## Organizational Aspects

| | |
|---|---|
| **Project:** | **SFB 501** |
| **Subproject:** | **Team 2** |
| **Documenttype:** | **Informelle Beschreibung** |
| **Documentname:** | **Systemarchitektur** |
| **Responsible:** | **Dellen, Kronenburg, Molter, Münch, Peper, Queins, Verlage** |
| **User:** | **Team2** |
| **Description:** | In diesem Bericht wird eine Kontrollsystem-Architektur beschrieben. |

| **Version#** | **Date** | **Creators** | **Remarks** |
|---|---|---|---|
| V1 | 17.01.96 | Münch | Initiale Version; entstanden aus dem Ergebnisdokument der Architekturgruppe |
| V2 | 05.02.96 | Münch | Überarbeitung von Kap. 2 |
| V3 | 20.02.96 | Münch | Überarbeitung |
| V4 | 22.02.96 | Münch | Überarbeitung |
| V5 | 29.02.96 | Münch | Überarbeitung |
| V6 | 05.03.96 | Münch | Einarbeitung der Ergebnisse des Reviews von Kap. 1-3 durch Team 2 am 05.03.96 |
| V7 | 08.03.96 | Münch | Überarbeitung |
| V8 | 11.03.96 | Münch | Überarbeitung von Kap. 4 |
| V9 | 12.03.96 | Münch | Überarbeitung von Kap. 4 |
| V10 | 15.03.96 | Münch | Überarbeitung |
| V11 | 19.03.96 | Münch | Überarbeitung /*Kap. 4 kann von Team2 reviewt werden */ |
| | | | |
| | | | |
| | | | |

# 1 Einleitung

Dieses Dokument beschreibt die Systemarchitektur des Kontrollsystems. Als Grundlage dient die Anforderungsbeschreibung. Die Systemarchitektur beschreibt abstrakt, aus welchen Komponenten das System aufgebaut sein kann, wie diese Komponenten in Beziehung zueinander stehen können und auf welchen grundlegenden Konzepten das System basiert.

Bei der Erstellung der Systemarchitektur standen im wesentlichen zwei Ziele im Vordergrund:

- *Skalierbarkeit*: Die Architektur soll leicht auf größere Systeme (ganze Stockwerke, Gebäude, ...) erweiterbar sein.
- *Generizität*: Die Architektur soll leicht an veränderte Rahmenbedingungen anpaßbar sein. Dies kann z. B. die zur Verfügung gestellte Funktionalität betreffen.

Die Systemkomponenten werden in weiteren Dokumenten genauer beschrieben.

# 2 Systemaufbau

## 2.1 Einleitung

Dieses Kapitel enthält eine Einordnung des Kontrollsystems in das umgebende Gesamtsystem. Es werden grundlegende Begriffe definiert, und der logische Aufbau des Kontrollsystems wird beschrieben.

## 2.2 Systemüberblick und Schnittstellen

Das Kontrollsystem dient der Steuerung von Gebäudeteilen bez. verschiedener physikalischer Größen. Abbildung 1 gibt einen Überblick über die Einbettung des Kontrollsystems in seine Umwelt (die schattierten Bereiche werden in einer ersten Systemvariante nicht realisiert).

Das System ist so ausgelegt, daß es sowohl zur Steuerung von real existierenden Gebäudeteilen dienen kann als auch mit einem Simulator, der das physikalische Verhalten der Umwelt simuliert, lauffähig ist. Für das Kontrollsystem ist es unsichtbar, ob die reale Welt oder eine simulierte Welt gesteuert wird.

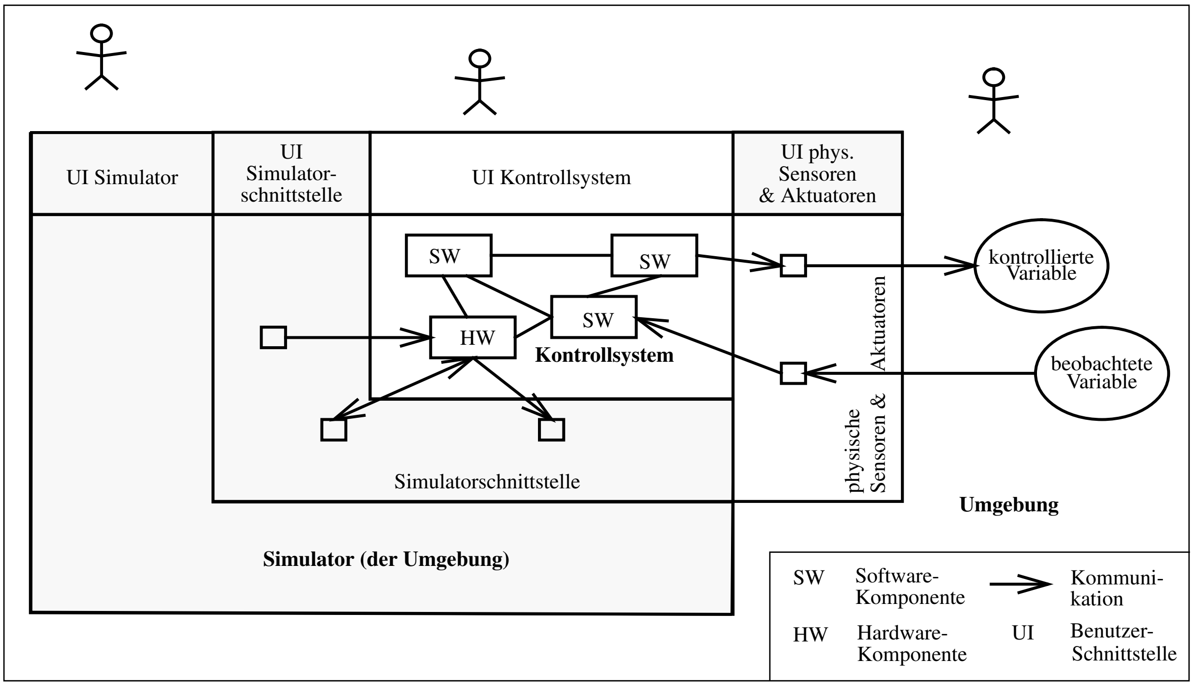


**Abb. 1: Kontrollsystem und Umgebung**

## 2.3 Grundlegende Definitionen

Zur Beschreibung der Systemstruktur werden zunächst grundlegende Begriffe definiert. Die Definition einiger Begriffe erfordert Vorwärtsverweise auf andere Definitionen innerhalb dieses Abschnitts.

### 2.3.1 Kontrollgröße (control quantity)

Eine *Kontrollgröße (engl.: control quantity)* ist eine physikalische Größe, die gesteuert werden kann (z. B.: Temperatur, Feuchtigkeit, ...). Darüber hinaus wird die Größe Safety (Schutz vor Zerstörung) als Kontrollgröße verstanden.

### 2.3.2 Kontrollbereich (control domain)

Ein *Kontrollbereich (engl.: control domain)* ist der Bereich, in dem eine (oder mehrere) Kontrollgrößen gesteuert werden (z. B.: Büroraum, Flur).

### 2.3.3 Komponente

Eine *Komponente* ist ein Teil des Kontrollsystems. Komponenten des Kontrollsystems können sowohl Software- als auch Hardware-Bestandteile beinhalten. Es gibt sechs Arten von Komponenten: *Aktuatoren, Sensoren, Eingabeeinheiten, Ausgabeeinheiten, Kontrollzellen* und *Kontrollgruppen*.

### 2.3.4 Aktuator (actuator, A)

Ein *Aktuator* ist eine Komponente des Kontrollsystems, mit dem die Umgebung beeinflußt werden kann. Hierfür bietet er eine Schnittstelle zur Kontrollhierarchie an.

### 2.3.5 Sensor (sensor, S)

Ein *Sensor* ist eine Komponente des Kontrollsystems, mit dem die Umgebung beobachtet werden kann. Hierfür bietet er eine Schnittstelle zur Kontrollhierarchie an.

### 2.3.6 Eingabeeinheit (input unit, IU)

Eine *Eingabeeinheit* ist eine Komponente des Kontrollsystems, mit der der Benutzer Eingaben an das Kontrollsystem machen kann. Eine Eingabeeinheit hat eine Schnittstelle zur Kontrollhierarchie.

### 2.3.7 Ausgabeeinheit (output unit, OU)

Eine *Ausgabeeinheit* ist eine Komponente des Kontrollsystems, mit der das Kontrollsystem Nachrichten an den Benutzer übermitteln kann. Eine Ausgabeeinheit hat eine Schnittstelle zur Kontrollhierarchie.

### 2.3.8 Kontrolleinheit (control unit, CU)

Als *Kontrolleinheiten* werden die logischen Grundeinheiten bezeichnet, aus denen die *Kontrollhierarchie* aufgebaut ist. Der Begriff “Kontrolleinheit” ist der Sammelbegriff für *Kontrollzelle, Kontrollgruppe* und *Kontrollregion*.

### 2.3.9 Kontrollhierarchie

Die Kontrolleinheiten sind hierarchisch angeordnet. Die hierbei verwendete Hierarchie wird im folgenden als *Kontrollhierarchie* bezeichnet.

### 2.3.10 Kontrollzelle (control cell, CC)

Die *Kontrollzellen* bilden die unterste Ebene in der Kontrollhierarchie. Eine Kontrollzelle steuert genau eine Kontrollgröße für einen Kontrollbereich.

Jede Kontrollzelle besitzt einen *Zellcontroller (engl.: cell controller)*, in dem die Steuerung der Kontrollgröße dieser Kontrollzelle stattfindet. Sofern keine Mißverständnisse auftreten, werden Kontrollzellen mit den ihnen zugehörigen Zellcontrollern identifiziert, d. h., falls Kontrollzellen Aktionen durchführen und bestimmte Zustände annehmen, so sind jeweils die zu den Kontrollzellen gehörenden Controller gemeint, die die Aktionen durchführen bzw. bestimmte Zustände annehmen.

### 2.3.11 Kontrollgruppe (control group, CG)

*Kontrollgruppen* aggregieren Kontrollzellen oder Kontrollgruppen.

Eine Kontrollgruppe besteht aus einer Anzahl von Kontrollzellen bzw. -gruppen und einem *Gruppencontroller (engl.: group controller, kurz GC)*, der die anderen Komponenten der Kontrollgruppe, insbesondere die Sohnkontrolleinheiten, koordiniert. Es ist möglich, daß eine Kontrollgruppe sowohl Kontrollzellen als auch Kontrollgruppen aggregiert. Sofern keine Mißverständnisse auftreten, werden Kontrollgruppen mit den ihnen zugehörigen Gruppencontrollern identifiziert, d. h., falls Kontrollgruppen Aktionen durchführen und bestimmte Zustände annehmen, so sind jeweils die zu den Kontrollgruppen gehörenden Controller gemeint, die die Aktionen durchführen bzw. bestimmte Zustände annehmen.

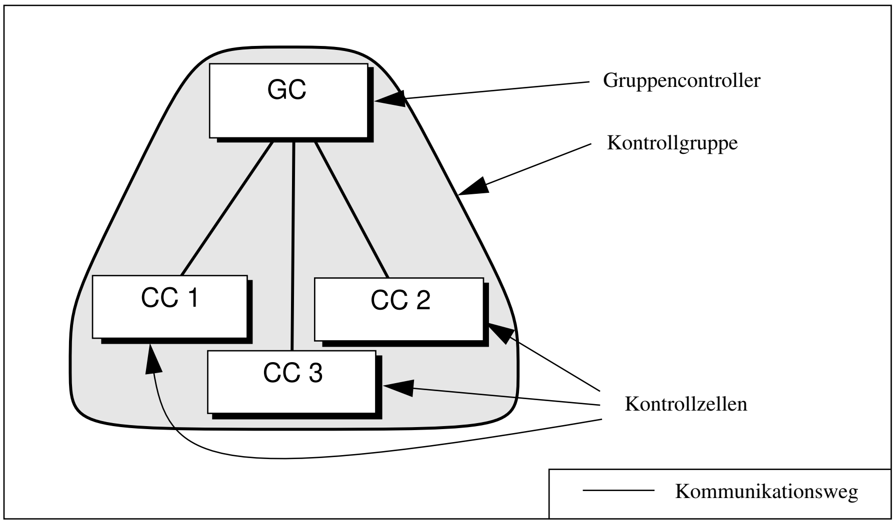


**Abb. 2: Aufbau einer Kontrollgruppe**

### 2.3.12 Kontrollregion (control region, CR)

Die *Kontrollregion* ist die oberste Ebene in der Kontrollhierarchie, d. h. sie ist die umfassendste Kontrolleinheit (z. B.: Gebäude, Universität). Sie besteht aus einer Anzahl von Kontrollgruppen und Kontrollzellen. Verschiedene Kontrollregionen beeinflussen sich nicht gegenseitig (siehe Abbildung 3).

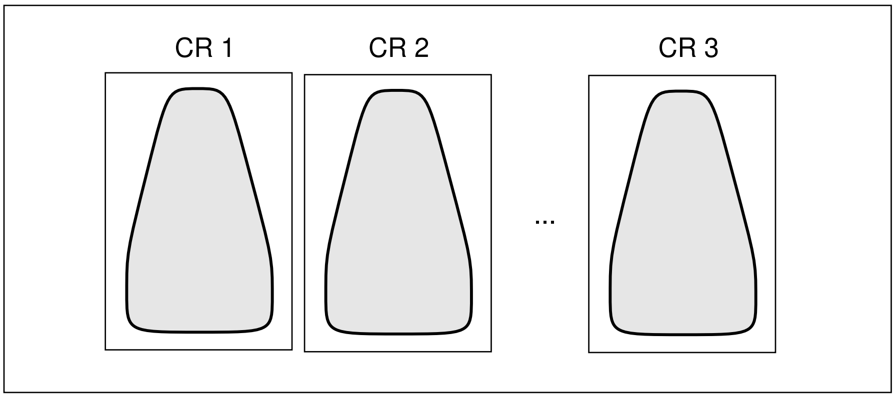


**Abb. 3: Verschiedene Kontrollregionen**

## 2.4 Logischer Systemaufbau

Das System umfaßt zum einen Komponenten, die hierarchisch strukturiert den *Kontrollsystemkern* bilden (CC, CG, CR), zum zweiten Komponenten, durch die der Kontrollsystemkern mit dem Benutzer kommuniziert (IU, OU), und zum dritten Komponenten, durch die der Kontrollsystemkern mit der physischen Außenwelt kommuniziert (S, A). Abbildung 4 stellt den logischen Systemaufbau dar.

