

elektrische Leitungswiderstände oder mechanische Reibungseffekte im Modell zu berücksichtigen sind, hängt immer vom Zweck des Modells ab. Deshalb wird es in der Regel im Rahmen des Entwurfes für ein und dasselbe mechatronische System mathematische Modelle mit *unterschiedlicher Modellgüte* bzw. Detaillierungsgrad (engl. *fidelity*) geben.

Low-Fidelity-Modelle Für den Reglerentwurf, stochastische Verhaltensaussagen oder einen Grobentwurf (Machbarkeitsstudien) werden üblicherweise *vereinfachte Modelle* (engl. *low fidelity*) genutzt. Diese Art von Modellen sind in der Regel lineare zeitinvariante Modelle (LTI-Systeme) mit niedriger Systemordnung. Im Falle von Mehrkörpersystemen wird dabei nur eine Auswahl von wenigen mechanischen Strukturfrequenzen (Eigenmoden) berücksichtigt. Die simulationstechnische Umsetzung zur Gewinnung von Zeitantworten bietet dabei mittels Einsatz von Standardalgorithmen kaum Probleme und soll in diesem Buch deshalb nicht weiter verfolgt werden.

High-Fidelity-Modelle Für die Entwurfsverifikation bzw. -validation (z.B. Empfindlichkeitsanalyse mit unterschiedlichen Experimentrahmen, statistisch Analyse mittels Monte-Carlo-Simulation) sind dagegen möglichst *detailgetreue Modelle* (engl. *high fidelity*) mit kleinstmöglichen Modellfehlern (Abb. 2.3) zu verwenden. Solche Modelle berücksichtigen in der Regel alle relevanten Nichtlinearitäten, breitbandiges dynamisches Systemverhalten und speziell die hochfrequenten mechanischen Strukturfrequenzen. Es entstehen so Modelle mit (sehr) hoher Systemordnung und beträchtlicher Komplexität. Weitere Fragen der Modellhandhabung, z.B. modulare Simulationsmodelle, spielen dabei eine Schlüsselrolle. Diese *High-Fidelity-Modelle* stellen hohe bis höchste Herausforderungen an die Simulationstechnik und können vielfach nur durch spezielle Vorkehrungen *korrekt* simuliert werden. Deshalb widmet sich das Kap. 3 diesen speziellen Aspekten und Lösungsansätzen zur Simulationstechnik von mechatronischen Systemen.

2.2 Systemmodellierung mit Strukturierter Analyse

Zielstellung In diesem Kapitel werden einige grundlegende Elemente der *Systemanalyse und -modellierung* eingeführt. Diese formalen Hilfsmittel erlauben es, ein System so zu *strukturieren*, dass es überschaubar wird und

dass seine *inneren Zusammenhänge* von einem Ingenieur durchschaubar werden. Das Sichtbarmachen von Strukturen ist eine Aufgabe, die im Allgemeinen ohne mathematische Formeln zu lösen ist, aber trotzdem nicht unterschätzt werden darf.

Qualitative Systemmodelle Aufgrund der Komplexität der betrachteten Systeme ist das Auffinden eines Systemmodells nämlich eine nichttriviale Aufgabe mit uneindeutigem Ausgang (es gibt unendlich viele Lösungen, je nachdem, welche Systemeigenschaften betrachtet werden). Die so entstehenden *qualitativen Systemmodelle* erlauben zwar keine numerischen Berechnungen, sie schaffen aber neben dem Erkennen von grundlegenden Zusammenhängen (Wirkungskreisläufe, Dynamik) überhaupt erst die Grundlage, um quantitative (mathematische) Modelle von klar abgegrenzten und überschaubaren Subsystemen zu erstellen. Auf der Basis dieser strukturierten Systemmodelle können gerätetechnische Entwurfsvarianten entwickelt werden. Erst nach Vorliegen dieser gerätetechnischen Entwurfsvarianten ist man überhaupt in der Lage eine physikalische, sprich quantitative Modellbildung zu betreiben. Erst zu diesem Zeitpunkt weiß man ja, welche gerätetechnischen Kandidaten zur Verfügung stehen, beispielsweise ein elektromechanischer oder ein servohydraulischer Antrieb. Nur in Lehrbüchern finden sich Standardaufgabenstellungen der Form „Gegeben ist – gesucht ist“.

Top-Down-Modellierung Die eben beschriebenen Arbeitsschritte der so genannten *Top-Down-Systemmodellierung* liegen immer am Beginn einer Produktentwicklung. Im Zuge der Anforderungsdefinition werden die Eigenschaften des zu entwickelnden Produktes nach obiger Vorgehensweise immer mehr detailliert. Die heute dazu verwendeten Methoden kommen häufig aus dem Bereich der Softwaretechnik (engl. *software engineering*) und lassen sich in *funktionsorientierte* und *objektorientierte* Methoden unterteilen (Vogel-Heuser 2003).

Funktionsorientierte Modelle Für den Entwurf *mechatronischer Systeme* bieten die *funktionsorientierten* Modellierungsmethoden den natürlichen Zugang, weil dabei in bekannter Weise Wirkflüsse und Eingangs-/Ausgangsbeziehungen im Mittelpunkt stehen. Deshalb werden nach einigen grundlegenden Begriffsdefinitionen im Folgenden einige Modellierungselemente der *Strukturierten Analyse* (engl. *SA – Structured Analysis*) vorgestellt, die wegen ihrer intuitiven Einfachheit in pragmatischer Weise für eine qualitative Modellierung nutzbar sind.



