

Vermittler (Mediator)

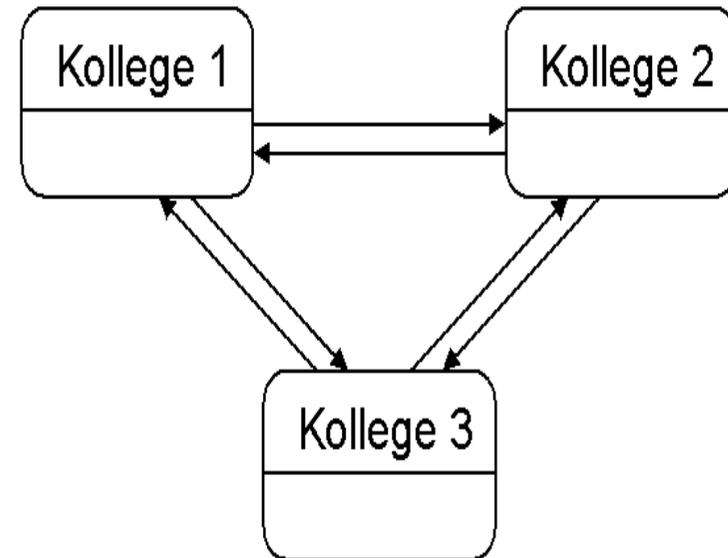
ein objektbasiertes Verhaltensmuster

Zweck

- ❑ Zusammenspiel einer Menge von Objekten wird durch ein Objekt gekapselt.
 - ❑ Vermittler fördert die lose Kopplung von Objekten
 - ❑ Dadurch kann das Zusammenspiel der Objekte, unabhängig von ihnen, variiert werden.
-

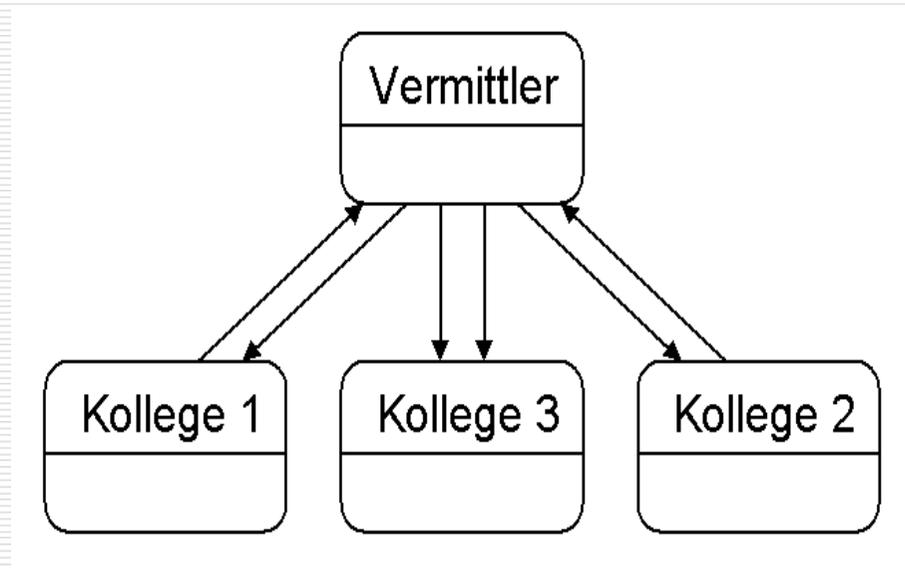
Motivation

- ❑ Objektorientierter Entwurf fördert die Verteilung von Verhalten zwischen Objekten. Das Ergebnis kann eine Objektstruktur mit vielen Beziehungen zwischen den Objekten sein; schlimmstenfalls muss jedes Objekt jedes andere Objekt kennen.
- ❑ Es entsteht ein Geflecht von Beziehungen, das unüberschaubar und nur schwer zu warten ist.



