Framework.

Um das entwickelte Konzept umsetzen zu können, werden diese beiden Szenarien im weiteren Verlauf detailliert dargestellt. Abbildung 5-18 veranschaulicht zunächst den Informationsfluss auf lokaler Seite:

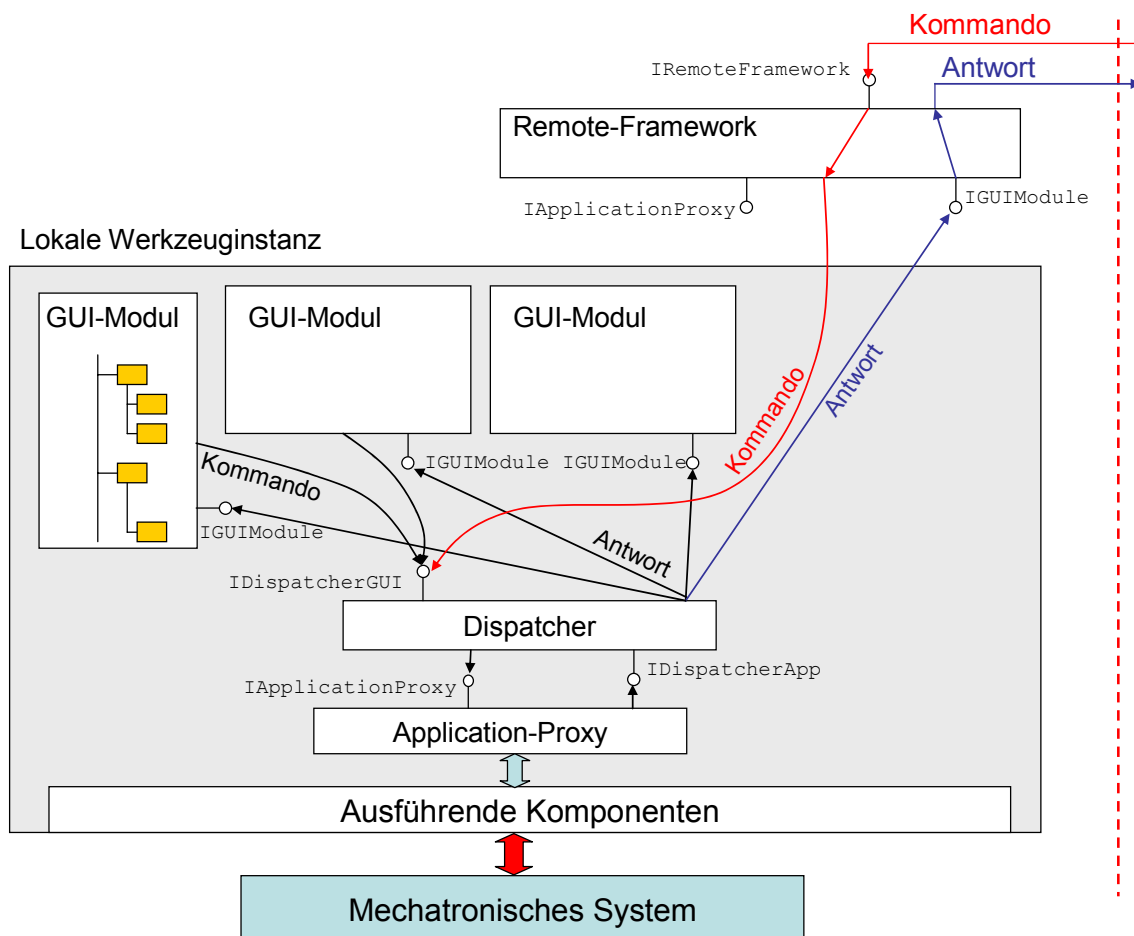


Abbildung 5-18: Informationsfluss auf der lokalen Seite

Funktionsaufrufe (Kommandos) von lokalen GUI-Komponenten werden in ein Aufrufobjekt verpackt und über die entsprechende Schnittstelle (vgl. Kapitel 5.3.2) an den Dispatcher übergeben. Dieser reiht die Befehle in eine Warteschlange ein und übergibt sie an den Application-Proxy, der die Entpackung und Interpretation der Aufrufobjekte übernimmt. Nach erfolgter Abarbeitung der Funktion in den ausführenden Komponenten des Werkzeugs nehmen Antworten den umgekehrten Weg zurück an die verantwortlichen GUI-Komponenten. Ein Aufruf auf der Remote-Seite wird analog an den dortigen Dispatcher übergeben. Je nachdem ob sich die Applikation im synchronisierten Modus befindet, werden die Antworten an das Remote-Framework, bzw. additiv an lokale GUI-Komponenten übermittelt. Den Informationsfluss auf Seiten der entfernten Werkzeuginstanz veranschaulicht Abbildung 5-19:

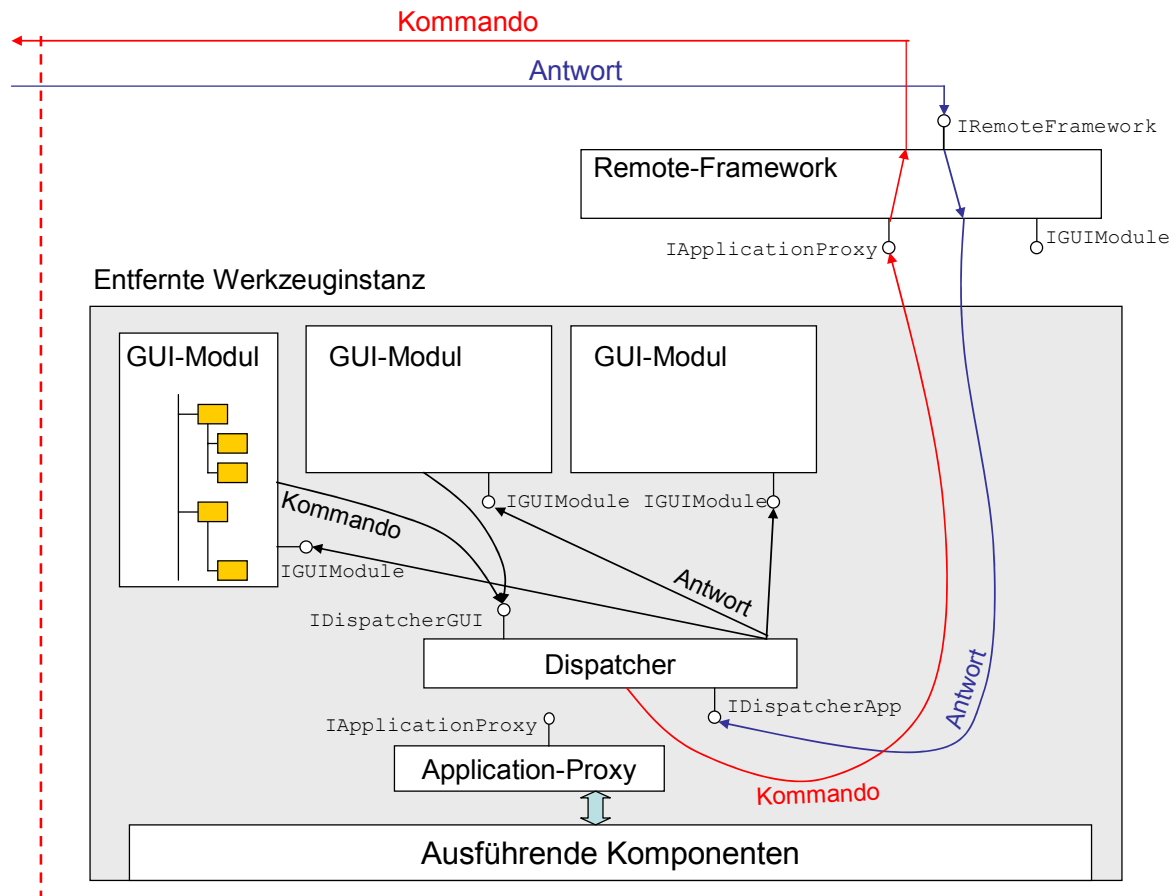


Abbildung 5-19: Informationsfluss auf der Remote-Seite

Auf der Remote-Seite ist keine Geräteanbindung über die ausführenden Komponenten vorhanden. Funktionsaufrufe werden deshalb vom Dispatcher nicht an den Application-Proxy, sondern an die parallele Schnittstelle des Remote-Frameworks übergeben. Hierzu wird lediglich der Schnittstellenzeiger entsprechend umgestellt. Antworten, die von der lokalen Instanz gesendet wurden, übergibt das Remote-Framework direkt an die entsprechende Schnittstelle des Dispatchers, so als ob die Antwort direkt vom Application-Proxy stammen würde. Für den Dispatcher selbst macht es also keinen Unterschied, ob er innerhalb der lokalen oder der Remote-Instanz arbeitet.

Die Vorgänge innerhalb der Werkzeugapplikation lassen sich mittels UML-Sequenzdiagrammen exakt beschreiben. Folgende vereinfachten Diagramme dienen dem Verständnis, der vollständige Nachrichtenfluss ist mittels Sequenzdiagrammen in Anhang E spezifiziert. Abbildung 5-20 zeigt zunächst die Sequenz für das Absetzen eines Funktionsaufrufs für die beiden Fälle lokale Instanz und entfernte Instanz:

Absetzen eines Kommandos

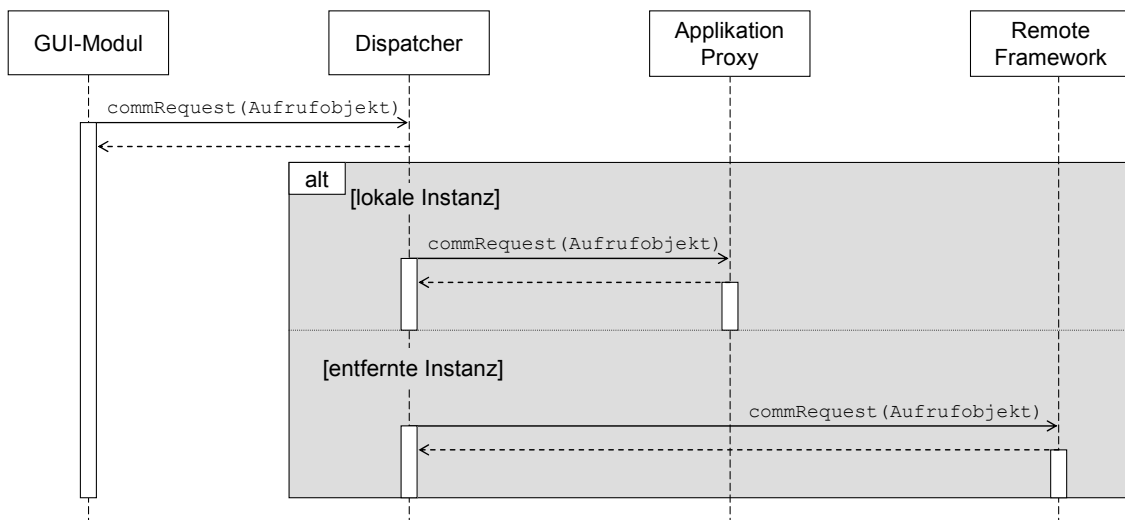


Abbildung 5-20: Ausführung einer Funktion (Funktionsaufruf)

Abbildung 5-21 zeigt die Sequenz für das Empfangen von Antworten sowohl auf lokaler als auch auf entfernter Seite:

Empfangen der Antwort

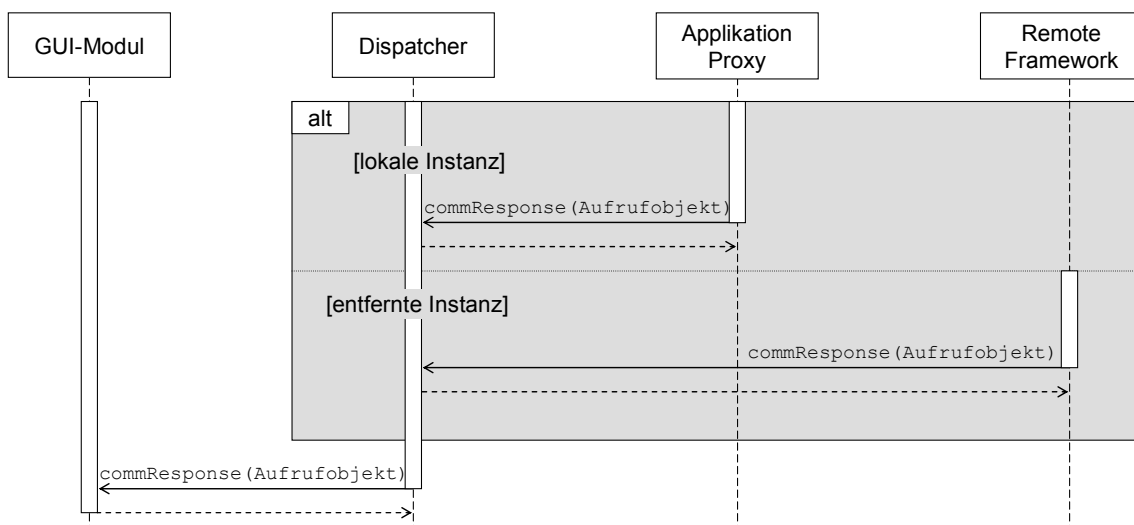


Abbildung 5-21: Ausführung einer Funktion (Empfang der Antwort)

Wie in den Diagrammen dargestellt, erfolgen sämtliche Funktionsaufrufe in asynchroner Form. Es sei nochmals betont, dass der Informationsfluss keinen harten Echtzeitanforderungen genügen muss. Diese liegen unterhalb der Ebene der ausführenden Komponenten vor. Sowohl für lokale als auch für entfernte Interaktionen gelten die zeitlichen Anforderungen gemäß Definition 4-2.

Bevor von der entfernten Werkzeuginstanz aus Interaktionen vorgenommen werden, muss der Verbindungsaufbau zwischen den beiden Instanzen des

Remote-Frameworks erfolgen. Dies beinhaltet nicht nur die Benutzeridentifikation und -Legitimierung unter Verwendung eines Session-Modells, sondern auch den automatischen Aufbau der Multimediaverbindung mit den voreingestellten Parametern des Initiators. Der Ablauf des Verbindungsauf- und Abbaus ist in Anhang E ebenfalls in Form von Sequenzdiagrammen spezifiziert.

5.3.3 Generisches Fehlerbehandlungskonzept

Bei der Übertragung von Nachrichten zwischen beiden Instanzen über weite Strecken und mehrere zwischengeschaltete Framework-Komponenten erhöht sich zwangsläufig die Anzahl möglicher Fehlerquellen. Fehler können beispielsweise durch Abbruch der Kommunikationsverbindung, nicht Vorhandensein angeforderter Ressourcen oder durch Abblocken einer Anforderung durch einen Benutzer entstehen. Die Entwicklung eines wirkungsvollen Fehlermanagements in verteilten Applikationen stellt für die Hersteller solcher Applikationen einen hohen Aufwand dar. Andererseits lassen sich nur so stabile Lösungen erzielen. Um den Aufwand zu reduzieren, realisiert das Remote-Framework ein generisches Fehlerbehandlungskonzept. Zu diesem gehört die Spezifikation eines einheitlichen Fehlerobjekts, welches an das Aufrufobjekt gekoppelt ist (vgl. Anhang E).

Von besonderer Bedeutung für die effiziente Fehlerbehandlung ist die Identifikation derjenigen Stelle, an welcher der Fehler aufgetreten ist. In Abhängigkeit davon müssen in verschiedenen Komponenten des Remote-Frameworks bzw. der Werkzeugapplikation Reaktionen auf den Fehler erfolgen (vgl. Abbildung 5-22):

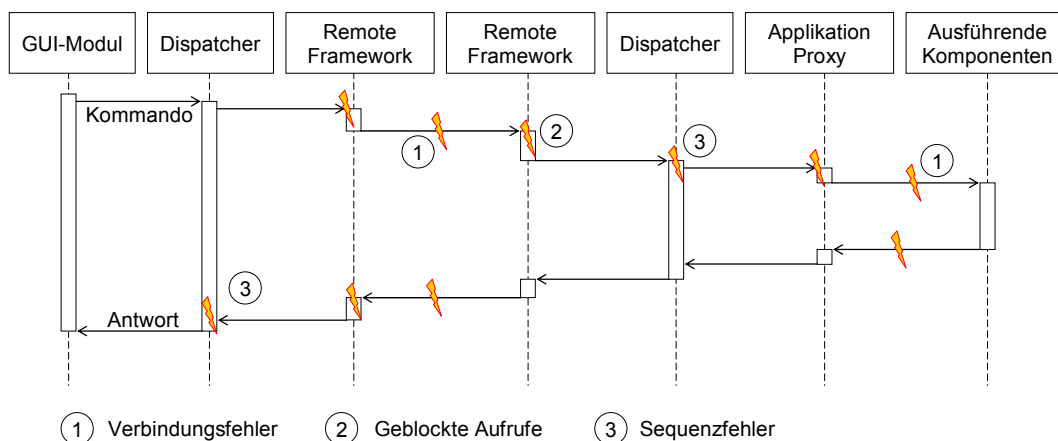


Abbildung 5-22: Beispiele für Fehler im Informationsfluss

Im Rahmen der Spezifikation für das Remote-Framework steht ein generisches Fehlerbehandlungskonzept zur Verfügung, welches sämtliche durch den verteilten Einsatz entstehenden Fehlerquellen abdeckt und Reaktionen vorschlägt. Das Konzept liegt in Form von Sequenzdiagrammen vor, die in Anhang E aufgeführt sind. Das Konzept deckt insbesondere diejenigen Fehlersituationen ab, welche

durch die Verwendung asynchroner Programmierschnittstellen im verteilten System entstehen.

5.3.4 Basistechnologien für die Remoting-Funktionalität

Für die Realisierung von Kommunikationsverbindungen in verteilten Applikationen hat sich heutzutage eine Vielzahl an Technologien etabliert. Einige wichtige Vertreter hieraus sind die DCOM-Technologie von Microsoft, CORBA der Object Management Group (OMG), oder RPC⁷⁹ in der Java-Welt. Gerade in Verbindung mit der Nutzung des Internets bzw. Standard-Technologien aus dem Internet-Bereich hat sich in jüngster Vergangenheit besonders eine Technologie etabliert: das SOAP⁸⁰-Protokoll, spezifiziert vom World-Wide-Web-Consortium und unterstützt durch nahezu alle namhaften Softwarehersteller.

SOAP spezifiziert die Kodierung von Nachrichten in Klartext-Form auf Basis von XML. Grundsätzlich stellt die Technologie ein zustandsloses, unidirektionales Paradigma für den Nachrichtenaustausch dar. In Kombination mit den unterlagerten Protokollen wie beispielsweise http und applikationsspezifischen Informationen sind auch komplexe Interaktionsmuster (beispielsweise request/response, request/multiple responses, etc.) möglich. SOAP adressiert weder die Semantik der transportierten applikationsspezifischen Daten, noch behandelt es Aspekte wie Routing der Nachrichten, zuverlässiger Datentransport oder Firewall-Resistenz. Jedoch stellt es ein standardisiertes Framework für den flexiblen Transport applikationsspezifischer Informationen zur Verfügung. Zudem beschreibt SOAP vollständig die Aktionen, die seitens der Kommunikationspartner für das Versenden und den Empfang von SOAP-Nachrichten auszuführen sind [W3C 03]. Die Spezifikation des Protokolls besteht neben einer umfangreichen Einführung aus zwei Teilen:

Teil eins definiert den so genannten SOAP-Envelope als universellen Rahmen für die Repräsentation des Inhalts einer SOAP-Nachricht sowie zusätzlicher Informationen für die Behandlung einer Nachricht (beispielsweise als Funktionsaufruf). Zudem werden die Rahmenbedingungen für die Bindung von SOAP an unterlagerte Protokolle definiert.

Teil zwei definiert ein Datenmodell für SOAP, im Speziellen ein Kodierungsschema in Form von XML-Schemata für Datentypen, die in verteilten Funktionsaufrufen übertragen werden. Außerdem ist die konkrete Realisierung der Bindung von SOAP an das http-Protokoll spezifiziert.

Abbildung 5-23 und Abbildung 5-24 zeigen ein Beispiel für eine SOAP-Anfrage (Funktionsaufruf) durch einen Client sowie die zugehörige SOAP-Antwort durch den Server.

⁷⁹ Remote Procedure Call [STEYER 99]

⁸⁰ Ursprünglich die Abkürzung für *Simple Object Access Protocol*, seit Version 1.2 nicht mehr offiziell da SOAP nicht mehr ausschließlich für den Zugriff auf Objekte dient [WIKIPEDIA 05].


```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <soapenv:Body>
    <ns1:add soap-
env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          xmlns:ns1="urn:CardValidator">
      <number xsi:type="xsd:string">1234 5678 9876 5432</number>
      <valid xsi:type="xsd:string">12/08</valid>
    </ns1:add>
  </soapenv:Body>
</soapenv:Envelope>

```

Abbildung 5-23: Beispiel der SOAP-Anfrage eines Clients an einen Server [W3C 03]

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <soapenv:Body>
    <ns1:addResponse soap-
env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          xmlns:ns1="urn:CardValidator">
      <addReturn xsi:type="xsd:string">true</addReturn>
    </ns1:addResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

Abbildung 5-24: Beispiel der SOAP-Antwort des Servers [W3C 03]

Obwohl SOAP häufig in Verbindung mit dem http-Protokoll eingesetzt wird, ist die Unabhängigkeit von unterlagerten Transportprotokollen als wichtiges Merkmal hervorzuheben. Diese Eigenschaft ermöglicht die flexible Anpassung an unterschiedliche Kommunikationsinfrastrukturen oder Firewall-Systeme. Beispielsweise ist neben http auch die Verwendung von TCP oder sogar Email möglich.

Aufgrund der aufgeführten Vorteile hat SOAP in der Automatisierungstechnik, wo ebenfalls reale Maschinen und Anlagen im Mittelpunkt stehen, einen hohen Akzeptanzgrad erreicht (vgl. Kapitel 3.2.1). Beispiele für den erfolgreichen Einsatz finden sich in [BENDER+ 01] oder [PNO 04]. Die Vorteile des Protokolls, vor dem Hintergrund des erfolgreichen Einsatzes in der Automatisierungstechnik führen zur Auswahl von SOAP als Kommunikationsprotokoll zwischen den verteilten Instanzen des Remote-Frameworks.

5.4 IT-Infrastrukturen und IT-Security

Bereits in den vorangegangenen Ausführungen wurde auf die enorme Bedeutung der IT-Sicherheit in Verbindung mit dem Einsatz von Remote-Technologien hingewiesen. Das nachfolgende Kapitel widmet sich dieser Problemstellung. Zunächst erfolgt die Betrachtung des bei der Nutzung von Remote-Technologien vorhandenen Bedrohungspotentials und im weiteren Verlauf eine kurze Übersicht geeigneter Gegenmaßnahmen. Aufgrund des großen Umfangs dieses eigenständigen Themas können hier nur Einblicke gegeben werden, deshalb widmet sich Anhang F der Zusammenstellung gängiger Normen und Richtlinien zur Erlangung von IT-Sicherheit. Abschließend wird ein Konzept erarbeitet, wie die Integration des Remote-Frameworks in entsprechend abgesicherte IT-Infrastrukturen erfolgen kann. Das Konzept ist innerhalb des Kommunikationsmoduls des Remote-Frameworks realisiert.

5.4.1 Identifikation des Bedrohungspotentials

Auf die besonderen Auswirkungen von Fehlern beim Fernzugriff auf reale mechatronische Geräte und Systeme wurde bereits mehrfach im Verlauf dieser Ausführungen hingewiesen. Im Zusammenhang mit der Verwendung von Remote-Technologien und der damit verbundenen Anbindung sicherheitskritischer Subnetze an öffentliche Netze stellen unautorisierte Eingriffe durch Dritte eine große Gefahr für die Entstehung solcher Fehler dar.

Angriffe auf Kommunikationsverbindungen können grundsätzlich in drei Kategorien gegliedert werden:

1. Passive Angriffe: Abhören bzw. Auslesen von Daten

Diese erfolgen zum Zwecke der Informationsgewinnung, bzw. zur Vorbereitung weiterführender Aktionen wie beispielsweise dem Ausspähen von Passwörtern.

2. Aktive Angriffe: Manipulation von Daten

Erfolgreiche aktive Angriffe ermöglichen letztendlich auch direkte Eingriff in den Datenhaushalt eines mechatronischen Systems.

3. Angriff auf die Funktionsfähigkeit des Systems

Hierbei geht es darum, durch ständiges Einspielen von (inkonformen) Datenpaketen die Dienste eines IT-Systems zu blockieren bzw. das System vollständig zum Absturz zu bringen (Denial of Services).

Die Liste der Angriffsmöglichkeiten ist äußerst umfangreich. Hinzu kommt, dass ständig neue Angriffsmuster und Methoden auftauchen. Die aufgezählten Kategorien von Angriffen betreffen dabei nahezu sämtliche Ebenen des ISO/OSI-Referenzmodells. Grund hierfür sind größtenteils fehlerhafte oder unsaubere Implementierungen der Protokoll-Stacks, aber auch Mängel in den Protokollspezifikationen selbst. Konkrete Beispiele für Angriffsformen sind:

Routing-Attacken, bei denen der Datenfluss über den Rechner des Angreifers gelenkt wird, Session Hijacking d. h. Übernahme von TCP-Verbindungen, Einspielen von Spionage- und Schadprogrammen oder Passwortattacken. für die Analyse des Bedrohungspotentials ist es zudem wichtig, den Kreis potentieller Angreifer zu kennen. So etwa belegen Untersuchungen wie in [SCHUHMANN 03] das hohe Gefahrenpotential durch Innentäter, d. h. Mitarbeiter des eigenen Unternehmens.

Aufgrund des großen Umfangs kann in dieser Arbeit nur ein kleiner Einblick in die Thematik Netzwerksicherheit vermittelt werden, weiterführend wird auf einschlägige Literatur wie beispielsweise [BUSCH 02], [DFAM 04], [IAONA 04], [KYAS 00] verwiesen. Aufgrund der dargestellten Gefahren ist es unablässlich, zusammen mit der Nutzung von Remote-Technologien auch entsprechende Maßnahmen zu ergreifen, um Kommunikationsverbindung abzusichern und Unberechtigte von internen Netzen auszuschließen. Weiterführend wird ein kurzer Überblick der Maßnahmen nach aktuellem Stand der Technik vermittelt.

5.4.2 Gegenmaßnahmen

Gegenmaßnahmen zur Absicherung der Kommunikation zwischen zwei privaten Netzen über öffentliche Netze lassen sich in zwei grundsätzliche Maßnahmen einteilen: Absicherung der Datenübertragung, beispielsweise durch Verschlüsselungsmechanismen sowie Analyse und Filterung von Datenpaketen an den Netzwerkgrenzen durch Firewalls. Hinzu kommen noch diverse ergänzende, jedoch nicht minder wichtige Mechanismen wie Revisions-Systeme, Intrusion-Detection-Systeme oder Honeypot-Systeme. Nicht minder wichtig beim Einsatz solcher Systeme ist eine durchdachte Netzwerkarchitektur, um letztendlich eine zuverlässige Abschottung gegen Gefahren von außen und innen zu erreichen.

In dieser Arbeit wird das Vorhandensein von Verschlüsselungs-Systemen, Firewall-Systemen und weiteren Mechanismen als gegebene Größe betrachtet, die in der Regel durch die Firmenpolitik bzw. die verantwortliche IT-Abteilung in einem Unternehmen festgelegt ist. Der Endanwender hat nur wenig Einfluss auf diese gegebenen Parameter. Für das Remote-Framework bedeutet dies, dass es flexibel ausgelegt sein muss, damit eine zuverlässige Funktionsweise in Verbindung mit unterschiedlichsten Sicherheitssystemen gewährleistet ist. Mit entsprechenden Lösungskonzepten beschäftigen sich die nachfolgenden Ausführungen.

5.4.3 Infrastrukturelle Ausprägungen

Die unkomplizierte Anwendung des Remote-Frameworks in bestehenden Netzwerk- und Sicherheitsarchitekturen ist von enormer Wichtigkeit. Auf Basis einer Studie, die sich mit der sicheren Anwendung von Ethernet und Remote-Technologien im Zusammenhang mit Automatisierungstechnischen Maschinen

und Anlagen beschäftigt [DFAM 04], werden deshalb zwei grundlegende Szenarien bestimmt. Diese stellen nach aktuellem Stand der Technik adäquate und standardisierte Möglichkeiten für den Aufbau der Kommunikationsverbindung zwischen den beiden Werkzeuginstanzen dar:

1. Aufbau der Kommunikationsverbindung über ein öffentliches Netz⁸¹ wie beispielsweise das Internet.

Das Unternehmensnetzwerk, bzw. kritische Subnetze sind mittels Firewalls gegenüber öffentlichen Netzen abgeschottet. Um sichere Kommunikation zu gewährleisten, ohne das interne Netz für den Zugang von außen zu öffnen, wird ein VPN⁸²-Tunnel zwischen den Endpunkten aufgebaut. Der VPN-Endpunkt kann dabei an verschiedensten Stellen eingerichtet sein: beispielsweise auf dem Entwicklungsrechner in Form einer Softwarekomponente oder als zentrales Gateway neben oder in einer Firewall. Abbildung 5-25 veranschaulicht dieses Szenario:

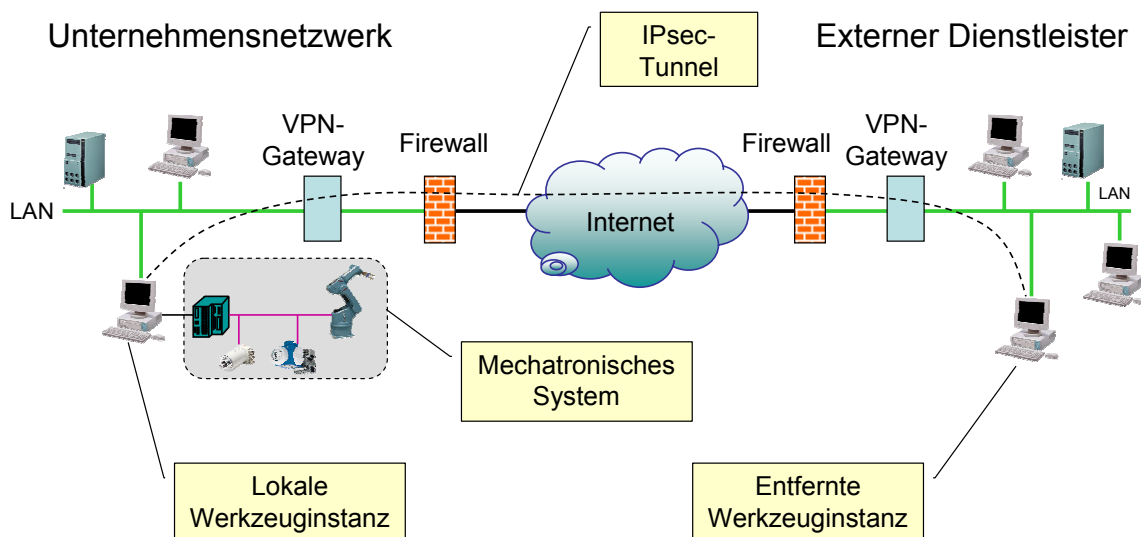


Abbildung 5-25: Verschlüsselte Kommunikation über öffentliche Netze

Der VPN-Tunnel ist mittels einer verschlüsselten Verbindung realisiert. Moderne Verschlüsselungstechniken gewährleisten dabei eine zuverlässige Absicherung gegen Angriffe aus öffentlichen Netzen. Es ist jedoch gerade bei Multimediaerverbindungen auf eine ausreichende Performance der Verschlüsselungskomponenten zu achten. Generell ist bei der Kommunikation über öffentliche Netze zu berücksichtigen, in wie fern diese die QoS-Anforderungen (vgl. Kapitel 5.2.4) von Multimediaübertragungen erfüllen.

⁸¹ Hierzu zählen auch umfangreiche Intranets innerhalb eines Unternehmens.

⁸² *Virtual Private Network*: Verbindung über ein öffentliches Netzwerk die durch Verschlüsselungsmaßnahmen die Merkmale einer privaten Verbindung aufweist.

2. Aufbau der Kommunikationsverbindung über dedizierte Leitungen wie beispielsweise ISDN oder analoge Telefonleitungen.

Auch das Telefonnetz stellt im Grunde ein öffentliches Netz dar. Die Verbindung wird jedoch exklusiv zwischen den entfernten Teilnehmern geschaltet und ist, außer in Fehlerfällen, für unberechtigte Dritte unzugänglich. Zu beachten ist bei diesem Szenario, dass über die Modemkopplung des Entwicklungsrechners eine Verbindung des internen Netzes mit einem Fremdnetz erfolgt. Der Rechner sollte demnach vom Unternehmensnetzwerk ganz getrennt oder durch eine Firewall sowie weitere Sicherheitssoftware (beispielsweise Virens Scanner) entsprechend abgesichert sein (vgl. Abbildung 5-26).

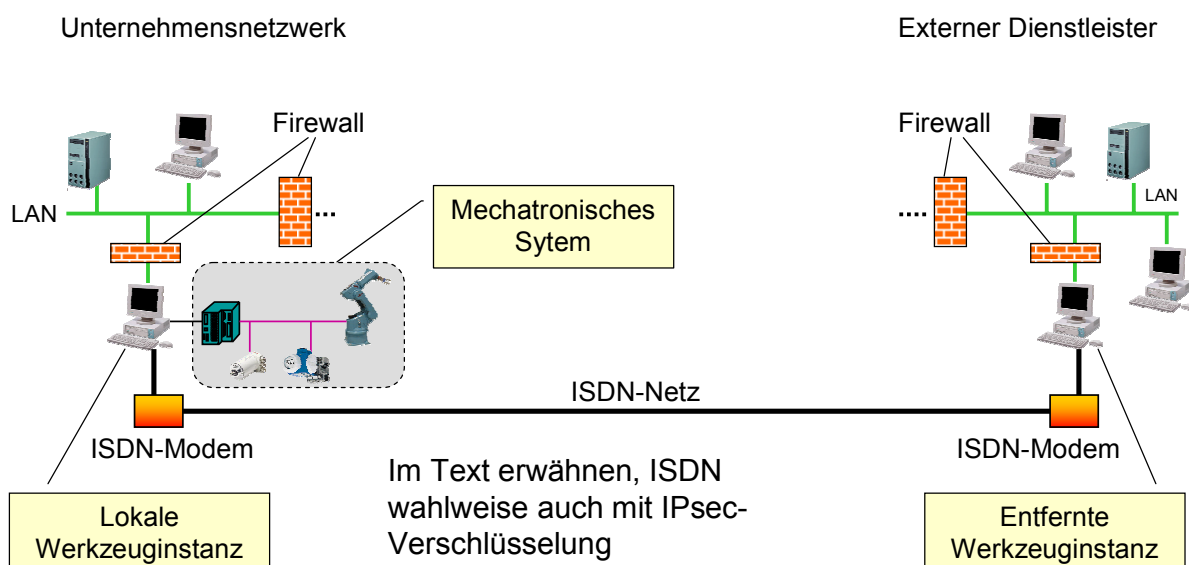


Abbildung 5-26: Kommunikation über dedizierte ISDN-Leitungen

Sollte der Anwendungsfall es erfordern, kann auch eine dedizierte Verbindung zusätzlich durch Verschlüsselungsmaßnahmen abgesichert werden. Dies kann beispielsweise durch den Einsatz von VPN-Komponenten zwischen Entwicklungsrechner und Modem, oder durch Einsatz softwarebasierter Verschlüsselungslösungen auf den Entwicklungsrechnern selbst erfolgen. Der Vorteil dedizierter Leitungen ist die garantierte Erfüllung der QoS-Anforderungen bei multimedialer Übertragung. Dem gegenüber stehen jedoch höhere Kosten insbesondere für hochperformante Verbindungen.

