

## Verschließbare Bereiche

---

### Was

---

Das Pattern „Verschließbare Bereiche“ bietet die Möglichkeit, UI-Elemente innerhalb einer Benutzeroberfläche in getrennte Bereiche zu verteilen.

### Wie

---

#### Aktion des Benutzers

Um ein bestimmtes UI-Element zu finden, überlegt sich der Benutzer, zu welchem Bereich es gehört. Er durchsucht die Überschriften der Bereiche, öffnet den passenden Bereich und durchsucht diesen. Verwendet der Benutzer das UI-Element nicht mehr, kann er den Bereich wieder schließen.

Der Benutzer öffnet bzw. schließt die Bereiche entweder per Mausklick auf den Balken, durch Tastatureingabe (Tab-Taste bzw. Pfeil-Tasten) oder durch Antippen des Balkens auf einem Touch-Bildschirm.

#### Reaktion des Systems

Der ausgewählte Bereich wird geöffnet bzw. durch nochmaliges Anklicken oder Antippen geschlossen.

#### Was muss bei der Gestaltung der Interaktion beachtet werden?

Jeder Bereich ist mit einem Balken zum Öffnen und Schließen versehen. Jeder Balken trägt die Bezeichnung des entsprechenden Bereichs als Überschrift. Das Öffnen eines Bereichs geschieht immer durch eine explizite Aktion, z.B. durch Anklicken oder Antippen des Balkens. Es sollte kein Hover-Effekt zum Öffnen verwendet werden (Ausnahme: „open on hover“ bei „Drag & Drop“-Operationen). Das Schließen des Bereichs geschieht ebenfalls durch eine explizite Aktion, also durch ein erneutes Anklicken/Antippen des Balkens. Zusätzlich kann in jedem Bereich ein Schließen-Knopf angebracht werden.

Der Benutzer kann die Bereiche einfach öffnen und wieder schließen. Er kann die Bereiche unabhängig voneinander öffnen und schließen. Er kann keinen, einen oder mehrere Bereiche gleichzeitig öffnen. Die Bereiche können je nach Menge der Inhalte unterschiedlich groß sein und sollten nicht größer sein als nötig. Haben alle Bereiche eine einheitliche und feste Größe, so sollten die Inhalte der Bereiche ähnlich umfangreich sein, damit es keine Leerflächen gibt. Für größere Bereiche, z.B. Arbeitsbereiche innerhalb einer Anwendung, bietet sich eine horizontale Ausrichtung an, da so der zur Verfügung stehende Platz optimal genutzt werden kann.

Die einzelnen Bereiche sollten einen sinnvollen Ausgangszustand haben. Beim Öffnen der Anwendung dürfen nicht alle Bereiche geschlossen, aber auch nicht alle Bereiche geöffnet sein. Es kann z.B. der erste, der wichtigste oder der zuletzt geöffnete Bereich geöffnet sein. Der Benutzer sieht dann die ersten, die wichtigsten bzw. die zuletzt angezeigten UI-Elemente und er weiß zugleich, dass sich die anderen Bereiche individuell öffnen und schließen lassen.

Der Inhalt eines Bereichs sollte in irgendeiner Form als zugehörig zum Überschrift-Balken gelayoutet werden und sich von den anderen Bereichen unterscheiden. Hierzu kann z.B. ein Farbschema verwendet werden.

Das Pattern darf nicht aus mehr als zwei Ebenen (Überschriften und Inhalte) bestehen. Das bedeutet, dass keine verschließbaren Bereiche ineinander verschachtelt sind. Der Benutzer verliert sonst leicht den Überblick, auf welcher Ebene er sich momentan befindet. Besteht die Notwendigkeit, UI-Elemente verschachtelt darzustellen, so sollte ein anderes Pattern verwendet werden.