Abb. 2.5. Anschauliche Definition des Begriffes „System“ (Bild adaptiert nach (Yourdon 1989), mit freundlicher Genehmigung von Ed Yourdon)

2.2.1 Begriffsdefinitionen

Definition 2.3. *System* – (Cellier 1991) „A system is characterized by the fact that we can say what belongs to it and what does not, and by the fact that we can specify how it interacts with its environment. System definitions can furthermore be hierarchical. We can take the piece from before, cut out a yet smaller part of it, and we have a new ‘system’.”, s. Abb. 2.5.

Definition 2.4. *System* – (Schneider 1999) „Ein System wird durch das Vorhandensein bestimmter Eigenschaften gekennzeichnet und durch folgende vier *Axiome* charakterisiert:

- **Strukturprinzip** Das System besteht aus einer Menge von Teilen, die untereinander und mit der (System-) Umgebung in wechselseitiger Beziehung stehen. Das System tritt über physikalische Größen, die den energetischen, stofflichen und informationellen Zustand des Systems beschreiben, mit seiner Umgebung in Wechselwirkung.
- **Dekompositionsprinzip** Das System besteht aus einer Menge von Teilen, die ihrerseits wieder in eine Anzahl in wechselseitiger Wirkung stehender Unterteile zerlegt werden können. Im Detail betrachtet, weisen die Unterteile wiederum eine gewisse Komplexität, d.h. allgemeine Systemmerkmale auf.
- **Kausalprinzip** Das System besteht aus einer Menge von Teilen, deren Beziehungen untereinander und deren Veränderungen selbst eindeutig determiniert sind. Im Sinne eines kausalen Wirkungszusammenhanges können spätere Zustände nur von vorangegangenen abhängen. Kausalität wird als die Logik von Abläufen verstanden.

- **Temporalprinzip** Das System besteht aus einer Menge von Teilen, deren Struktur und Zustand mehr oder weniger zeitlichen Veränderungen unterliegt. Temporalität ist die zeitliche Folge von Abläufen und Veränderungen.“

Die beiden Definitionen sind letztlich gleichwertig, wobei die zweite Definition den für technische Systeme wichtigen zeitlichen Aspekt mit ins Spiel bringt. Das Kausalitätsprinzip wird im Rahmen der Modellbildung leider manchmal in einem falschen Kontext zitiert, dazu erfolgt im Abschn. 2.3.8 eine Begriffsklärung.

2.2.2 Ordnungsprinzipien

Komplexe Systeme Eine wesentliche Aufgabe der Systemmodellierung besteht im Ordnen der verschiedenen Komponenten des Systems. Unter komplexen Systemen versteht man solche Systeme, die aus sehr vielen Komponenten bestehen. Erfahrungen zeigen, dass Menschen nur in der Lage sind, eine begrenzte Zahl von Elementen in einem Bild (grafische Symbole) gleichzeitig zu überblicken und den Inhalt (Semantik) zu verstehen. Im Folgenden werden die wichtigen Entwurfsprinzipien *Strukturierung*, *Dekomposition*, *Aggregation* und *Hierarchie* beschrieben, die es ermöglichen, trotz zunehmender Detaillierung der Modelle immer nur eine begrenzte Zahl von Systemkomponenten vor Augen zu haben.

Strukturierung

- Festlegen der Beziehungen zwischen den Gebilden eines Systems nach vorzugebenden Kriterien,
- ein gegebenes System nach bestimmten Kriterien so zerlegen, dass seine Beziehungen erkennbar werden.

Dekomposition

- „Zerlegung in Grundbestandteile“,
- Systeme werden in Subsysteme *dekomponiert*,
- im bestehenden Modell werden mehr Details sichtbar gemacht,
- Verfeinerung einer Struktur.

Aggregation³

- „Zusammenlagerung von Einzelementen“,
- Subsysteme werden zu einem System *aggregiert*.

³ Aggregation ist das Gegenteil der Dekomposition.

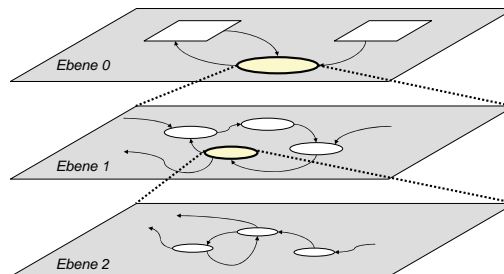


Abb. 2.6. Hierarchieebenen eines Systemmodells: *Dekomposition* von Ebene 0 → Ebene 1, *Aggregation* von Ebene 1 → Ebene 0

Hierarchie

- „(Pyramidenförmige) Rangordnung“,
- Systemhierarchie: Systemdefinitionen sind *hierarchisch*, d.h. man kann einen Teil des Systems herausschneiden und diesen Teil wieder als neues System betrachten. (s. Subsysteme),
- Hierarchieebene (engl. *level*): eine spezielle Betrachtungsebene eines Systems, im Allgemeinen stellt dies ein Subsystem dar,
- obere Hierarchieebene: ein globaler Blick auf das System
- untere Hierarchieebene: ein detaillierter Blick auf das System (Blick in das Innere).