Es sei angemerkt, daß durch den logischen Systemaufbau nicht festgelegt ist, welche Kontrolleinheiten mit den S, A, IU bzw. OU kommunizieren. Insbesondere kann es auch vorkommen, daß S, A, IU bzw. OU mit Nicht-Blatt-Kontrolleinheiten der Kontrollhierarchie kommunizieren.

Für Sensoren und Aktuatoren gilt folgende Regel: Sensoren können von verschiedenen Aktuatoren abgefragt werden, wohingegen Aktuatoren jeweils nur von einer ausgezeichneten Kontrolleinheit angesteuert werden. Der Grund hierfür ist, daß vor der Einstellung von Aktuatoren evtl. verschiedene Einstellungswünsche in Einklang gebracht werden müssen (siehe dazu “Konfliktlösung”).

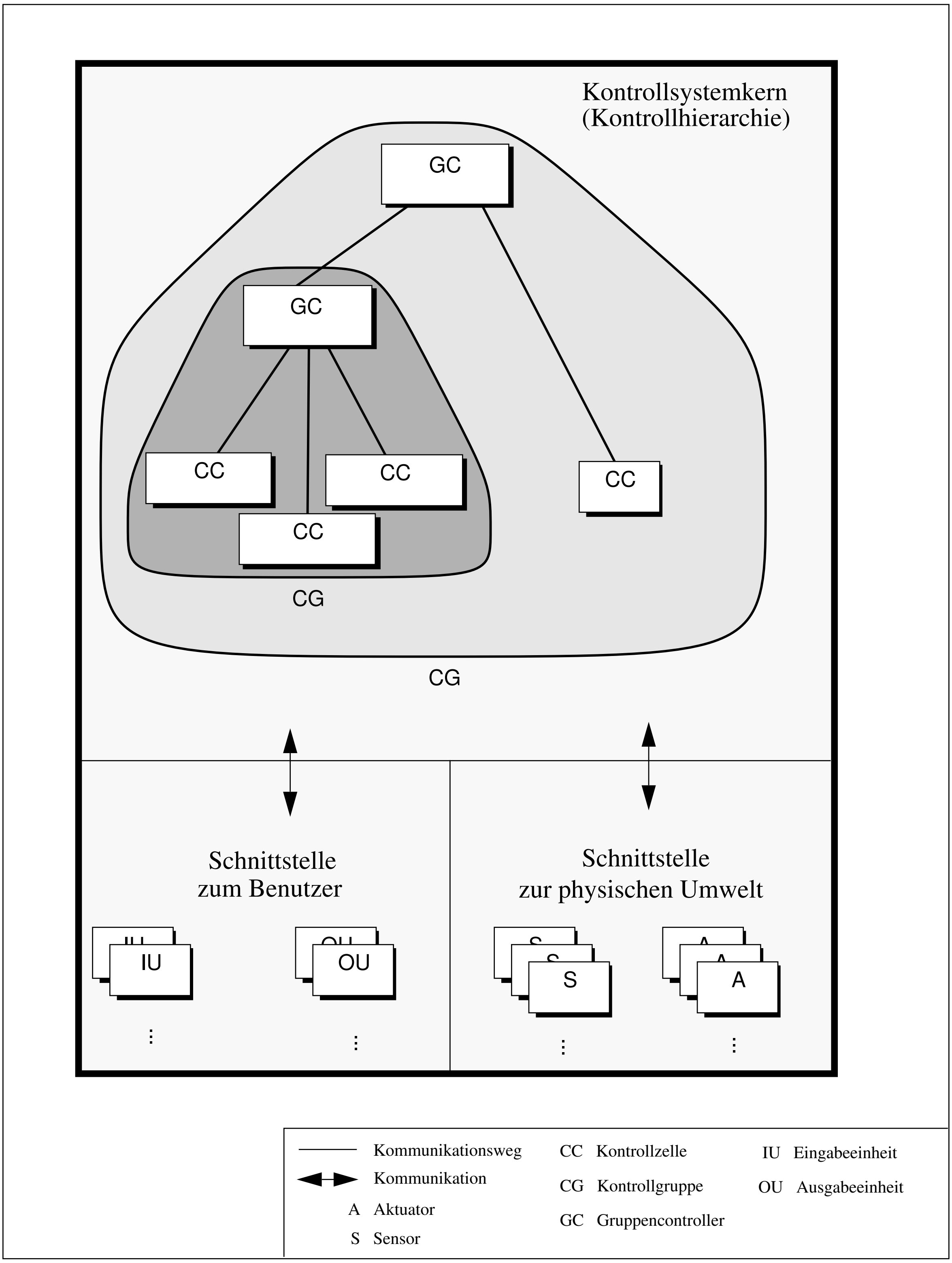


**Abb. 4: Logischer Systemaufbau**

# 3 Anwendungsspezifische Systemstrukturierung

## 3.1 Einleitung

Der Aufbau einer Kontrollhierarchie erfolgt in Abhängigkeit von einem konkreten Gebäude bzw. Gebäudeteil sowie den darin zu steuernden Kontrollgrößen. Mit der hier beschriebenen Architektur wird angestrebt, generisch für verschiedene Anwendungen in der Domäne “Hausautomation” Kontrollsysteme erzeugen zu können (d. h., daß eine möglichst einfache Anpassung der Architektur an unterschiedliche funktionale Anforderungen und räumliche Situationen angestrebt wird).

Kontrollhierarchien können nach verschiedenen Gesichtspunkten aufgebaut werden. Möglichkeiten hierfür werden im nächsten Abschnitt gegeben. Abschließend wird als konkretes Beispiel die Systemstruktur eines eingeschränkten Kontrollsystems angegeben.

## 3.2 Möglichkeiten der Aggregation von Kontrolleinheiten

Die Zusammenfassung von Kontrollzellen bzw. Kontrollgruppen zu neuen Kontrollgruppen kann nach verschiedenen Gesichtspunkten erfolgen:

### 3.2.1 Funktionale Strukturierung

Hierbei werden Kontrolleinheiten zusammengefaßt, die zur Realisierung einer Funktion beitragen (z. B.: Zusammenfassung aller Kontrolleinheiten, die zur Steuerung des Raumklimas dienen).

### 3.2.2 Räumliche Strukturierung

Hierbei werden Kontrolleinheiten zusammengefaßt, die zur Steuerung eines bestimmten Gebäudeteils dienen (z. B.: Zusammenfassung aller Kontrolleinheiten, die zur Steuerung eines Büroraums dienen).

### 3.2.3 Kombinierte Strukturierung

Bei dieser Art der Strukturierung werden sowohl räumliche als auch funktionale Gesichtspunkte kombiniert (z. B.: Zusammenfassung aller Kontrolleinheiten, die das Licht eines Büroraums steuern).

## 3.3 Einschränkung des Kontrollsystems

Um die Entwicklung des Kontrollsystems in seiner Komplexität beherrschbar zu machen, sollen in einer ersten Systemvariante nur bestimmte Funktionalitäten realisiert werden und der zu steuernde Gebäudeteil überschaubar sein. Hierfür wurde das Kontrollsystem eingeschränkt auf die Steuerung der Kontrollgrößen Temperatur, Lüftung und Safety innerhalb eines Büroraums (siehe Tabelle 1).

| **Mögliche Kontrollzellen zur Steuerung eines Büroraums** | **Name** | **In der Einschränkung betrachtete Kontrollzellen** |
|---|---|---|
| Temperatur | CC1 | X |
| Lüftung | CC2 | X |
| Feuchtigkeit | CC3 | |
| Beleuchtung-Türseite | CC4 | |
| Beleuchtung-Fensterseite | CC5 | |
| Beleuchtung-Arbeitsplatz | CC6 | |
| Security (Schutz vor Einbruch) | CC7 | |
| Safety | CC8 | X |

**Tabelle 1: Kontrollzellen**
**(X = Kontrollzelle der Einschränkung)**

Die weitere Beschreibung der Systemarchitektur ist ausgerichtet auf diese spezielle Einschränkung. Es ist geplant, in weiteren Systemvarianten sowohl die Funktionalität als auch die zu steuernden Gebäudeteile inkrementell zu erweitern, um Untersuchungen im Rahmen der Forschungsschwerpunkte des SFB 501 (wie z. B.: generische Methoden, domänenspezifische Entwicklung) durchzuführen.

## 3.4 Systemstruktur des eingeschränkten Kontrollsystems

Die untenstehende Abbildung zeigt den Aufbau der Kontrollhierarchie für das eingeschränkte Kontrollsystem (d. h. Steuerung der Kontrollgrößen Temperatur, Lüftung und Safety innerhalb eines Büroraums). Die Kontrollgruppe “Büroraum” ist räumlich strukturiert, die Kontrollgruppe “Klima” ist funktional strukturiert.

Es sei angemerkt, daß es sich hierbei nicht um den einzig möglichen Aufbau einer Kontrollhierarchie handelt. Andere Möglichkeiten können in weiteren Systemvarianten realisiert werden, sollen hier jedoch nicht weiter verfolgt werden.

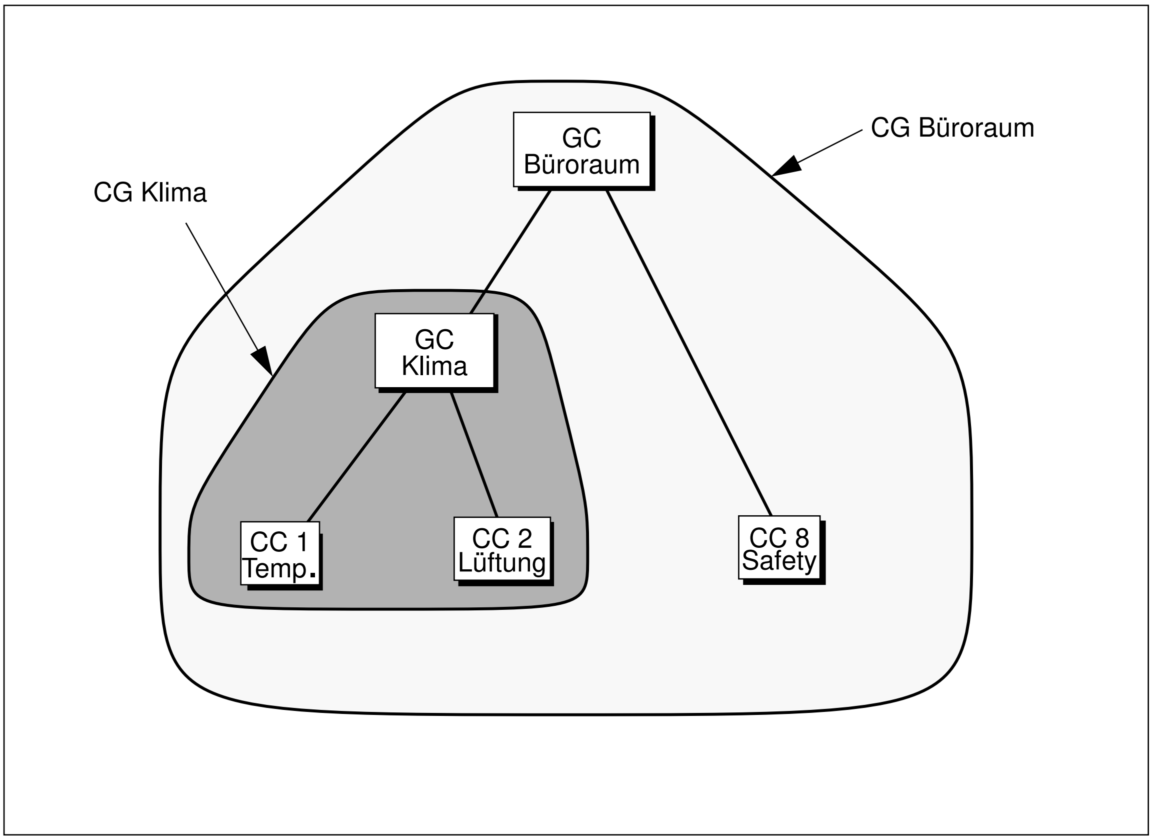


**Abb. 5: Aufbau der Kontrollhierarchie für das eingeschränkte Kontrollsystem**

# 4 Konfliktlösung

## 4.1 Einleitung

Kontrolleinheiten sind Aufgaben zugeordnet (z. B.: Steuerung eines Raumes oder einer Kontrollgröße). Bei der Ausführung der Aufgaben können Kontrolleinheiten parallel arbeiten. Hierbei kann es, falls zwei Kontrolleinheiten dieselbe Größe einstellen wollen, zu Konflikten kommen, die gelöst werden müssen.

In diesem Kapitel wird eine Methodik zur Lösung von Konflikten erläutert.

## 4.2 Konfliktarten

In der hier vorgestellten Methodik werden zwei Konfliktarten unterschieden, die in der Anwendungsdomäne “Gebäudeautomation” von Bedeutung sind: Aktuatorkonflikte und Vorgabenkonflikte.