Die entwickelten Szenarien stellen repräsentative Mechanismen zur Koppelung von verteilten Werkzeuginstanzen dar. Sie dienen dem Anwender als Ausgangsbasis und können leicht den lokalen Begebenheiten in den Unternehmen angepasst werden. Dennoch stellt die Einrichtung sicherer Verbindungen über Netzwerkgrenzen für den Anwender immer einen gewissen Aufwand dar. Insbesondere der Abstimmungsaufwand mit den IT-Verantwortlichen in den

Unternehmen ist erfahrungsgemäß nicht zu unterschätzen. Da dies dem beschriebenen spontanen Charakter der Kollaboration entgegenwirkt, entwickelt das nachfolgende Kapitel weitere Alternativen für die Kopplung von Applikationen zum Zwecke der verteilten Kollaboration.

5.4.4 Integration des Remote-Frameworks in IT-Sicherheitsstrukturen

In vielen Fällen liegt die Freischaltung von Kommunikationswegen über die Grenzen des Unternehmensnetzwerks hinweg nicht in der Entscheidungsbefugnis des Anwenders einer verteilten Werkzeugapplikation. Dies erschwert oder blockiert vor allem die Nutzung multimedialer Technologien. Insbesondere die Übermittlung echtzeitrelevanter Datenströme mit Hilfe des RTP-Protokolls stellt gewisse Anforderungen an die Offenheit von Netzwerken. Als Ausweg aus dieser Situation können beispielsweise Proxy-Lösungen dienen, deren Handhabung jedoch ebenfalls nicht dem eigentlichen Anwender, sondern der IT-Administration obliegt. Somit besteht wiederum eine Abhängigkeit von der Gesamtinfrastruktur.

Die Unabhängigkeit des Datenkanals vom Multimediakommunikationskanals bietet hinsichtlich dieses kritischen Punktes entscheidende Vorteile: Bei der Gestaltung des nicht echtzeitfähig ausgelegten Datenkanals (vgl. Definition 4-2) existieren Variationsmöglichkeiten bezüglich einer Anpassung an bestehende restriktive Sicherheitseinrichtungen. Der Anwender kann also in einer restriktiven Infrastruktur in jedem Fall die Werkzeugkopplung nutzen, auch wenn er im Extremfall auf multimediale Unterstützung verzichten muss. Als Alternative kann in vielen Fällen zumindest eine herkömmliche Audioverbindung über das Telefon eingerichtet werden. Zur Anpassung an verschiedene Sicherheitskonfigurationen werden für den Datenkanal drei Modi spezifiziert, die in Abhängigkeit von vorgegebenen Firewall-Konfigurationen nutzbar sind. Zusätzlich werden die Vor- und Nachteile des jeweiligen Modus hinsichtlich der Performance und der Firewall-Kompatibilität betrachtet.

1. Callback mittels http-Anfragen

Bei dieser Lösung stellt sowohl das lokale als auch das entfernte Framework sämtliche Funktionen in Form von Webservices, basierend auf dem http-Protokoll zur Verfügung. Der Aufruf der Webservices über http-Anfragen muss folglich auf beiden Rechnern möglich sein (vgl. Abbildung 5-27). Dies erfordert auf beiden Seiten die Freischaltung des http-Ports (typischerweise Port 80) für den Remote-Zugriff auf den Rechner mit der Werkzeuginstallation in der Firewall.

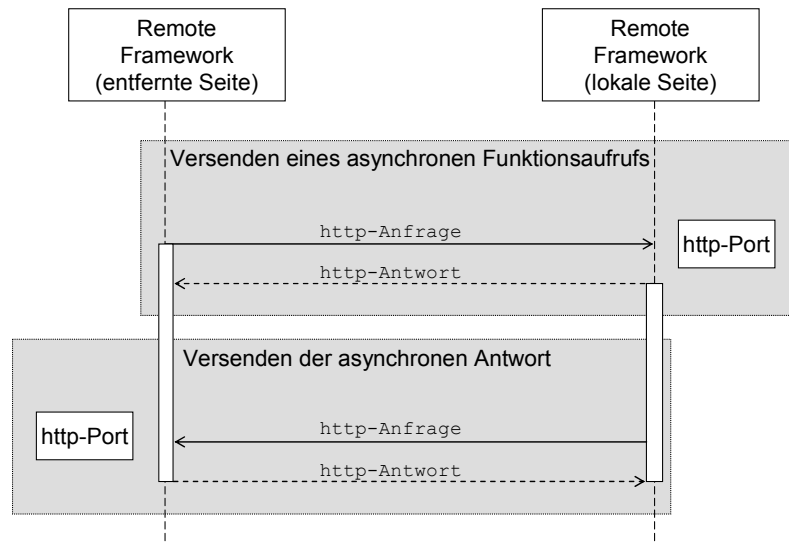


Abbildung 5-27: Modus 1: Callback unter Einsatz des http-Protokolls

Firewall-Konfiguration: Zugriff von außen über den http-Port erlaubt, Keine Content-Filterung oberhalb Schicht 5.

Vorteile: Einfachstes Szenario für die Nutzung von Webservices, performant, keine zyklische Abfrage der Antworten nötig, resistent gegen Filter auf Applikationsebene.

Nachteile: Jeweils neuer Verbindungsaufbau bei jedem Funktionsaufruf (wirkt sich bei einer Vielzahl kurzer Anfragen unter Umständen negativ auf die Performance aus).

2. SOAP über TCP

Es wurde bereits erwähnt, dass SOAP die Verwendung unterschiedlicher unterlagerter Protokolle unterstützt. Die Verwendung von TCP als verbindungsorientiertes Protokoll ermöglicht den Aufbau einer stehenden Verbindung aus einem Firmennetzwerk heraus. Die andere Seite muss die Annahme einer solchen Verbindung gestatten. Verbindungen von innen nach außen werden in aller Regel weniger restriktiv gehandhabt. Da moderne Firewalls den Status einer Verbindung speichern, ist nach dem Verbindungsaufbau der Kommunikationsfluss in beide Richtungen möglich, siehe Abbildung 5-28. Funktionen können über SOAP folglich auf beiden Seiten aufgerufen werden.

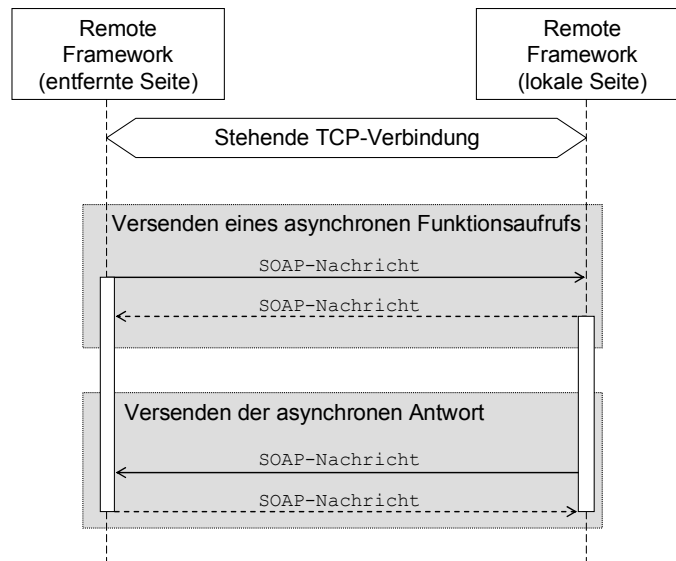


Abbildung 5-28: Modus 2: SOAP unter Verwendung des TCP-Protokolls

Firewall-Konfiguration: Verbindungsaufbau von innen nach außen auf beliebigen Ports erlaubt, kein Filterung auf Applikationsebene oberhalb Schicht 4.

Vorteile: Performante Lösung, keine zyklische Abfrage der Antworten nötig.

Nachteile: Funktioniert nicht, wenn die Firewall eine Filterung auf Applikationsebene vornimmt, permanente aktive Verbindung auch, bei wenigen Anfragen in großen Zeitabständen.

3. Passive Verbindung

Im passiven Modus erfolgt das Absetzen von http-Anfragen nur in eine Richtung, nach dem Client-Server-Prinzip. Der Client stellt keine Schnittstellen zur Verfügung, an die der Server Rückantworten senden muss, sondern fragt, nach absetzen eines Funktionsaufrufs, solange den Server zyklisch nach der asynchronen Antwort ab, bis diese zur Verfügung steht. Ein solches Vorgehen stellt die Firewall-freundlichste Lösung dar, da es im Grunde dem Web-Surfen mit einem Browser entspricht, der eine Web-Seite wiederholt aktualisiert. Das zyklische Abfragen der Antworten verursacht jedoch unnötiges Datenaufkommen und führt zu Verzögerung bei der Antwortübermittlung (vgl. Abbildung 5-29).

Firewall-Konfiguration: http-Anfragen von innen nach außen erlaubt (entspricht dem Abruf von HTML-Seiten mit einem Web-Browser). Keine Filterung auf Applikationsebene oberhalb Schicht 5.

Vorteile: Größtmögliche Firewall-Transparenz

Nachteile: Unnötiger Datenverkehr (insbesondere bei langen Zeiten der Inaktivität), Performance-Verlust durch Verzögerungen, Möglicherweise unverträglich mit Intrusion-Detection-Systemen.

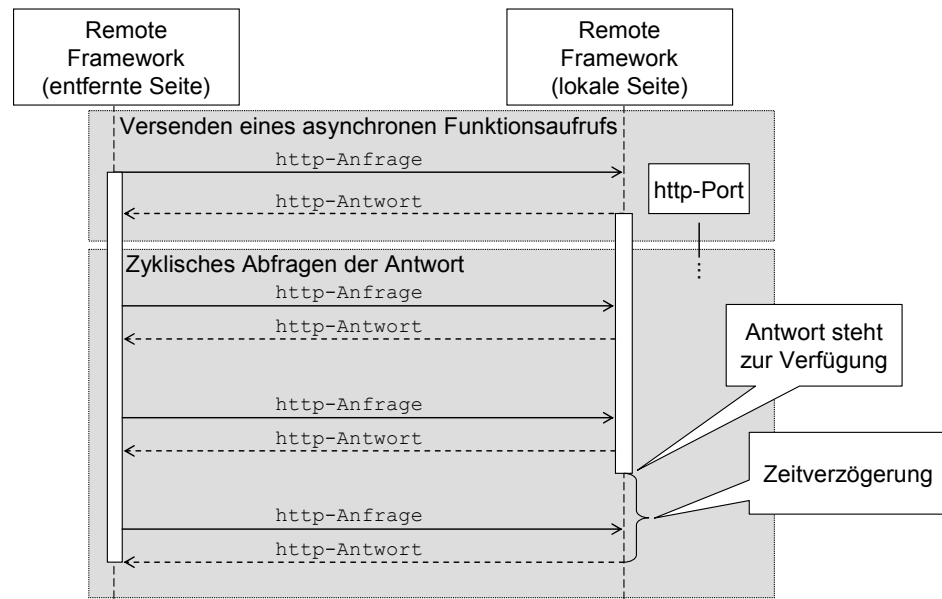


Abbildung 5-29: Modus 3: Zyklisches Abfragen der Antworten durch den Client

Das Kommunikationsmodul des Remote-Frameworks unterstützt sämtliche der spezifizierten Modi. Je nach Bedarf kann der Anwender vom zentralen GUI aus den Modus wechseln und das Framework auf diesem Weg an die vorgegebene Sicherheitsinfrastruktur anpassen. Grenzen sind jedoch letztendlich gesetzt, falls Sicherungssysteme eine Content-Filterung oberhalb des http-Protokolls auf Schicht 5 des ISO/OSI-Modells vornehmen. In diesem Fall erfolgt eine Analyse und evtl. Blockade der Inhalte eines http-Requests, bzw. einer http-Response. Dies entspricht einer sehr restriktiven Firewall-Konfiguration. Beispielsweise werden nur HTML-Seiten von der Firewall akzeptiert, XML-Inhalte hingegen geblockt. Datenaustausch mit standardisierten Mechanismen ist dann über solche Konfigurationen in der Regel nicht mehr zu realisieren. In einem solchen Fall müssen zusätzliche infrastrukturelle Maßnahmen getroffen werden (vgl. Kapitel 5.4.3).

5.5 Zusammenfassung

Das vorangegangene Kapitel beschreibt die Umsetzung des in Abschnitt 4 erarbeiteten Lösungskonzepts. Weiterhin erfolgt die Bereitstellung von Lösungsmustern um das Remote-Framework möglichst einfach für die verteilte Nutzung von Werkzeugen in den integrativen Phasen der Mechatronikentwicklung einzusetzen.

Das Kapitel löst zu Beginn die Problematik des granular gestuften Zugriffs auf die Funktionalität eines Werkzeugs von einem geografisch entfernten Standpunkt aus. Hierfür wurde die Sicherheitsnorm IEC 61508 als geeignete Ausgangsbasis identifiziert, um eine Einordnungssystematik für die Werkzeugfunktionalität in Zugriffsklassen zu entwickeln.

Im weiteren Verlauf erfolgte die Auswahl der für den vorliegenden Anwendungsfall benötigten Multimediakomponenten. Zudem wurde eine Schnittstelle für die Integration universeller Multimediaapplikationen in das Remote-Framework definiert. Abschließend erfolgte eine analytische Betrachtung der verwendeten Multimediatechnologien vor dem Hintergrund der zur Verfügung stehenden Performanz der Kommunikationsverbindung.

Der dritte Teil des Kapitels erarbeitet eine detaillierte Spezifikation für die Komponenten des Remote-Frameworks auf Basis des entwickelten Gesamtkonzepts. Die Spezifikation erfolgt mittels statischer und dynamischer Modelle und beinhaltet ein generisches Konzept für das Fehlermanagement in verteilten Werkzeugapplikationen. Den Abschluss bildet die Auswahl geeigneter Basistechnologien für die Umsetzung des Frameworks.

Der letzte Teil des Kapitels gibt zunächst einen Überblick über die Gefahren und mögliche Gegenmaßnahmen bei der Nutzung von Remote-Technologien. Vor diesem Hintergrund wurden Lösungsmuster und repräsentative Szenarien erarbeitet, um das System einfach in bestehende IT-Infrastrukturen integrieren zu können.

Das Kapitel stellt die Basis sowohl für die Realisierung des Remote-Frameworks als auch für die Integration von Werkzeugapplikationen in das Framework dar. Gleichzeitig wird der Integration des Frameworks in bestehende IT-Infrastrukturen Rechnung getragen.

6 Evaluierung im Rahmen einer industriellen Anwendung

Nachdem in den vorangegangenen Ausführungen eine Lösung für die verteilte Kollaboration in den integrativen Phasen der mechatronischen Produktentwicklung erarbeitet wurde, stellt dieses Kapitel ein industrielles Anwendungsbeispiel der neuen Lösung vor. Der gewählte Anwendungsfall konzentriert sich auf die Entwicklung von intelligenten automatisierungstechnischen Geräten der Güter produzierenden Industrie, als wichtige Vertreter mechatronischer Produkte. Der Anwendungsfall wird zunächst dargestellt sowie die spezifischen Randbedingungen erörtert. Das Kapitel beschreibt anschließend die Migration eines dort verbreiteten Entwicklungs-Tools hin zu einem verteilt einsetzbaren Werkzeug mit Hilfe des entwickelten Remote-Frameworks. Den Abschluss bildet die Bewertung der neuen Lösung auf Basis einer industriellen Evaluierung sowie weitere analytische Betrachtungen.

6.1 Entwicklung von Geräten mit Feldbusschnittstelle

Sowohl in der Fabrik- als auch in der Prozessautomation haben sich nach heutigem Stand der Technik Feldbussysteme etabliert. Automatisierungstechnische Geräte, wie beispielsweise Aktoren, Sensoren oder Controller, werden deshalb in aller Regel mit einer Feldbusschnittstelle ausgestattet. Der Einsatz von Feldbussen bietet gegenüber einer analogen Signalübertragung eine Reihe von Vorteilen wie geringer Verkabelungsaufwand, höhere Auflösung der zu übertragenden Größen, keine Messwertverfälschungen und erweiterte Funktionalität zur Diagnose und Instandhaltung [RATJE 97], [WEHRES 98]. Über die technischen Vorteile hinaus ist der Feldbus auch bei Wirtschaftlichkeitsuntersuchungen der konventionellen Verdrahtung überlegen. Zusammenfassend lässt sich die Prognose stellen, dass zukünftig nahezu alle automatisierungstechnischen Feldgeräte über eine Feldbusschnittstelle verfügen werden. Abbildung 6-1 stellt die moderne Feldbustechnik der früher üblichen konventionelle Punkt-zu-Punkt-Verdrahtung gegenüber.

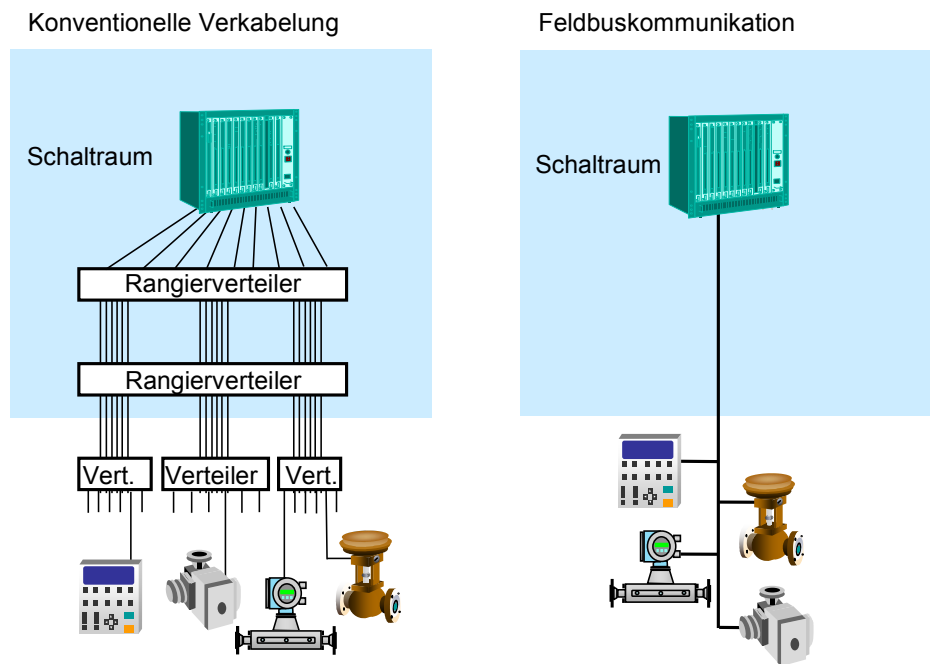


Abbildung 6-1: Vergleich zwischen konventioneller Verkabelung und Feldbustechnik

Beim Einsatz von Feldbustechnik in mechatronischen Geräten der Automatisierungstechnik kommt es nicht mehr allein auf die Korrektheit und die Zuverlässigkeit der Einzelgeräte an. Vielmehr müssen alle Geräte einer Anlage fähig sein, über das gemeinsame Übertragungsmedium korrekt zu kommunizieren.

6.1.1 Einführung in das Anwendungsgebiet

Tests sind ein wichtiges Instrument der Qualitätssicherung. Insbesondere bei Produkten, bei denen die exakte Einhaltung von Richtlinien oder Normen von hoher Bedeutung ist, sind Tests zur Überprüfung der Normkonformität unabdingbar. Im Bereich Automatisierungstechnik gilt dies in besonderem Maße für intelligente Feldgeräte, die über eine offene Feldbusschnittstelle verfügen. Die Konformität dieser Schnittstelle zu einer Kommunikationsspezifikation ist unabdingbare Voraussetzung für den fehlerfreien Einsatz des Feldgerätes innerhalb eines Multivendor-Feldbussystems, bei dem Geräte unterschiedlicher Hersteller gemeinsam betrieben werden. Dies setzt voraus, dass alle Geräte über eine Implementierung des Kommunikationsprotokolls verfügen, die konform zu einer gemeinsamen Kommunikationsspezifikation ist. Ob ein Gerät wirklich eine konforme Implementierung des Feldbusprotokolls beinhaltet, wird im Rahmen der so genannten *Konformitäts-* oder *Zertifizierungsprüfung* kontrolliert, eine wichtige Qualitätssicherungsaufgabe aller Feldbusorganisationen.

In Abhängigkeit von den testdurchführenden Parteien lassen sich grundsätzlich drei Klassen von Tests unterscheiden [KUTTIG+ 03]:

- *First-Party-Test*: Die Prüfung wird von den gleichen Personen durchgeführt, die auch an der Entwicklung des Gerätes beteiligt gewesen sind.

- *Second-Party-Test*: Die Prüfung wird vom Gerätehersteller durchgeführt, jedoch sind Prüfende und die an der Entwicklung des Gerätes Beteiligten, verschiedene Personen. Diese Form der Prüfung wird meist durch eine spezielle Prüfabteilung des Geräteherstellers vorgenommen.
- *Third-Party-Test*: Die Prüfung wird durch eine Prüfinstitution durchgeführt, die unabhängig vom Hersteller eines Gerätes ist (*Zertifizierungsprüfung*).

Eine Geräteprüfung gemäß dem First-Party-Test oder dem Second-Party-Test wird auch als *In-House-Test* bezeichnet, der Third-Party-Test als *Out-Door-Test*. Als problematisch für die Unternehmen erweist sich die Vorbereitung auf die *Zertifizierungsprüfung* in Form von vorbereitenden *In-House-Tests*. Derartige Prüfungen gehören üblicherweise nicht zu den Kernkompetenzen der Gerätehersteller. Sie erfordern neben einer detaillierten Kenntnis des verwendeten Kommunikationsprotokolls einen relativ hohen Aufwand für die Einarbeitung in die hierfür verwendeten hochspezialisierten Testmethoden und Testtools sowie das Vorhalten von entsprechendem Prüf-Know-how. Folglich sind die eigentlich erforderlichen Testaktivitäten bei den Geräteherstellern nur schwach ausgeprägt und die Durchführung der vorbereitenden Konformitätsprüfungen durch Gerätehersteller häufig mangelhaft. Dies wird durch die Erfahrung unabhängiger Prüflabore bestätigt, wonach die überwiegende Zahl der angelieferten Geräte den Zertifizierungstest nicht im ersten Anlauf besteht. Langwierige Korrekturschleifen sind die Folge, was einerseits die Produkteinführung verzögert und andererseits höhere Kosten für Nachentwicklung und erneute Prüfung induziert.

6.1.2 Rollenverteilung beim Zertifizierungstest

Die Entwicklung von intelligenten mechatronischen Feldgeräten gliedert sich, wie bereits dargestellt, in verschiedene Entwicklungsphasen. Zur Unterstützung der einzelnen Entwicklungsphasen kommt eine Vielzahl von Rechnerwerkzeugen zum Einsatz. Für die effiziente Nutzung dieser Werkzeuge ist ein nicht unerheblicher Einarbeitungsaufwand zu leisten, der sich nur dann bezahlt macht, wenn die Werkzeuge häufig genutzt werden. Bei Werkzeugen, die nur gelegentlich gebraucht werden, ist jeder Einsatz mit einer neuerlichen Einarbeitungsphase verbunden, worunter die Akzeptanz und damit die Verwendung eines Tools leidet. Die Kernkompetenz der Gerätehersteller liegt typischerweise im Bereich der Geräteentwicklung, also schwerpunktmäßig in den Phasen Analyse, Entwurf und Fertigung bzw. Implementierung. Auch entwicklungsbegleitende analytische Qualitätssicherungsmaßnahmen wie Modultests, Review-Verfahren und Funktionstests, insbesondere auf Applikationsebene, werden im Allgemeinen von den Geräteherstellern beherrscht und angewendet. Problematischer erweist sich die Situation bei Prüfungen, die nur in größeren und unregelmäßigen Zeitintervallen durchgeführt werden.

Da der Entwickler eines Gerätes ein natürliches Interesse daran hat, dass seine Entwicklungsarbeit erfolgreich gewesen ist, wird ein, wenn auch nur rudimentärer, First-Party-Test immer stattfinden. Eine Prüfung, die ausschließlich diese Art von Test beinhaltet, ist jedoch aus verschiedenen Gründen kritisch zu sehen: Ein Entwickler wird sich bei der Durchführung der Prüfung auf Aspekte konzentrieren, die aus seiner Sicht wichtig sind. Andere Aspekte, die ihm weniger bedeutungsvoll erscheinen oder die bei der Entwicklung gänzlich übersehen worden sind, werden mit hoher Wahrscheinlichkeit nicht geprüft, wodurch Fehler unerkannt bleiben. Fehlinterpretationen der Spezifikation und ein daraus resultierendes von der Spezifikation abweichendes Verhalten des Prüflings wird bei der Prüfung nicht als solches identifiziert, weil hier gegen die gleichermaßen fehlinterpretierte Spezifikation geprüft wird.

Zur Vorbereitung der Zertifizierungsprüfung, im Rahmen eines First- oder Second-Party-Tests, führen die Hersteller in aller Regel Vorprüfungen durch, für die spezifische Prüfwerkzeuge zum Einsatz kommen. Wie bereits oben erläutert ist eine solche Vorprüfung nur ein Teil innerhalb des gesamten Entwicklungsprozesses eines Feldgerätes. Folglich ist es außerordentlich schwierig, Know-how über den Test und die Bedienung der professionellen Prüfwerkzeuge so verfügbar zu halten, dass eine ausreichende Prüfungsdurchführung jederzeit und ohne nennenswerte Einarbeitung möglich ist. Aus diesen Gründen mangelt es auf Seiten der Feldgerätehersteller häufig sowohl an den spezialisierten Prüfwerkzeugen als auch am Expertenwissen über deren Anwendung.

Feldgeräte mit offenen Feldbusschnittstellen müssen üblicherweise hinsichtlich ihrer Konformität zum Kommunikationsstandard durch unabhängige Prüflabore zertifiziert werden. Das Prüflabor ist auf derartige Prüfungen spezialisiert und verfügt über entsprechende, häufig im Rahmen von Forschungsprojekten erstellte Prüfwerkzeuge. Darüber hinaus zeichnen sich Prüflabore durch ein hohes Maß an Expertise bezüglich der Bedienung dieser Prüfwerkzeuge und bezüglich der Prüfung selbst aus. Charakteristisch für solche Prüfinstitutionen ist, dass sie auf ein ausgewähltes Kommunikationssystem spezialisiert sind und die dort geprüften Geräte von unterschiedlichen Herstellern stammen. Die Anzahl der geprüften Geräte ist in der Regel deutlich höher, als dies bei Prüfabteilungen der Gerätehersteller der Fall ist. Damit ist zu erwarten, dass beim Third-Party-Test ein breiteres Spektrum an Prüferfahrung gesammelt werden kann und auf der Seite der Prüflabore detaillierte Kenntnisse der Kommunikationsspezifikation vorliegen. Prüferfahrung und detaillierte Kenntnis der Kommunikationsspezifikation bilden die Grundlage, um Prüfscenarien zu spezifizieren, die einen möglichst großen Umfang der Kommunikationsspezifikation abdecken.

Die bisherigen Ausführungen zeigen zudem die unterschiedlichen Rollen, die der Geräteentwickler und ein unabhängiger Experte bezüglich ihres Wissensstandes und der auszuführenden Tätigkeiten während des Entwicklungsprozesses einnehmen. Diese sind in Tabelle 6-1 veranschaulicht:

	Entwickler	Externer Prüfexperte
Wissensstand	<ul style="list-style-type: none"> ▪ Detailkenntnis der Implementierung ▪ Detailkenntnis der Hardware ▪ Detailkenntnis der Gerätefunktionalität ▪ Grundkenntnis der Prüfwerkzeuge 	<ul style="list-style-type: none"> ▪ Detailkenntnis der Koommunikationsspezifikation ▪ Detailkenntnis der Prüfspezifikation ▪ Umfangreicher Erfahrungsschatz ▪ Detailkenntnis der Prüfwerkzeuge
Tätigkeiten	<ul style="list-style-type: none"> ▪ Ausführen von Testfällen ▪ Bedienung des Gerätes (Upper-Test) ▪ Grundparametrierung 	<ul style="list-style-type: none"> ▪ Erstellen und Editieren von Testfällen ▪ Ausführen von Testfällen ▪ Herbeiführen spezieller Zustände: Spezielle Parametrierung, Kommandozeilenoperationen

Tabelle 6-1: Rollenverteilung der am Test Beteiligten Personen

6.1.3 Das Prüfwerkzeug *MoTest*

Die Durchführung von Tests zur Konformitätsprüfung der Feldbusschnittstelle stellt ein umfangreiches und zeitaufwändiges Unterfangen dar. Aufgrund der Vielzahl von Zuständen und Zustandsübergängen, die eine automatisierungstechnische Komponente beinhaltet, existiert eine große Anzahl von Testfällen die zur Konformitätsüberprüfung notwendig sind. Die Testfälle müssen zudem unter harten Echtzeitbedingungen durchgeführt werden, um auch das korrekte zeitliche Verhalten der Feldbusschnittstelle zu überprüfen. Zudem muss, wie bei jeder ordnungsgemäßen Testdurchführung, stets eine Reproduzierbarkeit der Tests gewährleistet sein. Hierfür ist es für jeden Testfall erforderlich, das Gerät durch entsprechende Befehle in einen definierten Zustand zu bringen. Allein die aufgezählten Umstände verdeutlichen, dass Testfälle in dieser Anwendungsdomäne nicht manuell durchführbar sind, sondern es der Unterstützung durch entsprechende Testwerkzeuge bedarf. Eine solche Unterstützung wird weiterhin dadurch motiviert, dass ein Werkzeug zusätzlich in der Lage ist, umfangreiches und komplexes Test-Know-how zu implementieren und für die testdurchführende Person zu kapseln.

Aufgrund dieser Tatsachen haben sich rechnerbasierte Werkzeuge zur Durchführung von Tests etabliert. Diese Werkzeuge stellen nicht nur die echtzeitfähige Ablaufumgebung für Testfälle dar, sondern bieten zudem die Möglichkeit der Protokollierung des Testdurchlaufs und der Auswertung der Ergebnisse. Ebenfalls üblich sind Mechanismen zur Verwaltung und Erstellung bzw. Editierung von Testfällen und Testsuiten. Im Rahmen dieses Kapitels konzentriert sich der Fokus auf das Prüfsystem *MoTest*⁸³. Es handelt sich hierbei um ein Rechnerwerkzeug, welches im Umfeld der Konformitätsprüfung von automatisierungstechnischen Geräten mit Feldbusschnittstelle weite Verbreitung erfahren hat.

⁸³ *Modulares Testsystem*

Dies betrifft sowohl die Seite der unabhängigen zertifizierenden Organisationen als auch die Seite der Geräteentwickler in den Unternehmen. MoTest stellt ein komplexes Prüfsystem mit umfangreichen Funktionen dar: Innerhalb einer Ablaufumgebung können verschiedene Module integriert werden, welche die Anbindung an unterschiedliche Kommunikationssysteme wie beispielsweise Profibus, CAN oder Ethernet-basierte Systeme ermöglichen. Parallel zur Ablaufumgebung stehen Module für die Verwaltung, Auswertung, Erstellung und Editierung von Testfällen und Testprojekten zur Verfügung. Zusätzlich existieren Module zur Konfiguration und Parametrierung eines SUT⁸⁴ sowie Module für den direkten manuellen Eingriff in das SUT.

Nachfolgende Abbildung 6-2 gibt einen Überblick der Komponenten des Prüfsystems.

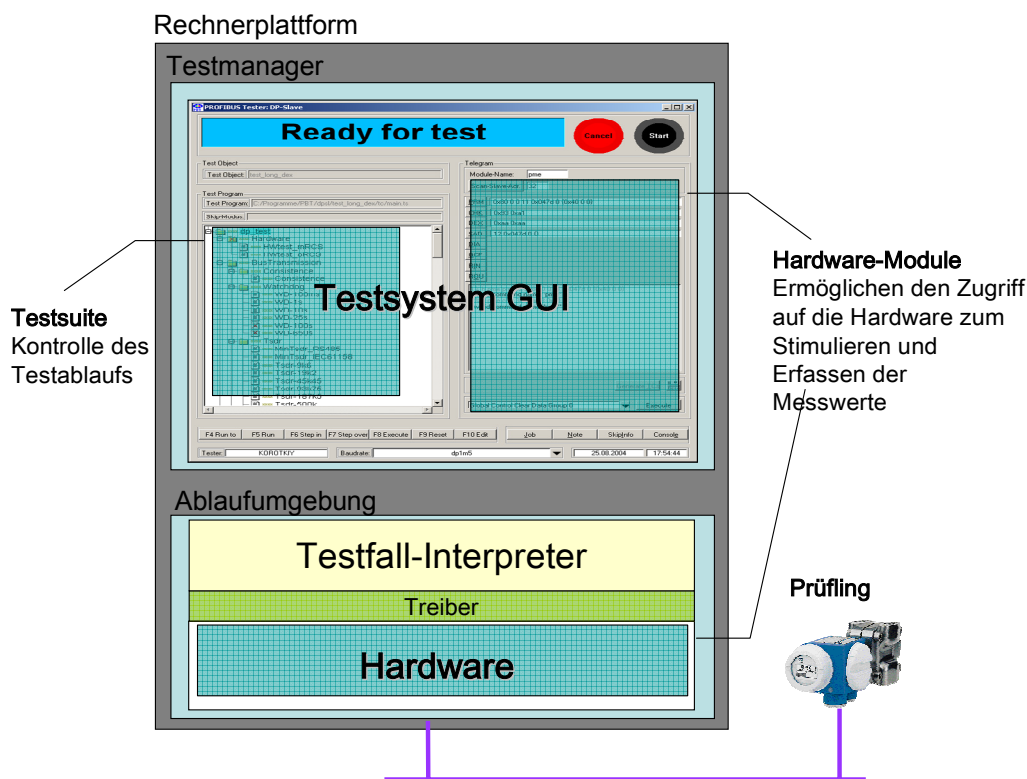


Abbildung 6-2: Aufbau des Prüfsystems MoTest

Die Abbildung zeigt schematisch den Aufbau des Tools: Im GUI des Testsystem stehen dem Prüfer sämtliche Funktionen zur Testdurchführung in Form von Modulen zur Verfügung. Diese können je nach Bedarf aktiviert werden. Testfälle werden vom Testmanager an die Ablaufumgebung übermittelt, dort interpretiert und unter Echtzeitbedingungen ausgeführt. Über die zuvor beschriebenen Treiber-Module werden aus dem Testfall heraus verschiedene Hardwareanschlutungen angesprochen, die eine physikalische Verbindung zum mechatronischen Gerät darstellen. Während der Durchführung eines Testfalls zeichnet die

⁸⁴ System Under Test

Ablaufumgebung ein Testprotokoll auf. Dieses wird nach erfolgtem Durchlauf an den Testmanager zurück übergeben und steht dort für die Auswertung zu Verfügung.