2.2.3 Modellierungselemente der Strukturierten Analyse

Einführende Betrachtungen Die aus der Softwaretechnik bekannte Methode der *Strukturierten Analyse* (engl. *SA – Structured Analysis*) bietet für den Systementwurf mechatronischer Systeme einen sehr natürlichen Zugang zur Systemmodellierung.

Im Folgenden wird eine vereinfachte Variante des YOURDONSchen SA-Ansatzes (Yourdon 1989) vorgestellt. Nach Erfahrungen des Autors reichen die vorgestellten Modellierungselemente völlig aus, um ohne großes methodisches Vorwissen und ohne spezielle Werkzeugunterstützung, überschaubare mechatronische Produktfunktionalitäten innerhalb kurzer Zeit in transparenter Weise zu erzeugen⁴.

⁴ Das vorgestellte pragmatische Vorgehen wird erfolgreich am Lehrstuhl des Autors seit mehr als 10 Jahren in verbindlicher Form im Rahmen aller studentischen Projekte (Studien- und Diplomarbeiten) eingesetzt.




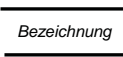
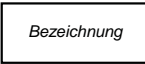
Eine erweiterte Form *Strukturierte Analyse mit Real Time Ergänzung (SA/RT)* (Hatley u. Pirbhai 1987; Hatley u. Pirbhai 1993), (Vogel-Heuser 2003) erlaubt eine strengere formale Spezifikation des zeitlich-kausalen Verhaltens und empfiehlt sich für größere (komplexere) Aufgabenstellungen.

Funktionsorientierte Modellierung Mittels strukturierter Analyse erhält man primär ein *funktionsorientiertes Modell*, weil ausgehend von den Funktionen deren *logisch-kausale* wirkungsmäßige Vernetzung über Daten bzw. Signale betrachtet werden. Es sei ferner vermerkt, dass die strukturierte Analyse zur vollständigen Systembeschreibung auch zusätzlich die *zeitlich-kausalen* Zusammenhänge über *Zustandsautomaten* modelliert.

Modellelemente

Die wichtigsten Modellelemente zur funktionsorientierten Modellierung sind in Tabelle 2.1 dargestellt.

Tabelle 2.1. Modellelemente der Strukturierten Analyse

Symbol	Bezeichnung	Erklärung
	Datenfluss (<i>data flow</i>)	Ein Datenfluss repräsentiert den Transport von (verallgemeinerten) <i>Datenpaketen</i> definierter Länge und Zusammensetzung.
	Steuerfluss (<i>control flow</i>)	Ein Steuerfluss repräsentiert den Transport von (verallgemeinerten) <i>Steuerdaten</i> definierter Länge und Zusammensetzung.
	Prozess, Funktion (<i>process</i>)	Eine Funktion wandelt eingehende Daten nach definierten Regeln in ausgehende Daten um. Bezeichnung: Prädikat + Objekt , z.B. <i>messe Geschwindigkeit</i>
	Speicher (<i>store</i>)	Ein Datenspeicher ist ein Bereich, in dem Datenelemente über eine bestimmte Zeit gespeichert werden.
	Endanwender (<i>terminator</i>)	Ein Endanwender repräsentiert eine Einheit (Funktion, Person, Gerät), die außerhalb des spezifizierten Systems liegt und Daten mit diesem System austauscht.

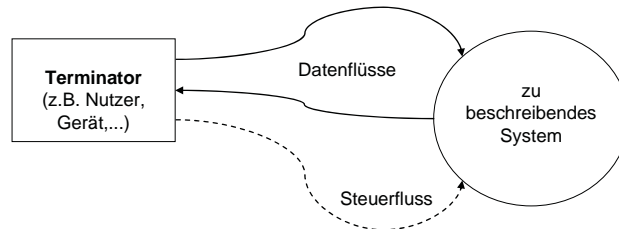


Abb. 2.7. Systemabgrenzung mittels Datenkontextdiagramm

Kontextdiagramm

Datenkontextdiagramm (engl. *DCD – Data Context Diagram*) Kontextdiagramme beschreiben die Außensicht eines Nutzers auf das zu entwickelnde System. Der Zweck des Systems wird in einem einzigen Systemprozess zusammengefasst. Dieser Systemprozess wandelt Eingaben von Terminatoren / Endanwendern in Ausgaben an Terminatoren / Endanwendern um (Abb.2.7). Das Kontextdiagramm beschreibt die Interaktion des Systems mit seiner Umwelt.

Datenflussdiagramm

Datenflussdiagramm (engl. *DFD – Data Flow Diagram*) Datenflussdiagramme sind das primäre Werkzeug zur Bestimmung der funktionalen Eigenschaften eines Systems. Es konkretisiert Eigenschaften des Systems in Form von Komponentenfunktionen (Prozesse), die durch Datenflüsse verbunden sind. Ein DFD enthält Prozesse, Datenflüsse und Datenspeicherstellen, aber keine Terminatoren (Abb. 2.8).

Prozesse Ein Prozess (auch Funktion, Aktivität, Aufgabe, engl. *process, function*) erzeugt aus einer Eingabe eine Ausgabe durch das Anwenden einer Operation. Prozesse haben Namen und Nummern.