### 4.2.1 Aktuatorkonflikte

*Aktuatorkonflikte* entstehen dadurch, daß zwei oder mehrere Kontrolleinheiten zu einem bestimmten Zeitpunkt unterschiedliche Einstellungen desselben Aktuators anstreben (z. B.: die Lüftungskontrollzelle strebt ein geöffnetes Fenster an und gleichzeitig strebt die Temperaturkontrollzelle ein geschlossenes Fenster an).

### 4.2.2 Vorgabenkonflikte

*Vorgabenkonflikte* entstehen, wenn eine Kontrolleinheit nicht die Einstellung eines konkreten Aktuators beeinflussen möchte, sondern den Zielwert (Vorgabewert) für eine Kontrollgröße (z. B.: Licht, Temperatur) setzen möchte, die von einer anderen Kontrolleinheit überwacht wird (z. B.: eine Lüftungskontrollzelle benötigt zum Erreichen guter Lüftungsverhältnisse einen anderen Temperatur-Zielwert als den, den die für die Temperatursteuerung zuständige Kontrollzelle verfolgt).[1]

## 4.3 Methodik zur Behandlung von Aktuatorkonflikten

In diesem Abschnitt wird eine Methodik beschrieben, mit deren Hilfe Aktuatorkonflikte bearbeitet und aufgelöst werden können.

### 4.3.1 Prinzipieller Ablauf

Ursache für einen Konflikt ist, daß zwei oder mehrere verschiedene Kontrolleinheiten bez. eines Aktuators unterschiedliche Einstellungen wünschen.

Will eine Kontrolleinheit einen Aktuatorwert einstellen, der sich von der aktuellen Einstellung unterscheidet, gibt es einen Konflikt, wenn für andere Kontrolleinheiten die aktuelle Einstel-

1. Im eingeschränkten Kontrollsystem treten keine Vorgabekonflikte auf, d. h. es kann nicht vorkommen, daß zwei Kontrolleinheiten verschiedene Vorgabewerte einstellen wollen. Da die Methodik zur Lösung von Vorgabekonflikten der Methodik zur Lösung von Aktuatorkonflikten weitestgehend entspricht, werden beide Konfliktarten in diesem Dokument berücksichtigt.

lung erforderlich ist. Für einen solchen Konflikt, im folgenden *Aktuatorkonflikt* genannt, muß eine Lösung gefunden werden. Dies geschieht auf die folgende Weise:

Für jeden Aktuatorkonflikt gibt es einen zuständigen Gruppencontroller, der alleine den Aktuator kontrolliert und der in der Kontrollhierarchie allen Kontrolleinheiten übergeordnet ist, die auf diesen Aktuator zugreifen wollen. Möchte eine Kontrolleinheit einen Aktuatorwert einstellen, so stellt diese einen Antrag, der über den Vater Stufe für Stufe in der Hierarchie nach oben an den für diesen Aktuatorkonflikt zuständigen Gruppencontroller weitergeleitet wird. Dieser Gruppencontroller ist verantwortlich für die Festlegung einer neuen Aktuatoreinstellung, d. h. er startet einen Konfliktlösungsablauf. Es gibt verschiedene mögliche Konfliktlösungsabläufe, z. B. können alle an diesem Aktuatorkonflikt beteiligten Kontrolleinheiten nach ihren Wünschen gefragt werden. Nachdem eine Aktuatoreinstellung festgelegt wurde, wird diese den beteiligten Kontrolleinheiten vom für den Aktuatorkonflikt zuständigen Gruppencontroller mitgeteilt.

Wesentliche Merkmale dieser Methodik sind:

- *Ansteuerung eines Aktuators durch genau eine Instanz*. Der Gruppencontroller, der bez. einer Konfliktbehandlung in der Hierarchie am höchsten steht, steuert als einzige Kontrolleinheit den Aktuator. Untergeordnete Instanzen können nur einen Antrag auf den Start einer Konfliktbehandlung stellen. Der Konfliktlösungsablauf für die Auflösung eines Konfliktes kann nur von *einer* Instanz gestartet werden, nämlich von dem Gruppencontroller, der bez. einer Konfliktbehandlung in der Hierarchie am höchsten steht.
- *Keine Unterbrechungen eines Konfliktlösungsablaufs*. Es wird festgelegt, daß ein Konfliktlösungsablauf, der einmal gestartet wurde, nicht unterbrochen werden kann. Beispiel: Die Kontrollzelle “Safety” möchte aufgrund einsetzenden Regens, daß das Fenster geschlossen wird. Falls gerade eine Konfliktlösung bezüglich der Fensteröffnung stattfindet, bringt die Kontrollzelle “Safety” entweder ihren Wunsch im Rahmen dieses Konfliktlösungsablaufes ein oder stellt nach Abschluß dieser Konfliktlösung einen Antrag an den zuständigen Gruppencontroller zum Schließen des Fensters.
- *Verschiedene Möglichkeiten der Konfliktlösung*. Die hier vorgestellte Methodik erlaubt verschiedene Möglichkeiten, Aktuatorkonflikte zu lösen (z. B. mittels Prioritäten, durch Kompromißfindung etc.)
- *Kommunikationsaufwand*. Die hier vorgestellte Methodik erfordert einen gewissen Kommunikations- und somit Zeitaufwand für die Lösung von Konflikten. Dieser Zeitaufwand wird jedoch als vertretbar betrachtet, da es für die gewünschte Funktionsfähigkeit des hier betrachteten Kontrollsystems nicht notwendig ist, „sehr schnell“, d.h. im Millisekundenbereich oder noch schneller auf Ereignisse zu reagieren. Stattdessen ist es ausreichend, wenn das Kontrollsystem innerhalb weniger Sekunden auf „besondere“ Ereignisse reagiert (zum Beispiel die Gefahr von Schäden für die Jalousie durch Wind). Dies kann mit der hier vorgeschlagenen Methodik realisiert werden.
- *Parallele Konfliktlösungen*. In einem Kontrollsystem können mehrere Aktuatorkonflikte (z. B. Fensteröffnung und Heizungsventil) gleichzeitig behandelt werden, indem parallele Konfliktlösungen laufen. Diese sind voneinander unabhängig.

### 4.3.2 Nachrichtentypen

In diesem Abschnitt werden die Typen von Nachrichten vorgestellt, die im Rahmen einer Konfliktlösung ausgetauscht werden.

1. Kontrolleinheiten können zunächst einen *Request (Antrag)* bez. eines Konfliktes an ihren Vater senden, mit dem sie zum Ausdruck bringen, daß aus ihrer Sicht ein Konfliktlösungsablauf gestartet werden soll, weil sie eine neue Einstellung für einen Aktuator möchten. Das Versenden eines Antrags bedeutet ausschließlich, daß eine CU einen Konfliktlösungsablauf starten möchte. Der Antrag enthält keine weiteren Informationen.[1]

Wurde ein Konfliktlösungsablauf gestartet, so können die folgenden vier Arten von Nachrichten zwischen den Kontrolleinheiten ausgetauscht werden (siehe Tabelle 2):

2. *Wish (Wunschnachricht)*: Einstellungswunsch von einer Kontrolleinheit auf niedrigerer Ebene, der dort aufgrund der lokalen Sichtweise berechnet wurde.
3. *Demand (Forderungsnachricht)*: eine aus Sicht einer niedrigeren Kontrolleinheit notwendige Aktuatoreinstellung. Diese Art von Nachricht wird dann verschickt, wenn ausgedrückt werden soll, daß ohne diese Einstellung das vorgegebene Ziel (z.B. ein Temperaturintervall) keinesfalls erreicht/eingehalten werden kann.
4. *Proposal (Vorschlagsnachricht)*: von einer höheren Kontrolleinheit berechneter Kompromiß, der jedoch noch nicht endgültig sein muß.
5. *Command (Vorgabenachricht)*: von einer höheren Kontrolleinheit berechnete Entscheidung, die eingehalten werden muß.

| | | Nachrichteninhalt | |
|---|---|---|---|
| | | normal | dringend |
| Sender | Sohn-CU | Wish | Demand |
| | Vater-CU | Proposal | Command |

**Tabelle 2: Nachrichtenarten beim Konfliktlösungsablauf**

Zusammengefaßt gibt es fünf Nachrichtentypen, einen zum Stellen eines Antrags auf Konfliktlösung und vier für den Konfliktlösungsablauf.

### 4.3.3 Arten von Kontrolleinheiten (Top-CUs, Middle-CUs, Bottom-CUs)

In der hier vorgestellten Methodik zur Konfliktlösung werden für jeden einzelnen Konflikt drei Arten von Kontrolleinheiten unterscheiden.[2]

---

1. Der Grund hierfür ist, daß vor der Entscheidung über die Aktuatoreinstellung die Einstellungswünsche aller beteiligten Kontrolleinheiten eingeholt werden sollen.

2. In der Beschreibung werden jeweils die Kontrollgruppen (CGs) mit den ihnen zugehörigen Controllern identifiziert, d. h., falls CGs Aktionen durchführen und bestimmte Zustände annehmen, so sind jeweils die zu den CGs gehörenden Controller gemeint, die die Aktionen durchführen bzw. bestimmte Zustände annehmen.

- Die *Top-CU* eines Konfliktes: Dies ist die in der Kontrollhierarchie am niedrigsten angeordnete Kontrolleinheit, unterhalb derer alle Kontrolleinheiten liegen, die den Aktuator beeinflussen wollen (siehe Abbildung 6). Pro Konflikt gibt es genau eine solche Kontrolleinheit. Die Top-CU ist verantwortlich für die Festlegung einer neuen Aktuatoreinstellung, d. h. sie startet einen Konfliktlösungsablauf und führt die Konfliktlösung durch.

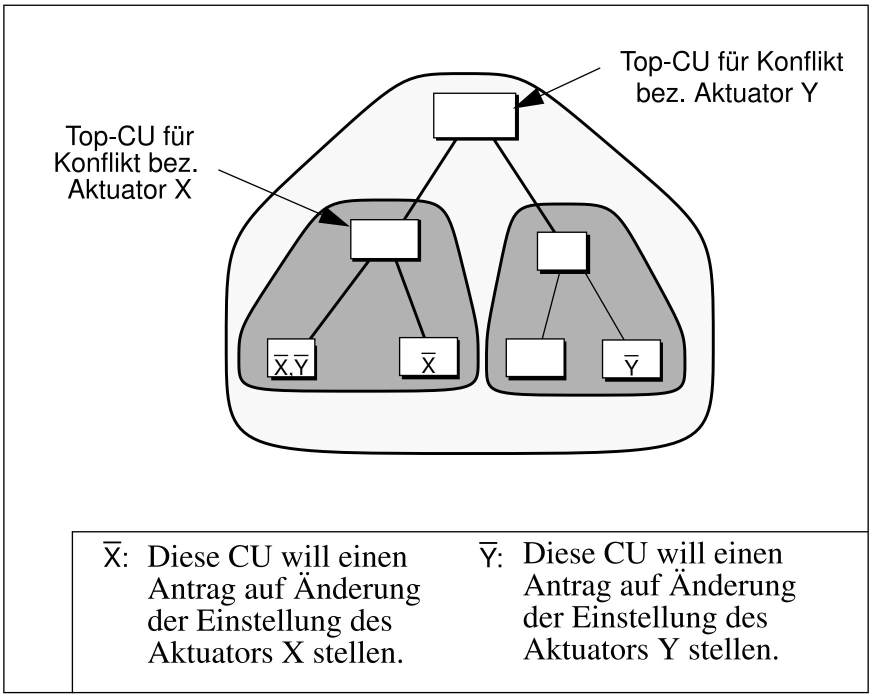


**Abb. 6: Beispiele für Top-CUs**

- *Bottom-CUs* eines Konfliktes: Dies sind die Kontrolleinheiten, die bez. eines Konfliktes in der Hierarchie an unterster Stelle stehen (siehe Abbildung 7).

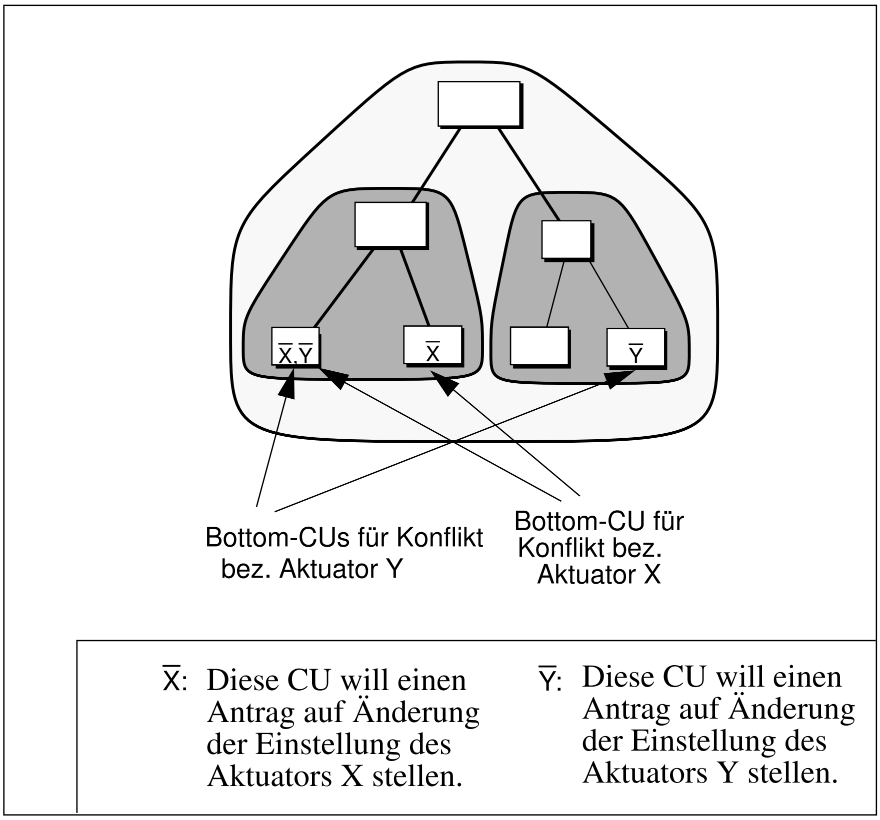


**Abb. 7: Beispiele für Bottom-CUs**

- *Middle-CUs* eines Konfliktes: Dies sind die Kontrolleinheiten, die bez. eines Konfliktes in der Hierarchie zwischen den Bottom-CUs und dem Top-CU liegen (z. B.: für den Aktuatorkonflikt bez. des Aktuators Y in Abbildung 8). Es kann vorkommen, daß es bez. eines Aktuatorkonflikts keine Middle-CUs gibt (z. B.: für den Aktuatorkonflikt in Abbildung 8 gibt es keine Middle-CUs)

.