### **Welche Bestandteile sind wichtig für die Interaktion?**

- Benutzeroberfläche
- UI-Elemente
- Bereiche
- Balken

### **Wie findet der Interaktionsgestalter diese Bestandteile im eigenen Kontext?**

Die UI-Elemente werden in mehrere inhaltlich-logische bzw. aufgabenorientierte Gruppen unterteilt. UI-Elemente sind alle Controls, Funktionen, Informationen usw., die in der Benutzeroberfläche verwendet bzw. angezeigt werden.

Für jede Gruppe von UI-Elementen wird ein eigener Bereich angelegt. Die Bereiche erhalten passende und prägnante Bezeichnungen, die für den Anwender leicht verständlich sind.

### **Wann**

---

Das Pattern kann eingesetzt werden, wenn es in einer Benutzeroberfläche eine große Anzahl von UI-Elementen gibt, aber nicht alle UI-Elemente zeitgleich auf dem Bildschirm dargestellt werden müssen. Es reicht aus, wenn der Benutzer weiß, wo sich die UI-Elemente befinden, und dass er leicht auf diese zugreifen kann. Alle UI-Elemente müssen zumindest zeitweise optional sein. Es darf keine UI-Elemente geben, die der Benutzer zur Erledigung seiner Aufgabe permanent benötigt.

Die UI-Elemente müssen sich in getrennte und klar bezeichnete Bereiche unterteilen lassen. Es sollte mehr als zwei Bereiche, aber auch nicht zu viele Bereiche (nicht mehr als zehn Bereiche) geben. Alle UI-Elemente sollten inhaltlich miteinander verwandt sein und von vergleichbarer Wichtigkeit sein, so dass die Darstellung auf einer Hierarchieebene sinnvoll ist.

Das Pattern „Verschließbare Bereiche“ ist eher für fortgeschrittene als für gelegentliche Nutzer einer Anwendung geeignet, denn die meisten UI-Elemente sind zunächst versteckt und die Benutzeroberfläche wird schnell unübersichtlich, wenn man mehrere oder alle Bereiche öffnet.

## Warum

---

Komplexe Anwendungen bieten oft zeitgleich eine Vielzahl von UI-Elementen innerhalb einer Benutzeroberfläche. Dies ist nicht sehr unübersichtlich und der Benutzer muss unter Umständen weit scrollen, um benötigte UI-Elemente zu erreichen.

Das Pattern „Verschließbare Bereiche“ ermöglicht eine platzsparende und übersichtliche Anordnung von UI-Elementen. Es sorgt dafür, dass die Benutzeroberfläche aufgeräumt ist, leicht zu bedienen ist und dass nur die jeweils benötigten UI-Elemente angezeigt werden. Da der Benutzer meist nur wenige UI-Elemente zeitgleich verwendet, bietet ihm das Pattern „Verschließbare Bereiche“ die Möglichkeit, nicht benötigte UI-Elemente wegzuschließen und zu ignorieren. Es findet keine visuelle Ablenkung durch nicht benötigte UI-Elemente statt und der Benutzer ist nicht durch die Anzahl der gleichzeitig dargestellten UI-Elemente überfordert.

Der Benutzer kann selbst entscheiden, wie viele und welche Bereiche der Anwendung er gleichzeitig geöffnet haben möchte und auf welche Inhalte er gleichzeitig zugreifen möchte. Die tatsächlich benötigten UI-Elemente und die aktuelle Aufgabe stehen so stets im Fokus des Benutzers. Wird ein weiteres UI-Element benötigt, kann dieses sehr schnell gefunden und bearbeitet werden. Verschiedene Datenquellen können in den getrennten Bereichen geöffnet bleiben. Dadurch ist es möglich, Teilbereiche der Anwendung und Prozessschritte, die sonst sequentiell bearbeitet werden, parallel zu öffnen und zeitgleich zu bearbeiten.