Datenflüsse Datenflüsse repräsentieren alle möglichen Arten von verallgemeinerten Informationen (Signale, Wirkflüsse) und können weiter zerlegt werden. Datenflüsse können binär, digital oder auch analog sein.

Schichtung (engl. *leveling*) Zerlegung eines Eltern-DFD in Kind-DFDs mit steigendem Detaillierungsgrad. Die Zerlegungsebenen können unterschiedlich tief sein.

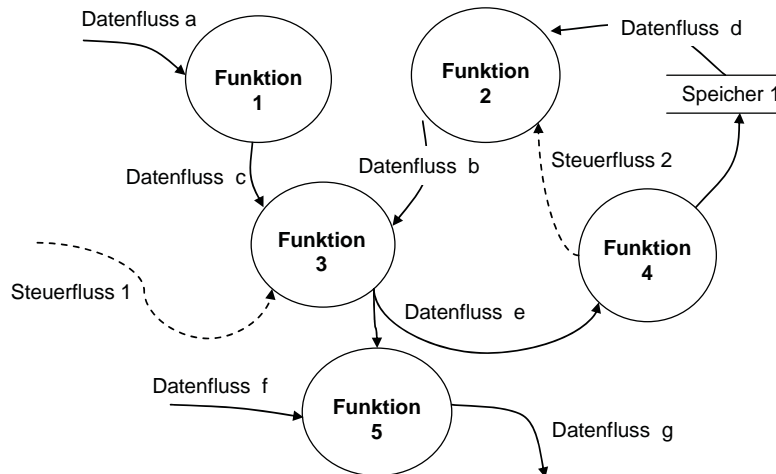


Abb. 2.8. Funktionsdekomposition mittels Datenflussdiagramm

Ausgleich (engl. *balancing*) Test auf inkonsistente Datenflüsse. Ein- und Ausgehende Datenflüsse von Eltern- und Kindprozessen müssen übereinstimmen. Datenflüsse ohne Quelle oder ohne Ziel erzeugen Inkonsistenzen, die automatisch oder manuell getestet werden können. Je Ebene sollten günstiger Weise *fünf bis sieben* (max. zehn) Prozesse (Funktionen) aufscheinen (eine größere Anzahl wird zu unübersichtlich und ist vom Entwerfer nur noch schwer durchschaubar).

Prozessspezifikation (engl. *PSPEC – Process Specification*) Ein Prozess wird solange zerlegt, bis eine kurze und eindeutige Beschreibung möglich ist. Als Beschreibungsmittel ist alles erlaubt, was der Eindeutigkeit des Inhaltes dient, z.B. Tabellen, verbale Beschreibung, Gleichungen, regelungstechnische Übertragungsfunktionen. *PSPECs* können auf allen Verfeinerungsebenen auftreten (Abb.2.9).

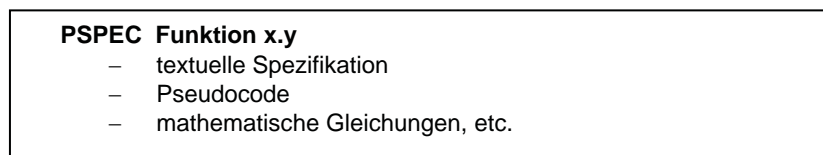


Abb. 2.9. Funktionsdefinition mittels Prozessspezifikation

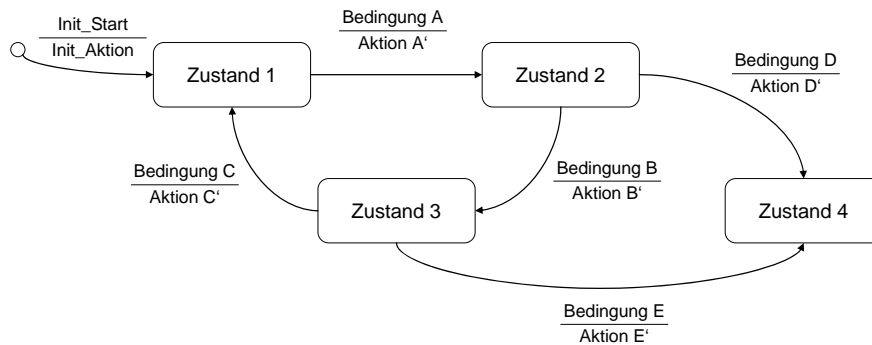


Abb. 2.10. Spezifikation der zeitlich-logischen Ausführung von Funktionen mittels Zustandsdiagramm (hier in Notation eines MEALY-Automaten)

Steuerspezifikation

Zustandsdiagramm (engl. *STD – State Transition Diagram*) Steuerspezifikationen beschreiben die Verarbeitung der Steuerflüsse. Üblicherweise lösen sie Zustandsübergänge aus oder werden mit anderen Steuersignalen verknüpft zu neuen Steuersignalen. Typische Beschreibungsmittel sind Zustandsübergangsdigramme, Entscheidungstabellen, verbale Beschreibungen. Jeder Prozess hat im Allgemeinen *seine* eigene Steuerspezifikation. In Abb. 2.10 ist beispielhaft ein Zustandsdiagramm in Notation eines *MEALY-Automaten* (Litz 2005) gezeichnet. Bei einem aktiven *Zustand 1* wird bei Erfüllen der *Bedingung A* der neue *Zustand 2* aktiviert (und gleichzeitig *Zustand 1* deaktiviert) und die *Aktion A'* ausgeführt.

