



Composition Filters – C++/CF

Marc Lörner, 19.05.2003

(gerd.loerner@t-online.de)



Übersicht

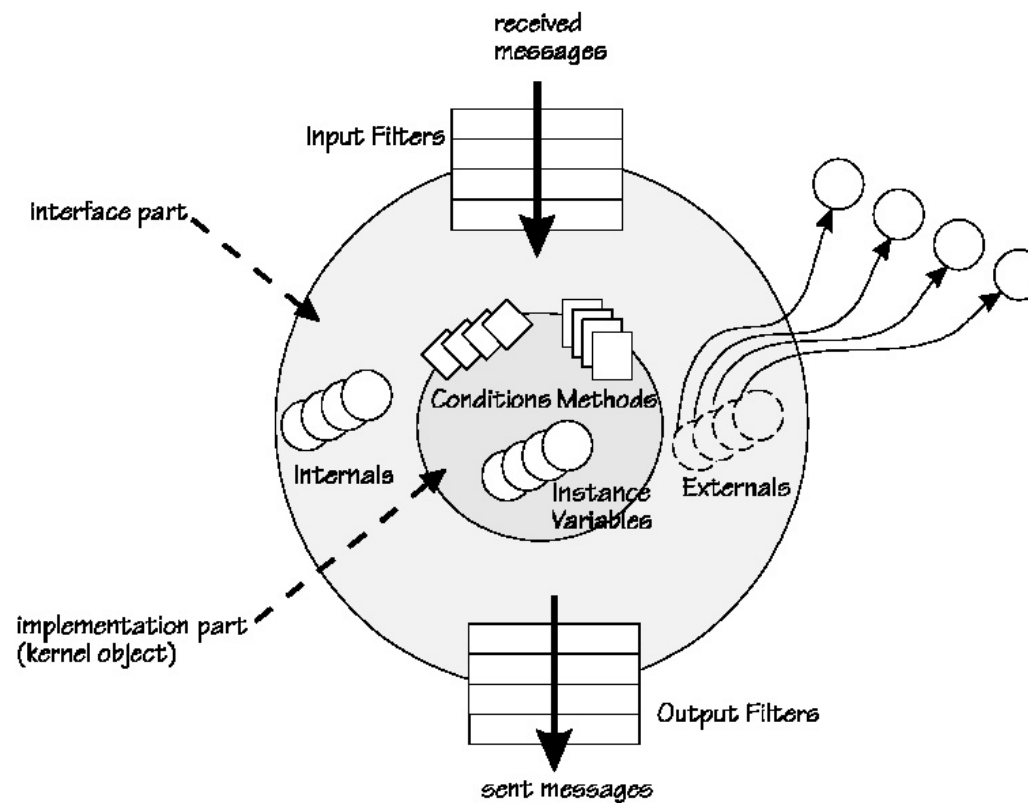
- Composition Filters (CF) Objektmodell
- C++/CF



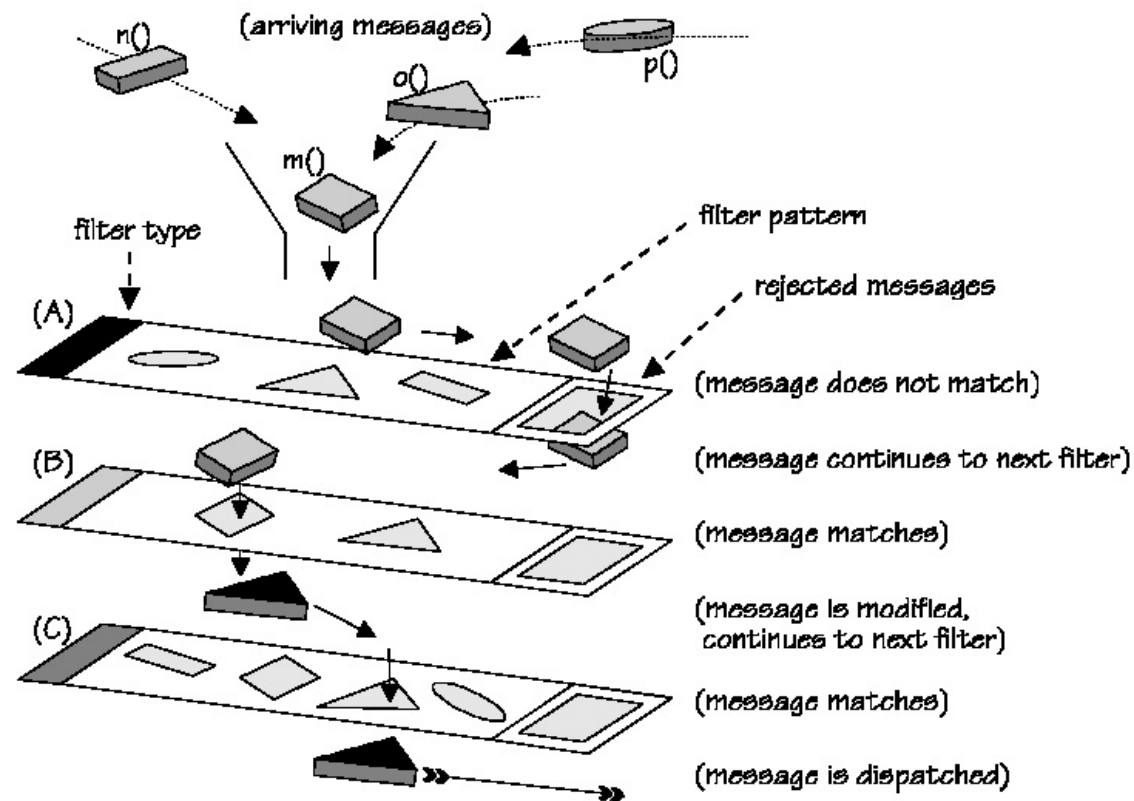
Allgemeines zum Objektmodell

- Erweiterung des OO-Konzepts
- Offenheit für Erweiterungen
- Feste Filter-Reihenfolge