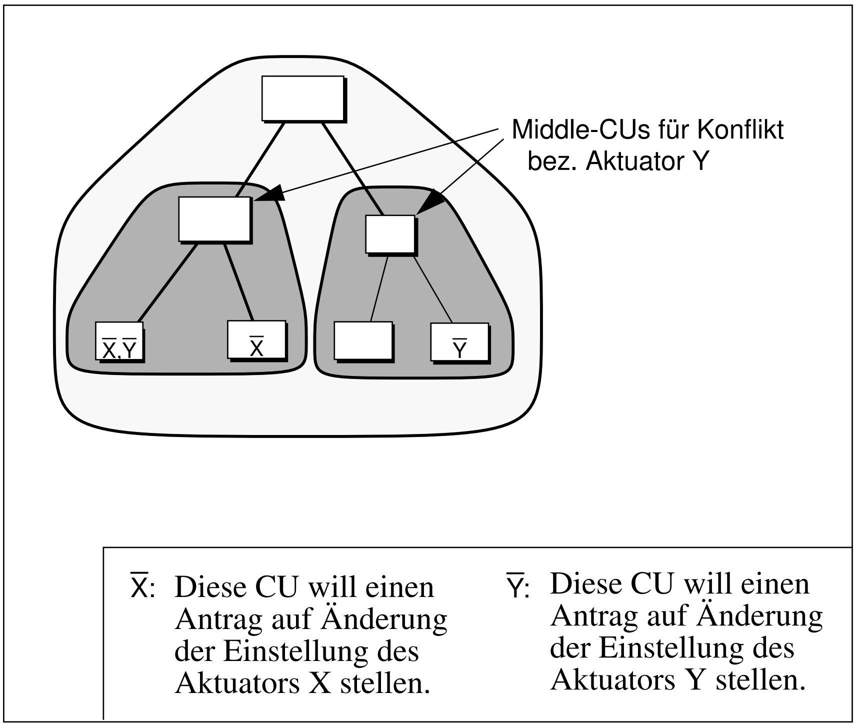


**Abb. 8: Beispiele für Middle-CUs**

### 4.3.4 Aufgaben der beteiligten Kontrolleinheiten

Den drei Arten von Kontrolleinheiten (Top-CUs, Middle-CUs, Bottom-CUs) sind während eines Konfliktlösungsablaufes unterschiedliche Aufgaben zugeordnet:

- *Aufgaben einer Top-CU:*
  - Sie kann als einzige Instanz einen Konfliktlösungsablauf starten.
  - Sie kann auf Antrag einer ihr untergeordneten Kontrolleinheit den Konfliktlösungsablauf starten.
  - Sie kann auch ohne Antrag einer ihr untergeordneten Kontrolleinheit den Konfliklösungsablauf starten (zum Beispiel periodisch).
  - Sie beendet den Konfliktlösungsablauf und legt die endgültige Einstellung des Aktuators fest.
  - Sie steuert den entsprechenden Aktuator, d. h. die Top-CU stellt als einzige CU die Aktuatorwerte ein.
  - Sie teilt nach Abschluß eines Konfliktlösungsablaufs die eingestellten Aktuatorwerte den ihr untergeordneten Kontrolleinheiten mit.
  - Sie ist eine Kontrollgruppe.
- *Aufgaben einer Bottom-CU*:
  - Sie kann einen Antrag auf den Start eines Konfliktlösungsablaufs stellen.

    - Sie äußert evtl. im Rahmen eines Konfliktlösungsablaufs Wünsche und Forderungen bez. der Aktuatoreinstellung.
    - Sie ist eine Kontrollzelle.
- *Aufgaben einer Middle-CU*:
    - Sie leitet Anträge, die sie von ihren Söhnen empfangen hat, an ihren Vater weiter.
    - Sie leitet Vorgaben und Vorschläge, die sie von ihrem Vater empfangen hat, an diejenigen Söhne weiter, die an dem Konflikt beteiligt sind.
    - Sie kann eine lokale Konfliktlösung mit ihren Söhnen aushandeln.
    - Sie ist eine Kontrollgruppe.

Für die drei Arten von Kontrolleinheiten (Top-CU, Middle-CU, Bottom-CU) gibt es drei unterschiedliche Zustandsgraphen, die beschreiben, wie sich eine Kontrolleinheit im Rahmen eines Konfliktlösungsablaufs verhält. Dabei spiegeln diese Graphen im wesentlichen wider, wie eine Kontrolleinheit auf eine eintreffende Nachricht reagiert. Diese Zustandsgraphen geben keinen konkreten Ablauf für eine Konfliktlösung an, sondern beschreiben allgemein, welche Möglichkeiten es gibt.

In den Zustandsgraphen wird nicht darauf eingegangen, aufgrund welcher Situation eine Kontrolleinheit einen Antrag an ihren Vater stellt oder weswegen eine Top-CU von sich aus einen Konfliktlösungsablauf startet, weil dies von Konflikt zu Konflikt und von Kontrolleinheit zu Kontrolleinheit an unterschiedliche Bedingungen geknüpft sein kann.

Empfängt eine Kontrolleinheit eine Nachricht eines bestimmten Typs, so wird dies in der Beschreibung der Prädikate an den Zustandsübergängen durch einen Pfeil links von dem Nachrichttentyp gekennzeichnet; sendet eine Kontrolleinheit eine Nachricht eines bestimmten Typs, so steht der Pfeil rechts von dem Nachrichtentyp.

Die Zustandsgraphen werden in den folgenden Abschnitten angegeben. Dabei wird stets nur ein Aktuatorkonflikt betrachtet; d.h., wenn gesagt wird, daß sich eine Kontrolleinheit in dem Zustand ***X*** befindet, so ist damit stets gemeint, daß sie sich bez. eines bestimmten Konfliktes in dem Zustand ***X*** befindet.

Werden innerhalb eines Konfliktlösungsablaufs Vorschläge oder Vorgaben an untergeordnete Kontrolleinheiten geschickt, so werden diese jeweils nur an Kontrolleinheiten geschickt, die am Konflikt als Middle- oder Bottom-CU beteiligt sind.

### 4.3.5 Zustandsgraph für die Top-CU eines Konfliktes

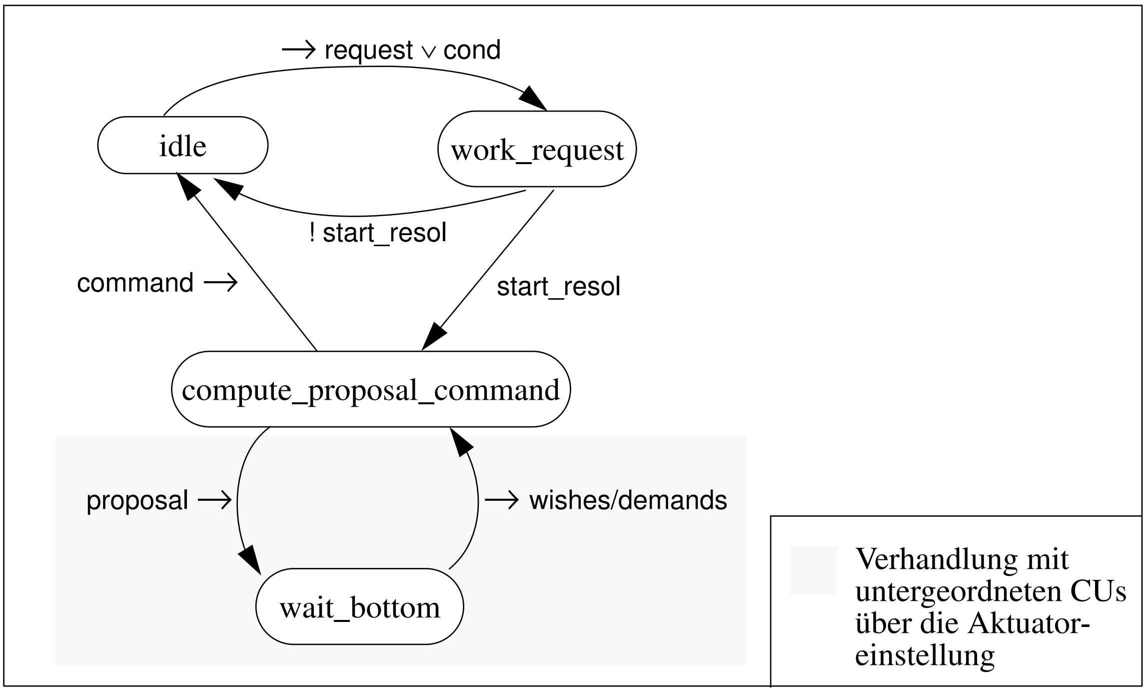


**Abb. 9: Zustandsgraph für die Top-CU**

Die Zustandsübergänge sind durch Prädikate definiert (siehe Tabelle 3).

| Name des Prädikats | Definition des Prädikats |
|---|---|
| → request | Falls die Top-CU einen Antrag eines Sohnes empfängt, wird dieses Prädikat zu true. |
| cond | Dies ist ein Prädikat, das eine Bedingung beschreibt, unter der die Top-CU von sich aus überprüft, ob ein Konfliktlösungsablauf gestartet werden soll. (Diese Bedingung kann zum Beispiel periodisch erfüllt werden.) |
| start_resol | Falls start_resol true ist (und die Top-CU im Zustand work_request), startet ein Konfliktlösungsablauf. Mit start_resol kann z. B. verhindert werden, daß unbedingt bei einem Request ein Konfliktlösungsablauf gestartet wird. |
| proposal → | Falls die Top-CU ihren Söhnen einen Vorschlag übermittelt, wird dieses Prädikat zu true. |
| → wishes/demands | Dieses Prädikat wird true, falls jeder Sohn im Rahmen eines laufenden Konfliktlösungsablaufs seinen Wunsch oder seine Forderung übermittelt hat und die Top-CU alle diese Nachrichten empfangen hat. |

**Tabelle 3: Prädikate (Top-CU)**

| Name des Prädikats | Definition des Prädikats |
|---|---|
| command $\longrightarrow$ | Falls die Top-CU ihren Söhnen eine Vorgabe übermittelt, wird dieses Prädikat zu true. |

**Tabelle 3: Prädikate (Top-CU)**

Die einzelnen Zustände sind in Tabelle 4 beschrieben.

| Zustand | Beschreibung |
|---|---|
| idle<br>(Initialzustand) | In diesem Zustand wartet die Top-CU auf eintreffende Anträge (von untergeordneten CUs) und überprüft, ob eine Situation vorliegt, die sie dazu veranlaßt, von sich aus einen Konfliktlösungsablauf zu starten. |
| work_request | In diesem Zustand entscheidet die Top-CU, ob ein Konfliktlösungsablauf gestartet wird oder nicht. Befindet sich die Top-CU in diesem Zustand, weil das Prädikat cond erfüllt wurde, so wird das Prädikat start_resol auf true gesetzt, ohne weitere Bedingungen zu betrachten und damit ein Konfliktlösungsablauf gestartet.<br>Ist die Top-CU aufgrund des Eintreffens eines Antrags in diesem Zustand, so kann der Wert von start_resol von weiteren Bedingungen abhängen, die konfliktspezifisch sein können. Es ist z. B. denkbar, daß vor dem Start eines Konfliktlösungsablaufs bestimmte Initialisierungsmaßnahmen erforderlich sind, die im Zustand work_request realisiert werden müssen. Ist die Initialisierung dann erfolgt, wird start_resol auf true gesetzt. |

**Tabelle 4: Zustände (Top-CU)**

| Zustand | Beschreibung |
|---|---|
| compute_proposal_command | In diesem Zustand kann die Top-CU Vorgaben oder Vorschläge ermitteln. Der erste Vorschlag an die Söhne wird aufgrund lokalen Wissens der Top-CU ermittelt (z. B.: die aktuelle Aktuatoreinstellung). Zur Berechnung von weiteren Vorschlägen bzw. Vorgaben können die von den Söhnen empfangenen Wünsche bzw. Forderungen verarbeitet werden.<br>Wenn der Wert für eine Vorgabe bestimmt wurde, wird entschieden, ob die Änderung, die durch die neue Vorgabe im Vergleich zur aktuellen Einstellung entsteht, so gravierend ist, daß sie sich in einer Änderung der Aktuatoreinstellung niederschlagen soll. Ist dies nicht der Fall, so wird als Vorgabe die aktuelle Einstellung versendet und der Aktuator nicht neu eingestellt. Ansonsten wird in diesem Zustand der Aktuator neu eingestellt und danach die neue Einstellung als Vorgabe versendet.<br>Beispiel: Als Ergebnis des Konfliktlösungsablaufes wird ein Öffnungsgrad von 11% für den oberen Fensterflügel ermittelt. Die aktuelle Einstellung beträgt 10%. In diesem Fall würde die aktuelle Einstellung von 10% beibehalten und als Vorgabe an die Söhne gesendet werden, da 1% Abweichung eine marginale Änderung des Öffnungsgrades ist. |
| wait_bottom | In diesem Zustand wartet die Top-CU solange, bis jeder Sohn seinen Wunsch oder seine Forderung gesendet und sie sie empfangen hat. Diese Werte speichert die Top-CU. |

**Tabelle 4: Zustände (Top-CU)**

Man beachte, daß es ist nicht möglich ist, daß eine Top-CU Vorschläge oder Vorgaben empfängt.