Datenlexikon

(engl. *DD – Data Dictionary*) Jeder Daten- und Steuerfluss sowie alle Speicherstellen müssen im Datenlexikon definiert werden. Flüsse sind entweder Primitive oder Nicht-Primitive, letztere bestehen aus Primitivengruppen. Ein Datenlexikon wird meist rechnerlesbar abgelegt und kann verbal, tabellarisch oder als Datenbank formuliert sein.

Architekturdiagramm

Implementierungstechnische Struktur Das Architekturdiagramm beschreibt eine implementierungstechnische Struktur, welche die funktionalen Zusammenhänge eines DFD realisiert (gerätetechnische Einheiten und

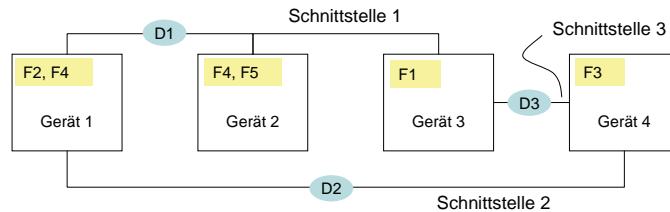


Abb. 2.11. Zuordnung von Funktionen und Datenflüssen zu gerätetechnischen Komponenten mittels Architekturdiagramm

ihre Verknüpfungen – *Gerätearchitektur*, Abb. 2.11). Zusätzlich wird die Verteilung der Funktionen auf Geräteeinheiten und die Verteilung der Datenflüsse auf Geräteschnittstellen eingetragen. Damit erhält man eine *semantische* Gerätebeschreibung, d.h. „welche Aufgabe(n) hat das Gerät zu erfüllen“. Ferner ergibt sich daraus eine transparente Begründung für die Existenz eines bestimmten Gerätes innerhalb einer Gerätearchitektur.

Systemmodell

Das vollständige qualitative *Systemmodell* der *Strukturierten Analyse* ist in Abb. 2.12 dargestellt. Es beschreibt in kompakter und semi-formaler Form unterschiedliche Sichten auf das System (logisch-kausal = funktional bzw. zeitlich-kausal = dynamisch), es besitzt unterschiedliche Hierarchiestufen (Kontext / DFD Ebene 1 / DFD Ebene 2/ ...) und beinhaltet eine oder mehrere mögliche Gerätearchitekturen (Entwurfsvarianten) mit Zuordnung Geräteeinheiten ↔ Funktionen.

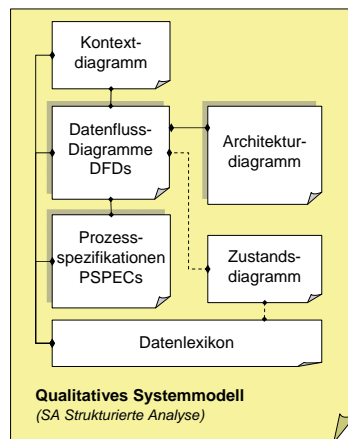


Abb. 2.12. Komplettes Systemmodell mittels Strukturierter Analyse

2.2.4 Produktbeispiel Autofokuskamera

Aufgabenstellung Gesucht ist ein qualitatives Entwurfsmodell für eine einfache *Autofokuskamera*. Dabei sollen insbesondere die mechatronischen Aspekte für eine optimale Bildaufnahme sichtbar gemacht werden. Ferner soll für mögliche gerätetechnische Realisierungen (Entwurfsvarianten) die funktionelle Zuordnung zur vorgeschlagenen Gerätetechnik aufgezeigt werden. Es handelt sich also um typische erste Entwurfsschritte im Rahmen der Entwicklung eines neuen Produktes.

Verbale Produktspezifikation „Die Autofokuskamera soll sehr bedienerfreundlich sein (Zielgruppe fotografischer Laie) und möglichst wenige Bedien- und Anzeigefunktionen beinhalten. Es sollen scharfe Bilder ohne spezielle Handeingriffe erzeugt werden (marktübliche Autofokusfunktion). Die Kamera soll mit handelsüblichen Filmrollen arbeiten (Schwarz-Weiß, Farbe). Es sollen handelsübliche, wieder aufladbare Akkus verwendbar sein. Die Kamera soll im Low-cost-Bereich angesiedelt sein, möglichst geringes Gewicht aufweisen und eine möglichst lange Betriebszeit mit voll aufgeladenem Akkusatz ermöglichen.“

Typischerweise werden Nutzeranforderungen (z.B. vom Marketing) rein verbal formuliert und beinhalten noch große Entwurfsfreiheiten. Diese Freiheiten sind in weiterer Folge mittels der formalen qualitativen Systemmodelle in Kooperation mit dem Auftraggeber (z.B. mit dem Marketing) einzuschränken.

Kontextdiagramm Hier ist bereits die wesentliche Produktaußensicht, d.h. wie der Nutzer die Kamera sieht, erkennbar (Abb. 2.13). Im Grunde genommen kann mit den vorhandenen Datenflüssen und einer PSPEC der Hauptfunktion F0 bereits eine erste Version des Bedienhandbuches geschrieben werden. Die Datenflüsse sind alphanumerisch kodiert, um eine spätere Referenzierung zu erleichtern (D0.x = Ebene 0).