6.1.4 Verteilte Kollaboration beim Zertifizierungstest

Die in Kapitel 6.1.1 geschilderten Probleme während der Vorbereitung auf den Zertifizierungstest sowie die identifizierte Rollenverteilung zwischen Prüfer und Entwickler stellen ein prädestiniertes Anwendungsgebiet für die gewinnbringende Nutzung von verteilter Kollaboration dar. Es lassen sich die zwei grundlegende Szenarien identifizieren:

1. Durchführung der Prüfung durch den Entwickler – Unterstützung durch den externen Prüfexperten

Für den Entwickler muss es das Ziel sein, durch entsprechende First-Party-Tests so viele Fehler wie möglich zu eliminieren. Umfangreiches Prüf-Know-how kann hier jedoch nicht vorausgesetzt werden. Der Entwickler nutzt folglich nur grundlegende Funktionen des Test-Tools. Das bedeutet, er führt eine Serie von Testfällen aus, die er für sinnvoll erachtet und bekommt Fehlermeldungen sowie zugehörige Prüfprotokolle angezeigt. Beim Auftreten von Sonderfällen stellt ihn die Interpretation der Protokolle vor Schwierigkeiten. Das Wissen über weiterführende manuelle Testschritte zur Fehlerlokalisierung kann auf Seiten des Entwicklers nicht vorausgesetzt werden. Diese Schritte können sinnvoller Weise in Zusammenarbeit mit einem erfahrenen Prüfexperten durchgeführt werden, der beispielsweise in einem unabhängigen Prüflabor tätig ist.

Ein ähnliches Szenario lässt sich während des Second-Party-Tests identifizieren. Der dortige Prüfer ist zwar erfahrener im Umgang mit Prüfwerkzeugen und der Interpretation von Prüfprotokollen, jedoch kann nicht die gleiche Erfahrung wie die eines unabhängigen Prüfexperten vorausgesetzt werden, der ungleich mehr Geräte verschiedenster Hersteller bearbeitet. In komplexen Sonderfällen wird auch der Second-Party-Prüfer an Grenzen stoßen. Durch die gemeinsame Durchführung der Tests mit einem externen Prüfexperten lassen sich Fehler besser eingrenzen und die Ursachen schneller identifizieren.

2. Durchführung der Prüfung durch den Prüfexperten – Unterstützung durch den Entwickler

Tauchen während des Gerätetests bei einer unabhängigen Prüfinstitution Fehler auf, so ist es das Ziel, nicht nur das Auftreten des Fehlers zu vermerken und den Test abzubrechen, sondern die Ursache des Fehlers zu identifizieren. Unter Umständen können Fehler sogar die Fortsetzung eines Tests verhindern, was dazu führt, dass nicht alle vorhandenen Fehler in einem Durchlauf aufgedeckt werden können, sondern wiederholte Korrekturschleifen entstehen. Deshalb liegt es auch im Interesse des Dienstleisters, in

diesem Falle der Prüfinstitution, Fehlerursachen zu finden und gegebenenfalls Fehler zu umgehen. Hierbei ist der Prüfer auf die Unterstützung des Entwicklers angewiesen. Durch verschiedene Einstellungen bzw. Interaktionen mit dem Gerät, die durchaus herstellerspezifischer Natur sein können, hat der Entwickler die Möglichkeit den Prüfer durch sein Detailwissen über die Implementierung zu unterstützen.

Das gleiche Szenario lässt sich auch auf Second-Party-Tests projizieren. Auch hier sind Prüf- und Entwicklungsabteilung oftmals geografisch verteilt. Durch enge Kollaboration von Prüfer und Entwickler an verteilten Frontends kann das TestszENARIO optimiert und Fehler detaillierter analysiert werden, als wenn der Prüfer dies alleine durchführen müsste.

6.2 Realisierung des verteilten Prüfwerkzeugs

Nachfolgend wird die prototypische Realisierung der verteilten Kollaboration mit dem Prüfwerkzeug MoTest unter Einsatz des spezifizierten Remote-Frameworks vorgestellt. Das Prüfsystem lag zunächst in einer Form vor, die den in Kapitel 5 gestellten Anforderungen an Modularität weitgehend entsprach. Dies bedeutet: Die Trennung zwischen Ablaufumgebung und Management-Komponenten ist ausgeprägt und die Komponenten der Benutzeroberfläche sind modular strukturiert. Die nachfolgenden Unterkapitel beschreiben zunächst die Implementierung der Schnittstellen des Remote-Frameworks sowie die Realisierung der werkzeugseitigen Komponenten, namentlich Dispatcher und modulares GUI. Anschließend beschreiben die Ausführungen die Realisierung der interpersonellen Kommunikation sowie die Umsetzung des Rollenmodells im modularen Frontend des Prüfsystems.

6.2.1 Schnittstellen des Remote-Frameworks

Nachdem in Kapitel 5 die Spezifikation der Schnittstellen des Remote-Frameworks erfolgte widmet sich dieser Unterabschnitt der Implementierung dieser Schnittstellen für ein konkretes Realisierungsbeispiel. Bereits bei der Schnittstellendefinition wurde die SOAP-Technologie als geeignetes Mittel für die Realisierung identifiziert. Dort wurde auch das http-Protokoll als standardisiertes Trägerprotokoll für die Übermittlung von SOAP-Nachrichten beschrieben. Ebenfalls wurden in Kapitel 5 verschiedene Realisierungsmöglichkeiten zur Koppelung zweier Instanzen des Remote-Frameworks aufgezeigt. Im Folgenden wird deshalb schwerpunktmäßig die Realisierung der Schnittstellen zwischen dem Remote-Framework und der Werkzeugapplikation, in diesem Fall das Prüfsystem MoTest, betrachtet.

Die Nutzung von SOAP-Funktionen in Verbindung mit http wird üblicherweise als Webservices bezeichnet. Dies bedeutet: ein http-Server verfügt über eine öffentliche Schnittstelle mit einem bestimmten Umfang an Funktionen. Soll eine

Funktion aufgerufen werden, so wird von einem http-Client ein http-Request mit einer SOAP-Nachricht an den Server geschickt. Die SOAP-Nachricht enthält den gemäß der SOAP-Spezifikation kodierten Funktionsaufruf, in Form eines XML-Strings. Der Server verarbeitet diesen String und startet die Ausführung der entsprechenden Funktion. Nach deren Abarbeitung wird die Antwort ebenfalls in Form einer SOAP-Nachricht als http-Response an den Client zurückgeschickt [SNELL+ 02].

Für die Nutzung von Webservices ist jedoch nicht der Einsatz eines vollständigen, öffentlichen http-Servers Voraussetzung. Vielmehr können Webservices auch für die interne Kommunikation zwischen Applikationen auf ein und derselben Rechnerumgebung eingesetzt werden. Zu diesem Zweck existieren so genannte http-Channels (http-basierte Kanäle), die nur für lokale Applikationen zugänglich sind. Die Nutzung von http für die interne Kommunikation wirft auf den ersten Blick die Frage auf, welche Vorteile der Einsatz eines solchen Protokolls mit entsprechendem Overhead beispielsweise im Vergleich zu TCP bringt. Neben der einfachen und einheitlichen Verwendbarkeit von Webservices sowohl für externe als auch interne Kommunikation korrespondiert das Request-Response-Prinzip von http in optimaler Form mit den spezifizierten asynchronen Funktionsschnittstellen. Des Weiteren findet bei einem SOAP-Aufruf in der Regel keine kontinuierliche Übermittlung von Datenströmen über die Funktionsschnittstelle statt, sondern Funktionen werden in erster Linie durch Benutzerinteraktionen in gewissen Zeitabständen aufgerufen. Gewöhnlich ist der Nutzdatenanteil auch weitaus größer als der http-Protokoll-Header. Verzögerungen durch den http-Overheads fallen also vergleichsweise wenig ins Gewicht.

Die Schnittstellen des Remote-Frameworks wurden auf einer Windows-Plattform mit Hilfe der Entwicklungsumgebung Visual Studio .NET der Firma Microsoft implementiert. Ein integrales Konzept der .NET-Umgebung ist die Unterstützung von WebServices bei der Realisierung verteilter Applikationen. Mit Hilfe dieser Unterstützung ist eine einfache Realisierung der Verbindung zwischen Prüfsystem und Remote-Framework auf der Basis von Webservices möglich.

6.2.2 Softwarearchitektur des Prüfsystems

Das Prüfsystem MoTest ist eine auf Windows-Plattformen lauffähige Applikation. Zum Zeitpunkt der Remote-Erweiterung implementierte die Applikation bereits eine strikte Trennung zwischen der Ablaufumgebung und den Management-Komponenten (GUI). Durch den Einsatz geeigneter Hardwareanschlüssen für feldbusfähige Geräte, genügt die Ausführung der Ablaufumgebung als hochpriorer Windows-Task den vorherrschenden Echtzeitanforderungen während des Tests gängiger Feldbussysteme. Für zukünftige striktere Anwendungen ist aufgrund der Trennung auch eine Portierung auf echtzeitfähige Betriebssysteme einfach möglich. Die Implementierung der Laufzeitumgebung erfolgte in

der Programmiersprache C++. Testfälle werden in der Skriptsprache TCL erstellt, und von der Laufzeitumgebung interpretiert.

Die Management-Komponenten von MoTest wurden mit Hilfe der .NET-Entwicklungsumgebung von Microsoft erstellt und nutzen die .NET-Laufzeitumgebung. Auf dieser Ebene der Applikation herrschen keine Echtzeitbedingungen vor. Management-Komponenten und Ablaufumgebung kommunizieren ebenfalls über Webservices. Die Schnittstellen sind asynchron ausgeführt, sodass die Ausführung von Testfällen nicht das GUI der Management-Komponenten blockiert (vgl. Abbildung 6-3).

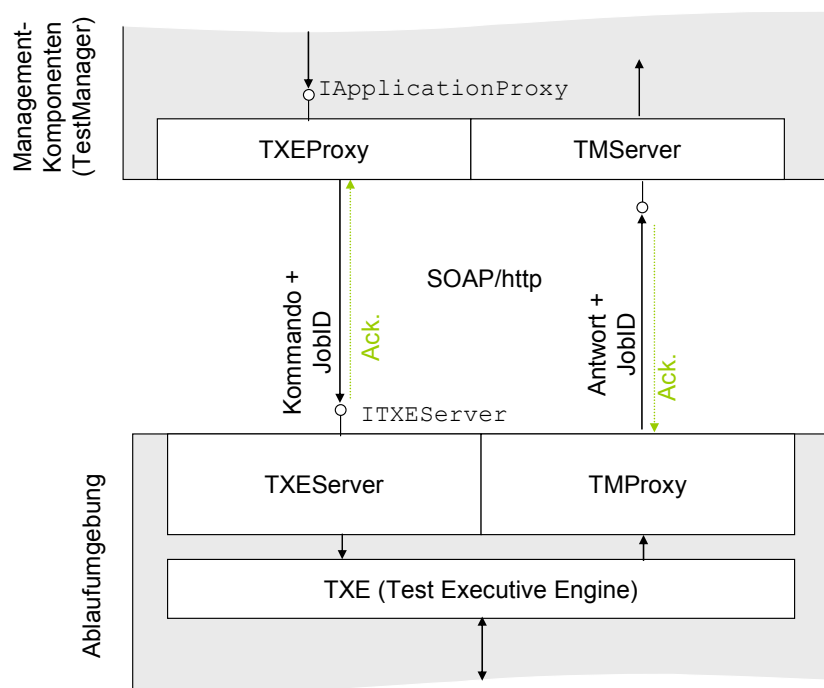


Abbildung 6-3: Basis-Architektur des Prüfsystems MoTest

Vor der Erweiterung der Applikation mit Remote-Funktionalität waren zwar modulare GUI-Elemente innerhalb der Management-Komponente vorhanden, diese hatten jedoch unmittelbaren Zugriff auf die Ablaufumgebung. Mit der Anbindung an das Remote-Framework wurde dieser direkte Zugriff abgeschafft und stattdessen die Dispatcher-Komponente in Form eines so genannten *Handlers* eingeführt. In diesem sind die im Verlauf dieser Arbeit spezifizierten Funktionen wie Message-Routing, Anbindung des Frameworks, Synchronisierungsfunktionen und Zugriffskoordination implementiert. Um die Realisierung des Handlers zu vereinfachen, wurden im Zuge der Umstellung sämtliche Schnittstellen zwischen dem Handler und den GUI-Komponenten an die für das Framework spezifizierte Aufrufschnittstelle angepasst. Auf diese Weise müssen im Handler keine weiteren Umsetzungsaktionen bei der Abarbeitung von Remote-Aufrufen erfolgen (vgl. Abbildung 6-4). Als positiver Nebeneffekt sind zudem einheitliche Schnittstellen für die Integration weiterer Module vorhanden.

Im Rahmen der prototypischen Umsetzung der erarbeiteten Lösung wurde bei der Implementierung darauf verzichtet, das Remote-Framework als eigenständige Applikation auszuführen. Um in der Erprobungsphase vor allem das Debugging zu erleichtern wurde das Framework als eigenständiges Modul innerhalb des Software-Projekts für das Prüfsystem implementiert (vgl. Abbildung 6-4).

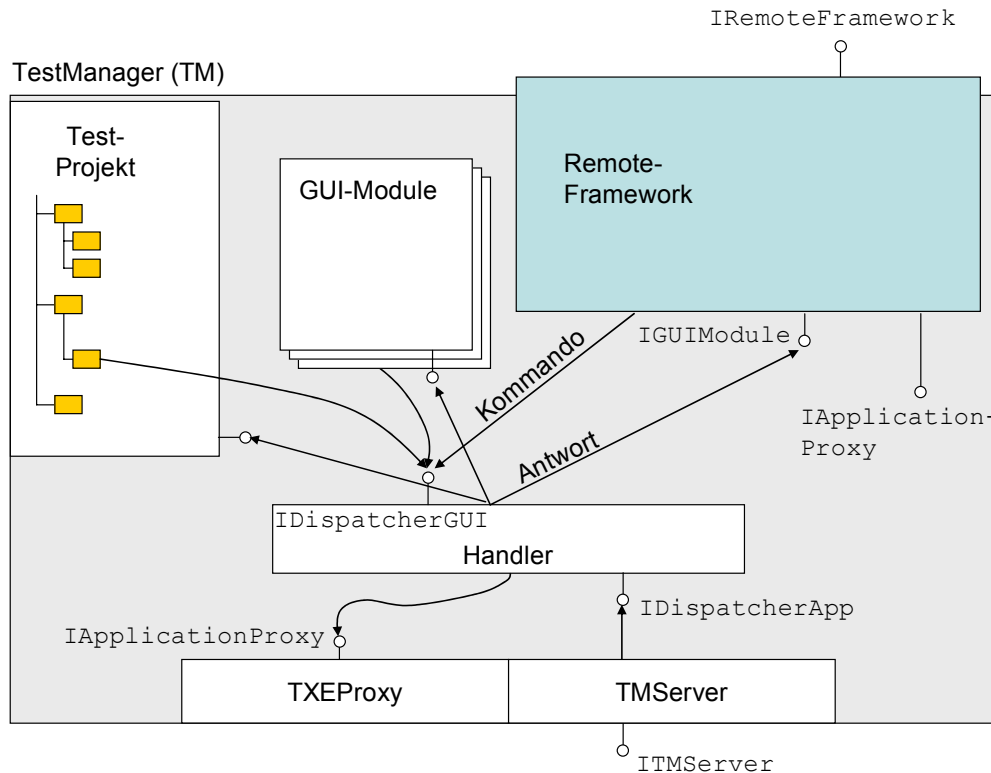


Abbildung 6-4: Verbindung von MoTest mit dem Remote-Framework

Trotz dieser implementierungstechnischen Abwandlung erfolgt die Anbindung des Frameworks an den Dispatcher des Prüfsystems gemäß den spezifizierten Schnittstellen.

6.2.3 Realisierung der interpersonellen Schnittstelle

Um die Multimediafunktionalität für das verteilte Prüfsystem zu realisieren erfolgte zunächst die Auswahl einer geeigneten Multimediaapplikation, welche die gewünschte Basisfunktionalität bereitstellt. Hierzu wurde eine Vielzahl von existierenden Applikationen und Bibliotheken hinsichtlich der in Kapitel 5.2 erarbeiteten Kriterien analysiert. Ein wesentlicher Faktor war die Festlegung auf das Betriebssystem Windows. Letztendlich fiel die Wahl auf das Programm *Netmeeting*, welches zusammen mit dem Web-Browser *Internet Explorer* von Microsoft ausgeliefert wird. *Netmeeting* beinhaltet sämtliche benötigte Grundfunktionalität und wird, ebenso wie der *Internet Explorer*, kostenlos von Microsoft zur Verfügung gestellt. Die Applikation beruht auf dem in Kapitel 5.2.4 vorgestellten ITU-Standard. Die Gründe die zur Entscheidung für *Netmeeting* als Multimedia-Applikation führten sind in Tabelle 6-2 aufgeführt:

Vorteile:	
Installation	Integriert in den Internet Explorer, kein zusätzlicher Installationsaufwand notwendig, praktisch auf jedem Windows-System vorhanden
Lizenzkosten	kostenlos
API	Umfangreiche, gut dokumentierte API auf Basis von DCOM-Schnittstellen
Support	Umfangreicher Web-Support in Form des Microsoft Developer Networks, Software-Updates erfolgen auf Wunsch automatisch über die integrierte Windows Update Funktion
Nachteile:	
Codecs	Beschränkte Auswahlmöglichkeiten für Codecs

Tabelle 6-2: Gründe für die Auswahl von Netmeeting

Der Nachteil der geringen Auswahlmöglichkeiten für verschiedene Codecs wird insofern ausgeglichen, dass Netmeeting Standard-Codecs verwendet, die von nahezu allen Multimediaapplikationen unterstützt werden. Der Multimedia-Wrapper für Netmeeting wurde als ein Modul des Remote-Frameworks ebenfalls auf Basis der .NET Laufzeitumgebung realisiert. Mechanismen zur Kapselung von DCOM-Schnittstellen für .NET-basierte Applikationen stellt die Entwicklungsumgebung bereit.

Abbildung 6-5 zeigt die Ausführung des GUI für die Multimediafunktionalität des Remote-Frameworks zusammen mit den GUI-Modulen von Netmeeting für die Darstellung der multimedialen Inhalte.

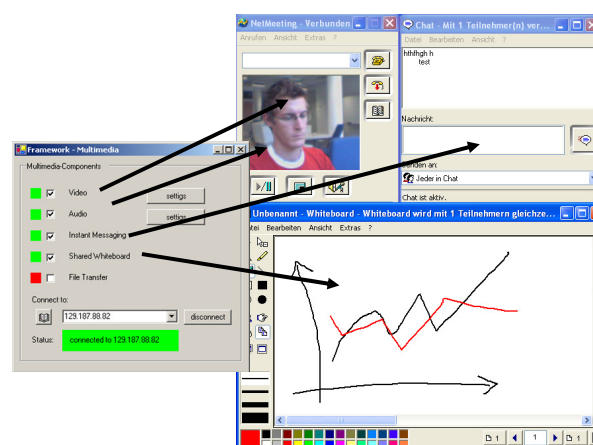


Abbildung 6-5: Multimedia-Steuerungs- und Ausgabefenster

6.2.4 Umsetzung des Rollenmodells im Prüfwerkzeug

In Kapitel 6.1.2 wurden diejenigen Rollen identifiziert, die während der integrativen Phasen von den Anwendern des Prüfsystems zum Test der

Feldbusschnittstelle mechatronischer Geräte eingenommen werden. Diesen Rollen gilt es nun die Funktionalität des Prüfsystems gegenüber zu stellen, was in Tabelle 6-3 erfolgt. Anschließend wird überprüft, in wie weit das modular ausgeführte GUI von MoTest diese Zuordnung umsetzen kann.

Grundfunktionalität	Zuständige Rollen
Erstellen und Verwalten von Testsuiten und Testprojekten	externer Experte
Ausführen von Testsuiten und einzelnen Testcases, Ergebniskontrolle	Entwickler, externer Experte
Parametrierung von Testcases	Entwickler
Editierung von Testcases	externer Experte
Auswertung von Logfiles	externer Experte
Konfiguration des Gerätes für den Test	Entwickler
Einlesen von Gerätebeschreibungen	Entwickler, externer Experte
Konsolenbefehle	externer Experte
Manuelles Auslösen von Feldbuskommandos	externer Experte

Tabelle 6-3: Zuordnung der Entwicklerrollen zur Basisfunktionalität von MoTest

Die Tabelle zeigt, dass der Entwickler hauptsächlich Funktionen nutzt um Testcases auszuführen und anschließend das Ergebnis zu begutachten. Zu diesem Zweck stellt das Prüfsystem ein GUI-Modul zur Verfügung welches die Testsuite in einem hierarchischen Baum abbildet. Über ein Kontextmenü werden Testcases oder Testgruppen gestartet. Das Bestehen oder Scheitern eines Tests wird innerhalb des Baumes durch grafische Symbole angezeigt. Da dem Entwickler in der Regel das nötige Hintergrundwissen über die Testspezifikation fehlt, bearbeitet dieser gewöhnlich keine Testcases oder nimmt tief greifende Analysen des Testprotokolls vor. Jedoch zählt die Parametrierung von Testcases zu seinen Aufgaben, da diese vom zu testenden Gerät abhängig sein kann. In diesem Zusammenhang ist auch das Einlesen einer Gerätebeschreibung zu sehen.

Ein externer Testspezialist besitzt tiefgreifendes Know-how über die Kommunikations- bzw. Prüfspezifikation. Er erstellt und editiert Testcases und ist in der Lage aus dem Logfile eines Testcases Rückschlüsse auf die Fehlerursache zu ziehen. Ebenso führt er elementare Operationen durch, um gezielt Zustände im Gerät herbeizuführen, die der Fehlersuche dienen können. Hierzu benötigt der Experte die Möglichkeit, manuell einzelne Befehle auf den Feldbus abzusetzen, bzw. die Hardware-Module des Prüfsystems über die Konsole manuell anzu-steuern.

Sämtliche in der Tabelle aufgeführten Funktionsgruppen von MoTest wurden im Zuge der Remote-Erweiterung in eigene GUI-Module ausgelagert. Je nach Relevanz sind die Module über eine zentrale Auswahlliste startbar. Dies ist vor allem dann der Fall wenn es sich um entwicklerspezifische Funktionen handelt, die ausgeführt werden, ohne dass der Anwender mit den Interna des Prüfsystems und der Prüfpezifikation vertraut ist. Module mit expertenbezogener, erweiterter Funktionalität sind erst über kontextabhängige Menüs oder weiterführende Schaltflächen startbar. Der Entwickler verliert sich folglich nicht in einer für ihn unnötigen Fülle von Funktions- und Modullisten. Module in beiden Frontends des verteilten Prüfsystems können jeweils individuell aktiviert werden. Auf diese Weise trägt das, entsprechend des Rollenmodells in Kapitel 4.2.2 gestaltete, verteilte Prüfsystem der Forderung nach angepassten Benutzerrollen Rechnung. Bei Bedarf kann das System zudem in den synchronisierten Modus geschaltet werden. Beide Parteien bekommen dann online jede Veränderung am Testprojekt angezeigt und können gleichberechtigt Aktionen ausführen. Im nicht synchronisierten Modus ist es möglich, Testprojekte ohne Zugriff der jeweils anderen Partei zu bearbeiten, auf diese Weise wird vorhandenes vertrauliches Wissen geschützt.

6.3 Bewertung des Remote-Frameworks

Um die Wertigkeit des entwickelten Remote-Frameworks unter Anwendung auf das Prüfsystem MoTest zu evaluieren wurden zwei Ansätze verfolgt: Zur Beurteilung grundlegender Parameter, wie beispielsweise Qualität der Audio/Videübertragung bei vorgegebener Bandbreite wurden Versuche unter Laborbedingungen durchgeführt. Um die industrielle Anwendbarkeit zu Demonstrieren wurde ein komplettes Testszenario bei der Firma Endress & Hauser, einem der führenden Hersteller von mechatronischen Geräten der Prozessindustrie, durchlaufen. Der Test wurde an einem Gerät am Firmenstandort Reinach in der Schweiz vorgenommen und durch das Zertifizierungslabor für Profibus Geräte am Standort München durchgeführt. Bestandteil war hierbei auch die Überprüfung der Anwendbarkeit des Zugriffsmodells auf das Prüfsystem.

6.3.1 Szenario der industriellen Anwendung

Im Sinne der Evaluierung der erarbeiteten Lösung erfolgte der Test des verteilten Prüfsystems MoTest zusammen mit einem Hersteller mechatronischer Geräte der Prozessindustrie. Die Firma Endress & Hauser ist einer der führenden Hersteller automatisierungstechnischer Geräte mit Feldbusanschlungen. Das Prüfsystem MoTest wird dabei von Endress & Hauser zur Vorbereitung des Zertifizierungstests für Profibus-fähige Geräte verwendet. Hierbei treten zwei Anwendungsszenarien auf:

1. Die Firma unterhält ein zentrales Feldbuslabor zur Unterstützung der Entwickler an den unterschiedlichen Firmenstandorten in Deutschland und in

der Schweiz. Die Entwickler benutzen das Prüfsystem MoTest vor Ort für die Durchführung von First-Party-Tests. Treten Probleme auf, wird das zentrale Labor hinzugezogen, meist reist dann ein Labormitarbeiter an. Zur Vorbereitung des Zertifizierungstests führt das Labor zudem Second-Party-Tests durch. Auch hier reisen zumeist Mitarbeiter an, oder das Gerät muss zwischen den Standorten verschickt werden.

2. Während des Zertifizierungstests stehen bei Problemen die Entwickler bzw. Labormitarbeiter in Kontakt mit der Zertifizierungsstelle für Profibus-Geräte an der Technischen Universität in München. Auch hier ist es gängige Praxis, dass Mitarbeiter nach München reisen um Problemfälle zu klären bzw. das Gerät zurückgeschickt werden muss.

In beiden Szenarien möchte das Feldbuslabor der Firma das verteilte Prüfsystem einsetzen um Reisen und Geräteversand zu reduzieren, die Firma Endress & Hauser stellt also einen idealen Partner für die Evaluierung dar. Um die Herausforderung der verteilten Kollaboration über Firmengrenzen hinweg zu bestehen wurde folgendes Testszenario für das verteilte Prüfsystem durchgeführt:

Am Schweizer Firmenstandort Standort Reinach wurde ein Rechner des Feldbuslabors mit dem neuen Prüfsystem und dem Remote-Framework ausgestattet. An diesem Rechner erfolgte die Anschaltung eines Durchflussmessers für die Prozessindustrie. Die Zertifizierungsstelle in München wurde ebenfalls mit dem verteilten Prüfsystem ausgestattet. Ziel war die Durchführung eines vollständigen Gerätetests durch die Zertifizierungsstelle.

Für die Durchführung des Tests wurde das in dieser Arbeit spezifizierte Klassifizierungsschema auf die Funktionen des Prüfsystems MoTest angewendet und dabei die Spezifika des zu testenden Gerätes berücksichtigt. Es handelt sich um das Durchflussmessgerät PROMASS 83 DP mit integrierter Profibus-Anschaltung. Da es sich bei diesem Gerät um einen Sensor handelt, von dem keinerlei Gefahr für Mensch und Umwelt ausgeht, ergab sich zwar keine restriktive Zugriffsbeschränkung, jedoch konnte aussagekräftig überprüft werden, inwiefern sich die Funktionalität des Prüfsystems sowie unterschiedlichste Testfälle in das Schema einordnen lassen. Das Prüfsystem MoTest stellt dabei eine besondere Herausforderung dar. Zwar sind die für die Testausführung zur Verfügung stehenden Funktionen klar definiert (vgl. Zustandsdiagramm in Abbildung 6-6), jedoch müssen gleichfalls die als Parameter übergebenen Testfälle zuverlässig von der Klassifizierung erfasst werden (vgl. Abbildung 6-7).

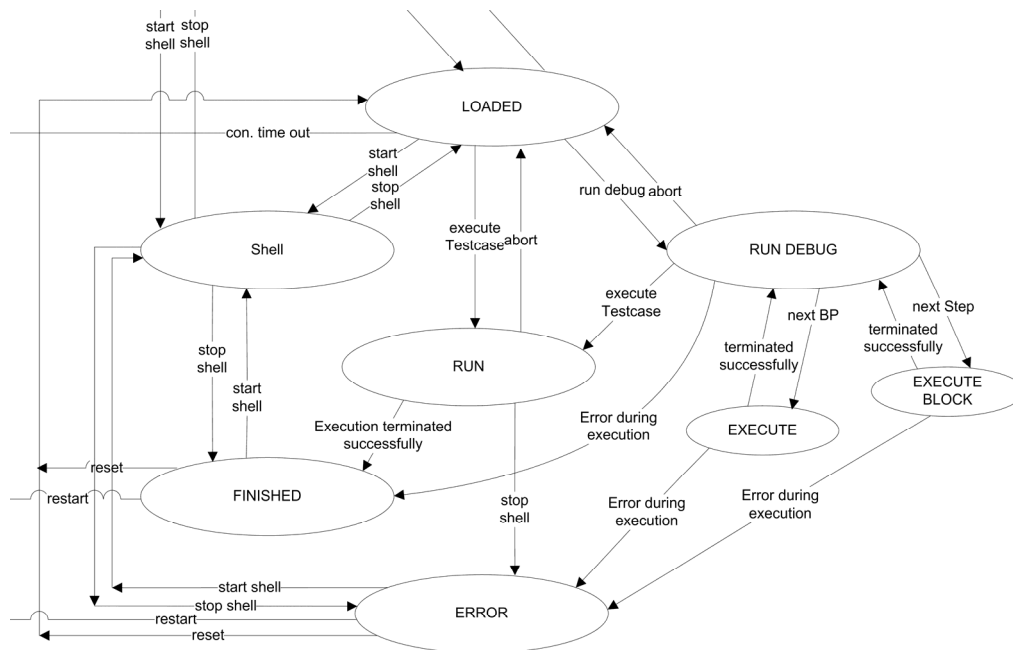


Abbildung 6-6: Ausschnitt aus dem Zustandsdiagramm des Prüfsystems

Anhang C zeigt in Form eines XML-Dokuments die vorgenommene Klassifizierung für das durchgeführte Test-Szenario. Die Klassifizierung der letztendlich am Gerät ausgeführten Funktionen konnte durch die Angabe von Funktionsnamen, Testcase-Namen sowie entsprechender Schlüsselwörter erfolgreich auf feingranularer Ebene durchgeführt werden. Für eine industrielle Umsetzung ist jedoch die Bereitstellung eines grafischen Tools zu empfehlen, welches zusätzlich vorgegebene Grundregeln für bestimmte Gerätetypen zur Verfügung stellt. Insbesondere für den unerfahrenen Anwender würde dies eine wertvolle Hilfestellung bedeuten.

```

tbftest str "\n\n #####\n"
tbftest str " \na: Slave in state WAIT_PRM"
switch $case {
  1 { tbftest str "\ncase 1: Slave supports not SYNC and not FREEZE"
      tbftest str "\n1. Send SLAVE_DIAG.req-PDU:"
      chk -mode wait { h pme dia $Adr } [list 0 [concat 0b0000?010
0x05 0 0xFF $RetIdent]]
      tbftest str "\n2. Send SET_PRM.req-PDU and watchdog control has
been activated:"
      chk -mode wait { h pme prm $Adr 0x88 $WDHigh $WDLow 0x0B
$Ident 0 $UsrPrmData} { 0 }
  }
}
  
```

Abbildung 6-7: Ausschnitt aus einem Testcase für Profibus DP - Geräte

6.3.2 IT-Infrastruktur

Bezüglich der Anpassungsfähigkeit des Remote-Frameworks an unterschiedliche IT-Infrastrukturen offenbarte der Test gewisse Grenzen, die ab einer gewissen Sicherheitsstufe des Firewall-Systems hervorgerufen werden. Bei der

von Endress & Hauser eingesetzten Firewall handelt es sich um einen Filter auf Applikationsebene, dies bedeutet im konkreten Fall, dass http-Transfers mit XML-Inhalten vom System abgeblockt werden. In diesem Fall ist zwar mit Hilfe der zur Verfügung stehenden Modi des Kommunikationsmoduls ein Verbindungsaufbau möglich, SOAP-Nachrichten können jedoch nicht übermittelt werden. Um ein solche System zu überwinden müsste der Schritt von der Verwendung standardisierter Applikationsprotokolle hin zu proprietären Protokollen erfolgen, die auf das Sicherheitssystem maßgeschneidert sind (in diesem Fall beispielsweise die Übermittlung von Funktionsaufrufen in HTML-Seiten). Dies ist jedoch nicht nur unpraktikabel für eine Vielzahl unterschiedlicher Systeme, sondern auch unzuverlässig, da aktuelle intelligente Sicherheitssysteme in der Lage sind, sich wiederholende Informationsmuster zu erkennen und anschließend zu blockieren.

Im konkreten Anwendungsfall wurde ein vom Netzwerk getrennter Laborrechner gemäß Szenario 2 in Kapitel 5.4.3 über ein analoges Modem (56 kbit/s) mit dem Internet verbunden und mit einer Firewall auf Paketebene abgesichert. Durch die Freigabe des http-Ports konnte die Kollaboration mit der Zertifizierungsstelle in München sowohl im aktiven als auch im passiven http-Modus durchgeführt werden. Auf beiden Rechnern wurden während der Versuchsdurchführung Logfiles mit genauen Zeitstempeln für den Aufruf und den Abschluss einer Benutzerinteraktion, bzw. einer Testfallausführung geschrieben.