Bei der Beschreibung einer konkreten Top-CU müssen vor allem folgende Aspekte genauer beschrieben werden:

- cond
- start_resol
- Verhalten im Zustand work_request
- Verhalten im Zustand compute_proposal_command

### 4.3.6 Zustandsgraph für eine Bottom-CU eines Konfliktes

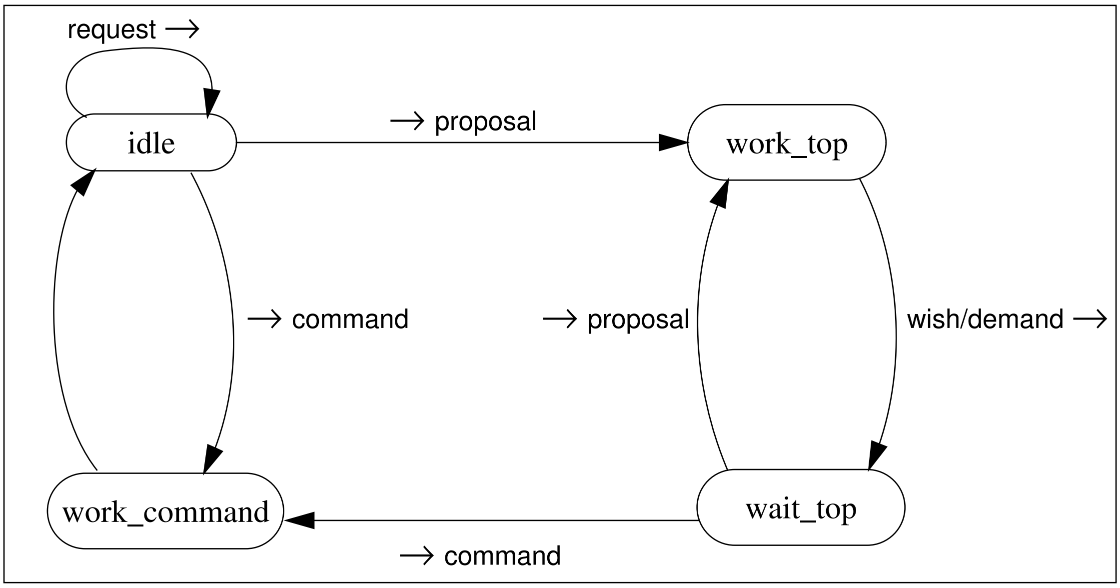


**Abb. 10: Zustandsgraph für eine Botton-CU**

Die Zustandsübergänge sind durch Prädikate definiert (siehe Tabelle 5).

| Name des Prädikats | Definition des Prädikats |
|---|---|
| → proposal | Die Bottom-CU empfängt einen Vorschlag des Vaters. |
| → command | Die Bottom-CU empfängt eine Vorgabe des Vaters. |
| wish/demand → | Die Bottom-CU sendet einen Wunsch oder eine Forderung an ihren Vater. |
| request → | Die Bottom-CU sendet einen Antrag an ihren Vater. |

**Tabelle 5: Prädikate (Bottom-CU)**

Der Übergang vom Zustand work_command in den Zustand idle findet unbedingt statt.

Die einzelnen Zustände sind in Tabelle 6 beschrieben.

| Zustand | Beschreibung |
|---|---|
| idle<br>(Initialzustand) | In diesem Zustand wartet die CU auf eintreffende Nachrichten (Vorschläge oder Vorgaben) ihres Vaters.<br>Nur in diesem Zustand darf die Bottom-CU einen Antrag (Request) an ihren Vater senden |
| work_top | In diesem Zustand verarbeitet die CU einen Vorschlag ihres Vaters und berechnet einen Wunsch oder eine Forderung. |

**Tabelle 6: Zustände (Bottom-CU)**

| Zustand | Beschreibung |
|---|---|
| work_command | In diesem Zustand verarbeitet die CU eine Vorgabe ihres Vaters. |
| wait_top | In diesem Zustand wartet die CU auf eintreffende Nachrichten (Vorschläge oder Vorgaben) ihres Vaters. |

**Tabelle 6: Zustände (Bottom-CU)**

Man beachte, daß es nicht möglich ist, daß eine Bottom-CU Anträge, Wünsche oder Forderungen empfängt. Selbst kann sie keine Vorschläge oder Vorgaben senden.

Bei der Beschreibung einer konkreten Bottom-CU müssen vor allem folgende Aspekte genauer beschrieben werden:

- Verhalten im Zustand work_top, d.h., wie ein Wunsch bzw. ein Forderung berechnet wird
- Verhalten im Zustand work_command
- Verhalten im Zustand idle, d.h., wann eine Request-Nachricht gesendet wird

### 4.3.7 Zustandsgraph für eine Middle-CU eines Konfliktes

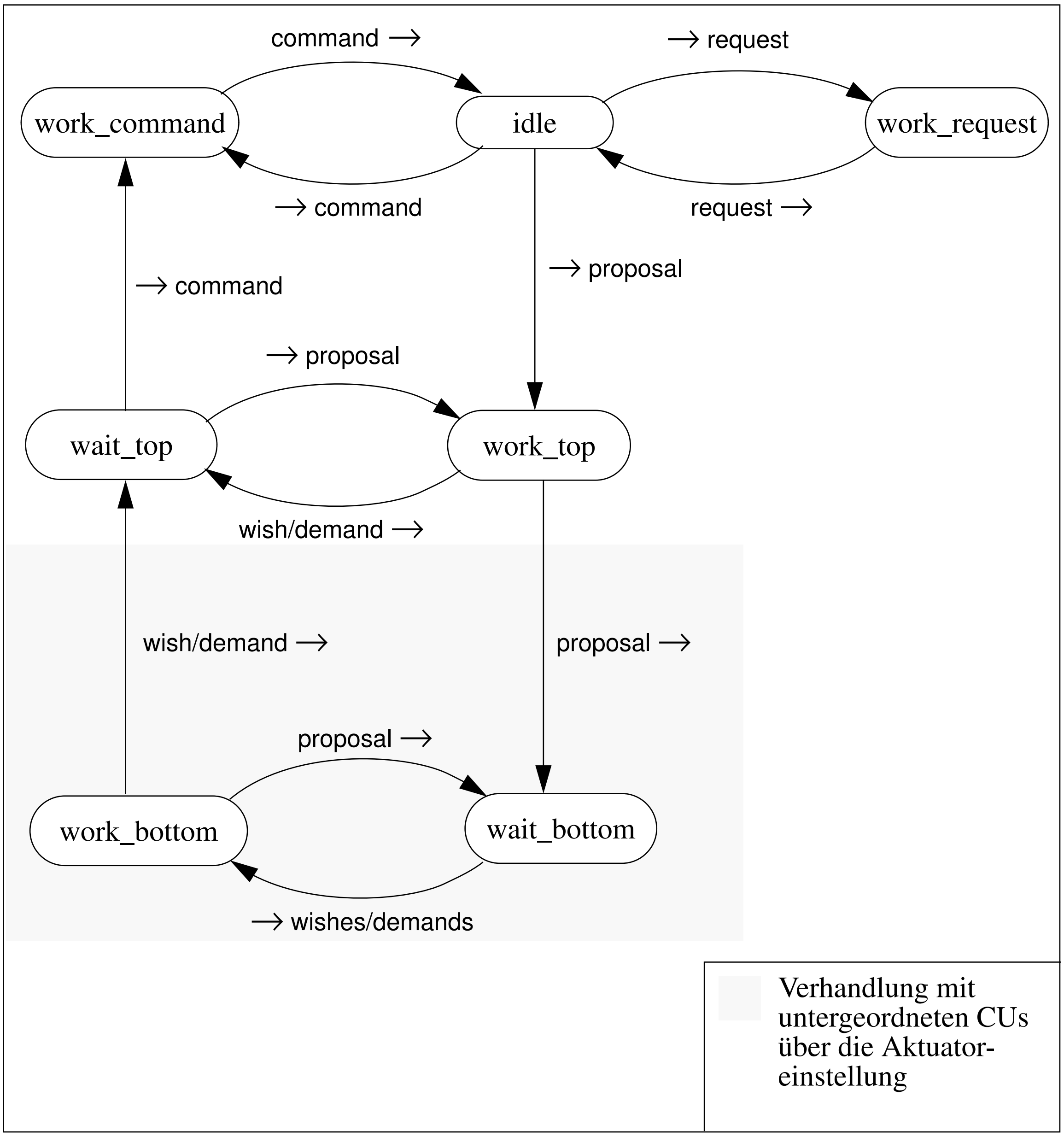


**Abb. 11: Zustandsgraph für eine Middle-CU**

Die Zustandsübergänge sind durch Prädikate definiert (siehe Tabelle 7).

| Name des Prädikats | Definition des Prädikats |
|---|---|
| ⟶ request | Die Middle-CU empfängt einen Antrag eines Sohnes. |
| request⟶ | Die Middle-CU übermittelt ihrem Vater einen Antrag. |
| ⟶ wishes/demands | Dieses Prädikat wird true, falls jeder Sohn seinen Wunsch oder seine Forderung übermittelt hat und die Middle-CU alle diese Nachrichten empfangen hat. |

**Tabelle 7: Prädikate (Middle-CU)**

| Name des Prädikats | Definition des Prädikats |
|---|---|
| $\longrightarrow$ proposal | Die Middle-CU empfängt einen Vorschlag ihres Vaters. |
| proposal $\longrightarrow$ | Die Middle-CU übermittelt ihren Söhnen einen Vorschlag. |
| $\longrightarrow$ command | Die Middle-CU empfängt eine Vorgabe ihres Vaters. |
| command $\longrightarrow$ | Die Middle-CU übermittelt ihren Söhnen eine Vorgabe. |
| wish/demand $\longrightarrow$ | Die Middle-CU übermittelt ihrem Vater einen Wunsch oder eine Forderung. |

**Tabelle 7: Prädikate (Middle-CU)**

Der Übergang von dem Zustand work_command in den Zustand idle findet unbedingt statt.

Die einzelnen Zustände sind in Tabelle 8 beschrieben.

| Zustand | Beschreibung |
|---|---|
| idle (Initialzustand) | In diesem Zustand wartet die Middle-CU auf eintreffende Nachrichten (Vorschläge oder Vorgaben ihres Vaters, Anträge ihrer Söhne). |
| work_request | In diesem Zustand leitet die Middle-CU einen eingegangenen Antrag eines Sohnes an den Vater weiter. |
| work_top | In diesem Zustand verarbeitet die Middle-CU einen Vorschlag ihres Vaters und berechnet entweder einen Wunsch oder eine Forderung, die sie dann an den Vater sendet, oder einen Vorschlag, den sie an ihre Söhne sendet. Dies ist abhängig davon, ob mit untergeordneten CUs verhandelt werden soll oder nicht. |
| work_command | In diesem Zustand verarbeitet die Middle-CU eine Vorgabe ihres Vaters. Eine Vorgabe wird in jedem Fall an die Söhne weitergegeben werden, damit diese über die aktuelle Aktuatoreinstellung informiert sind. |
| wait_top | In diesem Zustand wartet die Middle-CU auf eintreffende Nachrichten (Vorschläge oder Vorgaben) ihres Vaters. |
| wait_bottom | In diesem Zustand wartet die Middle-CU auf Wünsche und Forderungen ihrer Söhne. |
| work_bottom | In diesem Zustand verarbeitet die Middle-CU die von ihren Söhnen empfangenen Nachrichten und berechnet entweder einen Vorschlag, den sie dann an ihre Söhne sendet, oder einen Wunsch/Forderung, die sie an den Vater sendet. |

**Tabelle 8: Zustände (Middle-CU)**

Bei der Beschreibung einer konkreten Middle-CU müssen vor allem folgende Aspekte genauer beschrieben werden:

- Verhalten im Zustand work_command
- Verhalten im Zustand work_top
- Verhalten im Zustand work_bottom

### 4.3.8 Zentralistische und verhandlungsorientierte Konfliktlösungsabläufe

Die Beschreibung der Zustandsgraphen von Top-, Middle- und Bottom-CUs erlaubt viele mögliche Konfliktlösungsabläufe, die durch genauere Beschreibungen der Prädikate und des Verhaltens in den Zuständen festgelegt werden können.

Grob lassen sich zwei Richtungen von Konfliktlösungsabläufen unterscheiden:

- *Zentralistischer Konfliktlösungsablauf.* Es wird eine Aktuatorstellung ermittelt, ohne daß mit untergeordneten Kontrolleinheiten bez. der Aktuatorstellung verhandelt wird. Dies kann z. B. mit Hilfe von Prioritäten geschehen.
- *Verhandlungsorientierter Konfliktlösungsablauf.* Es wird eine Aktuatorstellung ermittelt, indem mit den untergeordneten Kontrolleinheiten über eine geeignete Einstellung verhandelt wird. Hierbei werden Vorschläge an die untergeordneten CUs gesendet, auf die diese mit Wünschen und Forderungen reagieren können. Anhand der Wünsche und Forderungen der untergeordneten CUs wird eine neue Aktuatoreinstellung ermittelt. Zur Ermittlung einer Aktuatoreinstellung kann es mehrere “Verhandlungsrunden” geben. Die Zustände, mit denen verhandlungsorientierte Konfliktlösungsabläufe realisiert werden können, sind in Abbildung 9 und Abbildung 11 schattiert unterlegt. Nachrichten müssen eindeutig gekennzeichnet (z. B. mittels IDs) sein, so daß klar ist, welche Wünsche/Forderungen zu welchem Vorschlag gehören. Dies ist insbesondere bei mehreren “Verhandlungsrunden” wichtig.