Motivation

- Das Entwurfsmuster Vermittler stellt für diese Problemstellung einen Lösungsansatz zur Verfügung.
- Auch hier kann jedes Objekt mit jedem anderen kommunizieren, aber ausschließlich über das Vermittlerobjekt



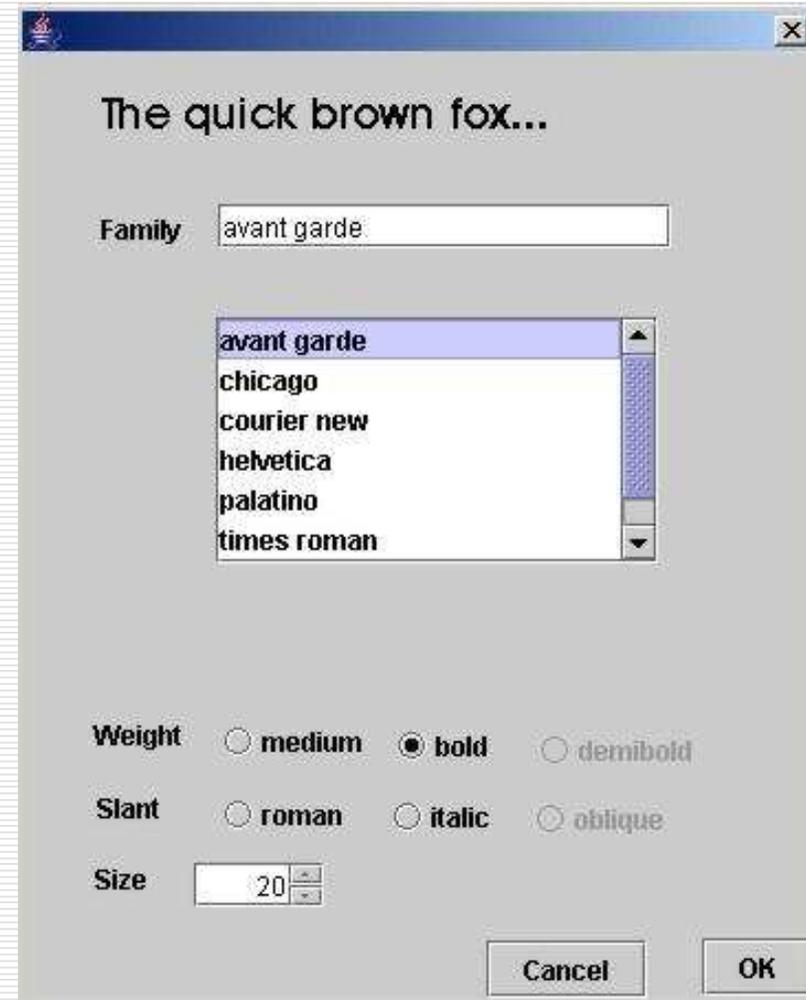
Motivation

- Dialogbox repräsentiert Ansammlung von Widgets wie Knöpfe, Menüs und Eingabefelder
- Zwischen den Widgets bestehen unterschiedliche Abhängigkeiten