6.3.3 Analyse der Reaktionszeiten

Die Analyse der Reaktionszeiten des Prüfsystems bei dem durchgeführten verteilten kollaborativen Test eines Gerätes, wurde durch das Mitschreiben von Protokolldateien auf beiden Rechnern mit den Werkzeuginstanzen vorgenommen. Von entscheidender Aussagekraft ist dabei das vom Remote-Anwender aufgezeichnete Protokoll, das lokale Protokoll dient der Absicherung der Messungen sowie zur Ermittlung weiterer Parameter. Bei der Messung sind bezüglich der Messgenauigkeit gewisse Einschränkungen in Kauf zu nehmen:

- Rechner unter dem Betriebssystem Windows sind zuverlässig nur bis zu einer Genauigkeit von einer Sekunde synchronisierbar [LEUNIG 04]. Laufzeitmessungen sind deshalb nur für einen vollständigen http-Request (inklusive Response) durchführbar.
- Die Auflösung der internen Uhr des Windows-Betriebssystem beträgt in der Praxis nicht mehr als 10 ms [MICROSOFT 05].

Diese Einschränkungen sind jedoch für die Beurteilung des Reaktionsverhaltens aus folgenden Gründen hinnehmbar:

- Die Übermittlungszeit der kurzen synchronen http-Response fällt bei der Übermittlung von Testcase- bzw. Logfiles im Kilobyte-Bereich nur wenig ins Gewicht.

- Gemäß den ergonomischen Anforderungen bekommt der Anwender vom Prüfsystem immer eine umgehende Reaktion auf eine Interaktion. Parallel dazu erfolgt die Abarbeitung der Interaktion in asynchroner Form. Die Ausführung von Testfällen nimmt ohnehin eine gewisse Zeit, mitunter im zweistelligen Sekundenbereich ein. Selbst bei einer Abweichung von maximal einer Sekunde entsteht deshalb ein aussagekräftiges Ergebnis. Ohnehin wurde der Vergleich zweier Zeitstempel von verschiedenen Rechnern nur zu Validierungszwecken der Messungen vorgenommen.

In den beidseitig geschriebenen Protokolldateien stehen folgende Zeitstempel zur Verfügung: Abschicken eines beliebigen Befehls (inklusive Testcase) per http-Requests, Empfang der synchronen http-Response, Empfang der asynchronen Fertigmeldung über die Ausführung des Befehls, falls vorhanden: Empfang des vollständigen Logfiles der Ausführung. Hieraus wurden die folgenden relevanten Parameter sowohl für die Ausführung einfacher Befehle als auch für die Ausführung von Testcases bestimmt: Zeitdauer für die Befehlsübermittlung, Ausführungsdauer des Befehls, Zeitdauer für die Übermittlung des Ergebnisses der Ausführung, Gesamtzeitdauer.

Die Ermittlung der Parameter wurde sowohl im aktiven als auch im passiven http-Modus durchgeführt. Eine Zusammenstellung zeigt Tabelle 6-4:

	Umfang [kByte]	Übermitt- lungszeit Funktion [s]	Bearbei- tungszeit [s]	Übermitt- lungszeit Ergebnis [s]	Gesamt- zeit [s]
aktiver Modus					
TXE - Kommandos					
Mittel	0,212	0,696	0,443	0,708	1,847
Std Abw.	0,25	0,216	0,105	0,032	0,178
Testcases					
Mittel	7,967	1,329	0,539	1,071	2,399
Std Abw.	10,694	0,612	0,194	0,143	0,703
passiver Modus (500ms poll-time)					
TXE - Kommandos					
Mittel	0,212	1,224	0,597	0,794	2,615
Std Abw.	0,25	0,09	0,401	0,075	0,09
Testcases					
Mittel	7,967	1,836	2,558	1,877	6,299
Std Abw.	10,694	0,22	2,078	1,334	0,22

Tabelle 6-4: Gemessene Parameter des Remote-Tests

Die Messungen basieren auf einer Internetverbindung über ein analoges Modem mit 56 kBit/s Datendurchsatz auf Seiten von Endress & Hauser sowie 100Mbit/s Internetanbindung auf Seiten der Zertifizierungsstelle. Die Testcases besitzen eine Dateigröße von minimal 512 Byte und maximal 44 kByte. Im Vergleich der beiden Modi ist zunächst klar der Effekt des Pollings zu erkennen. Im Schnitt liegen die Mittelwerte der Übertragungszeiten der Funktionsaufrufe um ca. 500 ms höher als im aktiven Modus. Die entschieden höhere Gesamtzeit liegt dabei in der unterschiedlichen Bearbeitungszeit begründet. Im Protokoll ließ sich bei einem der Testcases ein Ausreißer identifizieren, der sich stark auf den Mittelwert niederschlägt, was auch die hohe Standardabweichung belegt. Zusätzlich liegt aber auch die Übermittlungszeit für die Ergebnisprotokolle im Modus 3 deutlich höher als in Modus 1. Die vergleichsweise große Standardabweichung lässt auch hier auf einzelne Ausreißer schließen was durch die schwankende effektive verfügbare Bandbreite erklärt werden kann. Wird diese Einfluss sowie die Bearbeitungszeit herausgerechnet, so ergeben sich Verzögerungen im Bereich von 300 ms bis 800 ms im passiven Modus, verglichen mit dem aktiven Modus.

Die Zeiten bestätigen den subjektiven Eindruck der ausführenden Prüfexperten, wonach das Prüfsystem zu jeder Zeit ergonomisch bedienbar blieb. Die durch die Remote-Verbindung entstehenden Verzögerungen stellten im Vergleich zu den vorhandenen Bearbeitungszeiten keinen entscheidenden Nachteil dar. Im aktiven Modus war, gemäß den Aussagen der Beteiligten, flüssiges Arbeiten möglich. Lediglich im passiven Modus wurden leichte Verzögerungen wahrgenommen. Eine Verringerung der Abfrageintervalle in Kombination mit einer schnelleren Netzwerkanbindung auf Firmenseite würde hier jedoch für Verbesserungen sorgen. Als Fazit ist weiterhin zu bemerken, dass in Versuchen mit Application-Sharing-Tools selbst über ADSL-Verbindungen keine annähernd kurzen Reaktionszeiten erreicht wurden: die Übermittlung der grafischen Informationen nahm erheblich mehr Zeit in Anspruch und wäre über eine Verbindung mit 56 kBit/s gänzlich inakzeptabel. Als weiterer Vorteil erwies sich die Möglichkeit der wahlweisen Datenhaltung auf der entfernten Seite: einmal übermittelte Logfiles von Testcases konnten ohne weitere Transaktionen mit dem lokalen System verzögerungsfrei analysiert werden.

6.3.4 Qualitätsbetrachtungen des Multimediakanals

Von besonderem Interesse ist, neben den Reaktionszeiten des Datenkanals, auch das Gesamtverhalten des Systems bei gleichzeitiger Nutzung der Inter-Tool- und der Multimediakommunikation. Angesichts der vergleichsweise geringen Datenmengen (vgl. Tabelle 5-3, Tabelle 5-4 und Tabelle 6-4) ist bei breitbandigen Übertragungsmedien wie beispielsweise DSL-Leitungen keine Beeinträchtigung zu erwarten. Da aber auch von Kommunikationsszenarien über gebündelte bzw. einfache ISDN-Kanäle (64 kBit/s pro Kanal) auszugehen ist, erfolgt eine Untersuchung des Gesamtverhaltens in diesem Bereich. Hierzu werden in

unterschiedlichen Bandbreitenbereichen beide Kanäle des verteilten Testsystems genutzt und die prozentuale Auslastung der Verbindung mit einem Netzwerkmonitor aufgezeichnet. In Verbindung hierzu wird eine subjektive Beurteilung des Reaktionsverhaltens des Systems sowie der Multimediaqualität vorgenommen. Die Ergebnisse fasst Tabelle 6-5 zusammen:

Bandbreite	Auslastung [%] (unidirektional)	Eingestellte Qualität	Audioqualität (subjektiv)	Videoqualität (subjektiv)
1,544 MBit/s (DSL)	26	LAN	Telefonqualität	Gute Auflösung bei Standbild und Bewegungen, hohe Frame-Rate
128 kBit/s 2-Kanal ISDN	42	Modem/ISDN	Gute Qualität, leichtes Rauschen	Gutes Standbild, grobe Auflösung bei Bewegungen
64 kBit/s 1-Kanal ISDN	76	Modem/ISDN	Zufriedenstellende Qualität, Aussetzer beim Versenden von Testcases	Gutes Standbild, niedrige Frame-Rate, grobe Auflösung bei Bewegungen

Tabelle 6-5: Ergebnisse der Analyse des Bandbreitenbedarfs multimedialer Datenströme

Die Messungen entstanden während intensiver Sprachkommunikation und während der Aufnahme stark bewegter Bilder, was durch intensives Kameraschwenken simuliert wurde. Diese Art von Video stellt sicherlich ein Extrem dar, da sich üblicherweise nicht der vollständige Bildausschnitt verändert. Während der Multimediaübertragung wurden, von der Remote-Instanz des Testsystems, Testcases mit 44 kByte Dateigröße an die lokale Instanz geschickt. Die Tabelle offenbart zunächst, dass selbst bei niedrigen Bandbreiten wie 1-Kanal ISDN mit den heutigen modernen Kodierungsverfahren zumindest eine zufriedenstellende Qualität der Multimediaübertragung erzielbar ist. Durch die Anpassung der Video-Frame-Rate durch Netmeeting bleiben stets Bandbreitenreserven erhalten, die dem Datenkanal des Remote-Frameworks zur Verfügung stehen. Es zeigte sich jedoch in der Einstellung 64 kBit/s, dass das Versenden eines Testcases zu einer Unterbrechung der Sprachübertragung führt. Im Messprotokoll ist dies durch den schlagartigen Anstieg des Bandbreitenbedarfs auf 100 % erkennbar (vgl. Abbildung 6-8).

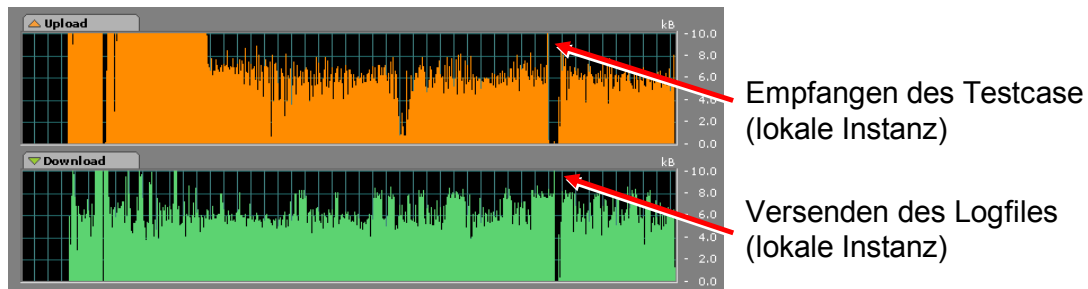


Abbildung 6-8: Messkurve der Auslastung bei 64 kBit/s

Im Videobereich konnte auch bei kleineren Bandbreiten eine gute Bildqualität erreicht werden (vgl. Abbildung 6-9). Hier sind schlimmstenfalls eine niedrige Frame-Rate, bzw. im Extremfall wiederholte Standbilder in Kauf zu nehmen. Treten keine schnellen Kameraschwenks auf, wirkt sich diese Tatsache nicht störend aus. Auch wenn in professionellen Videokonferenzen sicherlich höhere Ansprüche gestellt werden, ist die Qualität einer schmalbandigen Videoübertragung für den vorliegenden Anwendungsfall ausreichend. Die Verwendung hoher Bandbreiten ermöglicht hingegen ein hochqualitatives Video-Streaming.



Abbildung 6-9: Videobild eines Testgerätes im CIF-Format, bei ca. 6 kByte/s Datenstrom

6.4 Zusammenfassung

Im Rahmen der Evaluierung der erarbeiteten Lösung wurde das Remote-Framework prototypisch implementiert und im Rahmen einer industriellen Anwendung getestet. Es handelt sich dabei um den Test der Feldbusschnittstelle mechatronischer Geräte der Automatisierungstechnik unter Anwendung eines speziellen Prüfsystems. Das Prüfsystem wurde, gemäß der erarbeiteten Spezifikation, in das Remote-Framework integriert. Der Aufwand für diese Integrationsarbeiten blieb dabei im vorgesehenen Rahmen (vgl. Kapitel 5.4.4).

Die Evaluierung erfolgte zusammen mit dem Gerätehersteller Endress & Hauser in Reinach/Schweiz. Hierfür wurde im Labor der Firma ein Gerätetest durch die Zertifizierungsstelle für Profibus-Geräte in München mit Hilfe des verteilten Testsystems durchgeführt. Als schwierig erwies sich die Überwindung der am Firmenstandort vorhandenen äußerst restriktiven Sicherheitsmechanismen. Die dabei gemachten Erfahrungen lassen insbesondere für zukünftige Anbieter von kommerziellen Remote-Diensten die Notwendigkeit erkennen, verschiedenste Zugangspunkte für den Aufbau einer Kommunikationsverbindung von einem abgesicherten Standort aus anzubieten. Entsprechende Szenarien hierzu wurden in Kapitel 5 erarbeitet. Neben offenen, internetbasierten Zugangspunkten bedeutet dies vor allem die Bereitstellung von VPN-Verbindungen oder auch direkten Einwahlverbindungen auf Rechner, die von der Gegenstelle als vertrauenswürdig eingestuft werden.

Die über eine nur geringe Bandbreite von 56 kBit/s durchgeführten Messungen sowie die subjektive Beurteilung der testdurchführenden Personen belegen die Funktionsfähigkeit des entwickelten Konzepts für die verteilte Kollaboration. Die Firma Endress & Hauser plant deshalb, das Testsystem zusammen mit dem Remote-Framework sowohl für interne Second-Party-Tests, als auch für externe Third-Party-Tests zusammen mit der Zertifizierungsstelle einzusetzen.

Eine Qualitätsbeurteilung der Multimediaübertragung bei verschiedenen Bandbreiten führte zu dem Ergebnis, dass mit den heute verfügbaren modernen Kodierungsverfahren auch über schmalbandige Medien eine ausreichende Übertragungsqualität gewährleistet ist. Im Laborversuch traten bei gleichzeitig aktivem Daten- und Multimediakanal lediglich bei einer einkanaligen ISDN-Leitung Beeinträchtigungen der Sprachqualität auf.

7 Abschließende Betrachtung

Die resümierende Betrachtung der vorliegenden Arbeit und der darin erzielten Ergebnisse ist das Ziel dieses abschließenden Kapitels. Problemstellung und Lösung werden hierfür nochmals zusammenfassend dargestellt. Im Weiteren erfolgt eine Gesamtbewertung der erarbeiteten Ergebnisse sowie ein Ausblick auf weiterführende wissenschaftliche Arbeiten und Forschungstätigkeiten.

7.1 Zusammenfassung

Moderne mechatronischen Produkte integrieren Komponenten der drei klassischen Ingenieursdisziplinen Mechanik, Elektronik und Softwaretechnik/Regelungstechnik in abgeschlossenen Einheiten. Die enge Integration und vor allem das enge funktionale Zusammenspiel der Einzelkomponenten führt zu Produkten, mit erheblich gesteigertem Leistungsvermögen und Funktionsumfang: Auf engstem Raum lassen sich hochfunktionelle Einheiten realisieren. In Folge dieser Vorteile werden mechatronische Produkte heutzutage in vielen Bereichen eingesetzt und besitzen eine signifikante Marktrelevanz.

Eine besondere Herausforderung stellt jedoch die erfolgreiche Entwicklung mechatronischer Produkte, nicht zuletzt unter ökonomischen Gesichtspunkten dar. Verschiedenste Spezialisten aus den drei Disziplinen, oft an verteilten Standorten lokalisiert, müssen zu einem Entwicklungsverbund integriert werden. Dies verursacht insbesondere dann Schwierigkeiten, wenn durch Entwicklungstätigkeiten am realen Produkt, Prototypen oder Teilkomponenten eine Ortsbindung entsteht (Integrative Phasen des Entwicklungsprozesses). Beteiligte müssen dann Reisen unternehmen, bzw. ist der Versand von Geräten und Teilkomponenten notwendig. Gerade vor dem Hintergrund fortschreitender Globalisierung verursacht diese Problemstellung hohe Kosten sowie Verzögerungen im Entwicklungsprozess.

Verteilte Kooperation im Entwicklungsprozess mechatronischer Geräte stellt eine Lösung für die vorhandene Problemstellung dar. In den integrativen Phasen herrscht jedoch eine Reihe von Randbedingungen vor, die den Einsatz existierender Remote- und Kooperationssysteme verhindert. Es fehlen Lösungen, die den individuellen Anforderungen des Entwicklungsprozesses in den integrativen Phasen genüge leisten. Verteilte Kooperation in diesen Phasen muss den Einsatz hochkomplexer Rechnerwerkzeuge unterstützen, wobei stets von Interaktionen entfernter Beteiligter mit realen Geräten auszugehen ist, die eine Gefahr für Mensch und Umwelt darstellen können. Parallel dazu muss die Möglichkeit

einer integrierten interpersonellen Kommunikation zwischen den beteiligten Spezialisten zur Verfügung stehen. Um die Effizienz der Zusammenarbeit zu sichern, werden anwendungsfallorientierte Kooperationslösungen benötigt, die vom Anwender ohne viel Aufwand für Administration und Bedienung nutzbar sind. Nicht zuletzt muss die Kooperationslösung den Anforderungen an IT-Sicherheit genügen und an bestehende IT- und Sicherheitsinfrastrukturen von Unternehmen anpassbar sein.

Als Ausgangspunkt für die Entwicklung einer Kooperationslösung, die den aufgezählten Anforderungen entspricht, erfolgte Eingangs eine Analyse existierender Konzepte und Technologien für die verteilte Kooperation. Es stellte sich heraus, dass bisher keine Lösungen existieren, die der Problemstellung in vollem Umfang gerecht werden. Auch wenn für Teilbereiche Grundagententechnologien existieren, die jedoch allenfalls als Bausteine einer Gesamtlösung einsetzbar sind. Zu diesen zählen einzelne Technologien zur Übertragung multimedialer Daten.

Die Basis der in dieser Arbeit entwickelten Lösung ist ein umfassendes Rahmenkonzept für die Realisierung von verteilter Kooperation in den integrativen Phasen der mechatronischen Produktentwicklung. Es beinhaltet als Kern die Spezifikation eines Frameworks für die Koppelung rechnerbasierter Entwicklungswerkzeuge unter Einbeziehung realer Produktkomponenten. Dieses Framework trägt den besonderen Anforderungen der integrativen Phasen Rechnung. So beinhaltet es ein Modell für die feingranulare Kontrolle entfernter Interaktionen, zur Vermeidung der dadurch entstehenden Gefahren. Die Anpassbarkeit der Lösung im Sinne der Anwender wird durch ein Rollenmodell sichergestellt. Um die Kooperationslösung universell an verschiedenste Anwendungsfälle anzupassen beinhaltet das Framework die Möglichkeit der Integration unterschiedlichster Werkzeuge. Da die Nutzung von Remote-Technologien in vielen Firmen aufgrund der Sicherheitsinfrastruktur häufig mit Problemen verbunden ist, wurden Lösungen für den Einsatz des Frameworks in verschiedensten Sicherheitsarchitekturen entwickelt. Für die interpersonelle Kommunikation zwischen Spezialisten an unterschiedlichen Standorten wurde im Framework ebenfalls ein Integrationspunkt geschaffen. Mittels diesem können verschiedenste Multimedia-Applikationen eingebunden und auf einheitliche Art und Weise aus dem Framework heraus genutzt werden.

Die erarbeitete Lösung wurde prototypisch umgesetzt und zusammen mit einem Prüfwerkzeug für mechatronische Geräte der Automatisierungstechnik in einem industriellen Anwendungsszenario getestet. Innerhalb des Tests wurde die verteilte Kooperation zwischen dem Schweizer Standort eines namhaften Geräteherstellers und einer unabhängigen Zertifizierungsstelle für Feldbuskommunikation in München realisiert. Daneben wurden diverse Versuche unter Laborbedingungen mit unterschiedlichen Netzwerkbandbreiten und Sicherheitskonfigurationen unternommen.

7.2 Bewertung

Die prototypische Implementierung des Frameworks sowie die durchgeführte Evaluierung in Verbindung mit einem industriell eingesetzten Prüfwerkzeug belegen die umfassende Unterstützung der entwickelten Lösung für die verteilte Kooperation in den integrativen Phasen. Der Aufwand für die Einrichtung und Nutzung der erweiterten Kooperationsfunktionalität bleibt für den Anwender aufgrund der ganzheitlichen Integration sämtlicher benötigter Remote-Komponenten gering. Eine unkomplizierte Nutzung am eigenen Arbeitsplatz ist hierdurch möglich, zusätzlicher Installations- und Einrichtungsaufwand für Remote-Applikationen entfällt. Durch die Flexibilität des integrierten Kommunikationskanals wird eine umfangreiche Kompatibilität des Frameworks zu verschiedensten Sicherheitskonfigurationen und IT-Infrastrukturen erreicht. Die Konzeption der Inter-Tool-Schnittstelle auf Basis von XML-Webservices ermöglicht kurze Reaktionszeiten der verteilten Anwendung auch bei geringen Bandbreiten, wie etwa beim Einsatz analoger Modems. Sowohl die gemessenen als auch die subjektiv wahrgenommenen Reaktionszeiten liegen dabei um ein Vielfaches unter den Werten herkömmlicher Application-Sharing-Anwendungen, die auf einer Übertragung der grafischen Benutzeroberfläche basieren. Die verteilte Kooperation wird zudem durch die wahlweise Verwendung eines synchronisierten und eines nicht synchronisierten Modus unterstützt. Der nicht synchronisierte Modus ermöglicht es, zeitaufwändiges Bearbeiten umfangreicher Daten offline auszuführen.

Das entwickelte Modell zur feingranularen Abstufung der Zugriffsrechte für entfernte Interaktionen begegnet wirkungsvoll den Gefahren, die mit dem entfernten Zugriff auf reale mechatronische Komponenten verbunden sind. Anhand der Systematik des Modells ist ein Anwender in der Lage, die Gefahren abzuschätzen und die Funktionen der Entwicklungswerkzeuge entsprechend zu klassifizieren. Notwendige Beschränkungsmaßnahmen werden dann automatisch durch das Remote-Framework ausgeführt.

Der Implementierungsaufwand für die Erweiterung eines bestehenden Entwicklungswerkzeuges mit Remote-Kooperationsfunktionalität bleibt aufgrund des modularen Konzepts des Frameworks mit klar spezifizierten Schnittstellen in einem vertretbaren Rahmen. Die in der Lösung erarbeiteten Integrationsmuster geben dabei eine wertvolle Hilfestellung.

Im Rahmen eines industriellen Anwendungsszenarios erfolgte die Evaluierung der entwickelten Lösung. Dabei konnte die vollständige Erfüllung der eingangs definierten Anforderungen an die verteilte Kooperation in den integrativen Phasen des mechatronischen Entwicklungsprozesses bestätigt werden.

7.3 Ausblick

Gemäß dem in dieser Arbeit fokussierten Anwendungsbereich wurde für die entwickelte Lösung die Erfüllung der dort vorherrschenden Anforderungen in den Vordergrund gestellt. Durch das Hinzufügen weiterer Ausprägungen ist eine Nutzung in bisher nicht betrachteten Branchen und Anwendungsgebieten denkbar. Ein Ansatz wäre die Ausweitung des Kooperations szenarios auf mehrere Beteiligte in einer m:n-Konstellation. In dieser hätten nicht mehr alle Kooperationspartner grundsätzlich Interaktionsrechte, sondern es könnte beispielsweise zwischen aktiven Rollen und Beobachterrollen unterschieden werden. Beobachter hätten allenfalls die Möglichkeit im Rahmen der interpersonellen Kopplung Ihr Wissen einzubringen.

Eine weitere Möglichkeit wäre es, mehrere verteilte Werkzeuginstanzen zu vernetzen und hierdurch die verteilten Einzelkomponenten als ein Gesamtsystem zu betrachten. Dieser Gedanke führt zu Szenarien wie verteilte Inbetriebnahme oder verteilte Systemtests. Aufgaben aus der Ebene der Systemintegration könnten so bereits in die Ebene der Komponentenerstellung und des Komponententests vor verlagert werden. Eine wesentliche Herausforderung dieser Thematik ist die Realisierung von Echtzeitkommunikation und Zeitsynchronisation über Weitverkehrsnetze.

Im Hinblick auf die kommerzielle Nutzung des Remote-Frameworks wäre die Entwicklung eines Tools denkbar, welches den Anwender grafisch bei der Erstellung von Funktionsklassifizierungen für die sichere Remote-Kollaboration unterstützt. In diesem Zusammenhang bietet sich auch eine Untersuchung an, inwiefern sich gleichartige Werkzeuge und Anwendungsfälle in einer Art Szenarienbibliothek zusammenfassen lassen, die dem Anwender eine geeignete Ausgangsbasis zur Lösung seiner spezifischen Problemstellung verschafft. Als Ergebnis könnte eine Art Expertensystem stehen.

Als wichtiger Schritt auf dem Weg zu einer kommerziellen Nutzung der Lösung steht die Integration weiterer rechnerbasierter Entwicklungswerkzeuge der integrativen Phasen in das Remote-Framework. Beispiele hierfür sind: Feldbusmonitor, Diagnosegeräte der Automobilindustrie, HIL-Prüfstände u.v.m. Parallel zu diesen Aktivitäten scheint auch die Schaffung eines Verzeichnisdienstes zur Lokalisierung verschiedenster Dienstleister von Remote-Expertise auf Basis einer zentralen Datenbank ein sinnvolles Zukunftsziel.

Literaturverzeichnis

- [AHREND+ 04A] Ahrends, S.; Zäpf, J.: *Simulationsgestützte Testautomatisierung mit Hilfe einer HIL-Entwicklungsumgebung für den Einsatz in der Automobilindustrie*; VIP-2004 Workshop; Fürstfeldbruck, 2004.
- [AHREND+ 04B] Ahrends, S.; Zäpf, J.: *Hard- und Softwareplattform für Prüfsysteme in der Automobilindustrie mit Ergänzungen im Bereich Simulation und HIL*; VIP-2004 Workshop; Fürstfeldbruck, 2004.
- [ALLES+ 99] Allespach, M.: *Kooperatives Lernen im Betrieb. Zugänge aus Sicht der pädagogisch-psychologischen Kooperationsforschung*; Agenda-Verlag, Münster, 1999.
- [ANDERL 02] Anderl, R.: *Produktdatentechnologie I*; Skriptum zur Vorlesung; <http://www.iim.maschinenbau.tu-darmstadt.de/pdt1/frames/PDT1.html>; 2002.
- [ARMITAGE 00] Armitage, R.: *Quality of Service in IP-Networks*; Macmillan Technical Publishing, 2000.
- [BADACH 04] Badach, A.: *Voice over IP - Die Technik*; ISBN 3-446-22697-4; Carl Hanser-Verlag, München Wien, 2004.
- [BALZERT 00] Balzert, H.: *Lehrbuch der Software-Technik*; ISBN 3-8274-0480-0; Spektrum Akademischer Verlag GmbH, Heidelberg - Berlin, 2000.
- [BANGEMANN+ 03] Bangemann, T. et al.: *PROTEUS - a European Initiative for E-Maintenance Platform Development*; 9th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA2003; 2003.
- [BAUER 98] Bauer, L.: *Methoden und Werkzeuge für innovative Entwicklungsprozesse*; iwv Seminarberichte 41; ISBN 3-89675-041-0; Herbert Utz Verlag GmbH, München, 1998.

- [BENDER 01] Bender, K.; Bregulla, M.; Kuttig, F.: *Einheitliche Bedienung von Feldgeräten mittels Interet-Technologien*; VDI-Berichte 1608; VDI-Verlag GmbH, Düsseldorf, 2001.
- [BENDER 03] Bender, K.: *Skriptum zur Vorlesung Automatisierungstechnik*; Lehrstuhl für Informationstechnik im Maschinenwesen, TU München; München, 2003.
- [BENDER 04A] Bender, K.: *Skriptum zur Vorlesung Informationstechnik*; Lehrstuhl für Informationstechnik im Maschinenwesen, TU München; München, 2004.
- [BENDER 04B] Bender, K. (Hrsg.): *Embedded Systems - qualitätsorientierte Entwicklung*; ISBN 3-540-22995-7; Springer Verlag, Berlin, 2005.
- [BENDER+ 01] Bender, K. et al.: *Qualitätssicherung eingebetteter Software: Methoden und Best-Practices*; ISBN 3-8316-0024-4; Herbert Utz Verlag - Wissenschaft, München, 2001.
- [BENDER+ 02A] Bender, K. et al.: *DeKOS-Engineering-Modell für intelligente Automatisierungskomponenten*; ISBN 3-8316-0169-0; Herbert Utz Verlag - Wissenschaft, München, 2002.
- [BENDER+ 02B] Bender, K.; Kuttig, F.; Meyer, H.: *Intelligentes Engineering mittels einheitlicher Technologien*; atp - Automatisierungstechnische Praxis Nr. 2/2002; S. 32 - 38; Oldenbourg Industrieverlag GmbH, München, 2002.
- [BERGNER 99] Bergner, K. et al.: *Componentware - Methodology and Process*; 1999 International Workshop on Component-Based Software Engineering; <http://www.sei.cmu.edu/cbs/icse99/papers/41/position.html>; Carnegie Mellon University - Software Engineering Institute; 1999.
- [BETTENHAUSEN+ 02] Bettenhausen, K.-D.; Braune, A.; Rieger, B.: *Anforderungen an die Nutzung von Internettechnologien in der Automatisierung*; atp - Automatisierungstechnische Praxis Nr. 06/2002; Oldenbourg Industrieverlag GmbH, München, 2002.

- [BLACK 00] Black, U.: *QoS in Wide Area Networks*; ISBN 0130264970; Prentice Hall, New Jersey, 2000.
- [BMBF 00] N.N.: *Analyse und Evaluation der Softwareentwicklung in Deutschland*; Studie für das Bundesministerium für Bildung und Forschung; 2000.
- [BMI 05] N.N.: *V-Modell XT Release 1.01 Dokumentation*; http://www.kbst.bund.de/V-Modell/-_293/V-Modell-XT.htm; Bundesministerium des Inneren; 2005.
- [BÖHMER 02] Böhmer, W.: *VPN Virtual Private Networks*; ISBN 446-21532-8; Carl Hanser Verlag, München - Wien, 2002.
- [BOTELLÉ+ 04] Botellé, E., Lebrun, M., Zimmer, B.: *Simulationsumgebung für konzeptionelle Auslegung von multidisziplinären Systemen*; VDI-Berichte 1842; S. 61 - 70; ISBN 3-18-091842-X; VDI Verlag GmbH, Düsseldorf, 2004.
- [BRAUNE+ 99] Braune, A.; Hiller, O.; Janschek, K.: *Anwendungserfahrung bei der Nutzung Web-basierter Fernbeobachtung und Fernbedienug*; VDI-Berichte 1515; S. 127 - 136; ISBN 3-18-091515-3; VDI-Verlag GmbH, Düsseldorf, 1999.
- [BSI 03] N.N.: *Grundschutzhandbuch*; Bundesamt für Sicherheit in der Informationstechnik; 2003.
- [BULLINGER+ 96] Bullinger, H.-J.; Warschat, J.; Wörner, K.: *Future Potential of Rapid Product Development - A Concept Integration Organizational Aspects*; International Conference on RPD - Rapid Product Development; S. 13 - 24; Stuttgart, 1996.
- [BUSCH 02] Busch, C.; Wolthusen, S., D.: *Netzwerksicherheit*; ISBN 3-8274-1373-7; Spektrum Akademischer Verlag GmbH, Heidelberg – Berlin, 2002.
- [COLLABORATION 04] <http://www.collaboration-tools.com/>; 2004.

- [DFAM 04] Kuttig, F.: *IT-Sicherheit bei Einsatz von Ethernet in der Automatisierungstechnik*; Deutsche Gesellschaft für die Anwendung der Mikroelektronik e.V. (DFAM); ISBN 3-8163-0488-5; Frankfurt, 2004.
- [DOHMEN 02] Dohmen, W.: *Interdisziplinäre Methoden für die integrierte Entwicklung komplexer mechatronischer Systeme*; Dissertation; Technische Universität München; München, 2002.
- [DROSDOWSKI+ 63] Drosdowski, G.; Grebe, P.: *Duden Etymologie. Herkunftswörterbuch der deutschen Sprache*; Bibliographisches Institut; Dudenverlag, Mannheim, 1996.
- [DUDEN 04] Müller, W.: *Duden Bedeutungswörterbuch*; Bibliographisches Institut; ISBN 3-4110-4103-X; Dudenverlag, Mannheim, 2002.
- [EHLERS 00] Ehlers, K.: *Problemfelder der integrierten Elektronik-Entwicklung im Produktentstehungsprozess*; ATZ Automobiltechnische Zeitschrift 102, 10; S. 878 - 887; GWV Fachverlage GmbH, Wiesbaden, 2000.
- [EHRLENSPIEL 95] Ehrlenspiel, K.: *Integrierte Produktentwicklung*; ISBN 3-446-15706-9; Carl Hanser Verlag, München - Wien, 1995.
- [EICHELMANN 95] Eichelmann, B.: *Probleme der Kooperation in Forschungs- und Entwicklungsabteilungen*; iwB Seminarberichte 5; S. 31 - 38; ISBN 3-931327-05-1; Herbert Utz Verlag, München, 1995.
- [ELKO 04] *Elektronik Kompendium*; <http://www.elektronik-kompodium.de>; 2004.
- [ELLERMANN 02] Ellermann, U.: *Firewalls Klassifikation und Bewertung*; <http://www.cert.dfn.de/team/ue/fw/workshop/workshop.html>; DFN-Cert; 2002.
- [EVERSHEIM+ 96] Eversheim, W. et al.: *Verteilte Entwicklung - Erfahrungen mit dem Einsatz von fortschrittlichen Informations- und Kommunikationssystemen in der Automobilentwicklung*; VDI Berichte 1302; S. 175 - 196; ISBN 3-18-091302-9; VDI-Verlag GmbH, Düsseldorf, 1996.