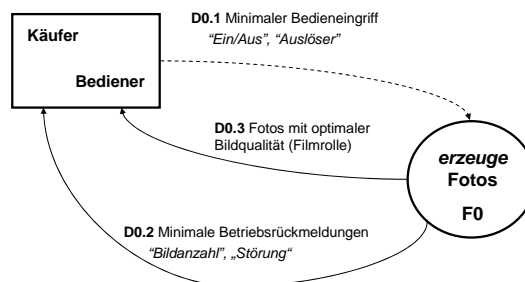


Abb. 2.13. Ebene 0 – Datenkontextdiagramm für Autofokuskamera

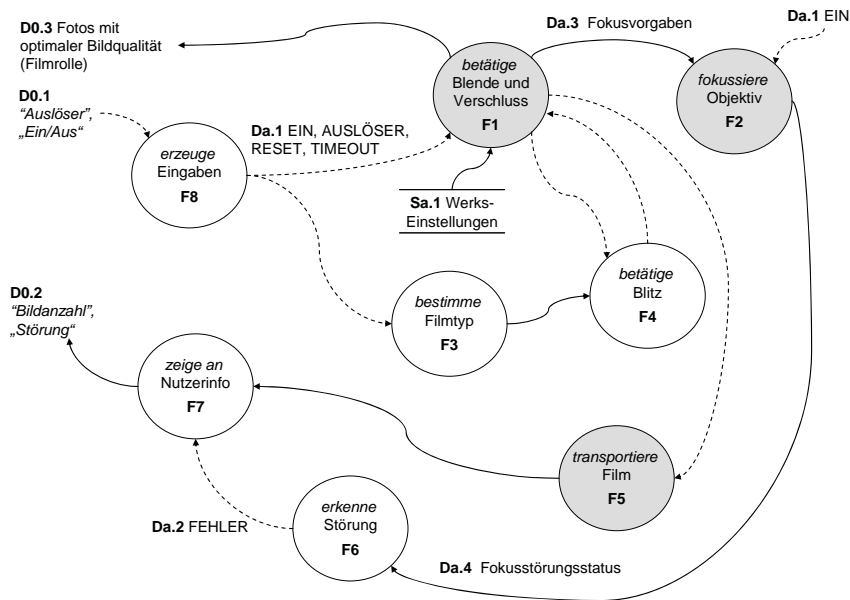


Abb. 2.14. Ebene a – Datenflussdiagramm für Hauptfunktion F0 *erzeuge_Fotos*. Mechatronische Funktionen sind grau hinterlegt (Beschriftung nur auszugsweise)

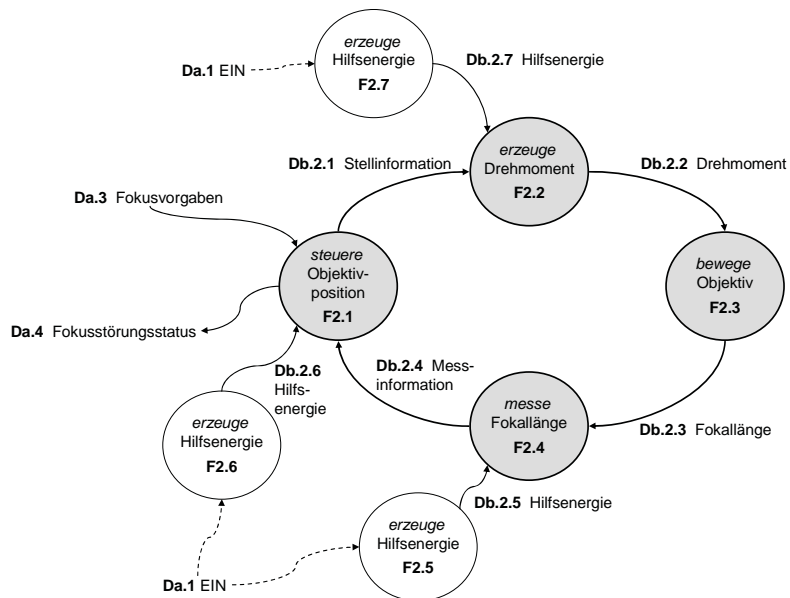


Abb. 2.15. Ebene b – Datenflussdiagramm für Funktion F2 *fokussiere_Objektiv*. Mechatronische Funktionen sind grau hinterlegt.

Datenflussdiagramme Bereits auf der *ersten Ebene-a* (Abb. 2.14), unterhalb des Kontextdiagramms, sind erste Kandidaten für mechatronische Funktionen sichtbar (F1, F2, F5), dort müssen offensichtlich massebehaftete Körper zielgerichtet bewegt werden. Mit acht Funktionen ist das DFD gerade noch lesbar, eine größere Anzahl würde das DFD sehr unübersichtlich machen.

Die Dekomposition ist natürlich alles andere als einzigartig, sie ist das Ergebnis einer subjektiven Systemsicht des Entwurfsingenieurs und sollte mit dem Auftraggeber abgestimmt sein. Die textuelle Darstellung von Datenflüssen und Funktionen erleichtert offensichtlich diesbezügliche Diskussionen über Fachgrenzen hinweg (bis hin zu technischen Laien). Die Funktionen und Datenflüsse sind alphanumerisch kodiert, um eine spätere Referenzierung zu erleichtern ($Da.x$ = Ebene a , Fi = Funktion i).