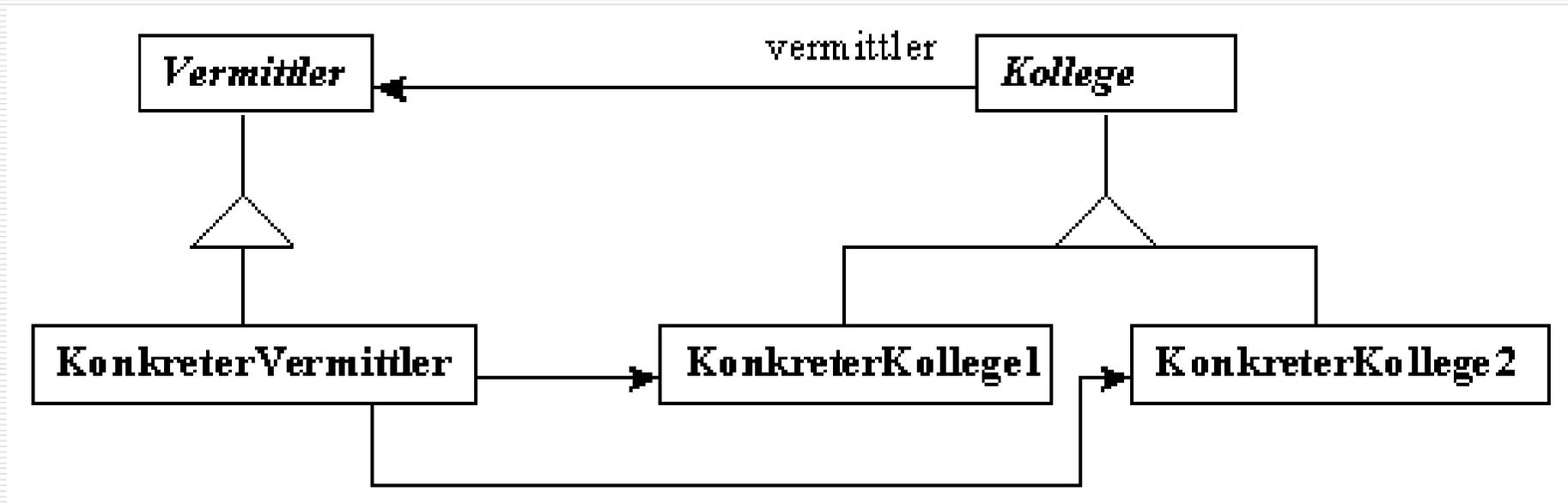


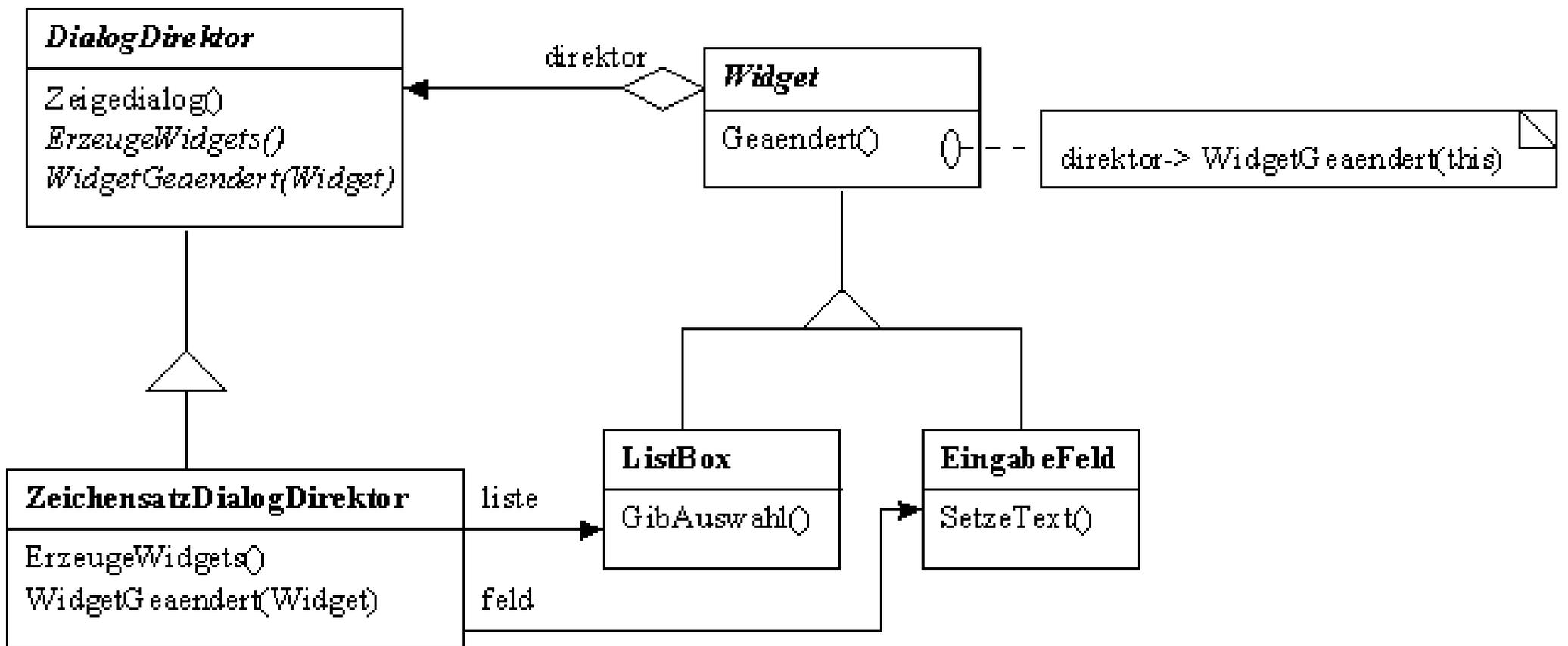
Motivation

- Das Gesamtverhalten wird in einem separaten Vermittlerobjekt gekapselt.
- Ein Vermittler kontrolliert und koordiniert die Interaktion von Objekten innerhalb einer Gruppe.