- [FAHNER+ 01] Fahner, H.; Feil, P.; Zseby, T.: *Mbone - Aufbau und Einsatz von IP-Multicast-Netzen*; ISBN 3-9209-9399-3; dpunkt Verlag, Heidelberg, 2001.
- [FALTIN+ 03] Faltin, N. et al.: *Internetgestützte Labore für die Automatisierungstechnik*; VDI-Berichte 1785; S. 87 - 96; ISBN 3-18-091785-7; VDI-Verlag GmbH, Düsseldorf, 2003.
- [FINGER 99] Finger, R.: *Prozesse der Konzeption, Realisierung und Einführung integrierter Telekooperationssysteme*; ISBN 3-89012-651-0; Eul-Verlag, Köln, 1999.
- [FRAUNHOFER 05] N.N.: *Audio und Multimedia - home of mp3*; Fraunhofer Institut Integrierte Schaltungen; <http://www.iis.fraunhofer.de/amm/index.html>; 2005.
- [GALL 03] Gall, H.: *Die IEC 61508, Erfahrungen bei der Prüfung und Zertifizierung sicherheitsrelevanter Komponenten und der Einfluss auf anwendungsabhängige Standards*; Tagungsband SPS/IPC/DRIVES 2003; S. 373 - 378; ISBN 3-8007-2793-5; VDE Verlag GmbH, Berlin - Offenbach, 2003.
- [GAUSEMEIER+ 00] Gausemeier, J.; Lückel, J.: *Entwicklungsumgebungen Mechatronik - Methoden und Werkzeuge zur Entwicklung mechatronischer Systeme*; Heinz Nixdorf Institut, Universität Paderborn; ISBN 3-931466-79-5; Bonifatius GmbH, Paderborn, 2000.
- [GAUSEMEIER+ 04] Gausemeier, J.; Michels, J., S.; Redenius, A.: *Modellierung und Planung von Produktentstehungsprozessen*; 2. Paderborner Workshop: Intelligente mechatronische Systeme; S. 33 - 42; Bonifatius GmbH, Paderborn, 2004.
- [GAUSEMEIER+ 99] Gausemeier, J.; Flath, M.; Riepe, B.: *Strategic Planning and Conceptual Design of Mechatronic Products*; Mechatronics and Advanced Motion Control, 3rd International Heinz Nixdorf Symposium; HNI Verlagsschriftenreihe, Nr. 49, Paderborn, 1999.
- [GÖBEL 04] Göbel, S.: *Ein LABCAR für alle Fälle*; VDI-Berichte 1842; S. 109 - 118; ISBN 3-18-091842-X; VDI Verlag GmbH, Düsseldorf, 2004.

- [GORONCY 04] Goroncy, J.: *Querdenken vermeidet Elektronikfehler*; VDI Nachrichten Nr. 24/2004; 13; VDI; VDI Verlag GmbH, Düsseldorf, 2004.
- [HABERFELLNER+ 02] Haberkellner, R. et al.: *Systems Engineering - Methodik und Praxis*; Verlag Industrielle Organisation, Zürich, 2002.
- [HEIMANN+ 01] Heimann, B.; Gerth, W.; Popp, K.: *Mechatronik Komponenten - Methoden - Beispiele*; ISBN 3-446-21711-8; Fachbuchverlag Leipzig, München, 2001.
- [HERMANNNS 97] Hermanns, O.: *Multicast-Kommunikation in kooperativen Multimedia-Systemen*; ISBN 3-8265-2348-2; Shaker Verlag, Aachen, 1997.
- [HOYER+ 03] Hoyer, H. et al.: *Telematik in der Lehre, eine Brückenkran-Regelung und ein inverses Pendel als Online-Experiment*; VDI-Berichte 1785; S. 79 - 86; ISBN 3-18-091785-7; VDI-Verlag GmbH, Düsseldorf, 2003.
- [IAONA 04] N.N.: *IAONA Handbook For Network Security V1.0*; <http://www.iaona-eu.com>; IAONA; 2004.
- [IEC 05] N. N.: *IEC 61508 part 0: Functional safety*; VDE-Verlag, Berlin - Offenbach, 2005.
- [IETF 05] N.N.: *Session Initiation Protocol (SIP)*; <http://www.ietf.org/html.charters/sip-charter.html>; 2005.
- [INGREY+ 94] Ingrey, A.; Lerévérénd, P.: *Handbuch Safety Integrity Level, IEC 61508/61511*; Pepperl + Fuchs GmbH; Mannheim, 38139.
- [ISERMANN 02] Isermann, R.: *Mechatronik; atp - Automatisierungstechnische Praxis Nr. 2*; S. 1; Oldenbourg Industrieverlag GmbH, München, 2002.
- [ISERMANN 99] Isermann, R.: *Mechatronische Systeme*; ISBN 3-540-64725-2; Springer Verlag, Berlin, 1999.

- [ISO 94] N.N.: *ISO/IEC 7498-1 : Open Systems Interconnection - Basic Reference Model*; <http://www.iso.org>; International Organisation for Standardization; 1994.
- [ITU 04] *International Telecommunication Union (ITU) - official website*; <http://www.itu.int/home/index.html>; 2004.
- [JHA+ 02] Jha, S.; Hassan, M.: *Engineering Internet QoS*; Artech House, 2002.
- [KALLENBACH 01] Kallenbach, E. et al.: *Integration mechatronischer Systeme*; VDI-Berichte 1631; S. 57 - 72; ISBN 3-18-091631-1; VDI-Verlag GmbH, Düsseldorf, 2001.
- [KALLENBACH 94] Kallenbach, E.: *Mechatronik - Übersichtsvortrag*; Mechatronik Workshop, Tagung Stuttgart, 9. und 10. Mai 1994; Düsseldorf, 1994.
- [KALLENBACH+ 97] Kallenbach, E.; Birli, O.; Saffert, E.; Schäffel, Chr.: *Zur gestaltung integrierter mechatronischer Produkte*; VDI-Berichte 1315; S.1 - 14; ISBN 3-18-091315-0; VDI-Verlag GmbH, Düsseldorf, 1997.
- [KALLMEYER 98] Kallmeyer, F.: *Eine Methode zur Modellierung prinzipieller Lösungen mechatronischer Systeme*; HNI - Verlagschriftenreihe, Band 42, Paderborn, 1998.
- [KIESEWETTER 97] Kiesewetter, T.: *Integrativer Produktentwicklungsarbeitsplatz mit Multimedia- und Breitbandkommunikationstechnik*; Fraunhofer IPK Berlin; ISBN 3-8167-4548-2; Berlin, 1997.
- [KLOTZBACH+ 03] Klotzbach, S.; Oedekoven, S.; Grassmann, O.: *Optimierung im mechatronischen Entwicklungsprozess*; VDI-Berichte 1753; S. 189 - 208; ISBN 3-18-091753-9; VDI-Verlag GmbH, Düsseldorf, 2003.
- [KOÇ 03] Koç, A.: *Entscheidungsunterstützung zur Planung der Software-Qualitätssicherung in mechatronischen Produkten*; Dissertation; Bender, K.; ISBN 3-8316-0250-6; Herbert Utz Verlag - Wissenschaft, München, 2003.

- [KOÇ+ 01] Koç, A., Jack, P.: *Qualitätsorientiertes Vorgehensmodell für intelligente Produkte der Automatisierungstechnik*; VDI-Berichte 1608; ISBN 3-18-091608-7; VDI Verlag GmbH, Düsseldorf, 2001.
- [KÖNIG 95] König, M.: *Optimierung der Kooperation in der Produktentwicklung*; iwv Seminarberichte 5; S. 1 - 15; ISBN 3-931327-05-1; Herbert Utz Verlag, München, 1995.
- [KRAUSE+ 00] Krause, F.-L.; Schulz, R.: *Integration praxisorientierter Telekooperationssysteme in verteilten Produktentwicklungsprozessen eines Automobilhersteller- /Zulieferer-Netzwerks*; VDI Berichte 1537; S. 145 - 168; ISBN 3-18-091537-4; VDI-Verlag, Düsseldorf, 2000.
- [KUTTIG+ 03] Kuttig, F.; Bechter, M.: *Verteiltes kooperatives Testen mit Hilfe der Internet-Technologien*; Tagungsband SPS/IPC/DRIVES 2003; S. 379 - 386; ISBN 3-8007-2793-5; VDE Verlag GmbH, Berlin - Offenbach, 2003.
- [KYAS+ 00] Kyas, O.; a Campo, M.: *IT-Crackdown – Sicherheit im Internet*; ISBN 3-8266-4080-2; MITP-Verlag, Bonn, 2000.
- [LANDWEHR 01] Landwehr, M.: *Labor- und Simulationswerkzeuge für die Analyse mechatronischer Systeme in der Feinwerktechnik*; Fortschritt-Berichte VDI Nr. 894; ISBN 3-18-389408-7; VDI Verlag GmbH, Düsseldorf, 2001.
- [LAUBER+ 99] Lauber, R.; Göhner, P.: *Prozessautomatisierung 1*; ISBN 3-540-65318-x; Springer Verlag, Berlin - Heidelberg, 1999.
- [LEBERT 03] Lebert, K.: *Testen der OBD-Funktionen von Steuergeräten mit HIL-Systemen*; VDI-Berichte 1753; S. 527 - 540; ISBN 3-18-091753-9; VDI-Verlag GmbH, Düsseldorf, 2003.

- [LEDER 99] Leder, L.: *Kooperative rechnerunterstützte Arbeit im Entwicklungs- und Konstruktionsbereich*; Praxis und Gestaltung kooperativer Arbeit; Zölch, M.; ISBN 3-7281-2687-x; Zürich, 1999.
- [LEUNIG 04] N.N.: *Zeitsynchronisation - Windows/Linux*; http://www.leunig.de/pro/netzwerk/funkuhr/neoclock_faqs.htm; Leunig GmbH; 2003.
- [LINDEMANN 00A] Lindemann U.: *Skript zur Vorlesung Methoden der Produktentwicklung*; TU München, Lehrstuhl für Produktentwicklung; München, 2000.
- [LINDEMANN 00B] Lindemann U.: *Skript zur Vorlesung Grundlagen der Produktentwicklung*; TU München, Lehrstuhl für Produktentwicklung; München, 2000.
- [LIPPOLD 01] Lippold, C.: *Eine domänenübergreifende Konzeptionsumgebung für die Entwicklung mechatronischer Systeme*; Shaker Verlag, Aachen, 2001.
- [LITTLE+ 90] Little, T., D., C.; Ghafoor, A.: *Synchronization and storage models for multimedia objects*; IEEE Journal on Selected Areas in Communications, 8, 3; S. 413 - 427; 1990.
- [LITZ+ 00] Litz, L. et al.: *Nutzung der Internettechnologie für die Automatisierungstechnik*; atp - Automatisierungstechnische Praxis Nr. 10/2000; S. 64 - 69; Oldenbourg Industrieverlag GmbH, München, 2000.
- [LOBIN 01] Lobin, H.: *Informationsmodellierung in XML und SGML*; Springer-Verlag, Berlin - Heidelberg - New York, 2001.
- [LUBICH 95] Lubich, Hannes., P.: *Towards a CSCW framework for scientific cooperation in Europe*; 1995.
- [MARQUART+ 03] Marquart, W.; Nagl, M.: *Arbeitsprozess-technische Unterstützung verfahrenstechnischer Entwicklungsprozesse*; atp - Automatisierungstechnische Praxis Nr. 04/2003; S. 45 - 58; Oldenbourg Industrieverlag GmbH, München, 2003.

- [MEIER 02] Meier, P.: *Vertikale Integration in der Automatisierungstechnik unter Verwendung objektorientierter Middleware*; Studienarbeit; Technische Universität München, Lehrstuhl für Informationstechnik im Maschinenwesen; Garching, 2002.
- [MEYER 04] Meyer, H.: *Modellbasierte Generierung einheitlicher Bedienobjekte für verteilte mechatronische Systeme*; Dissertation; TU-München Online Publikation; München, 2004.
- [MICROSOFT 95] N.N.: *The Component Object Model Specification*; <http://www.microsoft.com>; Microsoft Corporation; 1996.
- [MICROSOFT 04] N.N.: *Remote Desktop*; <http://www.microsoft.com/atwork/workspace/home/connect.aspx>; 2004.
- [MICROSOFT 05] N.N.: *Microsoft Developer Network*; <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfssystemtimespanclassmillisecondstopic.asp>; Microsoft Corporation; 2005.
- [MILBERG 96] Milberg, J.: *Kooperative Wertschöpfung Produkt, Prozess, Ressourcen*; Wettbewerbsfaktor Produktionstechnik - Aachener Perspektiven; S. 64 - 73; VDI-Verlag GmbH, Düsseldorf, 1996.
- [MUNKVOLD 03] Munkvold, B., E.: *Implementing collaboration technologies in industry*; ISBN 1-85233-418-5; Springer Verlag, London, 2003.
- [NAEDELE 03] Naedele, M.: *IT-Security for Automation Systems*; atp - Automatisierungstechnische Praxis Nr. 5; S. 84-91; Oldenbourg Industrieverlag GmbH, München, 2003.
- [NAMUR 01] N.N.: *NAMUR Empfehlung 91: Anforderungen an Systeme für anlagennahes Asset Management*; NAMUR-Gesellschaft; Leverkusen, 2001.
- [NETSCAPE 98] N.N.: *Introduction to SSL*; Internetquelle; <http://developer.netscape.com/docs/manuals/security/sslin/contents.htm>; 2001.

- [NITSCHKE 04] Nitsche, R.: *Systems Engineering beim Entwurf mechatronischer Systeme*; VDI-Berichte 1842; S. 9 - 18; ISBN 3-18-091842-X; VDI Verlag GmbH, Düsseldorf, 2004.
- [OESTEREICH 04] Oestereich, B.: *Die UML 2.0 Kurzreferenz für die Praxis*; ISBN 3-486-27604-2; Oldenbourg Wissenschaftsverlag GmbH, München, 2004.
- [OPCF FOUNDATION 04] N.N.: *OPC Foundation*; <http://www.opcfoundation.org>; 2004.
- [OTTERBACH 04] Otterbach, R.: *Effiziente Funktions- und Software-Entwicklung für mechatronische Systeme*; VDI-Berichte 1842; S. 119 - 126; ISBN 3-18-091842-X; VDI Verlag GmbH, Düsseldorf, 2004.
- [PNO 01] *PROFIBUS Guideline: Specification for PROFIBUS Device Description and Device Integration - Volume 3, FDT VI.2*; PROFIBUS Nutzerorganisation e. V.; Karlsruhe, 2002.
- [PNO 04] N.N.: *PROFINET Specification- Profinet CBA Architecture Description and Specification*; PROFIBUS Nutzerorganisation e. V.; Karlsruhe, 2004.
- [POHLMANN 00] Pohlmann, N.: *Firewall-Systeme*; ISBN 3-8266-4075-6; MITP-Verlag, Bonn, 2000.
- [PRITSCHOW+ 02] Pritschow, G.; Wälde, K.: *Übersicht über offene Prozessschnittstellen - Zukunftsperspektiven*; Fortschritt-Berichte VDI, Reihe 2, Nr. 593, IuK-Schnittstellen in der Produktionstechnik; S. 1 - 14; ISBN 3-18-359302-5; VDI-Verlag, Düsseldorf, 2002.
- [RATJE 97] Ratje, J.: *Der Feldbus kommt ins Rollen - Ein Situationsbericht*; atp - Automatisierungstechnische Praxis Nr. 39; S. 56 - 59; Oldenbourg Industrieverlag GmbH, München, 1997.

- [REICHWALD+ 97] Reichwald, R. et al.: *Telearbeit & Telekooperation: Bedingungen und Strategien erfolgreicher Realisierung*; Arbeitsbericht Nr. 12 (Mai 1997) des Lehrstuhls für Allgemeine und Industrielle Betriebswirtschaftslehre der Technischen Universität München; www.prof-reichwald.de; ISSN 0942-5098; 1997.
- [RING 95] Ring, K.: *Ein- und Verkauf von Dienstleistungen, Out-/Insourcing*; Dienstleistungen der Zukunft; S. 291 - 299; Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie; Berlin, 1995.
- [RÖMER 03] Römer, M.: *Rechnergestützte Analyse des Kommunikationsverhaltens feldbusfähiger Automatisierungsgeräte*; Dissertation; Lehrstuhl für Informationstechnik im Maschinenwesen; München, 2003.
- [RUSS+ 04] Russ, M.; Pöschl, M.: *Virtueller Funktionstest für die Entwicklung eingebetteter Software*; 2. Paderborner Workshop: Intelligente Mechatronische Systeme; S. 79 - 90; Bonifatius GmbH, Paderborn, 2004.
- [SCHÄFER 04] Schäfer, K.: *Entwicklung mechatronischer Systeme am Beispiel eines Flugzeugsystems*; VDI-Berichte 1842; S. 149 - 156; ISBN 3-18-091842-X; VDI Verlag GmbH, Düsseldorf, 2004.
- [SCHERER 98] Scherer, D.: *Internet wide software component development process and deployment integration*; 1998.
- [SCHERNIKAU 01] Schernikau, J.: *Gestaltung von mechatronikgerechten Organisationen in der Produktentwicklung*; Shaker Verlag, Aachen-Maastrich, 2001.
- [SCHMITZ+ 97] Schmitz, H.; Plöger, M.: *Konsequenter Einsatz der Simulation in allen Phasen der Fahrzeugsystementwicklung*; VDI Berichte 1315; S. 91 - 102; ISBN 3-18-091315-0; VDI-Verlag GmbH, Düsseldorf, 1997.
- [SCHÖN 00] Schön, A.: *Konzept und Architektur eines Assistenzsystems für die mechatronische Produktentwicklung*; Dissertation; Lehrstuhl für Konstruktion, Friedrich-Alexander-Universität Erlangen-Nürnberg; Erlangen, 2000.

- [SCHÖNER 02] Schöner, H.-P.: *Mechatronische Systeme im Automobil*; atp - Automatisierungstechnische Praxis Nr. 2; S. 34 - 39; Oldenbourg Industrieverlag GmbH, München, 2002.
- [SCHUHMANN 03] Schumann, D.: *Innentäter – Die unterschätzte Gefahr*; VDI-Nachrichten 4.7.2003; [HTTP:// www.vdi-nachrichten.com/vdi_nachrichten/](http://www.vdi-nachrichten.com/vdi_nachrichten/); 2003.
- [SFB453 04] N.N.: *Wirklichkeitsnahe Telepräsenz und Teleaktion, Sonderforschungsbereich 453 der DFG*; <http://www.lsr.ei.tum.de/>; Prof. Dr.-Ing. G. Färber; 2004.
- [SFB467 04] N.N.: *Sonderforschungsbereich 467 Wandlungsfähige Unternehmensstrukturen*; <http://www.sfb467.uni-stuttgart.de/projekte/c4/tc4.html>; 2004.
- [SIEGLER 98] Siegler, R.: *Mechatronischer Entwicklungsprozess am Beispiel Werkzeugmaschinen*; iw Seminarberichte 41; Milberg, J.; ISBN 3-89675-041-0; Herbert Utz Verlag GmbH, München, 1998.
- [SIEWERT+ 99] Siewert, J. R.; Feussner, H.; Etter, M.: *Telekommunikation im chirurgischen Alltag*; Tele- und computergestützte Chirurgie; S. 29 - 38; ISBN 3-540-65342-2; Springer-Verlag, Berlin - Heidelberg, 1999.
- [SMITH+ 01] Smith, D., J.; Simpson, K., G., L.: *Functional Safety*; ISBN 0-7506-5270-5; Butterworth-Heinemann, Oxford/England, 2001.
- [SNELL+ 02] Snell, J.; Tidwell, D.; Kulchenko, P.: *Webservice-Programmierung mit SOAP*; ISBN 3-89721-159-9; O'Reilly, Beijing, 2002.
- [SPRINGER 01] Springer, J.: *Telekooperation - vernetzte Arbeit mit integrierten Informations- und Kommunikationssystemen, 2001 0400/3.Jd 380*; Habilitationsschrift; Prof. H. Luczak; ISBN 3-8265-8456-2; Shaker Verlag, Aachen, 2001.
- [STEINBACH 04] Steinbach, E.: *Skriptum zur Vorlesung Medientechnik*; <http://www.lkn.ei.tum.de>; Lehrstuhl für Kommunikationsnetze, Technische Universität München; München, 2004.

- [STEYER 99] Steyer, R.: *Java 2 - Kompendium*; ISBN 3-8272-5564-3; Markt & Technik Verlag GmbH, München, 1999.
- [STOLLER-SCHAI 03] Stoller-Schai, D.: *E-Collaboration: Die Gestaltung internetgestützter kollaborativer Handlungsfelder*; Dissertation; Universität St. Gallen; Difo-Druck, Bamberg, 2003.
- [STOOSS 04] Stooss, R.: *VoIP-Performance-Analyse - Messkonzepte und Vorgehensweise*; LAN-Line, Ausgabe 9/2004; ISSN 0942-4172; Awi Verlagsgesellschaft mbH, Grasbrunn, 2004.
- [STURZEBECKER 00] Sturzebecher, D.: *A portable and flexible Framework for CSCW Systems*; Dissertation; 2000.
- [STÜTZEL 02] Stützel, B.: *Vergleichende Analyse des Umfelds und der Entwicklung mechatronischer Systeme*; Diplomarbeit; Lehrstuhl für Informationstechnik im Maschinenwesen; München, 2002.
- [SUN 04] N.N.: *Offizielle JAVA-Webseite der Firma SUN Microsystems*; <http://java.sun.com/>; 2004.
- [THINKOFIT 04] <http://www.thinkofit.com/>; 2004.
- [TODESCO 01] Todesco, R.: *Hyperbibliothek, Stichwort "Kollaboration"*; http://www.hyperkommunikation.ch/lexikon/lexikon_index.htm; 2001.
- [UHLMANN+ 02] Uhlmann, E.; Berger, R.: *IuK-Schnittstellen in der Produktionstechnik*; Fortschritt-Berichte VDI, Reihe 2, Nr. 593; S. 105 - 118; ISBN 3-18-359302-5; VDI-Verlag, Düsseldorf, 2002.
- [VA 99] Schätz, B. et al.: *Abschlußbericht der Vordringliche Aktion Entwicklung, Produktion und Service von Software für eingebettete Systeme in der Produktion*; Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie; 1999.
- [VDI 03] N.N.: *VDI-Richtlinie 2206: Entwicklungsmethodik für mechatronische Systeme*; Beuth Verlag, Berlin, 2003.

- [VDI 04] *Mechatronischer Systementwurf*; VDI-Berichte 1842; S. 61 - 70; ISBN 3-18-091842-X; VDI Verlag GmbH, Düsseldorf, 2004.
- [VDI 05] Steusloff, H.: *Globalisierung erzwingt das TeleEngineering*; VDI Nachrichten 6.5.05; VDI Verlag GmbH, Düsseldorf, 2005.
- [VDI 97] *Mechatronik im Maschinen- und Fahrzeugbau*; VDI Berichte 1315; ISBN 3-18-091315-0; VDI-Verlag GmbH, Düsseldorf, 1997.
- [VDI2187 02] *VDI/VDE 2187: Einheitliche Anzeige- und Bedienoberfläche für digitale Feldgeräte.*; Beuth-Verlag, Berlin, 2002.
- [VDMA 04] Preiml, H. et al.: *Leitfaden Software-Ergonomie*; Aus der Reihe Methoden und Verfahren; VDMA; ISBN 3-8163-0480-x; VDMA Verlag GmbH, Frankfurt am Main, 2004.
- [V-MODELL 97] N. N.: *Modell Entwicklungsstandard für IT - Systeme des Bundes*; <http://www.v-modell.iabg.de>; 1997.
- [VNC 04] N.N.: *RealVNC Ltd.*; <http://www.realvnc.com/>; 2004.
- [W3C 00] Bray, T.; et la.: *Extensible Markup Language (XML) 1.0 (Third Edition)*; W3C Recommendation; <http://www.w3c.org/XML/Core/#Publications>; World Wide Web Consortium; 2000.
- [W3C 03] N.N.: *SOAP Version 1.2 Part 0: Primer*; W3C Recommendation; <http://www.w3.org/TR/2003/REC-soap12-part0-20030624/>; World Wide Web Consortium; 2003.
- [WEHRES 97] Wehres, V.: *Wirtschaftliche und technische Auswirkungen der digitalen Feldtechnik auf die Prozessleittechnik*; atp - Automatisierungstechnische Praxis Nr. 3; S. 30 - 39; Oldenbourg Industrieverlag GmbH, München, 1997.
- [WEISS 02] Weiss, M.: *TCP/IP-Handbuch*; ISBN 3-7723-5026-7; Franzis Verlag GmbH, Poing, 2002.
- [WIKIPEDIA 05] *Online Lexikon*; <http://de.wikipedia.org>; 2005.

- [WINKLER 98] Winkler, L.: *Vortrag zur 13. Internationalen Wissenschaftlichen Konferenz, Hochschule Mittweida*; http://telecom.htwm.de/telecom/artikel/trendlinien_der_kt/sld032.htm; 1998.
- [WONDERWARE 04] Wonderware Corporation: *Suite Voyager Product Overview*; <http://www.wonderware.com/products/-visualization/sv/>; 2004.
- [X.ORG 04] N.N.: *X.org Foundation*; <http://www.x.org>; 2004.
- [XU 00] Xu, C.: *Interaction and collaboration mechanisms for distributed communities and groups in educational settings*; Dissertation; Technische Universität München; München, 2000.
- [ZITTERBART 02] Zitterbart, M.: *Skriptum zur Vorlesung Next Generation Internet, Sommersemester 2002*; <http://www.tm.uka.de/lehre/SS02/vorlesungen/NGI/ngi01-Einfuehrung-2up.pdf>; Institut für Telematik, Universität Karlsruhe; Karlsruhe, 2002.
- [ZWICKER 98] Zwicker, E.: *Unterstützung der unternehmensübergreifenden Produktentwicklung durch den Einsatz moderner Informationstechnologien*; Fortschritt-Berichte VDI , Reihe 20, Nr. 288, Rechnerunterstützte Verfahren; ISBN 3-18-328820-6; VDI-Verlag, Düsseldorf, 1998.

Abbildungsverzeichnis

Abbildung 1-1: Struktur der vorliegenden Arbeit	3
Abbildung 2-1: Struktur mechatronischer Systeme (in Anlehnung an [HEIMANN+ 01]).....	6
Abbildung 2-2: Phasenkonzept für Produktentwicklungsprozesse [HABERFELLNER+ 02].....	7
Abbildung 2-3: Das mechatronische Basissystem [KALLMEYER 98], [LIPPOLD 01]	13
Abbildung 2-4: 3-Ebenen-Vorgehensmodell der Mechatronik [BENDER 04B], [RUSS 04].....	15
Abbildung 2-5: Werkzeuge in den integrativen Phasen (basierend auf [Gausemeier+ 00])....	17
Abbildung 2-6: Teilaufgaben im Verlauf des Entwicklungsprozesses	19
Abbildung 2-7: Ergänzung der Kernkompetenz durch externe Ressourcen	22
Abbildung 2-8: Geografische Verteilung externer Ressourcen	24
Abbildung 2-9: Durchflussregler	26
Abbildung 3-1: Strukturierung von Kommunikationsressourcen für die E-Collaboration.....	34
Abbildung 3-2: Phasenmodell einer idealtypischen Ko-Aktion [STOLLER-SCHAI 03]	40
Abbildung 3-3: Die Grundlegende MACS-Architektur [STURZEBECKER 00]	42
Abbildung 3-4: Einordnung von Multimediastandards nach [ELKO 04], [WINKLER 98].....	43
Abbildung 3-5: Einsatz eines Konferenz-Servers	45
Abbildung 3-6: Beispiel für ein Desktop-Sharing-Werkzeug.....	47
Abbildung 4-1: Entwicklung des Konzepts für ein universelles Remote-Framework.....	53
Abbildung 4-2: Grundlegende Faktoren einer angepassten Remote-Lösung	54
Abbildung 4-3: Zugriff verschiedener Experten auf ein mechatronisches System.....	56
Abbildung 4-4: Einordnung des Kommunikationsszenarios	58
Abbildung 4-5: Gesamtarchitektur der Lösung.....	62
Abbildung 4-6: Fachgebietsunabhängige Klassifizierung mechatronischer Komponenten ...	66
Abbildung 4-7: Screenshot des Live-Videobildes eines Stellventils	68
Abbildung 4-8: Filterkonzept für Remote-Interaktionen	70
Abbildung 4-9: Modulare GUI-Elemente eines Rechnerwerkzeugs.....	74
Abbildung 4-10: Die Betriebsmodi lokaler Server und entfernter Client	75
Abbildung 4-11: Konzept für flexibles Message-Routing auf lokaler- und Remote-Seite.....	75
Abbildung 4-12: Anforderungen an sichere Kommunikation, nach [POHLMANN 00].....	77
Abbildung 4-13: Vorgehensweise zur Realisierung eines verteilten Werkzeugs	79
Abbildung 5-1: Risikograph-Analyse nach IEC 61508 [INGREY+ 04]	83
Abbildung 5-2: Einordnung der Werkzeugfunktionalität in Zugriffsklassen	84
Abbildung 5-3: Stationen eines Funktionsaufrufs.....	86
Abbildung 5-4: Risikograph für die Einordnung von Funktionen in Zugriffsklassen.....	88
Abbildung 5-5: Definition des Wurzelements.....	90
Abbildung 5-6: Definition des Grundelements für eine Klassifizierung	90
Abbildung 5-7: Definition des FunctionType-Elements	91
Abbildung 5-8: Handhabung von Funktionsparametern	91
Abbildung 5-9: Ausschnitt aus dem Logfile einer entfernten Werkzeuginstanz	92
Abbildung 5-10: Ausschnitt aus dem Logfile einer lokalen Werkzeuginstanz.....	93
Abbildung 5-11: Multimediakomponenten für das Remote-Framework.....	95
Abbildung 5-12: Schnittstelle zwischen Remote-Framework und Multimedia-Wrapper	96
Abbildung 5-13: A/V-Übertragungsprotokolle im TCP/IP-Protokollstack	98
Abbildung 5-14: Funktionsweise der A/V-Kommunikation, nach [STOOS 04]	99

Abbildung 5-15: Einflussfaktoren auf die Ende-zu-Ende-Verzögerung [BADACH 04]	102
Abbildung 5-16: Objektmodell des Remote-Frameworks mit verteilten Werkzeugen.....	105
Abbildung 5-17: Schnittstelle zwischen Werkzeug und Remote-Framework	106
Abbildung 5-18: Informationsfluss auf der lokalen Seite	108
Abbildung 5-19: Informationsfluss auf der Remote-Seite	109
Abbildung 5-20: Ausführung einer Funktion (Funktionsaufruf)	110
Abbildung 5-21: Ausführung einer Funktion (Empfang der Antwort).....	110
Abbildung 5-22: Beispiele für Fehler im Informationsfluss	111
Abbildung 5-23: Beispiel der SOAP-Anfrage eines Clients an einen Server [W3C 03].....	113
Abbildung 5-24: Beispiel der SOAP-Antwort des Servers [W3C 03].....	113
Abbildung 5-25: Verschlüsselte Kommunikation über öffentliche Netze	116
Abbildung 5-26: Kommunikation über dedizierte ISDN-Leitungen	117
Abbildung 5-27: Modus 1: Callback unter Einsatz des http-Protokolls.....	119
Abbildung 5-28: Modus 2: SOAP unter Verwendung des TCP-Protokolls.....	120
Abbildung 5-29: Modus 3: Zyklisches Abfragen der Antworten durch den Client.....	121
Abbildung 6-1: Vergleich zwischen konventioneller Verkabelung und Feldbustechnik	124
Abbildung 6-2: Aufbau des Prüfsystems MoTest	128
Abbildung 6-3: Basis-Architektur des Prüfsystems MoTest.....	132
Abbildung 6-4: Verbindung von MoTest mit dem Remote-Framework	133
Abbildung 6-5: Multimedia-Steuerungs- und Ausgabefenster	134
Abbildung 6-6: Ausschnitt aus dem Zustandsdiagramm des Prüfsystems	138
Abbildung 6-7: Ausschnitt aus einem Testcase für Profibus DP - Geräte	138
Abbildung 6-8: Messkurve der Auslastung bei 64 kBit/s	143
Abbildung 6-9: Videobild eines Testgerätes im CIF-Format, bei ca. 6 kByte/s Datenstrom	143

Tabellenverzeichnis

Tabelle 3-1: Remote-Funktionalität in der Automatisierungstechnik [BRAUNE+ 99]	36
Tabelle 3-2: Einordnung von E-Learning-Systemen nach [XU 00]	38
Tabelle 3-3: Synchron und Asynchrone Kommunikation in den konstruktiven Phasen.....	39
Tabelle 3-4: Die vier Ebenen des KOMFORCE-Referenzmodells [GAUSEMEIER+ 00].....	41
Tabelle 3-5: Werkzeuge zur Unterstützung von verteilter Kooperation [MUNKVOLD 03]	44
Tabelle 3-6: Bewertung existierender CSCW-Lösungen	50
Tabelle 4-1: Klassifikation der Ausprägungen von Interaktion, mit Beispielen [LUBICH 95].	57
Tabelle 4-2: Zeitbedingungen in kooperativen Multimediasystemen [HERMANN 97].....	60
Tabelle 4-3: Rollenmodell für verteilte Werkzeuge in den integrativen Phasen	66
Tabelle 4-4: Funktionen der Remote-Komponenten.....	71
Tabelle 5-1: Funktionsklassen für die Beschränkung des Remote-Zugriffs	85
Tabelle 5-2: Relevanz der existierenden Tool-Kategorien nach [STOLLER-SCHAI 03].....	94
Tabelle 5-3: Bandbreitenanforderungen der wichtigsten Videoformate [STEINBACH 04]	100
Tabelle 5-4: Audio-Codecs mit Bandbreitenanforderung und MOS-Wert [BADACH 04]	101
Tabelle 6-1: Rollenverteilung der am Test Beteiligten Personen	127
Tabelle 6-2: Gründe für die Auswahl von Netmeeting	134
Tabelle 6-3: Zuordnung der Entwicklerrollen zur Basisfunktionalität von MoTest.....	135
Tabelle 6-4: Gemessene Parameter des Remote-Tests	140
Tabelle 6-5: Ergebnisse der Analyse des Bandbreitenbedarfs multimedialer Datenströme ..	142

Stichwortverzeichnis

- 3-Ebenen-Modell 8, 14
- Aktoren 16
- Anwendungen, asynchrone 11
- Anwendungen, synchrone 11
- Application-Sharing 46
- Authentizität 48
- Bereich, spezifizierender 14
- CAD-Werkzeuge 39
- CERT 204
- CIF 100
- Client-Server-Prinzip 120
- Common Criteria* 203
- CORBA 112
- CSCW 30, 34, 78
- DCOM 112
- DeKOS 36, 64
- Desktop-Sharing 46
- DIN ISO 9000 21
- Dispatcher 63
- Distance-Learning 38
- DTD 89
- Echtzeit 60
- Echtzeitrechner 16
- E-Collaboration 43
- E-Learning 37
- Entwicklungsprozess 13
- EQUAL 14
- Expertise 18
- FDT 64
- Fernwirktechniken 47
- Firewall 115
- Gerät 7
- Gerät, mechatronisches 6
- Grundschutzhandbuch 201
- GSTOOL 202
- GUI 63
- H.323 98
- Hardware-in-the-Loop 16
- Hardware-In-The-Loop 30
- Honeypot-Systeme 115
- HTML-Seiten 120
- IEC 61508 26, 82
- IEC 61511 82
- IETF 43
- Innovationszyklen 20
- Integrationsgrad 14
- Integrität 48
- Intrusion-Detection-Systeme 115
- ISDN 43, 101
- ISO 7498-2 203
- ISO-OSI-Referenzmodell 43
- ITSEC 204
- IT-Sicherheit 32
- ITU 43
- Jitter-Puffer 103
- KMU 18, 22
- Koaktion 10
- Ko-Begriffe 9
- Kollaboration 8, 9
- KOMFORCE-Referenzmodell 41
- KOMFORCE-Referenz-Modell 105
- Kommunikationskanäle 72
- Komponenten-Ebene 14
- Komponentenintegration 30
- Kontrollmodul 70
- Kooperation 8, 9
- Koordination 9
- MACS 41
- MBONE 44
- Mechatronik 5
- Mechatronische Systeme 12
- MOS-Wert 101
- MPEG 44
- Multimediakomponenten 72, 95

- Multimedia-Komponenten 62
Multimediaverbindung 32
NAMUR 64
NE 91 64
Offline 11
Online 11
OPC 48
Orange Book 204
Outsourcing 19
Phasen, integrative 15
Phasenmodell der Ko-Aktion 40
Produkt, mechatronisches 6
Produktkomplexität 17
Produktzyklen 20
PROTEUS 36
Prozess, technischer 6
QCIF 100
QoS-Anforderungen 97
Qualitätssicherung 23, 30
Rechnerwerkzeuge 16
Remote-Desktop 46
Remote-Technologien 35
Revisions-Systeme 115
Risikoanalyse 203
Risikograph-Analyse 83
Risikomatrix-Analyse 83
Risikoparameter 87
RPC 112
RTP 99
SCADA 47
Sensoren 16
Serialisierungsverzögerung 102
Setting, kollaboratives 57
SFB Improve 39
SGML 89
SIL-Klasse 26
SIL-Klassen 82
Simulationsanwendungen 16
SINA 203
SIP-Protokoll 98
SOAP 112, 119
Sprachsegment 102
Subsystemebene 15
System, mechatronisches 6
System-Ebene 14
Systemintegration 23
Telechirurgie 37
TeleTest 55
UML 104
UML-Sequenzdiagramme 109
VDE Norm 3000 203
VDI 2206 18
VDI/VDE 2187 64
Vertraulichkeit 48
V-Modell 14
V-Modell XT 65
Vorgehensmodelle 14
VPN-Tunnel 116
Wrapper 61
XML 36, 89
XML-Schemata 89
XSL 89

Anhang A Anwendungsszenarien

Beispiel 1:

Das universelle Werkzeug für Laborarbeiten an mechatronischen Systemen der Feinwerktechnik mit der Bezeichnung TESLA⁸⁵ dient zur Durchführung von Messungen und Analysen. Das Werkzeug ist aus modularen Hard- und Softwarekomponenten aufgebaut und kann beliebig skaliert werden, was für den Anwender erhöhte Komplexität bedeutet. Neben der Möglichkeit zur Durchführung von Messungen im Echtzeitraster beinhaltet es Module zur Datenaufbereitung und Auswertung wie beispielsweise Spektralanalyse und Modalanalyse. Im Zusammenhang mit dem Gebrauch solcher Analyse-Module wird die Notwendigkeit betont, die Auswirkung von Einstellungen, beispielsweise Filterfrequenzen, sofort vor Ort beobachten zu können [LANDWEHR 01]. Ein Anwender, der kein unmittelbarer Spezialist in diesem Gebiet ist, sondern nur an den Ergebnissen interessiert ist, wird mit hohem Aufwand konfrontiert. Durch Kollaboration mit einem externen Experte aus der Mess- und Analysetechnik kann dieser Aufwand erheblich verringert werden.