Die Funktion F7 *fokussiere_Objektiv* ist in der *zweiten Ebene-b* weiter dekomponiert (Abb. 2.15). Man erkennt hier schon deutlich eine geschlossene mechatronische Wirkungskette, die Kodierung von Funktionen und Datenflüssen ist wie gehabt ($Db.x$ = Ebene b , $F2.j$ = Unterfunktion j von F2).

PSPECs Obwohl die Funktionsbezeichnungen der DFD-Funktions-elemente bereits sehr gut im Klartext interpretierbar sind, empfiehlt es sich, die Funktionsinhalte noch weiter zu spezifizieren.

Im vorliegenden Fall (Abb. 2.16) ist für die Funktion F2 *fokussiere_Objektiv* eine detaillierte verbale Spezifikation gegeben und mit wichtigen numerischen Leistungsparametern ergänzt. Allerdings ist zum jetzigen Zeitpunkt offenbar der konkrete Parameter nicht bekannt und deshalb mit *TBD = to be defined* angegeben. Der konkrete Wert muss dann im weiteren Entwurfsprozess spezifiziert werden.

Zustandsdiagramm In diesem Diagramm (Abb. 2.17) ist das Ablaufmodell mit den Betriebsarten der Kamera sichtbar. Man beachte, dass die Überführungsbedingungen des Zustandsautomaten aus der Menge der Kontrollflüsse genommen werden müssen.

PSPEC F2 *fokussiere_Objektiv*

Das Objektiv soll automatisch so bewegt werden, dass eine scharfe Abbildung sichergestellt wird. Dazu soll ein ausgewähltes Objekt vor dem Objektiv als Referenz gewählt werden.

Die Fokallänge soll auf TBD [mm] genau positioniert werden. Der automatische Einstellvorgang soll nach max. TBD [sec] beendet sein.

Abb. 2.16. PSPEC für Unterfunktion F2 *fokussiere_Objektiv* (*TBD = to be defined*)

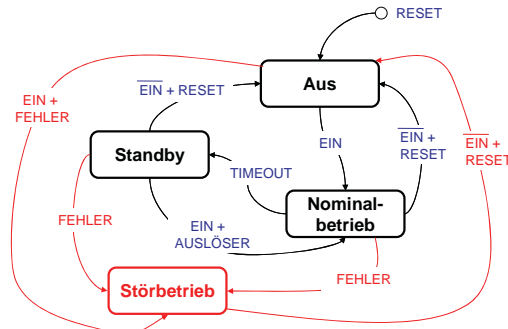


Abb. 2.17. Zustandsdiagramm für Hauptfunktion F0 *erzeuge_Fotos* (auf die Spezifikation der Aktionen der MEALY-Automatendefinition nach Abb. 2.10 wurde hier aus Übersichtlichkeitsgründen verzichtet)

Ebene	Datenfluss	Typ	Beschreibung
0	D0.1 Minimaler Bedieneingriff	C	Bedienhandlung durch Nutzer
	D0.2 Minimale Betriebsrückmeldung	D	Rückmeldung an Nutzer
	D0.3 Fotos mit optimaler Bildqualität	D	Ausgabe: belichtete und lichtgeschützt aufgerollte Filmroll
a	Da.1 EIN, AUSLÖSER, RESET, TIMEOUT	C	interne Steuersignale
	Da.2 FEHLER	C	internes Steuersignal
	Da.3 Fokussvorgaben	D	Betriebsvorgaben, Standardeinstellung durch Werk
b	Db.2.1 Stellinformation	D	berechnetes Stellsignal
	Db.2.2 Drehmoment	D	Drehmoment am Objektiv
	Db.2.3 Fokallänge	D	physikalische Fokallänge
	Db.2.4 Messinformation	D	gemessene Fokallänge
	Db.2.5 Hilfsenergie	D	Hilfsenergie für Messung
:	:	:	:

Abb. 2.18. Datenlexikon für Hauptfunktion F0 *erzeuge_Fotos* (Auszug)

Datenlexikon In dieser kompakten Darstellung (Abb. 2.18) können alle Datenflüsse näher beschrieben werden (*D* = Datenfluss, *C* = Kontrollfluss).

Entwurfsvarianten In Abb. 2.19 und Abb. 2.20 sind *zwei* verschiedene *Entwurfsvarianten* (Alternativen) für die Funktion F2 *fokussiere_Objektiv* dargestellt. In beiden Fällen werden die gleichen funktionellen Eigenschaften mittels unterschiedlicher Gerätetechnik realisiert. Die gerätetechnische Konzeption liegt völlig in der Hand des Entwurfsingenieurs, beispielsweise ist die Entwurfsentscheidung, zwei Mikrokontroller zu verwenden, in entsprechender Weise zu begründen.