Eine Kombination dieser Ansätze innerhalb eines Konfliktlösungsablaufs ist möglich, da diese Ansätze auf mehrere Hierarchieebenen ausgedehnt werden können. Es kann z. B. bis zu einer bestimmten Ebene in der Hierarchie über einen Aktuatorwert verhandelt werden, und ab dieser Ebene wird nicht mehr mit untergeordneten CUs verhandelt, sondern direkt ein Wert bestimmt und an den Vater weitergeleitet.

Alle möglichen Konfliktlösungsabläufe gleichen sich hinsichtlich ihres Starts und ihres Endes:

- *Start eines Konfliktlösungsablaufs.* Jeder Konfliktlösungsablauf hat zur Voraussetzung, daß ein Antrag an die Top-CU geschickt wird oder die Bedingung cond zu true ausgewertet wurde. Darauf muß die Top-CU entscheiden, ob ein Konfliktlösungsablauf in der derzeit vorliegenden Situation gestartet wird.
  Falls ein neuer Konfliktlösungsablauf gestartet wird, kann die Top-CU aufgrund ihrer lokalen Informationen einen neuen Aktuatorwert bestimmen oder mit den ihr untergeordneten CUs über eine Einstellung “verhandeln”, wobei als erster Vorschlag der aktuelle Aktuatorwert an die Söhne geleitet wird.
- *Ende eines Konfliktlösungsablaufs.* Ein Konfliktlösungsablauf endet stets damit, daß die Top-CU der Aktuator einstellt und danach allen beteiligten Kontrolleinheiten Vorgabenachrichten schickt, die die endgültige Einstellung des Aktuators enthalten.

Die Entscheidung, ob ein eher zentralistischer oder ein eher verhandlungsorientierter Konfliktlösungsansatz gewählt wird, ist wesentlich von der Art eines konkreten Konflikts abhängig. Tabelle 9 zeigt Vor- und Nachteile der Ansätze.

<table>
<tr><th></th><th>Zentralistischer Ansatz</th><th>Verhandlungsorientierter Ansatz</th></tr>
<tr><td rowspan="4">Vorteile</td><td colspan="2">Ansatz auf mehrere Hierarchieebenen erweiterbar</td></tr>
<tr><td colspan="2">Ansatz unabhängig von einem konkreten Konflikt formuliert</td></tr>
<tr><td>Einfacher</td><td rowspan="2">“demokratische” Lösungsfindung (durch Aushandeln einer Lösung)</td></tr>
<tr><td>Schneller (eignet sich daher besser für zeitkritische Konfliktlösungen)</td></tr>
<tr><td rowspan="2">Nachteile</td><td rowspan="2">“diktatorische” Lösungsfindung</td><td>Aufwendiger</td></tr>
<tr><td>Langsamer</td></tr>
</table>

**Tabelle 9: Vor- und Nachteile**

### 4.3.9 Integration von Benutzervorgaben

In diesem Abschnitt werden Benutzervorgaben betrachtet, in denen der Benutzer einen konkreten Aktuatorwert einstellen will.

Benutzervorgaben haben höchste Priorität. Sie können vom Kontrollsystem nicht geändert werden, bis der Raum nicht mehr als benutzt gilt (d. h., bis das in der Anforderungsbeschreibung definierte Prädikat room_used den Wert “false” annimmt).

Macht der Benutzer bestimmte Vorgaben bez. der Einstellung von Aktuatorwerten, so werden die von der Eingabeeinheit an die diesen Aktuator steuernde Kontrolleinheit (Top-CU) weitergeleitet (diese ist eindeutig). Es gibt dann zwei Möglichkeiten:

1. Für diesen Aktuatorkonflikt läuft gerade keine Konfliktlösung. Dann kann die Benutzervorgabe direkt als Aktuatorwert eingestellt werden und danach werden Vorgabenachrichten an alle untergeordneten CUs verschickt, damit diese den aktuellen Wert kennen.

2. Für diesen Aktuatorkonflikt läuft gerade eine Konfliktlösung. Dieser Konfliktlösungsablauf wird fortgesetzt bis zu dem Zeitpunkt, an dem die Top-CU eine endgültige Entscheidung über die Einstellung trifft. Anstelle einer berechneten Einstellung wird die Benutzervorgabe eingestellt und danach werden Vorgabe-Nachrichten an alle untergeordneten CUs verschickt, damit diese den aktuellen Wert kennen.

### 4.3.10 Übermittlung von Aktuatorwerten zwischen Kontrolleinheiten

Im Konfliktlösungsablauf enthalten die Nachrichtentypen Vorschlag, Vorgabe, Wunsch und Forderung Information bez. der Einstellung eines Aktuators. Diese Information wird einheitlich in Form von *Intervallen* übermittelt, wobei den Aktuatorwerten eines Aktuators jeweils eine totale Ordnung zugrunde gelegt wird, bez. derer man sie miteinander vergleichen kann.

Im Konfliktlösungsablauf kann die Übermittlung von Aktuatorwerten mittels Intervallen auf drei Arten geschehen:

1. *Bestimmtes Intervall* ([X,Y]; $X \leq Y$). Ein bestimmtes Intervall erlaubt die Übermittlung beliebiger Teilintervalle. Dies kann z. B. dann der Fall sein, wenn eine CU nach einem Einstellungswunsch gefragt wird und es für ihre Ziele ausreicht, daß die Einstellung des Aktuatorwerts in einem bestimmten Intervall liegt. Dann gibt die CU dieses Intervall in der Wunschnachricht zurück.

Die folgenden zwei Arten der Übermittlung von Aktuatorwerten sind Spezialfälle von bestimmten Intervallen:

2. *Minimales Intervall* ([X,X]). Ein minimales Intervall wird verwendet, falls eine konkrete Aktuatoreinstellung übermittelt werden soll. Vorgaben beinhalten immer minimale Intervalle, da sie einen bestimmten Aktuatorwert übermitteln.
3. *Maximales Intervall* ([minimaler_Aktuatorwert,maximaler_Aktuatorwert]). Ein maximales Intervall wird verwendet, falls keine Einschränkungen an die Aktuatoreinstellung gemacht werden. Dies kann z. B. dann der Fall sein, wenn eine CU nach einem Einstellungswunsch gefragt wird, aber gerade keinen hat. Dann gibt die CU in der Wunschnachricht ein maximales Intervall zurück.

### 4.3.11 Konfliktlösungsszenarien

In diesem Abschnitt werden zwei Szenarien angegeben, mit denen mögliche Konfliktlösungsabläufe dargestellt werden sollen. Die Szenarien dienen der Verdeutlichung des Unterschiedes zwischen einem zentralistischen und einem verhandlungsorientierten Konfliktlösungsablauf. Bei den Szenarien handelt es sich um vereinfachte, beispielhafte Abläufe, die nicht als Ausgangspunkt für einen konkreten Entwurf gedacht sind.

Die Ausgangssituation ist in Abbildung 12 dargestellt, wobei ein Konflikt zwischen einer Temperaturkontrollzelle und einer Lüftungskontrollzelle betrachtet wird, die beide den oberen Fensterflügel eines Büroraums steuern wollen.[1] Die koordinierende Kontrolleinheit ist die Klimakontrollgruppe. Es sei keine Kontrolleinheit auf höherer Ebene zu berücksichtigen, d. h. der Gruppencontroller Klima ist die Top-CU. In den Szenarien werden aus Gründen der Vereinfachung keine Forderungen verschickt, sondern nur Wünsche.

1. In der Einschränkung des Kontrollsystems existiert ein solcher Konflikt auch, an ihm ist jedoch zusätzlich die Kontrollzelle “Safety” beteiligt.

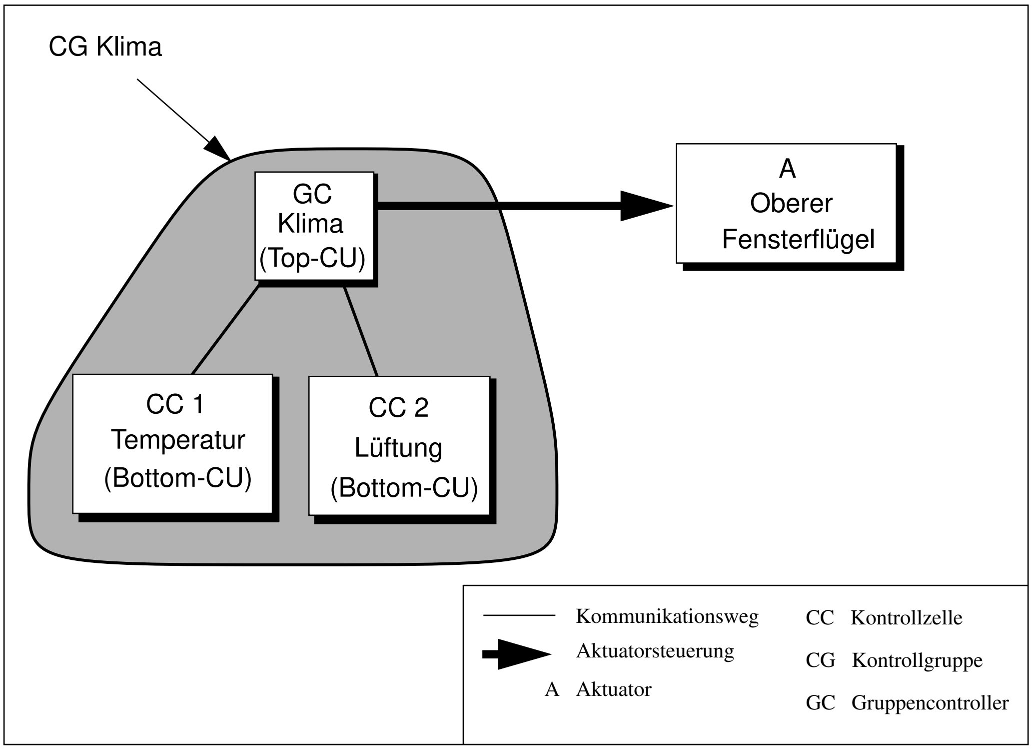


**Abb. 12: Ausgangssituation**

- *Szenario 1: Beispiel eines zentralistischen Konfliktlösungsablaufs*. Falls der GC Klima einen Antrag auf Konfliktlösung erhalten hat, schickt er die aktuelle Einstellung an seine Söhne, die daraufhin ihre Wünsche äußern. Dies ist erforderlich, da bei einer zentralistischen Konfliktlösung zumindest ein Wunsch der Top-CU vorliegen sollte. Danach trifft der GC Klima eine Entscheidung bez. der Aktuatoreinstellung, teilt diese den Söhnen als Vorgabe mit und steuert den Aktuator mit dem entsprechenden Wert. Ein solches Szenario könnte aus folgenden Schritten bestehen:

  (a) Die CC Lüftung berechnet eine neue Wunscheinstellung des oberen Fensterflügels: [80%, 100%].

  (b) Die CC Lüftung sendet eine Antragsnachricht an den GC Klima (man beachte, daß die Antragsnachricht keinen konkreten Wert hinsichtlich der Einstellung des Aktuators enthält).

  (c) Der GC Klima stellt fest, daß ein Konfliktlösungsablauf gestartet werden kann, und bestimmt als ersten Vorschlag die aktuelle Einstellung: [0%, 0%].

  (d) Diese aktuelle Einstellung ([0%, 0%]) wird als Vorschlagsnachricht vom GC Klima an die CC Lüftung und an die CC Temperatur gesendet.

  (e) Die beiden Söhne empfangen den Vorschlag und berechnen ihre Wünsche: CC Temperatur wünscht [0%, 0%], CC Lüftung wünscht [80%, 100%].

  (f) Die beiden Söhne senden ihre Wünsche als Wunschnachrichten an den Gruppencontroller: [0%, 0%] von der CC Temperatur und [80%, 100%] von der CC Lüftung.

(g) Der GC Klima empfängt die Wünsche. Für den GC Klima hat der Einstellungswunsch der CC Lüftung in der aktuellen Situation Priorität über den Wunsch der CC Temperatur. Daher entscheidet sich die GC Klima für den Aktuatorwert 80%. Der obere Fensterflügel wird zu 80% geöffnet.

(h) Der GC Klima sendet das Ergebnis der Konfliktlösung in Vorgabenachrichten an die betroffenen Kontrollzellen, d. h., der GC Klima sendet an die Kontrollzellen Temperatur und Lüftung, daß der obere Fensterflügel auf 80% eingestellt ist.

- *Szenario 2: Beispiel eines verhandlungsorientierten Konfliktlösungsablaufs*. Falls der GC Klima einen Antrag auf Konfliktlösung erhalten hat, schickt er die aktuelle Einstellung an seine Söhne, die daraufhin ihre Wünsche äußern. Hat der GC Klima zum ersten Mal alle Wünsche seiner Söhne empfangen und gibt es eine nichtleere Schnittmenge zwischen den angegebenen Intervallen, so wählt der GC Klima einen Wert aus diesem Schnitt aus und sendet diesen als Vorgabe an die Söhne. Ist die Schnittmenge der Intervalle leer, bestimmt der GC Klima einen Kompromiß, den er den Söhnen als Vorschlag sendet. Hat der GC Klima zum zweiten Mal alle Wünsche seiner Söhne empfangen, so bestimmt er einen Kompromiß, den er den Söhnen als Vorgabe sendet und veranlaßt eine entsprechende Einstellung des Aktuators. Ein solches Szenario könnte aus folgenden Schritten bestehen:

(a) Die CC Lüftung berechnet eine neue Wunscheinstellung des oberen Fensterflügels: [80%, 100%].