# Illustration

Bilder der Beispielimplementierung in „Excalibur“  
 (Prototyp; erstellt im Rahmen des Forschungsprojekts FUN-NI für einen Versuch bei  
 HanseNet, Februar 2010):

Abbildung 1: Ausgangszustand der Anwendung - der erste Bereich „Detailinformationen“ ist geöffnet:

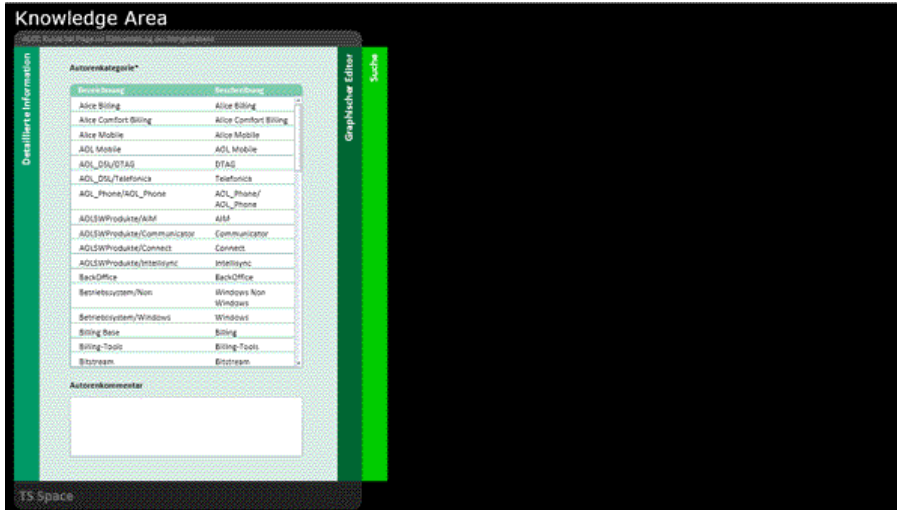


Abbildung 2: Die zwei Bereiche „Detailinformationen“ und „Graphischer Editor“ sind geöffnet:

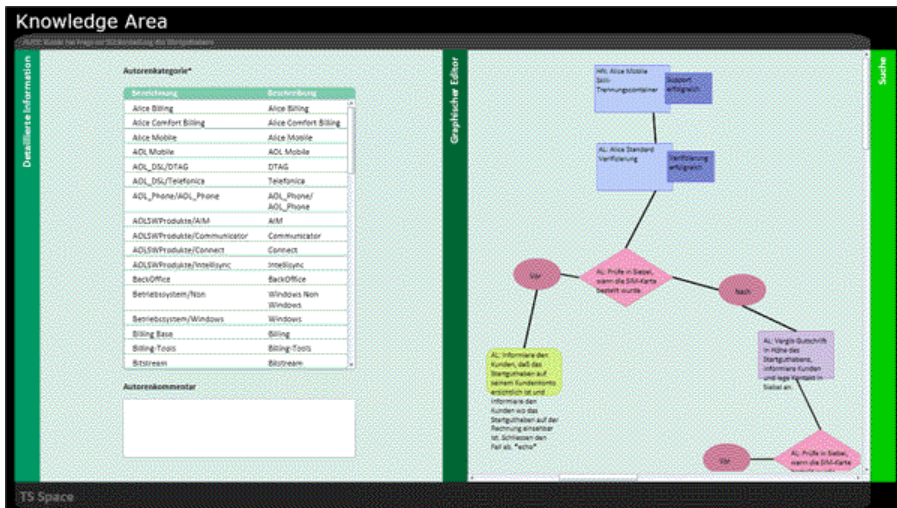


Abbildung 3: Alle Bereiche der Anwendung sind geöffnet:

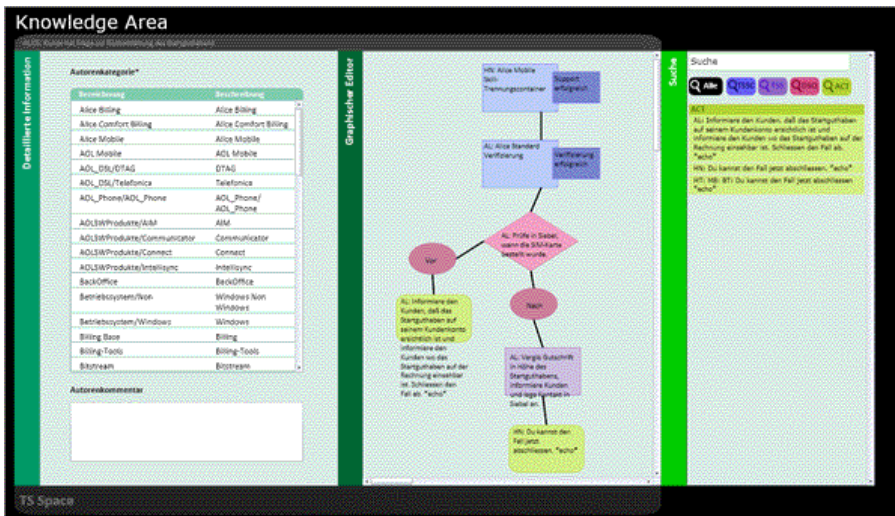


Abbildung 4: Der Bereich „Graphischer Editor“ ist geöffnet:

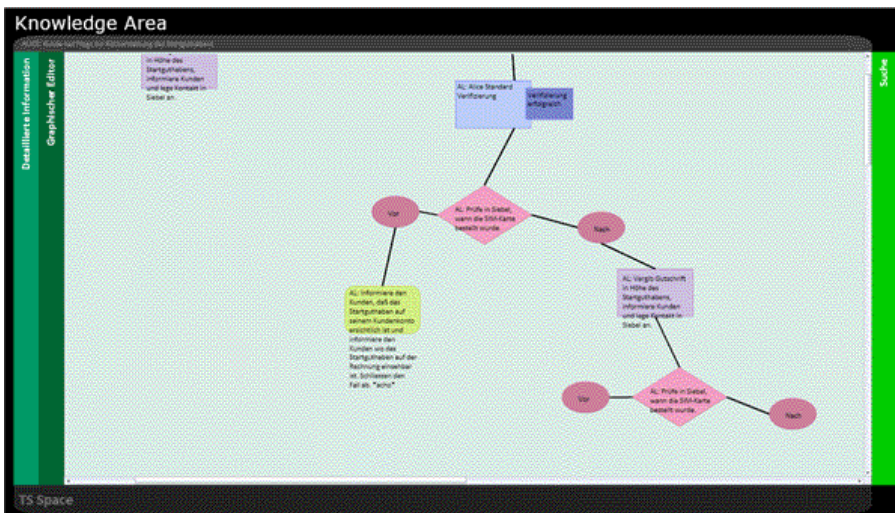
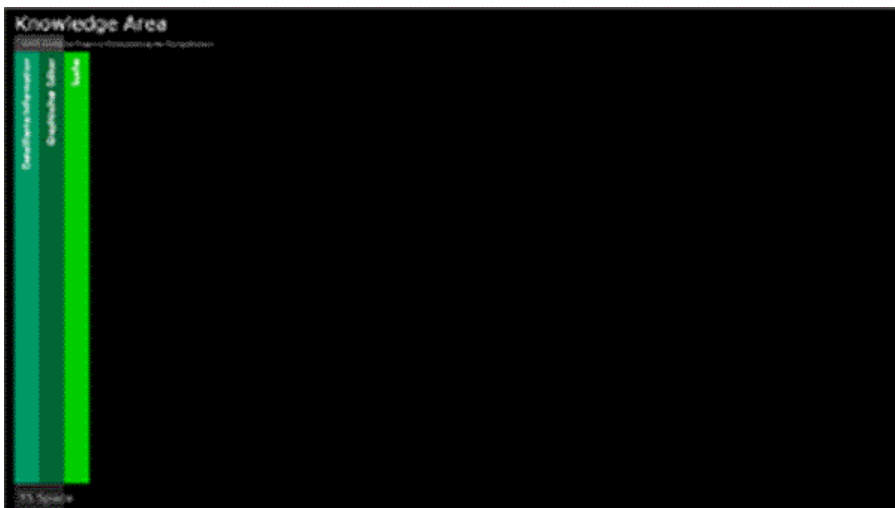