Beispiel 2:

Bei Hardware-In-The-Loop-Techniken erfolgt der Einbau des realen mechatronischen Produktes bzw. Teilkomponenten davon in eine komplexe Simulationsumgebung. Diese so genannten HIL-Umgebungen verfügen über umfangreiche Hard- und Softwarekomponenten deren Bedienung nicht nur die notwendige Kenntnis über den Prüfstand selbst sondern auch über die fachlichen Hintergründe erfordert. Beispiele hierzu sind Optimierungsaufgaben unter Einsatz numerischer Optimierungsverfahren [KLOTZBACH+ 03] oder der Test von On-Board-Diagnose-Steuergeräten im Automobilbau (ETAS LabCar-Umgebung) [LEBERT 03]. Um Aussagekräftige Ergebnisse der HIL-Simulation zu erhalten ist der richtige Umgang mit der Simulationsumgebung Voraussetzung (beispielsweise Einstellung von Prüfstandsparametern). Die Durchführung solcher Tests erfolgt wesentlich effizienter wenn sie in Kollaboration mit einem Prüfstandsexperten erfolgt, der für die HIL-Umgebung verantwortlich ist, während sich der Entwickler vollständig auf sein Produkt konzentrieren kann.

Beispiel 3:

In der Investitionsgüterindustrie haben sich offene automatisierungstechnische Systeme etabliert. Diese zeichnen sich durch einheitliche, herstellerunabhängige Kommunikationsschnittstellen wie beispielsweise Feldbusanbindung aus. Um ein sicheres und zuverlässiges Zusammenspiel aller Teilkomponenten zu erreichen ist die Konformität zu den vorhandenen, oftmals sehr komplexen Kommunikationsstandards unabdingbar. Die Erreichung dieser Konformität erfordert

⁸⁵ Test Site Laboratory Automation

idealerweise schon in der Implementierungsphase die Durchführung von Konformitätstests mit komplexen Test-Werkzeugen im Rahmen von Vortests [KUTTIG+ 03], [RÖMER 03]. Dabei ist nicht nur das komplexe Prüfwerkzeug an sich zu handhaben, sondern es muss gleichfalls die z. T. komplexe Testspezifikation eingehalten werden um aussagekräftige Vortests durchzuführen. Eine weitere schwieriger Aufgabe ist die Analyse der Testprotokolle zur Eingrenzung eines Fehlers. Ein unabhängiger Experte eines Prüflabors, der über ungleich mehr Erfahrung verfügt kann sein breites Wissen im Rahmen der verteilten Kollaboration einbringen und so den Entwickler wirkungsvoll unterstützen.

Beispiel 4:

Im Automobilbau werden Steuergeräte verschiedenster Zulieferer oftmals schon in Form von Prototypen testweise verbaut. In modernen Automobilen sind diese, zur Erfüllung der Gesamtfunktionalität, bisweilen hochgradig miteinander vernetzt. Treten während der Integrationsphase der Geräte Probleme auf, wird oftmals die Expertise des Zulieferers zur Identifikation eines Fehlers oder Durchführung einer Korrektur benötigt. Um das Gesamtverhalten des Geräteverbunds zu überprüfen kann das Gerät jedoch nicht einfach herausgelöst an den Zulieferer zurückgeschickt werden. Ein Entwickler müsste folglich anreisen was bei spontanem Auftreten von Problemen Komplikationen aufwirft. Verteilte Kollaboration zwischen dem Inbetriebnehmer beim Fahrzeughersteller und dem Entwickler des Zulieferers würde die Effizienz stark erhöhen.

Ein ähnliches Problem findet sich in den späteren Phasen des Produktzyklus bei der After-Sales-Diagnose in Werkstätten. Diese sind im Umgang mit der umfangreichen IT in modernen Fahrzeugen häufig überfordert, weshalb laut einer Studie bis zu 50% der Steuergeräte umsonst ausgetauscht werden. Die gemeinschaftliche Durchführung der Diagnose mit einem entfernten Spezialisten kann diese Quote minimieren.

Beispiel 5:

Zu Fehlersuche aber auch zur Optimierung während der Inbetriebnahme automatisierungstechnischen Anlagen werden im Zusammenhang mit feldbusfähigen Geräten Analysewerkzeuge wie Busmonitore eingesetzt. Mit diesen Werkzeugen ist möglich, auf feingranularer Ebene die Kommunikation zwischen den vernetzten mechatronischen Komponenten aufzuzeichnen. Viele Fehler lassen sich nur auf der untersten Kommunikationsebene identifizieren. Dies setzt jedoch voraus, dass der Systemintegrator vor Ort über detaillierte Kenntnisse auf diesen Ebenen verfügt. Ansonsten ist er auf die Unterstützung eines Spezialisten für Feldbuskommunikation angewiesen.

Anhang B Rollendefinitionen

<p>Rollenbezeichnung: Spezialist für Teilkomponente V-Modell XT: Softwarearchitekt, Softwareentwickler Bergner: Komponentenentwickler</p>
<p>Wer kann dieser Rolle übernehmen:</p> <ul style="list-style-type: none"> ▪ Entwickler/Implementierer der Komponente ▪ Zulieferer der Komponente
<p>Kennzeichen der Rolle:</p> <ul style="list-style-type: none"> ▪ Kennt die Implementierung (Whitebox-Charakter) ▪ Kennt das Soll-Verhalten der Teilkomponente ▪ Besitzt Erfahrung mit der Komponente
<p>Aktionen:</p> <ul style="list-style-type: none"> ▪ Eingriff auf unterster Ebene (Parametrierung, Konfiguration,...) ▪ Code-Modifikation , Konsolenbefehle
<p>Ausprägungen der Applikation für diese Rolle:</p> <ul style="list-style-type: none"> ▪ Editiermöglichkeit und Transfermöglichkeit für Programmcode ▪ Editierung von Speicherzellen und interner Parameter ▪ Steuerung der Komponente, bzw. des Systems
<p>Rollenbezeichnung: Spezialist für offene Standards (z. B. Kommunikationsstandards) V-Modell XT: Softwarearchitekt, Softwareentwickler Bergner: Komponentenentwickler</p>
<p>Wer kann dieser Rolle übernehmen:</p> <ul style="list-style-type: none"> ▪ Externer Spezialist (z. B. Kompetenz-Zentrum) ▪ Zulieferer
<p>Kennzeichen der Rolle:</p> <ul style="list-style-type: none"> ▪ Kennt die Implementierung und Spezifikation von Standards ▪ Kennt das Sollverhalten von Standardkomponenten ▪ Kein Wissen über das Gesamtsystem oder die Implementierung der Teilkomponente ▪ Kennt allgemeine und Wiederkehrende Probleme bei der Anwendung von Standards
<p>Aktionen:</p> <ul style="list-style-type: none"> ▪ Eingriff auf der Koppelungsebene (Blackbox) ▪ Protokollierung und des Datenverkehrs ▪ Herbeiführen von Zuständen in der Komponente bzw. im System
<p>Ausprägungen der Applikation für diese Rolle:</p> <ul style="list-style-type: none"> ▪ Steuerung des Systems, bzw. des Prozessablaufs ▪ Konsolensteuerung für elementare Befehle ▪ Protokollierungs- und Analysefunktionalität

<p>Rollenbezeichnung: Spezialist für Gesamtsystem V-Modell XT: Systemarchitekt, Systemintegrator Bergner: Systemarchitekt</p>
<p>Wer kann dieser Rolle übernehmen:</p> <ul style="list-style-type: none"> ▪ Lokaler Systemintegrator ▪ Lokaler Systemspezifizierer
<p>Kennzeichen der Rolle:</p> <ul style="list-style-type: none"> ▪ Kennt nicht die Implementierung der Teilkomponenten (Blackbox-Charakter) ▪ Kennt das Soll-Verhalten des Systems (Spezifikation des Gesamtsystems) ▪ Kennt die globalen Auswirkungen von Interaktionen
<p>Aktionen:</p> <ul style="list-style-type: none"> ▪ Eingriff nur auf der Applikationsebene der Teilkomponenten (Blackbox) ▪ Kontrolle der globalen Interaktionen ▪ Vergleich des globalen Soll-Ist-Verhaltens
<p>Ausprägungen der Applikation für diese Rolle:</p> <ul style="list-style-type: none"> ▪ Steuerung des Systems, bzw. des Prozessablaufs ▪ Protokollierungs- und Analysefunktionalität auf Systemebene ▪ Ausführung von Testabläufen

<p>Rollenbezeichnung: Prüfspezialist (Spezifikations-Spezialist) V-Modell XT: Prüfer, Anforderungsanalytiker (Auftraggeber und Auftragnehmer)</p>
<p>Wer kann dieser Rolle übernehmen:</p> <ul style="list-style-type: none"> ▪ Lokale Prüfabteilung ▪ Externe Prüfinstitution
<p>Kennzeichen der Rolle:</p> <ul style="list-style-type: none"> ▪ Kennt nicht die Implementierung der Teilkomponenten (Blackbox-Charakter) ▪ Kennt das Soll-Verhalten des Systems (Spezifikation des Gesamtsystems) ▪ Detailkenntnisse über die zugrunde liegende Spezifikation
<p>Aktionen:</p> <ul style="list-style-type: none"> ▪ Vergleich des globalen Soll-Ist-Verhaltens einer Teilkomponente bzw. des Gesamtsystems
<p>Ausprägungen der Applikation für diese Rolle:</p> <ul style="list-style-type: none"> ▪ Steuerung des Systems, bzw. des Prozessablaufs ▪ Ausführung von Testabläufen ▪ Protokollierung und Analyse der Testergebnisse

Anhang C Zugriffsklassen

C-1 Erklärungen und Beschränkungsmaßnahmen

Bezeichner	Aktion des Remote-Frameworks
1	Uneingeschränkter Remote-Zugriff: Die Funktion darf ohne Einbeziehung der lokalen Person ausgeführt werden. Diese bekommt lediglich eine Meldung im Ausgabefenster angezeigt.
2a	Lesender Zugriff mit Benachrichtigung der lokalen Person: Die lokale Person wird über die Funktionsausführung explizit mittels einer Dialogbox benachrichtigt.
2b	Lesender Zugriff erst nach Freigabe durch die lokale Person: Die lokale Person wird über die Funktionsanfrage in Form einer Dialogbox informiert. Sie kann die Ausführung der Funktion zulassen oder ablehnen.
3a	Schreibender Zugriff mit Benachrichtigung der lokalen Person: Die lokale Person wird über die Funktionsanfrage in Form einer Dialogbox informiert und explizit auf den schreibenden Zugriff hingewiesen.
3b	Schreibender Zugriff nach Freigabe durch die lokale Person: Die lokale Person wird über die Funktionsanfrage in Form einer Dialogbox informiert und explizit auf den schreibenden Zugriff hingewiesen. Sie kann die Ausführung der Funktion zulassen oder ablehnen.
3c	Schreibender Zugriff nach Freigabe durch die lokale Person für jedem Einzelschritt: Die lokale Person wird über die Ausführung eines jeden Einzelschrittes einer Funktion in Form einer Dialogbox informiert und explizit auf den schreibenden Zugriff hingewiesen. Sie kann die Ausführung des Schrittes zulassen, den Schritt überspringen oder die weitere Funktionsausführung abbrechen.
4a	Schreibender Zugriff nach zusätzlicher Freigabe durch den Bediener des Systems vor Ort: Die lokale Person wird über die Funktionsanfrage in Form einer Dialogbox informiert und explizit auf den schreibenden Zugriff hingewiesen. Sie kann die Ausführung der Funktion zulassen oder ablehnen, darf dies jedoch erst nach Rücksprache mit einem vor Ort stehenden Bediener des Systems tun. Das Personal vor Ort hat so die Möglichkeit, zusätzliche Absicherungsmaßnahmen zu treffen.

4b	<p>Schreibender Zugriff nach zusätzlicher Freigabe durch den Bediener des Systems vor Ort für jedem Einzelschritt:</p> <p>Die lokale Person wird über die Ausführung eines jeden Einzelschrittes einer Funktion in Form einer Dialogbox informiert und explizit auf den schreibenden Zugriff hingewiesen. Sie kann die Ausführung des Schrittes zulassen, den Schritt überspringen oder die weitere Funktionsausführung abbrechen, darf dies jedoch erst nach Rücksprache mit einem vor Ort stehenden Bediener des Systems tun. Das Personal vor Ort hat so die Möglichkeit, zusätzliche Absicherungsmaßnahmen zu treffen.</p>
5	<p>Kein Remote-Zugriff</p> <p>Die Ausführung der Funktion wird vom Remote-Framework in jedem Fall geblockt, die lokale Person wird über diesen Vorgang informiert.</p>

C-2 Modellierung der Zuordnungssystematik mit XML-Schemata

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: -->
<!-- Last Changes: 2005-04-07 by Florian Kuttig -->
<!--
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: -->
<xs:schema xmlns="http://www.itm.tum.de/2005/cls"
xmlns:xs="http://www.w3.org/2001/XMLSchema" targetName-
space="http://www.itm.tum.de/2005/cls" elementFormDe-
fault="unqualified" attributeFormDefault="unqualified">
  <!--
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: -->
    <!-- Declaration of Document Element -->
    <!--
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: -->
      <xs:element name="CLASSIFICATIONS">
        <xs:annotation>
          <xs:documentation>Comment describing your root
element</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="FILEINFO"
type="FileInfoType"/>
            <xs:element name="CLASSIFICATION"
type="ClassificationType" minOccurs="1" maxOccurs="unbounded"/>
            <xs:element name="DESCRIPTION"
type="DescriptionType" minOccurs="0"/>
          </xs:sequence>
          <xs:attribute name="NAME" type="xs:string"
use="required"/>
        </xs:complexType>
      </xs:element>
    <!--
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: -->
      <!-- Declaration of FileInfoType -->
      <!--
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:: -->
        <xs:complexType name="FileInfoType">
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="AUTHOR" type="xs:string"
use="required"/>
              <xs:attribute name="DATE" type="xs:date"
use="required"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    <!--

```

```

        <xs:attribute name="VERSION" type="xs:float"
use="required"/>
        <xs:attribute name="FILETYPE" type="xs:string"
use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<!--
:~::~:
:: -->
    <!-- Declaration of ClassificationType          -->
    <!--
:~::~:
:: -->
    <xs:complexType name="ClassificationType">
        <xs:sequence>
            <xs:element name="FUNCTION" type="FunctionType"
maxOccurs="unbounded"/>
            <xs:element name="DESCRIPTION"
type="DescriptionType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="NAME" type="xs:string"
use="required"/>
    </xs:complexType>
    <!--
:~::~:
:: -->
    <!-- Declaration of FunctionType          -->
    <!--
:~::~:
:: -->
    <xs:complexType name="FunctionType">
        <xs:sequence>
            <xs:element name="PARAM" type="ParamType" minOc-
curs="0" maxOccurs="unbounded"/>
            <xs:element name="DESCRIPTION"
type="DescriptionType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="CLASS" type="ClassType"
use="required"/>
        <xs:attribute name="NAME" type="xs:string"
use="required"/>
    </xs:complexType>
    <!--
:~::~:
:: -->
    <!-- Declaration of ParamType          --
>
    <!--
:~::~:
:: -->
    <xs:complexType name="ParamType">
        <xs:sequence>
            <xs:element name="DESCRIPTION"
type="DescriptionType" minOccurs="0"/>
        </xs:sequence>

```



```

        <xs:attribute name="CLASS" type="ClassType"
use="optional"/>
        <xs:attribute name="NAME" type="xs:string"
use="required"/>
        <xs:attribute name="RANGE" type="xs:string"
use="optional"/>
    </xs:complexType>
    <!--
:: -->
    <!-- Declaration of DescriptionType -->
    <!--
:: -->
    <xs:simpleType name="DescriptionType">
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <!--
:: -->
    <!-- Declaration of RangeType -->
    <!--
:: -->
    <xs:simpleType name="RangeType">
        <xs:union>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:pattern
value="\d{0,}?.?\d{0,}..\d{0,}?.?\d{0,}"/>
                </xs:restriction>
            </xs:simpleType>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:pattern value="\d{0,}?.?\d{0,}"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:union>
    </xs:simpleType>
    <!--
:: -->
    <!-- Declaration of ClassTypes -->
    <!-- Class-Types represent Access-Classes -->
    <!--
:: -->
    <xs:simpleType name="ClassType">
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="1"/>
            <xs:enumeration value="2a"/>
            <xs:enumeration value="2b"/>
            <xs:enumeration value="3a"/>
            <xs:enumeration value="3b"/>
            <xs:enumeration value="3c"/>
            <xs:enumeration value="4a"/>
            <xs:enumeration value="4b"/>
        </xs:restriction>
    </xs:simpleType>

```

```
        <xs:enumeration value="5"/>
        <!-- Class 1: unrestricted access -->
        <!-- Class 2a: Read-Only, local user informed about
the enquiry -->
        <!-- Class 2b: Read-Only, local user must explic-
iteluy allow the enquiry -->
        <!-- Class 3a: Write-Access, local user informed
abot the action -->
        <!-- Class 3b: Write-Access, local user must ex-
plicitely allow the action -->
        <!-- Class 3c: Write-Access, local user must ex-
plicitely allow every single step of the action -->
        <!-- Class 4b: Write-Access, local system-operator
must explicitely allow the action -->
        <!-- Class 4b: Write-Access, local system-operator
must explicitely allow every single step of the action -->
        <!-- Class 5: no remote-access -->
    </xs:restriction>
</xs:simpleType>
</xs:schema>
```

C-3 Klassifizierung der MoTest-Funktionalität (Ausschnitt)

```

<?xml version="1.0" encoding="utf-8"?>
<cls:CLASSIFICATIONS NAME="MoTest_POFIBUS"
xmlns:cls="http://www.itm.tum.de/2005/cls"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.itm.tum.de/2005/cls Classifica-
tions_V1.0.xsd">
  <FILEINFO AUTHOR="Florian Kuttig" DATE="2005-03-20"
FILETYPE="Classification" VERSION="1.0">MoTest Classifica-
tion</FILEINFO>

  <CLASSIFICATION NAME="PROFIBUS-Testing: Flow-Sensors">
    <FUNCTION CLASS="3a" NAME="startTC">
      <PARAM CLASS="3b" NAME="TCName" RANGE="CheckReset">
        <DESCRIPTION>Initiate Reset-Function and test
successful reboot of device</DESCRIPTION>
      </PARAM>
      <PARAM CLASS="3a" NAME="InitialStep" RANGE="1..5">
        <DESCRIPTION>Start with an initial step lower
than step 5</DESCRIPTION>
      </PARAM>
      <PARAM NAME="setProfibusAdress" CLASS="3b">
        <DESCRIPTION>changes the adress of the de-
vice</DESCRIPTION>
      </PARAM>
      <DESCRIPTION>Executes a Testcase</DESCRIPTION>
    </FUNCTION>

    <FUNCTION CLASS="3a" NAME="stopTC">
      <DESCRIPTION>Stops a Testcase</DESCRIPTION>
    </FUNCTION>

    <FUNCTION CLASS="2a" NAME="loadTC">
      <DESCRIPTION>loads a testcase into the
testsuite</DESCRIPTION>
    </FUNCTION>

    ...

    <DESCRIPTION>Access-Restrictions for testing devices of
type: Flow Sensor</DESCRIPTION>
  </CLASSIFICATION>

  <CLASSIFICATION NAME="PROFIBUS-Testing: Linear Actuators">
    <FUNCTION CLASS="4a" NAME="startTC">
      <PARAM CLASS="4b" NAME="TCName" RANGE="CheckReset">
        <DESCRIPTION>Initiate Reset-Function and test
successful reboot of device</DESCRIPTION>
      </PARAM>

      <PARAM CLASS="5" NAME="InitialStep" RANGE="1..5">

```

```
        <DESCRIPTION>Start with an initial step higher
than step 5</DESCRIPTION>
        </PARAM>

        <PARAM NAME="checkMaximumSetpoint" CLASS="4b">
        <DESCRIPTION>Drive Actuator to maximum posi-
tion</DESCRIPTION>
        </PARAM>

        <PARAM NAME="setProfibusAdress" CLASS="3b">
        <DESCRIPTION>changes the adress of the de-
vice</DESCRIPTION>
        </PARAM>

        <DESCRIPTION>Executes a Testcase</DESCRIPTION>

    </FUNCTION>

    <FUNCTION CLASS="3a" NAME="stopTC">
        <DESCRIPTION>Stops a Testcase</DESCRIPTION>
    </FUNCTION>

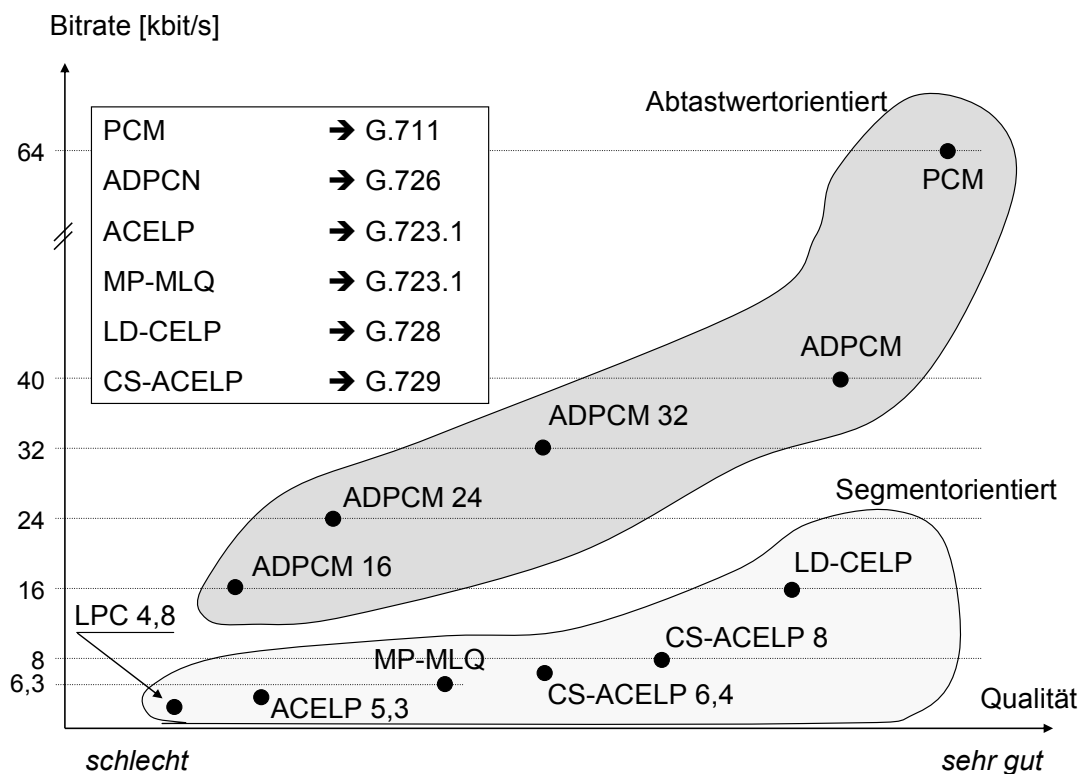
    <FUNCTION CLASS="2a" NAME="loadTC">
        <DESCRIPTION>loads a testcase into the
testsuite</DESCRIPTION>
    </FUNCTION>

        ...