Wichtig ist hier die Zuordnung der Funktionen und Datenflüssen des DFD zu den *Geräteeinheiten*. Damit wird klar, welche Aufgaben dieser Geräte zu erfüllen haben und zum anderen werden hier auch unmittelbar die *Hardwarechnittstellen* mit ihren Dateninhalten sichtbar gemacht. Man beachte, dass *erst hier* eine weitergehende *physikalische Modellierung* beginnen kann (s. Abschn. 2.3). Erst an dieser Stelle sind nämlich *Kandidaten* für konkrete gerätetechnische Einheiten bekannt, z.B. DC-Motor oder Piezomotor. Die Maßgabe, dass solche Geräteelemente verwendet werden, begründet sich im dargestellten Entwurfsbeispiel eindeutig aus den Funktionen als Konkretisierung der Nutzeranforderungen. Insofern ist eine derart abgeleitete Gerätearchitektur nachvollziehbar und klar begründet, keines der Geräteelemente „fällt irgendwie vom Himmel“. Unbenommen bleibt natürlich die konkrete Auswahl sachlich zu begründen. Im vorliegenden Fall ergeben sich für die beiden Varianten DC-Motor und Piezomotor ganz *unterschiedliche Eigenschaften*, z.B. Geräteaufwand (wenige Elemente bei Piezo), Hilfsenergie (Hochvolt bei Piezo), Bewegungsverhalten, Masse, Kosten. Eine endgültige Auswahl kann nur nach einer Analyse aller Verhaltens- und Leistungseigenschaften erfolgen. Als Ausgangsbasis dient aber für alle diese weitergehenden Untersuchungen das qualitative funktionelle Systemmodell.

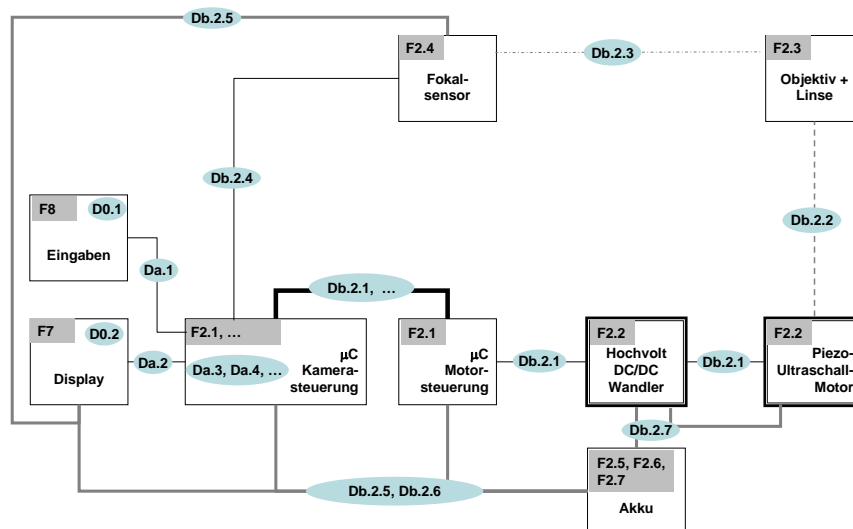


Abb. 2.19. Entwurfsvariante A für Funktion F2 *fokussiere_Objektiv* mit DC-Motor und Getriebe – Architekturdiagramm mit Zuordnung Funktionen/Datenflüsse ↔ Gerätetechnik

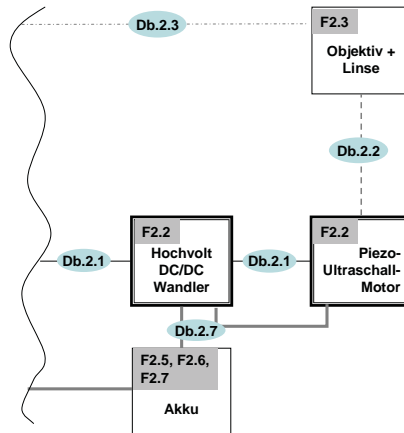


Abb. 2.20. Entwurfsvariante B für Funktion F2 *fokussiere_Objektiv* mit *getriebe-losem Piezo-Ultraschallmotor* – Architekturdiagramm (Ausschnitt) mit Zuordnung Funktionen/Datenflüsse ↔ Gerätetechnik

Nomenklatur Die Strukturierte Analyse kennt keine standardisierte Nomenklatur, lediglich bei der Verwendung von rechnergestützten Werkzeugen sind strikte Syntaxregeln einzuhalten. Im vorliegenden Beispiel wurde eine sehr einfache und pragmatische Nomenklatur verwendet, die sich aus der langjährigen Erfahrung des Autors im industriellen Bereich und im Lehrbetrieb als sehr nützlich erweisen hat. Als wichtigstes Syntaxelement hat sich die *alphanumerische Kodierung* erwiesen, womit eine einfache, prägnante und eindeutige Benennung von Modellgrößen möglich wird. Im vorliegenden Beispiel sieht man auch sehr schön, dass die Strukturierte Analyse leicht *ohne spezielle Werkzeuge* anwendbar ist und häufig verwendete Textverarbeitungswerkzeuge durchaus ausreichen.

2.2.5 Alternative Modellierungsmethoden

Objektorientierte Systemmodellierung mit UML

Unified Modeling Language – UML Neben der bereits angesprochenen *Strukturierte Analyse mit Real Time Ergänzung (SA/RT)* nach *Hatley* und *Pirbhai* sei aus Gründen des ingenieurmäßigen Zuganges und der internationaler Akzeptanz als einzige ernsthafte Alternative für eine *umfassende* Systemmodellierung der *objektorientierte* Formalismus *Unified Modeling Language (UML)* genannt (Oestereich 2006), (Vogel-Heuser 2003).