(b) Die CC Lüftung sendet eine Antragsnachricht an den GC Klima (man beachte, daß die Antragsnachricht keinen konkreten Wert hinsichtlich der Einstellung des Aktuators enthält).

(c) Der GC Klima stellt fest, daß ein Konfliktlösungsablauf gestartet werden kann, und bestimmt als ersten Vorschlag die aktuelle Einstellung: [0%, 0%].

(d) Diese aktuelle Einstellung ([0%, 0%]) wird als Vorschlagsnachricht vom GC Klima an die CC Lüftung und an die CC Temperatur gesendet.

(e) Die beiden Söhne empfangen den Vorschlag und berechnen ihre Wünsche: CC Temperatur wünscht [0%, 0%], CC Lüftung wünscht [80%, 100%].

(f) Die beiden Söhne senden ihre Wünsche als Wunschnachrichten an den Gruppencontroller: [0%, 0%] von der CC Temperatur und [80%, 100%] von der CC Lüftung.

(g) Der GC Klima empfängt die Wünsche und stellt fest, daß sich die Intervalle nicht überschneiden. Der GC Klima berechnet daraufhin einen Kompromiß. Dies kann beispielweise durch Mittelwertbildung, aber auch mit Hilfe komplexerer Algorithmen geschehen. Der gefundene Kompromiß ist das Intervall [40%, 60%].

(h) Der Kompromiß wird in Form von Vorschlagsnachrichten allen Söhnen mitgeteilt, d. h. der CC Temperatur und der CC Lüftung wird als Vorschlag das Intervall [40%, 60%] mitgeteilt.

(i) Die Kontrollzellen prüfen, ob der Wert für sie akzeptabel ist, indem beispielsweise andere Steuerungsstrategien benutzt werden. Falls dem so ist, senden sie das gleiche Intervall wieder als Wunsch an den GC Klima.

Andernfalls können sie gegenüber dem GC Klima einen neuen Wunsch äußern. Die CC Lüftung ist mit dem Komromiß nicht einverstanden, sie beharrt auf ihrer Wunsch-Einstellung ([80%,100%]), während die Temperatur-Kontrollzelle den Vorschlag akzeptiert.

(j) Die Wünsche werden dem GC Klima in Wunschnachrichten mitgeteilt, die zusätzlich die Kennung der in Schritt (h) übermittelten Vorschlagsnachricht tragen (um eine eindeutige Zuordnung der Wunschnachrichten zu den Vorschlagsnachrichten zu gewährleisten). Die CC Temperatur sendet das vorgeschlagene Intervall [40%, 60%] in einer Wunschnachricht zurück, während die CC Lüftung das Intervall [80%, 100%] in einer Wunschnachricht an den GC Klima schickt.

(k) Der Gruppencontroller kann anhand der Kennung die neuen Wunschnachrichten den von ihm selbst ausgesendeten Vorschlagsnachrichten zuordnen und als Antwort in der aktuellen Abgleichrunde erkennen. Er entscheidet sich für die untere Intervallgrenze des von der CC Lüftung geforderten Intervalls als neue Einstellung. Der obere Fensterflügel wird zu 80% geöffnet.

(l) Der GC Klima sendet das Ergebnis der Konfliktlösung in Vorgabenachrichten an die betroffenen Kontrollzellen, d. h., der GC Klima sendet an die Kontrollzellen Temperatur und Lüftung, daß der obere Fensterflügel auf 80% eingestellt ist.

## 4.4 Methodik zur Behandlung von Vorgabenkonflikten

Vorgabenkonflikte entstehen, wenn eine Kontrolleinheit nicht die Einstellung eines konkreten Aktuators beeinflussen möchte, sondern den Zielwert (Vorgabewert) für eine Kontrollgröße (z. B.: Licht, Temperatur) setzen möchte, die von einer anderen Kontrolleinheit überwacht wird. Vorgabekonflikte werden analog zu Aktuatorkonflikten behandelt, d. h., es werden Top-CU, Middle-CUs und Bottom-CUs bez. eines Vorgabekonflikts identifiziert, und bei Wunsch auf Änderung werden Anträge an die Top-CU gestellt, die dann einen Konfliktlösungsablauf starten kann. Der wesentliche Unterschied ist also, daß nicht ein Aktuatorwert Gegenstand der Konfliktlösung ist, sondern ein Vorgabewert. Vorgabewerte Vorgabewerte sind Top-CUs zugeordnet, in denen sie gespeichert sind.

Der Konfliktlösungsablauf für Vorgabewerte ist dahingehend eingeschränkt, daß keine Verhandlungen bez. eines Vorgabewerts erlaubt sind[1]. Die der Top-CU untergeordneten CUs können einmal nach ihrem Wunsch gefragt werden, und dann entscheidet die Top-CU, welcher Vorgabewert gültig ist. Diesen teilt sie den untergeordneten CUs mit. Der Ablauf läßt sich allgemein durch folgende Schritte beschreiben:

1. Eine der Top-CU untergeordnete Kontrolleinheit wünscht eine neue Einstellung eines Vorgabewerts. Sie sendet einen Antrag an ihren Vater, der Stufe für Stufe in der Kontrollhierarchie an die für diesen Vorgabewert zuständige Top-CU weitergeleitet wird.
2. Die Top-CU hat den letzten Vorgabewert gespeichert und teilt diesen den untergeordneten CUs als Vorschlag mit.

---

1. Dies ist eine Entwurfsentscheidung, die getroffen wurde, um die Behandlung von Vorgabekonflikten einfach zu halten.

3. Die untergeordneten CUs senden ihre Wünsche/Forderungen für den Vorgabewert an die Top-CU.
4. In der Top-CU wird über den einzustellenden Vorgabewert entschieden. Der festgelegte Vorgabewert wird den untergeordneten CUs als Vorgabe zugesandt.
5. Die untergeordneten CUs richten sich in ihrer weiteren Regelung nach dem Vorgabewert, den sie erhalten haben. Die Änderung eines Vorgabewerts kann zu neuen Wünschen bez. der Einstellung von Aktuatoren und somit zu neuen Konfliktabläufen von Aktuatorkonflikten führen.

# 5 Fehlerbehandlung

In diesem Abschnitt wird ein Konzept für die Fehlerbehandlung vorgestellt. Dabei sollen unter anderem folgende Fragen beantwortet werden:

- Wo können Fehler auftreten?
- Welche Arten von Fehlern können auftreten?
- Wie können Fehler erkannt werden?
- Welche Fehler können erkannt werden, welche nicht?
- Auf welche Fehler wird wie reagiert?

Im Rahmen dieses Fehlerbehandlungskonzeptes wird zunächst eine Fehlerklassifikation vorgenommen, durch die beschrieben wird, welche Arten von Fehlern betrachtet werden. Anschließend wird ein Fehlermodell vorgestellt, mit dessen Hilfe unter anderen gezeigt wird, wie Fehler innerhalb des Kontrollsystems behandelt werden. Wesentliche Eigenschaft dieses Fehlerbehandlungskonzeptes ist, daß es komponentenorientiert ist, d.h. die Betrachtungen erfolgen stets aus der lokalen Sicht einer Komponente.

## 5.1 Fehlerklassen

Die Fehler, die in einem Kontrollsystem auftreten können, werden in drei Klassen aufgeteilt. Die Zuordnung eines Fehlers zu einer der Klassen erfolgt aus der Sicht einer Komponente des Kontrollsystems und orientiert sich im wesentlichen daran, wie ein Fehler von einer Komponente festgestellt wird.

- *Originäre Fehler*:
  Dies sind aus der Sicht einer Komponente die Fehler, die sie selbst lokal erkennt. Hierzu gehören zum Beispiel Hardware-, Kommunikations-, Betriebssystem- und Implementierungsfehler, die eine Komponente selbst feststellt.
- *Dienstausfall-Fehler*:
  Dies sind aus der Sicht einer Komponente die Fehler, die ihr von anderen Komponenten gemeldet werden, mit denen sie in einer „benutzt”-Relation steht. Dienstausfall-Fehler erkennt eine Komponente also nicht selbständig, sondern sie werden einer Komponente von einer anderen Komponente mitgeteilt.
  Dies ist zum Beispiel der Fall, wenn in einem Sensor ein originärer Fehler festgestellt wurde und der Sensor einer Kontrolleinheit mitteilt, daß er nur noch eine Dienstleistung mit eingeschränkter Qualität zur Verfügung stellen kann. Aus der Sicht der Kontrolleinheit handelt es sich um einen Dienstausfall-Fehler.
- *Unerkannte Fehler*:
  Dies sind Fehler, die im Kontrollsystem auftreten, jedoch von keiner Komponente des Kontrollsystems erkannt werden.

Bemerkungen:

- In einer Komponente des Kontrollsystems, die keine Funktionalitäten anderer Komponenten benutzt, können nur originäre oder unerkannte Fehler auftreten.
- Ein originärer Fehler liegt zum Beispiel auch vor, wenn eine Komponente mit Hilfe eines Plausibilitätstests feststellt, daß eine andere untergeordnete Komponete sich fehlerhaft verhält, ohne daß dies die untergeordnete Komponente erkennt. An diesem Beispiel wird deutlich, daß es für die Zuordnung eines Fehlers zu einer der Klassen wichtig ist, aus der Sicht

welcher Komponente man die Situation betrachtet. Aus der Sicht der untergeordneten Komponente handelt es sich um einen unerkannten Fehler.

## 5.2 Fehlermodell

Das Fehlermodell für den Kontrollsystem-Prototypen ist wie die Einteilung der Fehlerklassen komponentenorientiert. Grundlage für das Fehlermodell ist die Idee, daß jede Komponente eine bestimmte Menge von Funktionalitäten zur Verfügung stellt. Für jede einzelne Funktionalität können zusätzlich verschiedene Qualitäten unterschieden werden, mit denen die Funktionalität zur Verfügung gestellt wird. Funktionalitäten, die sich nur in ihrer Qualität unterscheiden, werden in einer Dienstklasse zusammengefaßt. Stellt eine Komponente eine Funktionalität mit einer hohen Qualität zur Verfügung, so impliziert dies, daß auch die niederwertigen Qualitäten der entsprechenden Dienstklasse zur Verfügung stehen. Welche Qualitäten einer Funktionalität zur Verfügung stehen, hängt von der konkreten Situation ab und wird durch den *Fehlerzustand* einer Komponente beschrieben.

Um den Fehlerzustand einer Komponente beschreiben zu können, müssen folgende Informationen verfügbar sein, d.h. es müssen die folgenden Fragen beantwortet werden können:

- Was benötigt eine Komponente, um eine Funktionalität mit einer bestimmten Qualität nach außen verfügbar zu machen?
  Dieses Wissen ist unabhängig von der konkreten Situation, in der sich das Kontrollsystem befindet.
- Welche Funktionalitäten von (Sub-)Komponenten (z.B. Sensoren, Aktuatoren) stehen mit welcher Qualität zur Verfügung?
  Dies ist das situationsabhängige Wissen, das beschreibt, was augenblicklich benutzt werden kann.
- Welche Funktionalitäten kann die Komponente mit welcher Qualität zur Verfügung stellen?
  Dies ist die Schlußfolgerung, die jede Komponente durch Vergleich der Informationen aus den ersten beiden Punkten ziehen muß. Dies ist situationsabhängiges Wissen, das beschreibt, was augenblicklich von der Komponente an Funktionalitäten bereitgestellt werden kann.

Aufbauend auf diesen Informationen wird für jede Komponente des Kontrollsystems nun ein Fehler-Zustandsgraph angegeben, bei dem die Zustände beschreiben, welche Funktionalitäten augenblicklich von der Komponente bereitgestellt werden (Wissen aus Punkt c)). Ein Übergang von einem Zustand in einen anderen kann stattfinden, wenn sich die Menge an Informationen, die sich aus Punkt b) ergibt, verändert. Es ist also auch notwendig, daß jede Kontrolleinheit die Informationen, die b) zugeordnet werden, speichert.
Es ergibt sich also folgender genereller Ablauf bei einer Fehlerbehandlung:

Nachdem eine Kontrolleinheit einen originären Fehler festgestellt hat oder ihr ein Dienstausfall-Fehler mitgeteilt wurde, überprüft sie, welche Funktionalitäten mit welcher Qualität sie noch zur Verfügung stellen kann. Dabei können zum Beispiel folgende Folgerungen gezogen werden:

- Der Fehler ist nicht relevant, weil zum Beispiel ein ausgefallener Aktuator zur Zeit nicht benötigt wird.

- Der Fehler kann intern, d.h. durch die Kontrolleinheit maskiert werden, weil zum Beispiel Alternativlösungen oder ein zweiter funktionsfähiger Aktuator existieren.
- Der Fehler reduziert die nach außen zur Verfügung gestellte Dienstqualität.

Das Zurücksetzen von Fehlern kann auf zwei Arten erfolgen. Entweder geschieht dies dadurch, daß eine Kontrolleinheit periodisch versucht, die Funktionalität einer untegeordneten Komponente zu verwenden, bis sie feststellt, daß die untergeordnete Komponenete einen bisher nicht zur Verfügung gestellten Dienst wieder anbietet. Die zweite Möglichkeit ist, daß von außen die Mitteilung kommt, daß eine Fehlersituation nicht mehr gegeben ist. In beiden Fällen findet wie beim Feststellen eines Fehlers eine Neubewertung der Situation statt, die zu einer Veränderung des Fehlerzustandes führen kann, d.h. die Kontrolleinheit kann Funktionalitäten wieder zur Verfügung stellen.