        <DESCRIPTION>Access-Restrictions for testing devices of
type: Linear Actuators Sensor</DESCRIPTION>
    </CLASSIFICATION>
</cls:CLASSIFICATIONS>
```

Anhang D Multimediale Kommunikation

D-1 Kodierungsverfahren für Audio und Video



ADPCM: Adaptive Differential PCM, CELP: Code-Excited-Linear-Prediction, ACELP: Algebraic-Code-Excited Linear-Prediction, LD-CELP: Low-Delay CELP, LPC: Linear Predictive Coding, MP-MLQ: Multipulse Maximum Likelihood Quantization, PCM: Pulse Code Modulation

Nach [BADACH 04]

Videoformat	Gängige Codecs
HDTV	MPEG-2
Videoproduktion	MPEG-2
High Quality video distribution (DVD)	MPEG-2
Intermediate quality video distribution (www)	MPEG-1, MPEG4, Quicktime, Real G2
Video Konferenz über ISDN/Internet	H.261/H.263
Videotelefonie über drahtloses oder drahtgebundenes Modem	H.263

Nach [STEINBACH 04]

D-2 Interpretation der MOS-Skala

MOS-Wert	Bedeutung
5 = exzellent	Keinerlei Anstrengung zum Verständnis der Sprache notwendig; totale Entspannung möglich
4 = gut	keine Anstrengung notwendig, Aufmerksamkeit nötig
3 = ausreichend	leichte, moderate Anstrengung nötig
2 = notdürftig	merkbare, deutliche Anstrengung nötig
1 = schlecht	trotz Anstrengung keine Verständigung

MOS = Mean Opinion Score

Nach [BADACH 04]

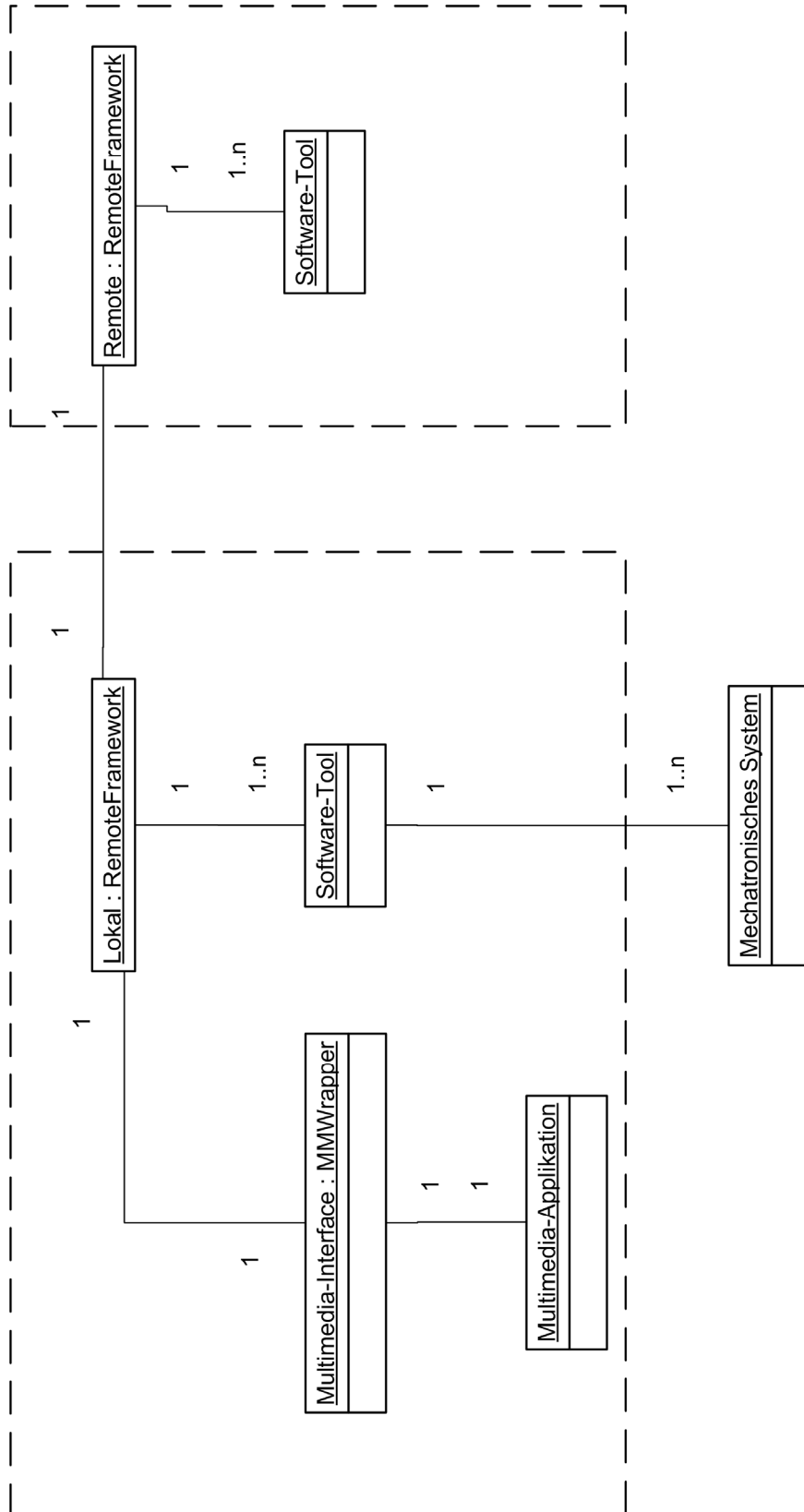
D-3 Performance-Parameter multimedialer Kommunikation

Delay	Laufzeit der Pakete durch das Netzwerk, Meist als Round Trip Time gemessen. Stellt nur einen Teil der gesamten Laufzeit dar, hinzu kommt das Delay in den Endgeräten. Sollte 250 ms nicht überschreiten.
Delay Jitter	Beschreibt die Laufzeitschwankungen der Pakete durch das Netzwerk. Entsteht durch das Vermittlungsprinzip „Store and Forward“ in einem paketvermittelten Netz. Sollte 30 ms nicht überschreiten.
Packet Loss	Prozentsatz der verlorenen Pakete. Wichtig ist dabei auch die Aussage, in welcher zeitlichen Verteilung der Paketverlust innerhalb der Verbindung aufgetreten ist.
Consecutive Packet Loss	Beschreibt den Verlust einer Sequenz von Paketen. Aussagekräftig ist besonders die Häufigkeitsverteilung des Consecutive Packet Loss, d.h. die Häufigkeit des Verlustes von verschiedenen langen Sequenzen.

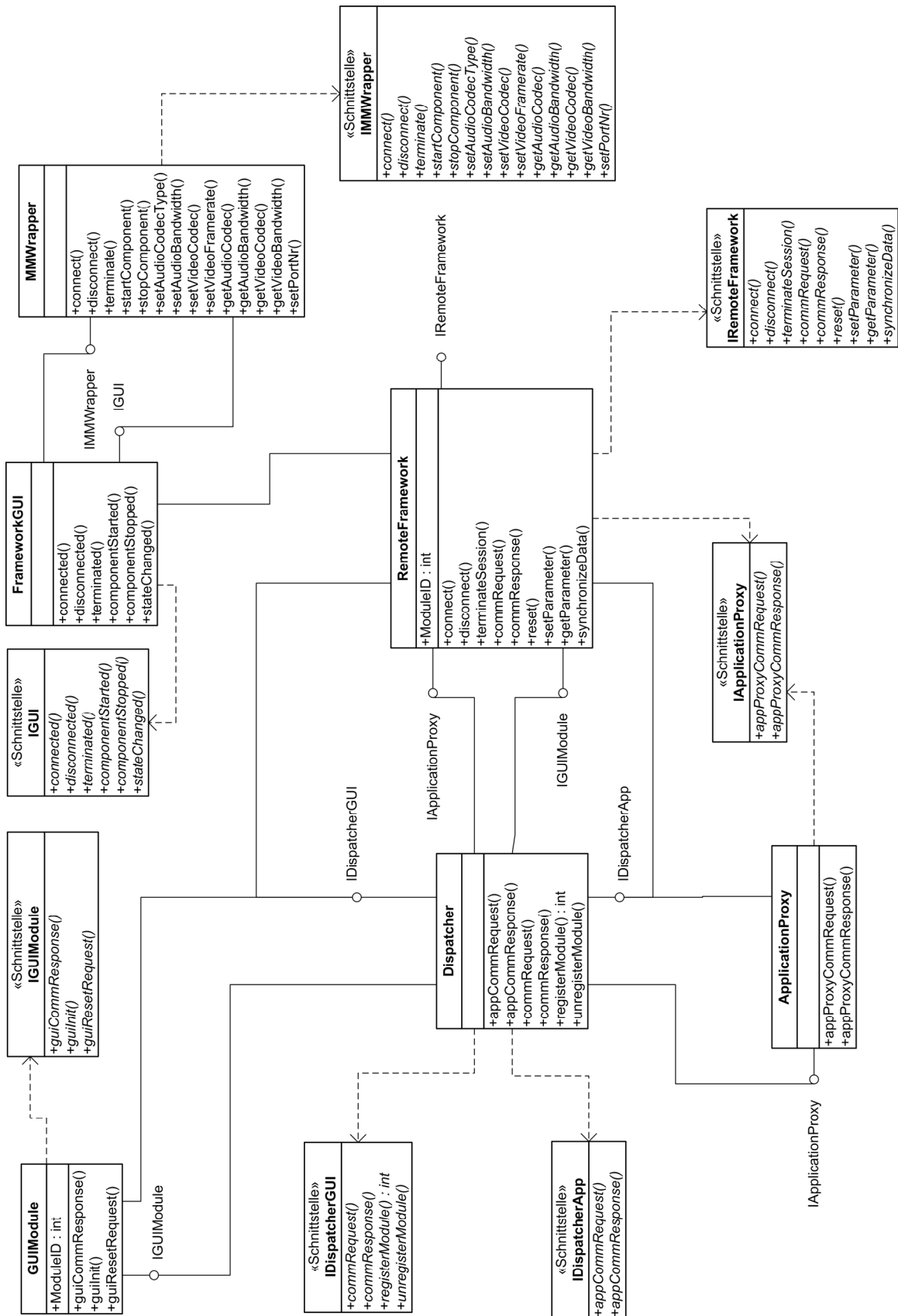
Nach [STOOSS 04]

Anhang E Spezifikation des Remote-Frameworks

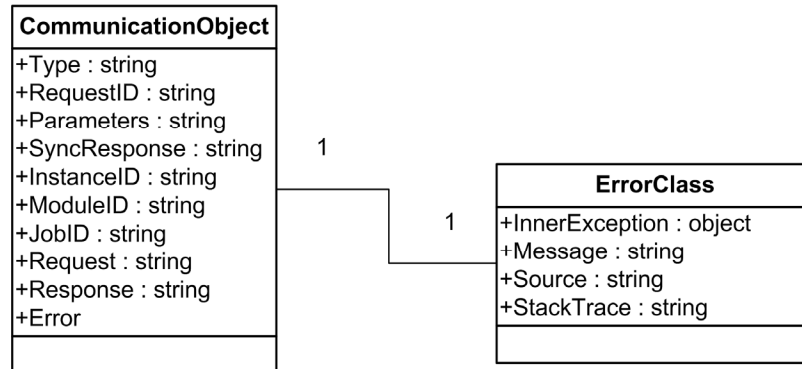
E-1 Übersicht: Objektdiagramm



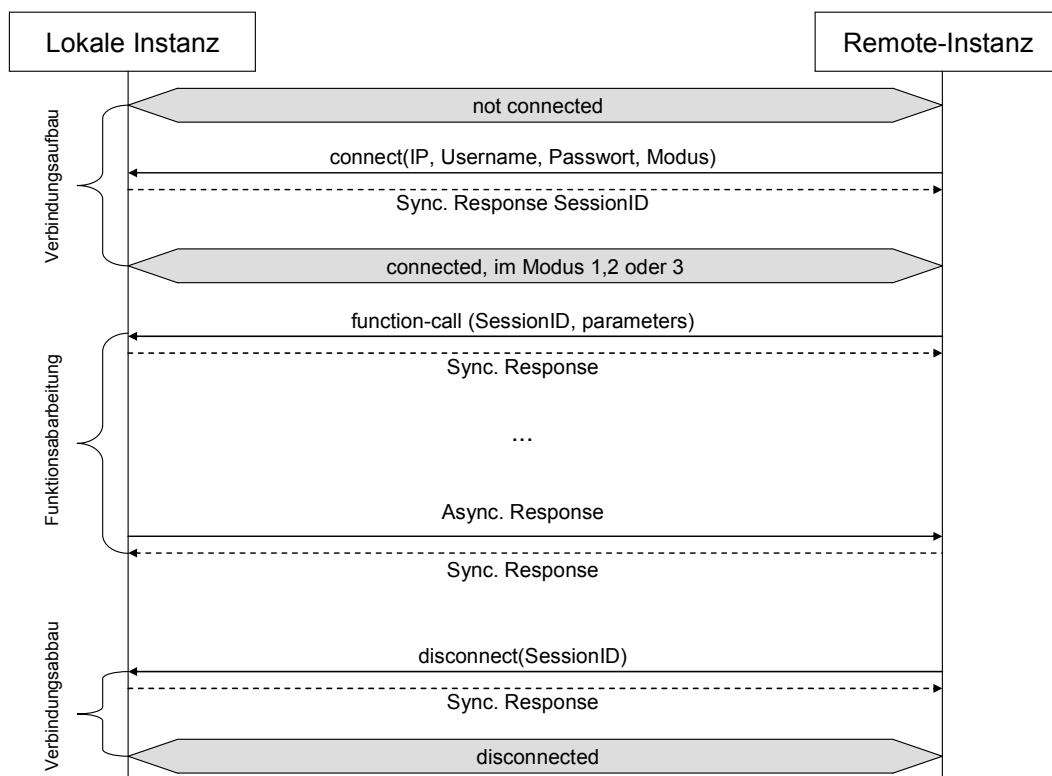
E-2 Klassendiagramm des Gesamtsystems



E-3 Aufrufobjekt

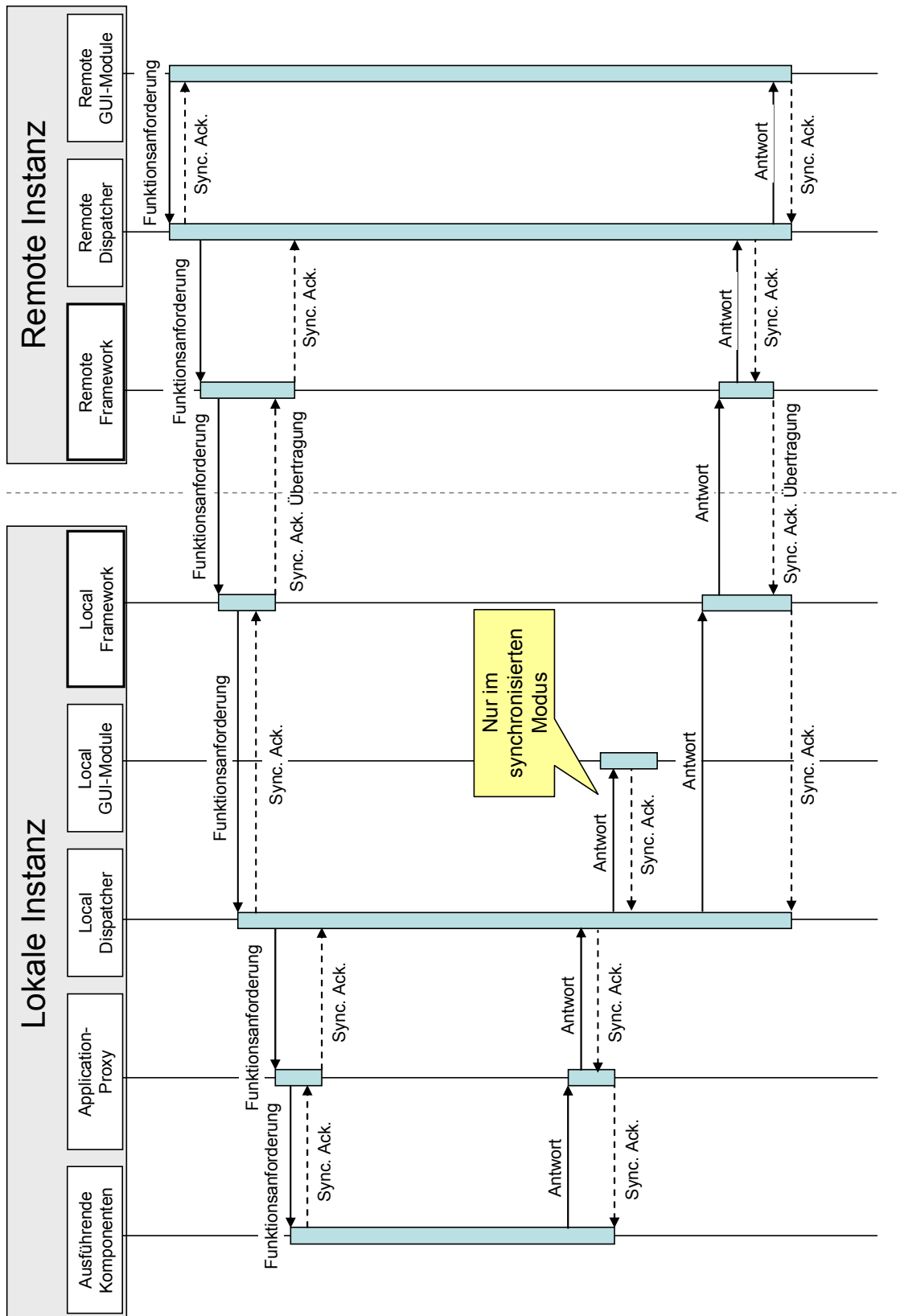


E-5 SFC für Verbindungsauf- und Abbau

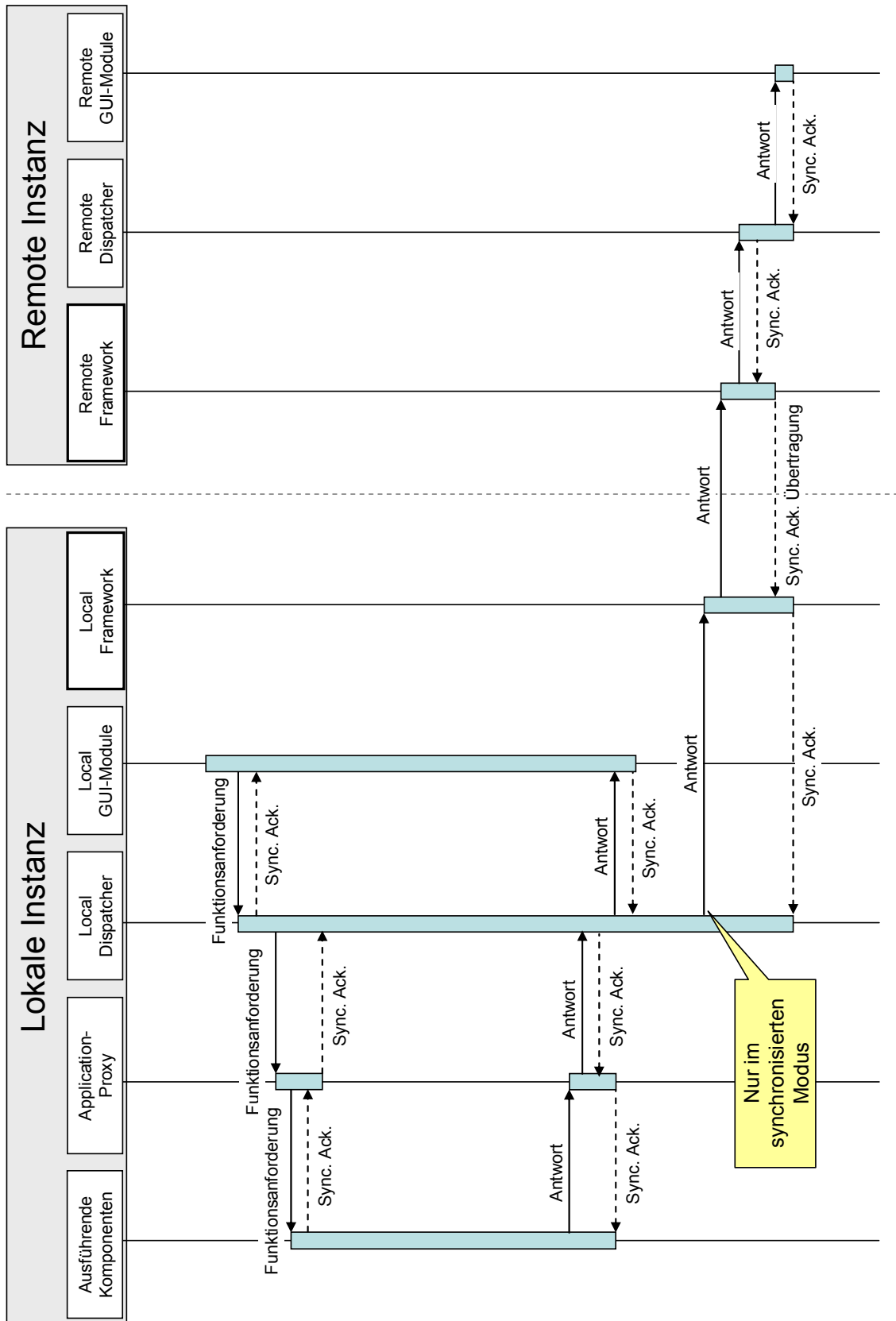


Am Beispiel der Remote-Instanz als Verbindungsinitiator

E-4 Nachrichtenfluss zwischen den Komponenten

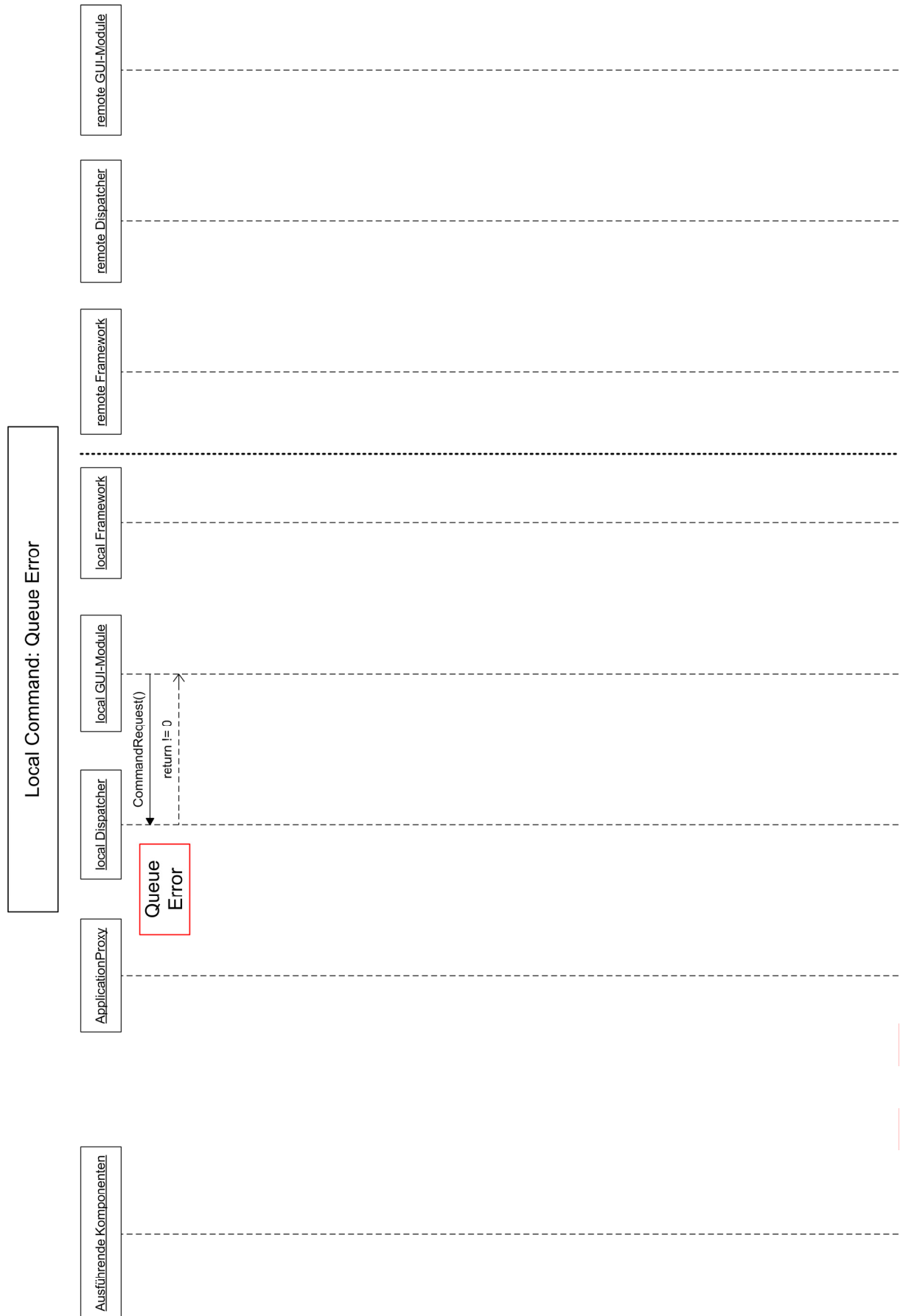


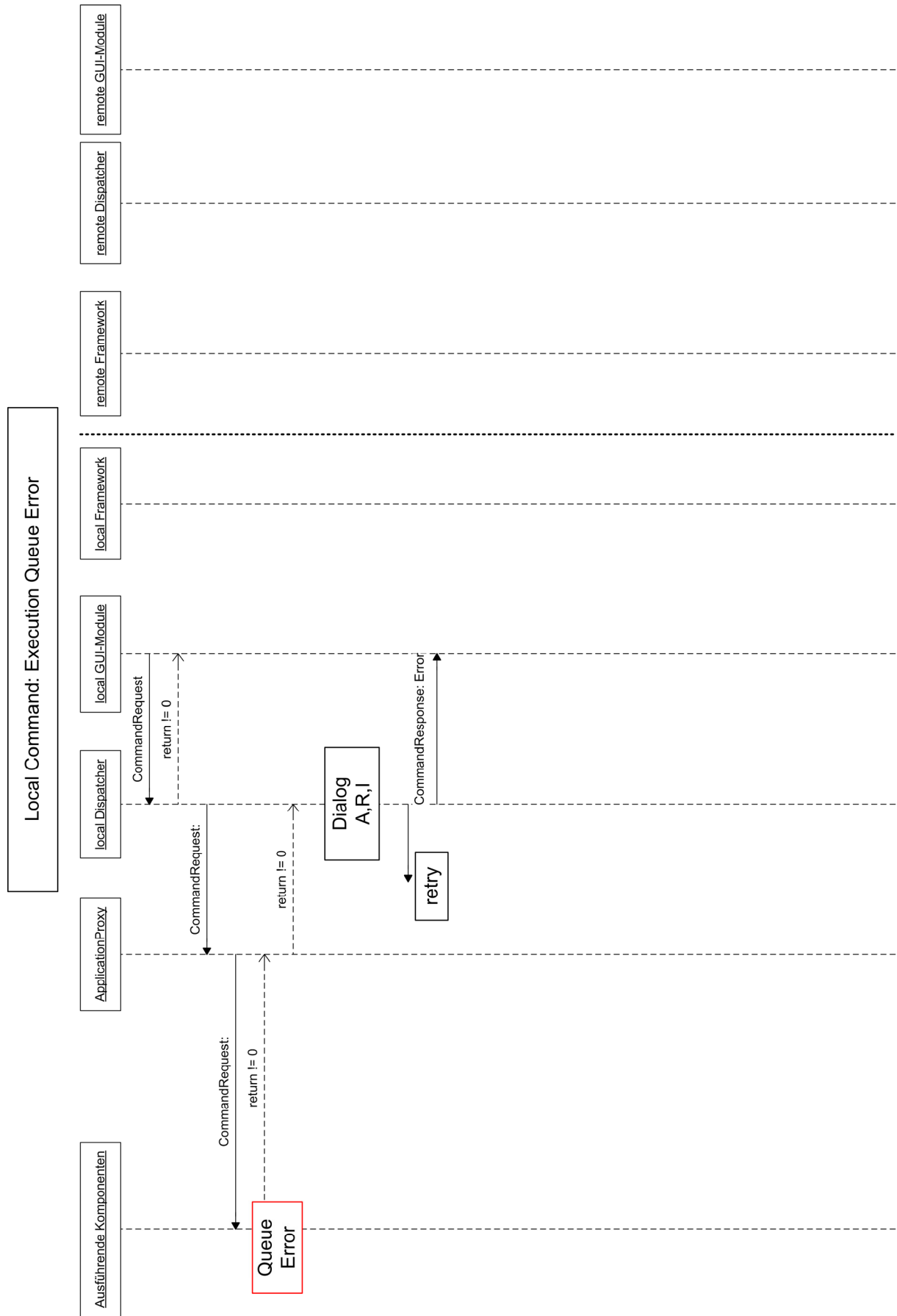
Funktionsaufruf von der Remote-Seite

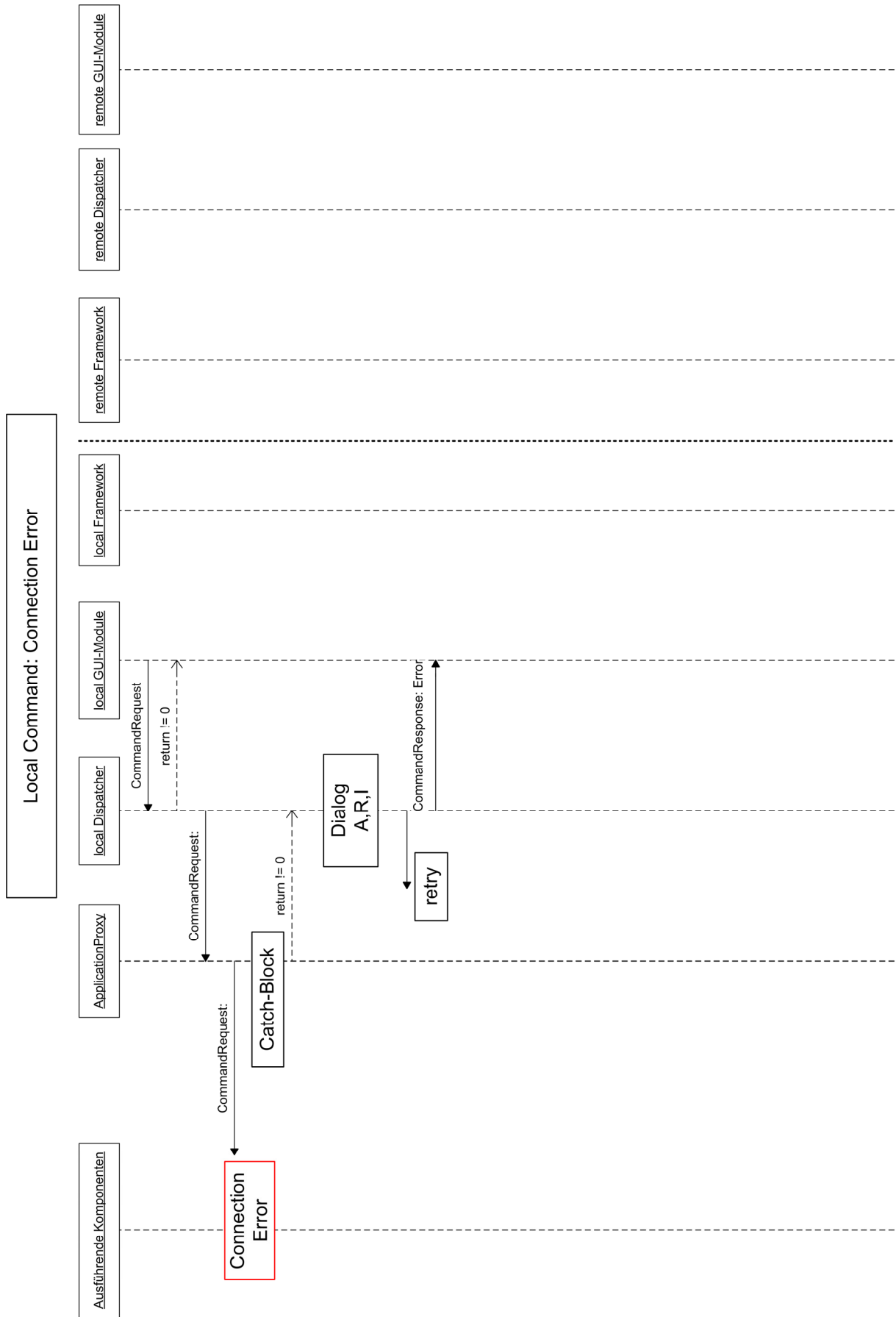


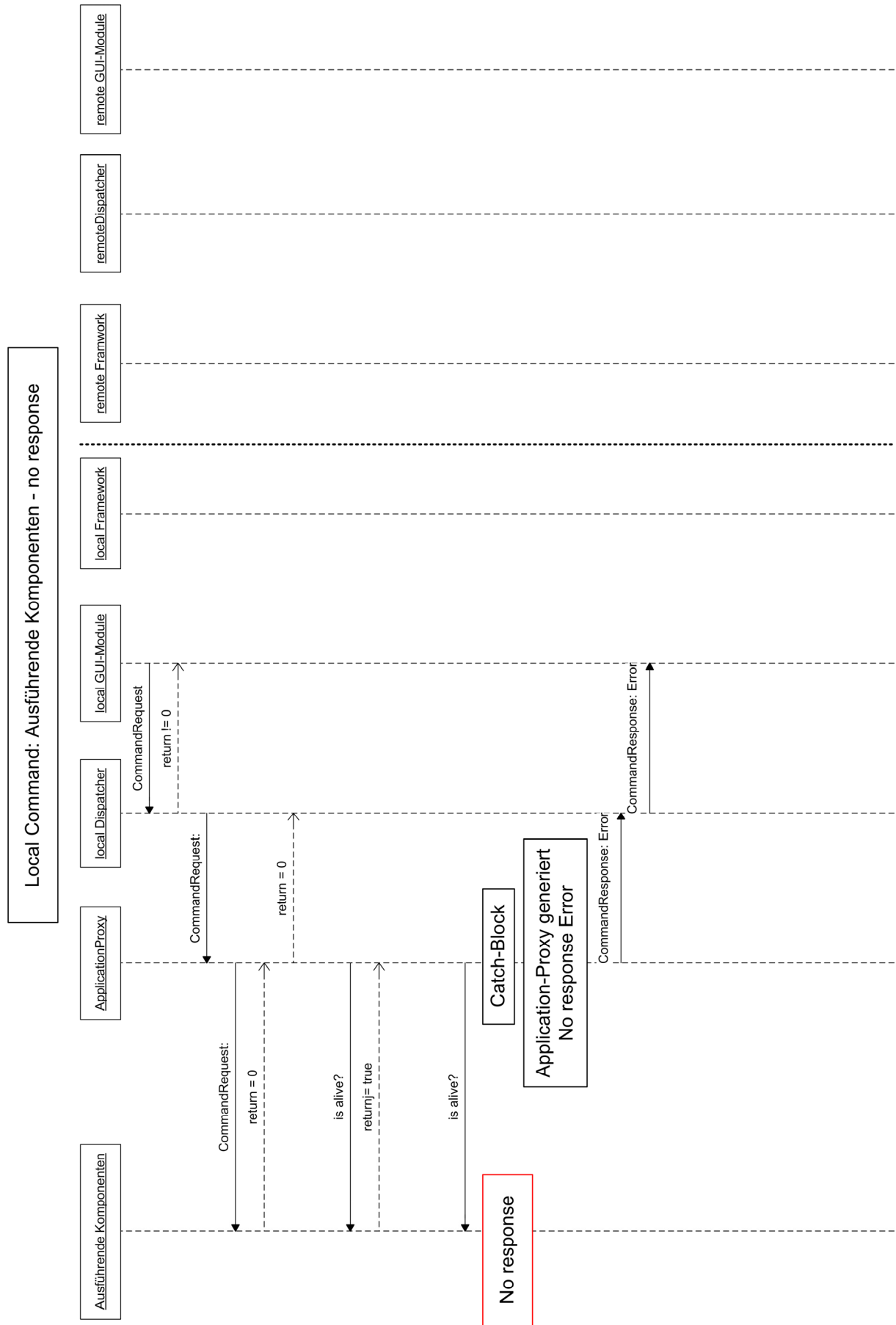
Funktionsaufruf auf lokaler Seite

E-7 SFCs der Fehlerbehandlung

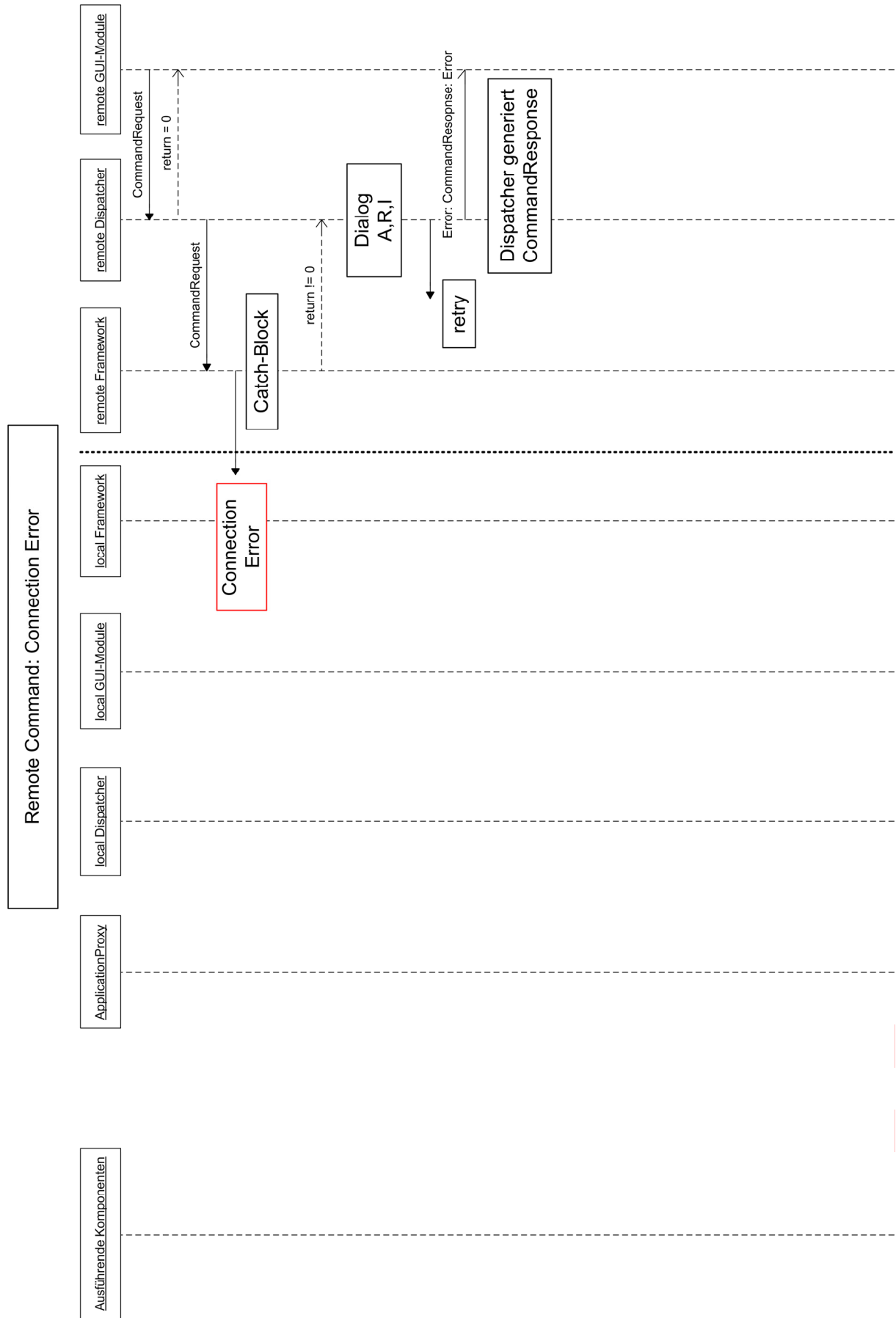


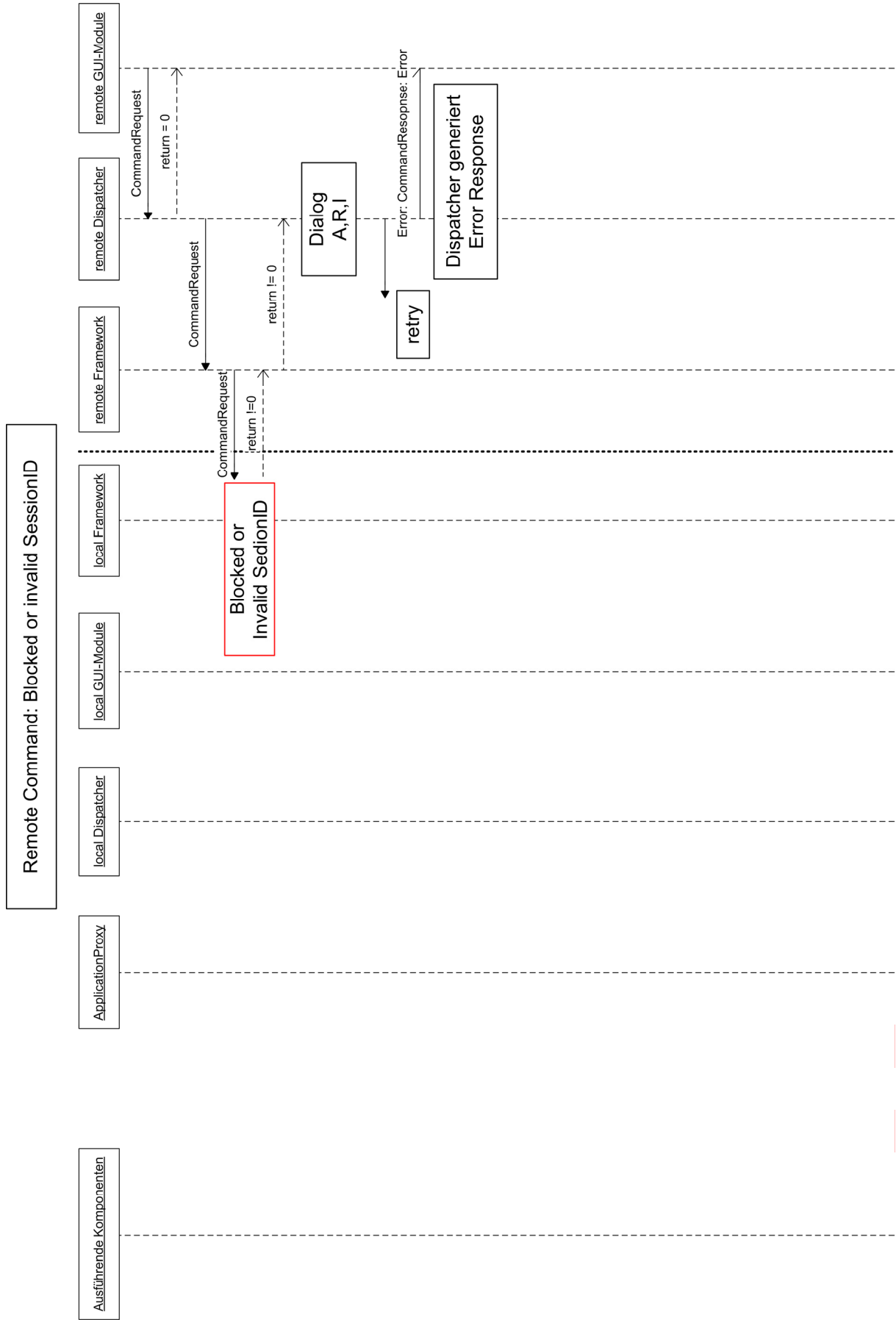


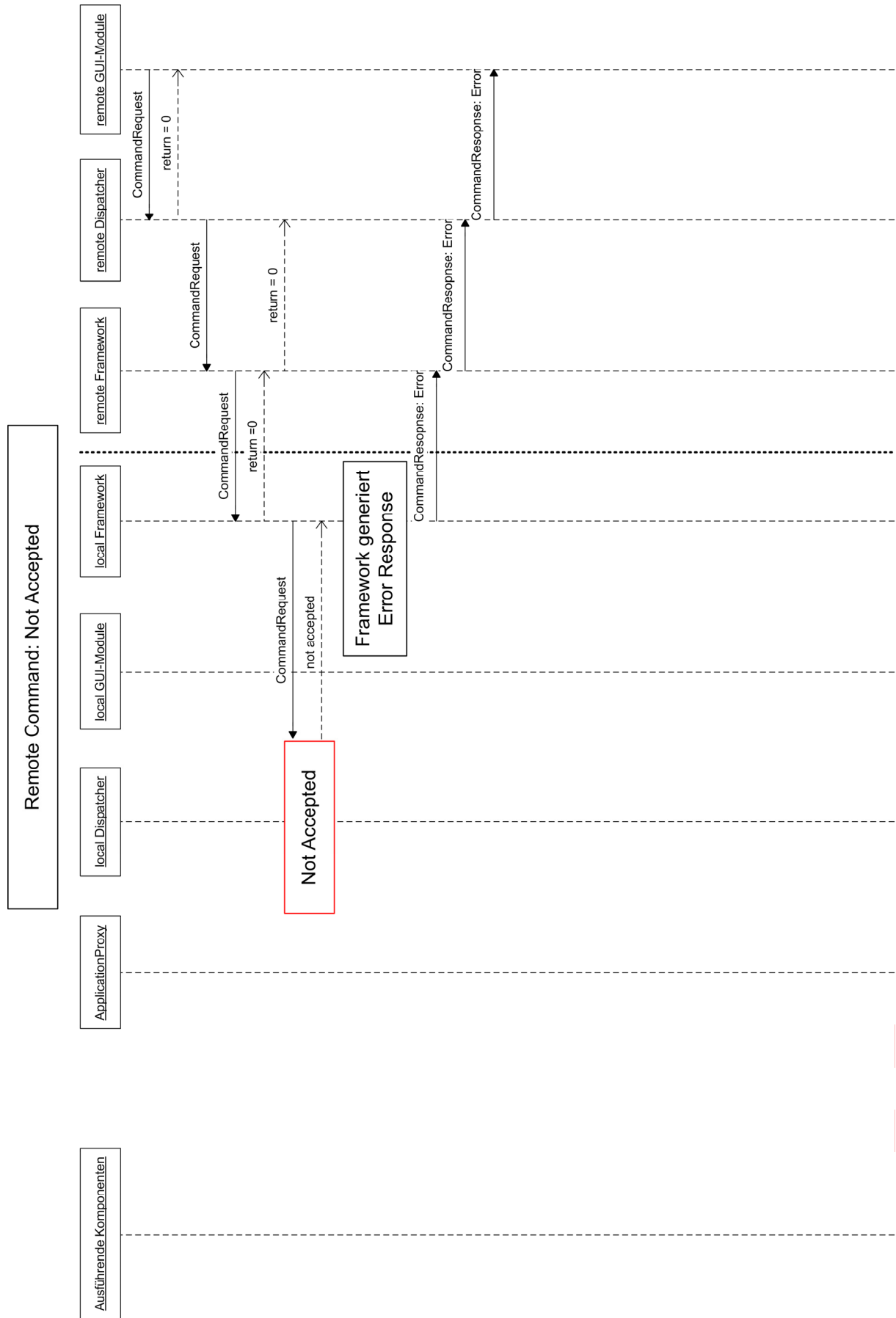


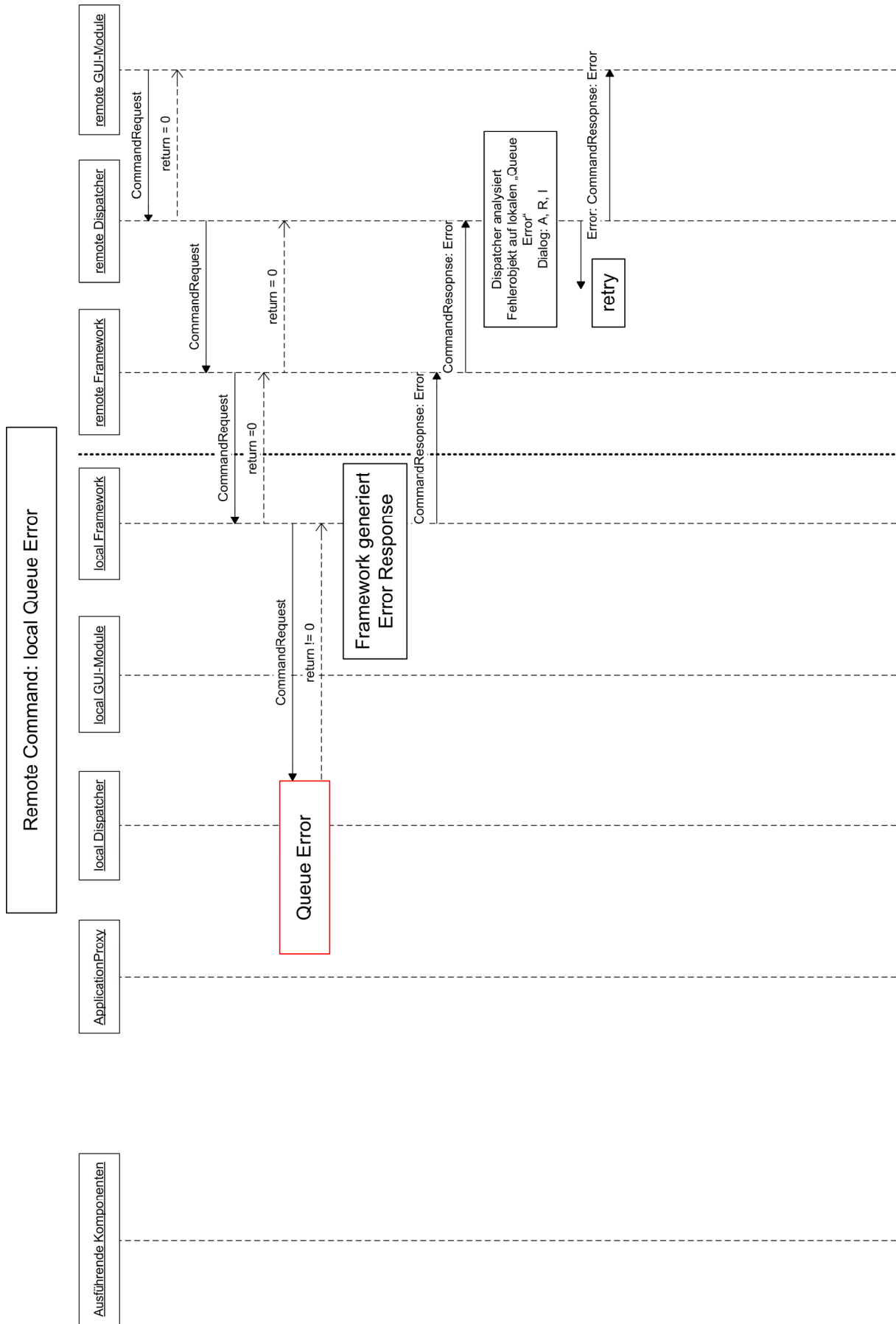


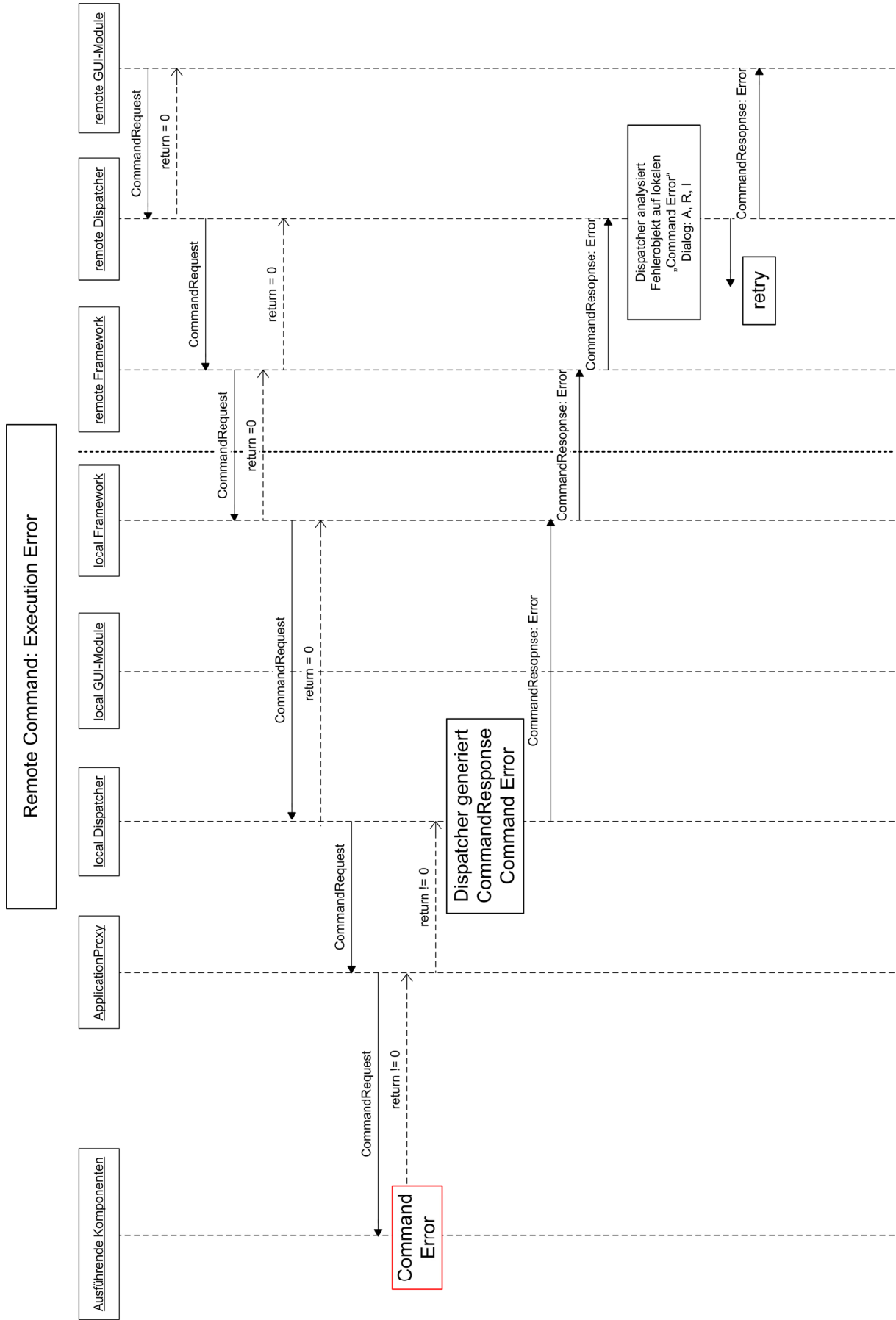


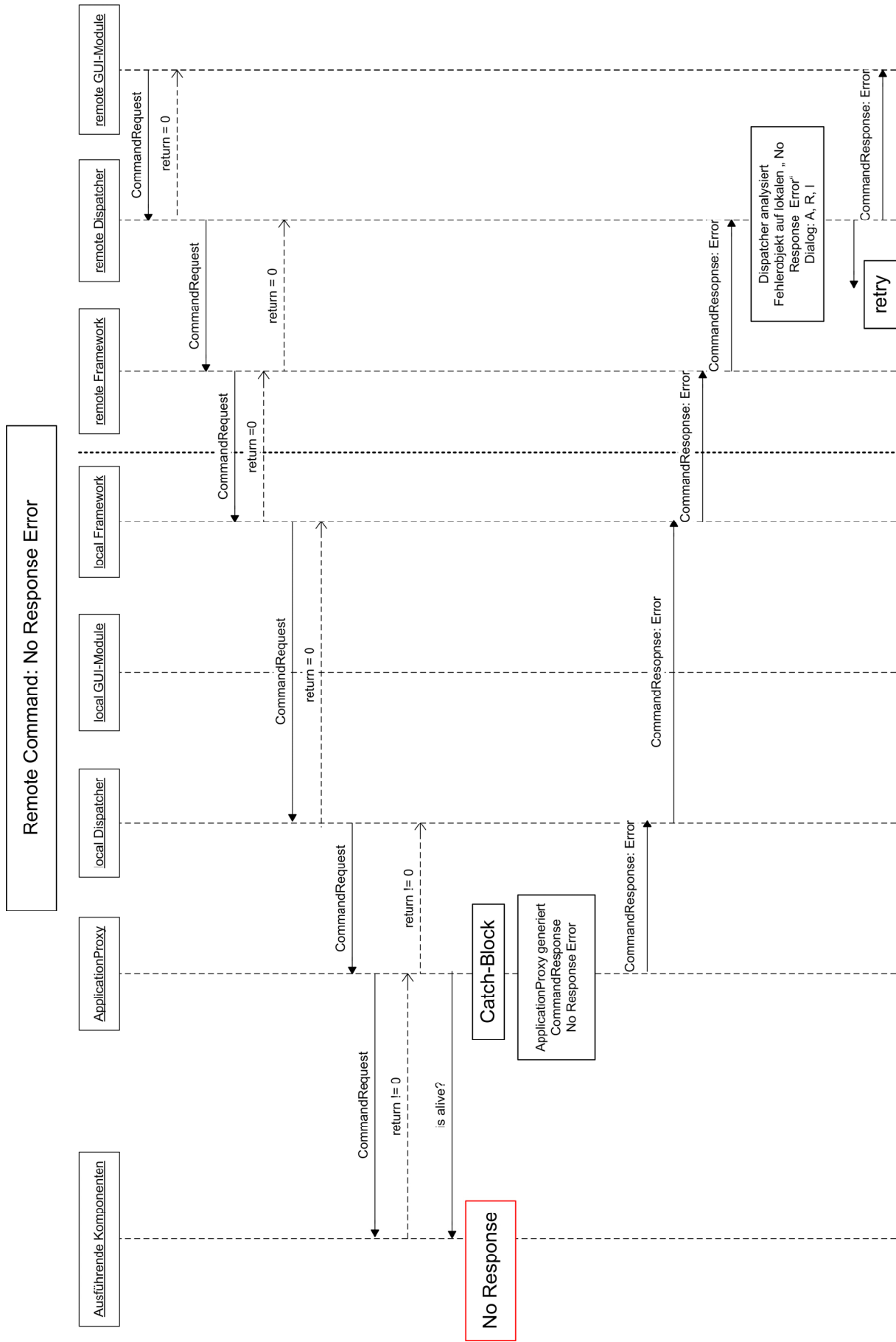


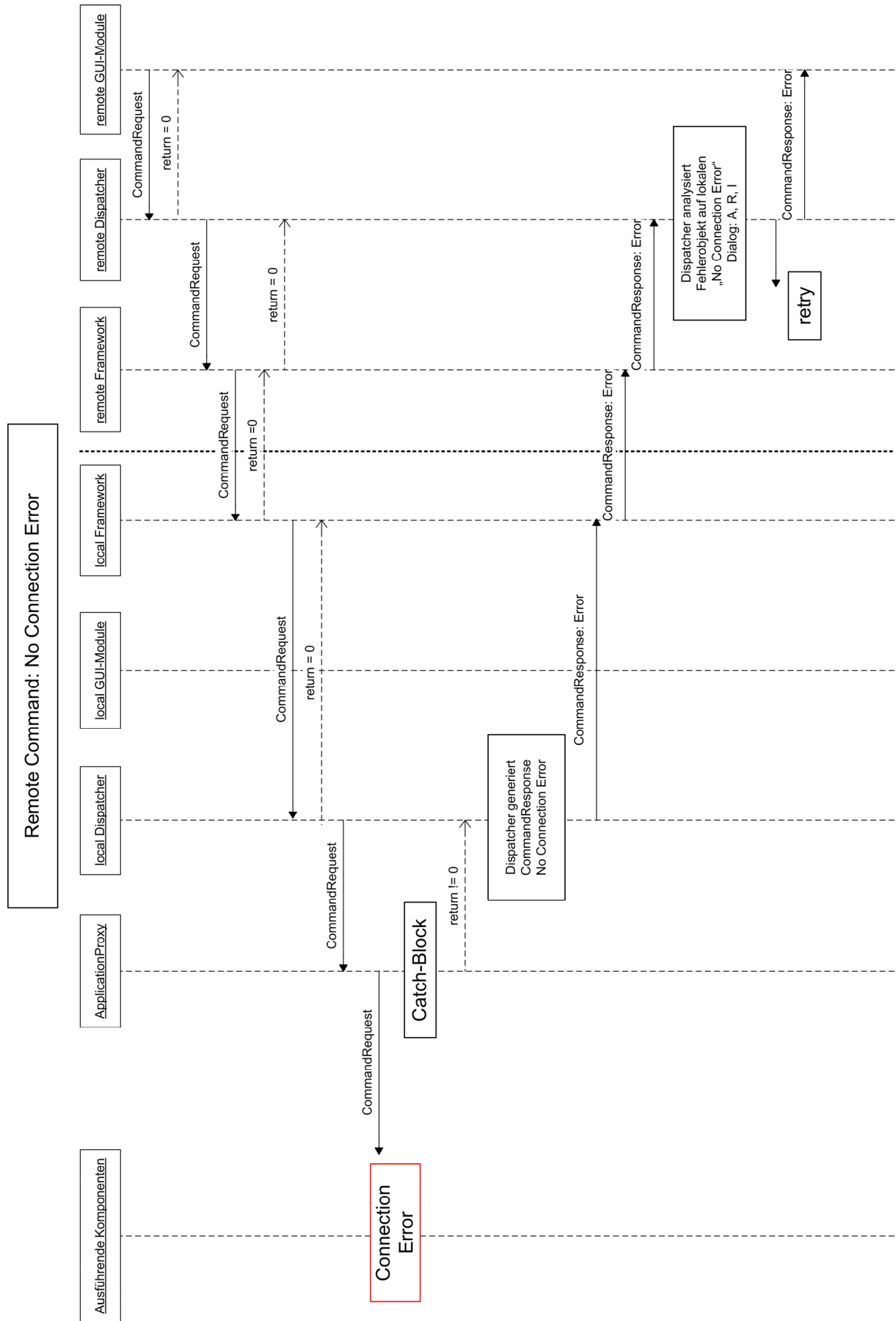


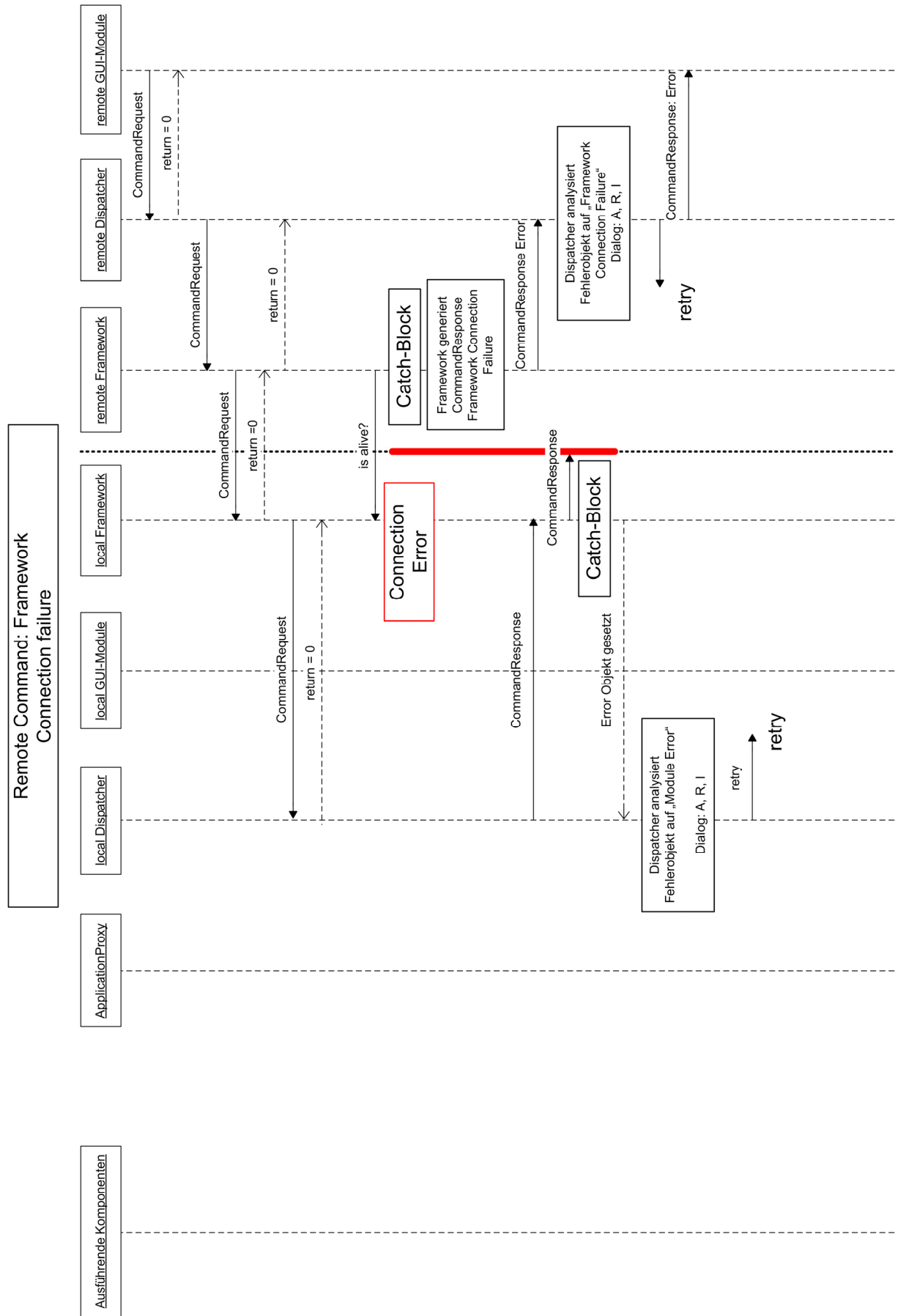












Anhang F Normen und Richtlinien IT Sicherheit

Grundschutzhandbuch des BSI

Das Grundschutzhandbuch wird vom BSI in digitaler (kostenfrei) und in gedruckter Form zur Verfügung gestellt und ist in auf der Homepage des BSI (www.bsi.de) erhältlich. Das Handbuch wurde erstmals im Jahre 1998 veröffentlicht und wird seither jedes Jahr fortgeschrieben, wobei die Anforderungen von Anwendern mit einfließen.

Das Grundschutzhandbuch adressiert Unternehmen (und staatliche Einrichtungen) mit mittlerem Schutzbedarf, wobei das Werk auch als Basis für hochschutzbedürftige Systeme dienen kann.

Das IT-Grundschutzhandbuch beschreibt detailliert die Standard-Sicherheitsmaßnahmen, die praktisch für jedes IT-System zu beachten sind. Es umfasst:

- Standardsicherheitsmaßnahmen für typische IT-Systeme mit "normalem" Schutzbedarf
- Eine Darstellung der pauschal angenommenen Gefährdungslage
- Ausführliche Maßnahmenbeschreibungen als Umsetzungshilfe
- Eine Beschreibung des Prozesses zum Erreichen und Aufrechterhalten eines angemessenen IT-Sicherheitsniveaus
- Eine einfache Verfahrensweise zur Ermittlung des erreichten IT-Sicherheitsniveaus in Form eines Soll-Ist-Vergleichs.

Die Maßnahmenbeschreibungen im Handbuch sind dabei so detailliert gestaltet, dass sie als konkrete Umsetzungshinweise dienen können. Da die Informationstechnik sehr innovativ ist und sich ständig weiterentwickelt, ist das vorliegende Handbuch auf Aktualisierbarkeit und Erweiterbarkeit angelegt. Das Bundesamt für Sicherheit in der Informationstechnik aktualisiert auf der Grundlage von Anwenderbefragungen das Handbuch ständig.

Neben dem Grundschutzhandbuch stellt das BSI eine ganze Reihe weiterer Informationen und Tools zur Verfügung, die das Thema IT-Sicherheit betreffen. Da der IT-Grundschutz als Basis für die Sicherheit bei Remote-Zugriff auf Automatisierungssysteme angesehen werden kann, werden die wichtigsten Themen kurz vorgestellt. Nähere Informationen sind über die Homepage des BSI erhältlich.

BSI Grundschutztool

Mit der Entwicklung des BSI Tools IT-Grundschutz (GSTOOL) stellt das BSI eine innovative und ergonomisch handhabbare Software bereit, die den Anwender bei Erstellung, Verwaltung und Fortschreibung von IT-Sicherheitskonzepten entsprechend dem IT-Grundschutz effizient unterstützt. Nach Erfassung benötigter Informationen steht dem Anwender ein umfangreiches Berichtssystem zur Verfügung, mittels dessen er strukturierte Auswertungen über alle relevanten Daten durchführen und diese auch auf Papier oder elektronisch ausgeben kann.

Leitfaden IT-Sicherheit

Analog zur Entwicklung in der Informationstechnik wurde das IT-Grundschutzhandbuch komplexer und umfassender. Besonders kleine und mittlere Institutionen mit beschränkten finanziellen und personellen Möglichkeiten wünschen sich deshalb einen leicht überschaubaren Einstieg in die Thematik. Der Leitfaden IT-Sicherheit gibt einen kompakten und allgemeinverständlichen Überblick über die wichtigsten IT-Sicherheitsmaßnahmen. Im Mittelpunkt stehen organisatorische Maßnahmen und die Veranschaulichung von Gefahren durch Praxisbeispiele. Auf technische Details wird bewusst verzichtet.