Struktur





Anwendbarkeit

- Menge von Objekten, arbeiten in komplexer Weise zusammen. Abhängigkeiten zwischen Objekten sind unstrukturiert und schwer nachvollziehbar.
 - Ein Objekt bezieht sich auf viele andere Objekte und arbeitet mit denen zusammen. Dadurch ist es schwierig diese Objekte wiederzuverwenden.
 - Ein auf mehrere Klassen verteiltes Verhalten soll, ohne viele Unterklassen zu bilden, maßgeschneidert werden.
-

Teilnehmer

- Vermittler
 - Definiert eine Schnittstelle für den Umgang mit den Kollegenobjekten.

 - KonkreterVermittler
 - kennt und verwaltet seine Kollegenobjekte.
 - implementiert durch Koordination der Kollegenobjekte das Gesamtverhalten.

 - KollegenKlassen
 - Jede Kollegenklasse kennt ihre Vermittlerklasse.
 - Jedes Kollegenobjekt arbeitet mit seinem Vermittler zusammen und nicht mit seinen Kollegenobjekten.
-

Interaktion

- Kollegen senden und empfangen Anfragen von/zu einem Vermittlerobjekt.
 - Vermittler leitet diese Anfragen zwischen den richtigen Kollegenobjekten weiter. Dadurch wird das Gesamtverhalten implementiert.
-