# 6 Initialisierung und Konfigurierung

In diesem Abschnitt wird die Konfigurierung und Initialisierung des Kontrollsystems behandelt.

## 6.1 Initialisierung

Unter Initialisierung wird das Festlegen von Startwerten und Startsituationen verstanden, d.h. bestimmte Variablen erhalten einen Initialwert zugewiesen, und für Zustandsgraphen wird ein Initialzustand festgelegt. Die Initialisierung kann auf mehreren Ebenen erfolgen, je nachdem, für welchen Bereich eine Variable oder ein Zustand sichtbar ist. Dabei wird die Initialisierung lokal durch die jeweilige Kontrolleinheit vorgenommen, weil hierzu lokales Wissen, wie zum Beispiel Implementierungsdetails, benötigt wird. Die Initialisierung ist ein einmaliger Vorgang, der beim Start des Systems erfolgt.

## 6.2 Konfigurierung

Unter Konfigurierung wird der Aufbau des Kontrollsystems verstanden, d.h. das Erzeugen der einzelnen Instanzen von Kontrolleinheiten, logischen Aktuatoren und Sensoren sowie das Vernetzen dieser Komponenten.
Bei dem zu entwickelnden Prototypen erfolgt die Konfigurierung nur einmal beim Start des Systems. Bei späteren Realisierungen des Kontrollsystems sollte es auch möglich sein, das System dynamisch während des Laufes zu (re)konfigurieren.

Es wird folgender Ablauf für die Konfiguration des Systems vorgeschlagen. Wesentlich bei diesem Konzept ist, daß das Konfigurieren in die Hierarchie des Kontrollsystems eingebettet wird.
Jeder Kontrolleinheit ist eine Konfigurationskomponente zugeordnet. Sie besitzt Informationen darüber, welche Umgebung die Kontrolleinheit benötigt, und enthält einen Namensdienst, der innerhalb der Kontrolleinheit eindeutige Namen, z.B. Temperatursensor_1, verwaltet. Wird nun eine Kontrolleinheit instanziiert, so stellt die Konfigurationskomponente dieser Kontrolleinheit die erforderliche Umgebung her, indem sie überprüft, ob die benötigten Aktuator-, Sensor- und Kontrolleinheits-Instanzen auf niedrigerer Hierarchiestufe bereits existieren. Dabei überprüft die Konfigurationskomponente zunächst mit Hilfe des Namensdienstes, ob die benötigte Instanz bereits lokal bekannt ist. Wenn ja, so kann mit der entsprechenden Instanz Kontakt aufgenommen werden, und die benötigten Verbindungen können erzeugt werden. Ist die benötigte Instanz noch nicht lokal bekannt, so kann sie entweder erzeugt werden, falls die Konfigurationskomponente dazu befugt ist, oder sie leitet die Namensabfrage an die Konfigurationskomponente der Vater-Kontrolleinheit weiter. Diese Namensabfrage impliziert die Aufforderung, die gewünschte Instanz zu erzeugen, falls sie noch nicht existieren sollte. Diese Namens-Abfrage kann sich über mehrere Ebenen der Hierarchie fortpflanzen, bis eine Ebene erreicht ist, auf der der gewünschte Name bekannt ist bzw. auf der die gewünschte Instanz erzeugt werden kann.

**Bemerkungen:**

Dieser Vorschlag ermöglicht es, das Wissen über die benötigte Umgebung einer Kontrolleinheit lokal bei der Kontrolleinheit selbst zu verwalten. Jede Kontrolleinheit kann die Instanziierung der Elemente ihrer Umgebung (Sensoren, Aktuatoren, Kontrolleinheiten auf niedrigerer Hierarchieebene) anstoßen, aber die Modalitäten ihrer Erzeugung bzw. die Entscheidung, wer die Instanziierung letztendlich vornimmt, werden dabei gekapselt.

Es muß sichergestellt sein, daß es für jede Instanz, die innerhalb des Kontrollsystems benötigt wird, genau eine Kontrolleinheit gibt, die diese Instanz erzeugen darf. Dies sollte die Kontrolleinheit sein, die der „kleinste gemeinsame Vorfahre“ aller Kontrolleinheiten ist, die diese Instanz benötigen.

Dieses Verfahren für die Erzeugung ähnelt dem Mechanismus bei der Konfliktlösung.

# 7 Benutzer-Interaktion

Bei der Interaktion zwischen Benutzer und System kann man zwei Richtungen unterscheiden: zum einen Eingaben, die der Benutzer vornimmt, um das Verhalten des Systems zu beeinflussen; zum anderen Meldungen des Systems, mit denen es den Benutzer über Ereignisse oder interne Zustände informiert. Mit *Benutzer* ist in diesem Abschnitt stets der aktuelle Raumbenutzer gemeint.

## 7.1 Interaktion System -> Benutzer

Wie die Ausgabe von Nachrichten an den Benutzer erfolgt, hängt im wesentlichen von zwei Faktoren ab. Zum einen ist es von Bedeutung, welche physikalischen Ausgabegeräte zur Verfügung stehen, zum Beispiel ein einzeiliges Display oder ein Rechner. Zum anderen kann die Ausgabe von der aktuellen Situation abhängen; zum Beispiel ist es nicht sinnvoll, in einem Raum eine Meldung auszugeben, auf die schnell reagiert werden muß, wenn sich keine Person in dem Raum befindet.

Um diese Aspekte von dem eigentlichen Kontrollablauf in einer Kontrolleinheit, d.h. dem Kontrollalgorithmus, der in dem Controller abläuft, loszulösen, gibt es in jeder Kontrolleinheit eine Komponente, die für die jeweilige Kontrolleinheit Ausgabe-Funktionalität zur Verfügung stellt. Diese Instanz kann das lokale Wissen verwenden, um Ausgaben situationsgerecht zu bearbeiten. Einfache Ausgabeeinheiten, wie sie zum Beispiel in Kontrollzellen vorkommen können, werden ihre Meldungen zum Beispiel standardmäßig an die Ausgabeeinheit ihrer Vater-Kontrolleinheit weiterleiten.

**Bemerkungen:**

> Man beachte, daß es bei diesem Konzept strukturelle Ähnlichkeit zum Fehlerbehandlungskonzept gibt. Dort gab es originäre Fehler, die von einer Kontrolleinheit selbst entdeckt wurden; diesen Fehlern entsprechen hier Ausgaben, die von einem Controller an die eigene Ausgabekomponente gesendet werden. Den Dienstausfall-Fehlern entsprechen bei der System->Benutzer-Interaktion Ausgabemeldungen, die eine Ausgabekomponente von der Ausgabekomponente ihrer Sohn-Kontrolleinheit erhält.
>
> Es muß noch geklärt werden, ob es möglich ist, daß verschiedene Ausgabeeinheiten auf dasselbe physikalische Ausgabegerät zugreifen dürfen, oder ob es wie bei den Aktuatoren nur eine Einheit gibt, die dies darf.

## 7.2 Interaktion Benutzer -> System

Wie in 7.1 kann es verschiedene physikalische Möglichkeiten geben, wie ein Benutzer mit dem Kontrollsystem kommuniziert. Auch hier ist es sinnvoll, daß die einzelnen Komponenten des Kontrollsystems von der konkreten physikalischen Realisierung in einem Raum oder einer anderen Räumlichkeit abstrahieren. Für die einzelnen Komponenten ist letztendlich nur der Wert einer Benutzereingabe entscheidend, nicht aber die Art und Weise, wie der Benutzer die Eingabe gemacht hat. Aus diesem Grund werden die Eingaben eines Benutzers den betroffenen Komponenten über *abstrakte Eingabeeinheiten* übermittelt.

Eine abstrakte Eingabeeinheit wird in der Regel eine echte Teilmenge der Funktionalitäten anbieten, die die ihr zugeordnete physikalische Eingabeeinheit zur Verfügung stellt. Die physikalische Eingabeeinheit wird in der Regel im wesentlichen durch Hardware realisiert, wohingegen die abstrakte Eingabeeinheit im wesentlichen durch Software implementiert wird.

Der Einsatz einer solchen abstrakten Eingabeeinheit kann dann z.B. so aussehen: eine Kontrollzelle teilt der entsprechenden abstrakten Eingabeeinheit mit, daß sie darüber informiert werden will, wenn der Benutzer Änderungen an der Einstellung des oberen Fensterflügels vornehmen möchte. Für jede mögliche Benutzereingabe gibt es eine solche abstrakte Eingabeeinheit, die als einzige Komponente des Kontrollsystems direkt mit der physikalischen Eingabeeinheit verbunden ist.

Benutzereingaben werden direkt an die Kontrolleinheiten geleitet, die sie benötigen und durchlaufen nicht erst evtl. mehrere Hierarchieebenen.

**Bemerkung:**

> Sensorwerte werden ebenfalls unmittelbar an die Kontrolleinheiten geleitet, die sie benötigen.
> Somit gibt es ein einheitliches Konzept für die Behandlung von Informationen, die von außen an das System herangetragen werden.

Man beachte, daß Eingaben oder Vorgaben eines Benutzers nicht direkten Einfluß auf die physikalischen Aktuatoren haben, sondern es werden die Wünsche eines Benutzers dem System mitgeteilt. Letztendlich entscheidet jedoch das Kontrollsystem, d.h. die in ihm realisierte Kontrollstrategie, wie die Aktuatoren eingestellt werden.

Das folgende Bild soll die Abläufe zur Entgegennahme von Benutzereingaben bzw. bei der Weiterleitung von Ausgaben verdeutlichen:

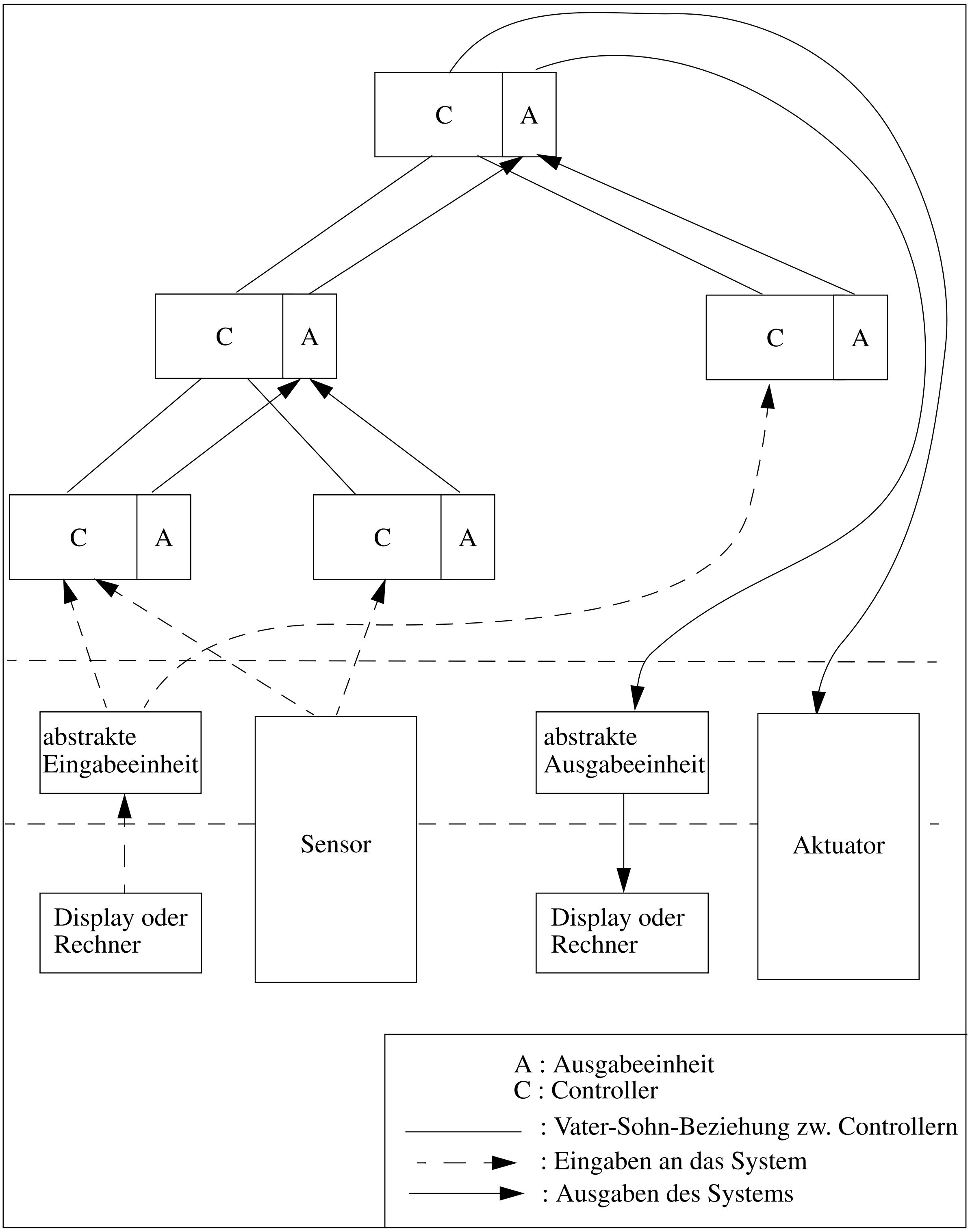


**Abb. 13: Behandlung von Ein- und Ausgaben**

# 8 Beziehungen zw. Anforderungen und Systemarchitektur

Dieser Abschnitt soll die Verfolgbarkeit zwischen dem Anforderungsdokument und der Systemarchitektur beschreiben. /*** Bisher nicht formuliert ***/.