Risikoanalyse auf der Basis von IT-Grundschutz

Das IT-Grundschutzhandbuch des BSI enthält Standard-Sicherheitsmaßnahmen aus den Bereichen Organisation, Personal, Infrastruktur und Technik, die bei normalen Sicherheitsanforderungen in der Regel angemessen und ausreichend zur Absicherung von typischen IT-Landschaften sind. Viele Anwender, die bereits erfolgreich mit dem IT-Grundschutzhandbuch arbeiten, stehen vor der Frage, wie sie mit Bereichen umgehen sollen, deren Sicherheitsanforderungen deutlich über das normale Maß hinausgehen. Wichtig ist dabei, dass die zugrundeliegende Methodik möglichst wenig zusätzlichen Aufwand mit sich bringt und möglichst viele Ergebnisse aus der IT-Grundschutz-Vorgehensweise wieder verwendet. Vor diesem Hintergrund hat das BSI eine Methodik zur Risikoanalyse auf der Basis des IT-Grundschutzhandbuchs erarbeitet. Diese Vorgehensweise bietet sich an, wenn Unternehmen oder Behörden bereits erfolgreich mit dem IT-Grundschutzhandbuch arbeiten und möglichst nahtlos eine ergänzende Sicherheitsanalyse an die IT-Grundschutz-Analyse anschließen möchten. Hierfür kann es verschiedene Gründe geben:

- Die Sicherheitsanforderungen des Unternehmens bzw. der Behörde gehen teilweise deutlich über das normale Maß hinaus (hoher oder sehr hoher Schutzbedarf).
- Die Institution betreibt wichtige Anwendungen oder Komponenten, die (noch) nicht im IT-Grundschutzhandbuch des BSI behandelt werden. Hierzu sind auch automatisierungstechnische Anlagen zu zählen.

Die Vorgehensweise richtet sich sowohl an Anwender der Informationstechnik (IT-Sicherheitsverantwortliche und -beauftragte) als auch an Berater und Exper-

ten. Häufig ist es allerdings empfehlenswert, bei der Durchführung von Risikoanalysen auf Expertensachverstand zurück zu greifen.

SINA-Projekt

Die Sichere Inter-Netzwerk Architektur (SINA) ist eine vom BSI entwickelte IT-Architektur zum Austausch von hoch schützenswerten Informationen in unsicheren Netzen. Durch die Kombination von Thin-Client/Server Verarbeitung und Virtual Private Network (VPN)-Technologie sowie den weitgehenden Einsatz von Open Source Software, können mit SINA flexible, hochsichere Systemlösungen realisiert werden. SINA umfasst eine wachsende Familie von modularen Komponenten zur Absicherung verschiedenster Anwendungsszenarien, deren Funktionalität stetig erweitert wird.

Risikoanalysen, Risikobestimmung

Handelt es sich bei dem zu schützenden System um ein hochschutzbedürftiges System so ist nach [BÖHMER 02] die Durchführung einer Risikoanalyse angezeigt. Bei der traditionellen Risikoanalyse werden zunächst die Gefährdungen ermittelt und mit Eintrittswahrscheinlichkeiten bewertet, um dann die geeigneten IT-Sicherheitsmaßnahmen auszuwählen und anschließend noch das verbleibende Restrisiko bewerten zu können. Ziel ist es unter anderem, aus den Ergebnissen der Analyse Aussagen über das Kosten-Wirksamkeitsverhältnis der Sicherheitsmaßnahmen abzuleiten.

Als Basis dient die DIN, VDE Norm 3000 als allgemeinen und anerkannten Grundsatz zur Risikobestimmung. Weiterführend sei hier auf entsprechende Literatur verwiesen. Das BSI bietet ebenfalls eine Vorgehensweise zur vereinfachten Risikoanalyse auf Basis des IT-Grundschutzes an.

ISO 7498-2

In der ISO Norm 7498-1 erfolgt die Definition des ISO/OSI-Referenzmodells. Teil zwei der Norm definiert eine Sicherheitsarchitektur durch die Beschreibung von Sicherheitsdiensten zum Schutz des Datenaustausches und die Zuordnung der definierten Dienste zu den Schichten des OSI-Modells.

Common Criteria (CC)

Die *Common Criteria for Information Security Evaluation* (CC) sind Kriterien zur Prüfung und Bewertung der Sicherheit von Informationstechnik. Anhand solcher gemeinsamen Kriterien werden vorhandene Sicherheitsmaßnahmen einer Prüfung und Bewertung unterzogen. Daraus lässt sich eine vergleichbare Aussage über die Vertrauenswürdigkeit von Schutzmaßnahmen für die zu schützenden Werte gewinnen. Die *Common Criteria* sind eine Weiterentwick-

lung und Harmonisierung von Kriterien auf europäischer Ebene (ITSEC), amerikanischen Kriterien (Orange Book, auch TCSEC) und kanadischen Kriterien (CT-CPEC).

Die CC gliedern sich in drei Teile:

Teil 1: Einführung und allgemeines Modell:

Grundlagen der IT-Sicherheitsevaluierung, Geltungsbereich der CC, Schutzprofile, Sicherheitsvorgaben

Teil 2: Funktionale Sicherheitsanforderungen

Katalog von Funktionalitätsanforderungen, Aufzeigen von Zusammenhängen zwischen Bedrohung, Sicherheitszielen und funktionalen Anforderungen.

Teil 3: Anforderungen an die Vertrauenswürdigkeit

Definition von Vertrauenswürdigkeitsstufen

CERT Guide

Das amerikanische CERT (Computer Emergency Response Team) und das CERT Coordination Center (CERT CC) beschäftigen sich gemeinsam mit der Abwehr von Angriffen auf Computer-Netzwerke (z.B. durch Hacker, Viren, etc). Das CERT (www.cert.org) ist dabei eine weltweit anerkannte Organisation wenn es um das Thema Computersicherheit geht. Auf der Homepage des CERT finden sich stets aktuelle Informationen und Artikel zum Thema Sicherheit bzw. Informationen zu aktuellen Sicherheitsproblemen (z.B. neue Viren). Angesprochen wird hier vor allem die technische Umsetzung von Computer- und Netzwerksicherheit in der Praxis. Seit dem Jahr 2001 hat auch das BSI eine CERT-Anlaufstelle für Bundesbehörden eingerichtet (www.bsi.de).

Security-Bulletins

Analog zu den Web-Seiten des CERT existieren eine Vielzahl weiterer Security-Bulletins und Foren im Internet. Zu nennen wäre beispielsweise die Security-Web-Seite des Heise-Verlags (www.heise.de). Behandelt werden hier vor allem aktuelle Themen und Vorfälle rund um das Thema Sicherheit wie beispielsweise aktuelle Sicherheitslöcher, neue Viren etc. Diese Art der Informationsquellen dienen weniger als Unterstützung für konzeptionelle Tätigkeiten als der ständigen Wartung und Administration eines Systems im Hinblick auf IT-Sicherheit.

Nach [DFAM 04]