Vermittler

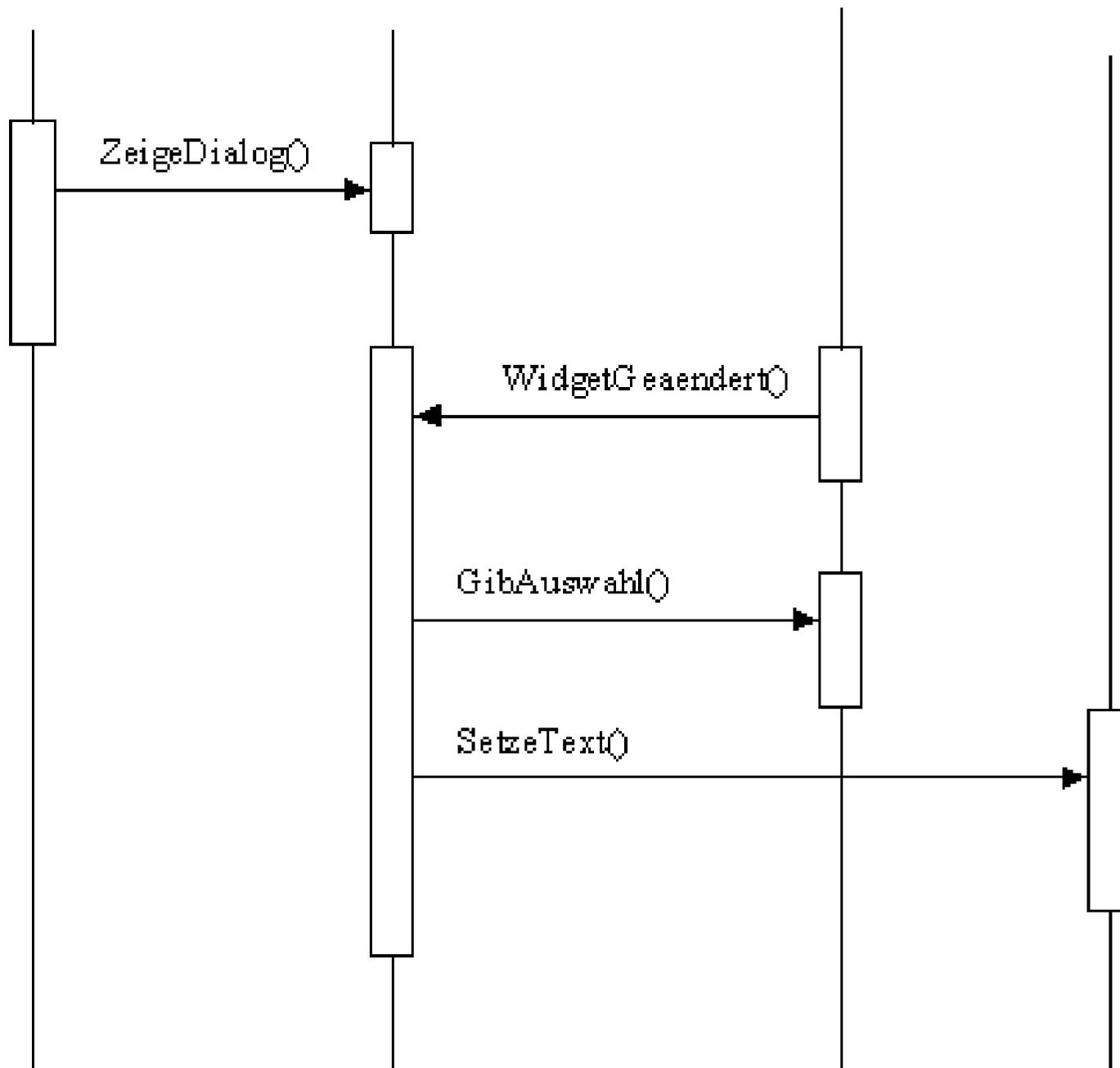
Kollegen

einKlient

einZeichensatzDialogDirektor

eineListBox

einEingabefeld



Interaktion

- ❑ Die Listbox meldet ihrem Direktor, daß sie sich geändert hat.
 - ❑ Der Direktor holt sich die Auswahl von der Listbox.
 - ❑ Der Direktor gibt die Auswahl an das Eingabefeld weiter.
 - ❑ Dann aktiviert der Direktor jene Radio-buttons die vom Eingabefeld abhängen und entsprechende Aktionen auslösen können.
-

Konsequenzen

- ❑ Schränkt Unterklassenbildung ein.
 - ❑ Entkoppelt Kollegenobjekte.
 - ❑ Vereinfacht die Struktur eines Objekts.
 - ❑ Vermittlermuster abstrahiert davon wie Objekte zusammenarbeiten.
 - ❑ Zentralisiert die Steuerung.
-

Implementierung

- Weglassen der abstrakten Vermittlerklasse.
 - Diese ist nur dann notwendig, wenn die Kollegenobjekte mit mehr als einem Vermittler zusammenarbeiten sollen.
-

Implementierung

- Interaktion von Kollegen- und Vermittlerklassen
 - Die Benachrichtigung von Kollegenklassen zur Vermittlerklasse kann als Beobachtermuster implementiert werden.
 - Kollegenobjekte benutzen eine spezialisierte Schnittstelle zum Vermittler und übergeben sich selbst. So kann der Vermittler feststellen von wem er benachrichtigt wurde.
-

Beispielcode

- ❑ Java benutzt auch das Vermittlermuster.
 - ❑ Java hat Listboxen, Eingabefelder und auch Buttons. Die Vermittlerklasse ist entweder eine von Applet abgeleitete Klasse oder eine von Frame abgeleitete Klasse.
 - ❑ Die Kollegenobjekte benachrichtigen den Vermittler unter Zuhilfenahme der ListenerInterfaces, dieses Interface wird durch Mehrfachvererbung in den Vermittler mit aufgenommen. Der Vermittler reagiert dann darauf und koordiniert.
-

Beispielcode

```
□ public class ZeichensatzDialogDirektor extends Applet
    implements ItemListener{
    private Button m_button;
    private List m_list;
    private TextField m_textfeld;

    public void Init(){
        m_button = new Button("OK");
        m_list = new List(4, false);
        m_list.addItem("chicago");
        m_list.addItem("courier");
        m_list.addItem("palatino");
        m_list.addItem("times roman");
```

Beispielcode

```
        m_textfeld = new TextField(20);
        add(m_button);
        add(m_list);
        add(m_textfeld);
        addItemListener(this);
    }

    public void itemStateChanged(ItemEvent e) {
        String cmd = e.getItemSelectable();
        m_textfeld.setText(cmd);
    }
}
```

Bekannte Verwendungen

- Sowohl ET++ als auch die THINK-C-Klassen, sowie auch Borland Delphi benutzten die Vermittler in Dialogen.
 - Smalltalk/V für Windows
Anwendungsarchitektur basiert auf der Vermittlerstruktur.
-

Verwandte Muster

- Das Fassadenmuster gibt eine vereinfachte Schnittstelle zu einem komplexen System. Sein Protokoll ist unidirektional. Der Vermittler ermöglicht eine Zusammenarbeit von Objekten. Sein Protokoll ist multidirektional.
 - Kollegenobjekte können mit Hilfe des Beobachtermuster den Vermittler benachrichtigen.
-