Abbildung 5: Alle Bereiche sind geschlossen:



## Bekannte Verwendungen

---

-

## Checkliste

---

|  | Ja                       | Nein                     |
|--|--------------------------|--------------------------|
| • Haben die Bereiche einen sinnvollen Ausgangszustand (geöffnet/geschlossen) und eine sinnvolle Ausgangsgröße? | <input type="checkbox"/> | <input type="checkbox"/> |
| • Kann der Benutzer die Bereiche einfach öffnen und schließen?   | <input type="checkbox"/> | <input type="checkbox"/> |
| • Können mehrere Bereiche gleichzeitig geöffnet werden?  | <input type="checkbox"/> | <input type="checkbox"/> |
| • Gibt es nur eine Inhaltsebene (und keine verschachtelten Bereiche)?  | <input type="checkbox"/> | <input type="checkbox"/> |
| • Sind die UI-Elemente inhaltlich-logisch bzw. aufgabenorientiert gruppiert?                                   | <input type="checkbox"/> | <input type="checkbox"/> |
| • Sind die UI-Elemente so verteilt, dass in den Bereichen kein Platz durch Leerflächen verschwendet wird?      | <input type="checkbox"/> | <input type="checkbox"/> |

## Implementierung

---

-

## Vertrauen (Confidence)

---

- Entwurf einer neuen Idee
- Übertragene Idee aus folgendem Kontext:
- Bewährtes Pattern in folgendem Kontext:
- Wird (gerade) evaluiert
- Evaluiert von/in: a3 systems/Excalibur GKE

## Ähnliche Patterns

---

- Zusammensetzung („has a“):
- Spezialisierung („is a“): Spezialisierung des Patterns „Closable Panels“ (auch bekannt als „Collapsible Panels“). Dieses Pattern wird in der Regel für Menüs oder für kleine Teilbereiche einer Anwendung verwendet (es ist dann eingebettet in einen größeren Container). Die Stackpanels beim Pattern „Closable Panels“ sind meist vertikal ausgerichtet.
- Empfohlene Verwendung zusammen mit:
- Äquivalente Alternative („kann ersetzt werden durch“):
- Schließt sich gegenseitig aus („darf nicht verwendet werden mit“):

Abgrenzung gegenüber anderen bekannten Patterns:

- „Local Zoom“ und „Details on Demand“: Bei diesen Patterns kann immer nur ein Element genauer betrachtet werden.
- „Module Tabs“: Es kann immer nur ein Tab geöffnet werden. Die Tabs haben eine identische Größe, so dass bei unterschiedlichen Inhaltsmengen Leerflächen entstehen.
- „Accordion“ und „Accordion Menu“: In der Regel ist nur jeweils ein Bereich des Accordions geöffnet. Accordion Menus sind meist vertikal ausgerichtet.

## Literaturangaben

---

...

## Auch bekannt als

---

...

## Tags

---

Benutzeroberfläche, Menü, Objektsteuerung, Layout, Interface Design, Organisation von Elementen

## Autoren

---

Hartmut Schmitt

## Gebrauchshistorie

---

Angewendet in Excalibur - Call-Center-Abteilung der HanseNet Telekommunikation GmbH.

## Danksagungen

---

Die Arbeiten an dem Pattern wurden durch das vom BMBF geförderte Projekt FUN-NI finanziert (BMBF-Förderkennzeichen 01 IS 09007). Wir danken der HanseNet Telekommunikation GmbH, die die Durchführung des Experiments unterstützt hat.

## Version

---

1.3

## Kommentare

---

...