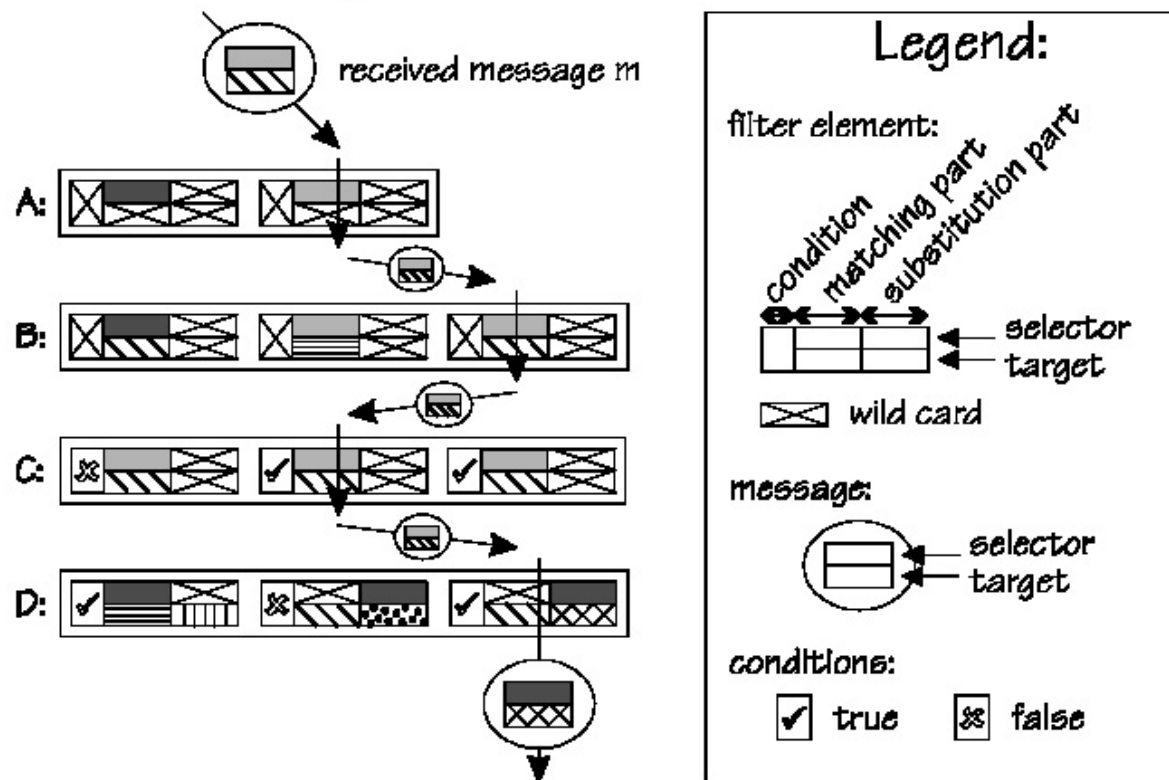
Das CF-Objektmodell



Arbeitsweise der Filter (1)



Arbeitsweise der Filter (2)





Welche Arten von Filtern gibt es?

- Error Filter
- Dispatch Filter
- Send Filter
- Meta Filter
- Wait Filter
- ...



Was ist C++/CF?

- Erweiterung von C++
- Ermöglicht Verwendung von „alten“ C++-Klassen



Definition eines Filters

```
aFilterIdentifier : aFilterClass =  
    {aCond=>aTarget.aSelector(aParameter,...),...};
```

Operatoren:

=>: Akzeptiere jede Nachricht, die im Matching-Part passt

~>: Akzeptiere die Nachrichten, die nicht im Matching-Part
passen



Bedingte Filter - Conditions

Verschiedene Formen:

- TRUE
- Condition-Methode
- Kombination logischer Ausdrücke



Beispiel für die Anwendung eines Filters

```
Class Point
  conditions
    Initialized; //Ist das Objekt initialisiert
  methods
    MoveTo(int , int);
    GetX returns int ;
    GetY returns int ;
  inputfilters
    disp : Dispatch = {TRUE=>inner.MoveTo ,
                       Initialized=>inner.* };
End;
```



C++-Code des Beispiels

```
Class Point {  
    public:  
        Point() : X(0),Y(0),InitializeDone(FALSE) {}  
        int Initialized() { return InitializeDone;};  
        void MoveTo(int x, int y) {  
            ...  
            InitializeDone = TRUE;  
        }  
        int GetX() { return X;};  
        int GetY() { return Y;};  
    private:  
        int X,Y, InitializeDone;  
};
```



Ableiten in C++/CF

```
Class ReferencePoint
  internals
    MyPoint : Point;
  methods
    Display;
  inputfilters
    disp : Dispatch = {TRUE=>MyPoint.* ,
                       TRUE=>inner.*};
End;
```



Dynamische Ableitung

```
Class Person
  internals
    atHome : Home;
    atWork : Work;
  conditions
    nineTILfive;
  inputfilters
    disp : Display = {nineTILfive=>atWork.* ,
                      TRUE=>atHome.*};
End;
```



Pseudo-Variablen von C++/CF

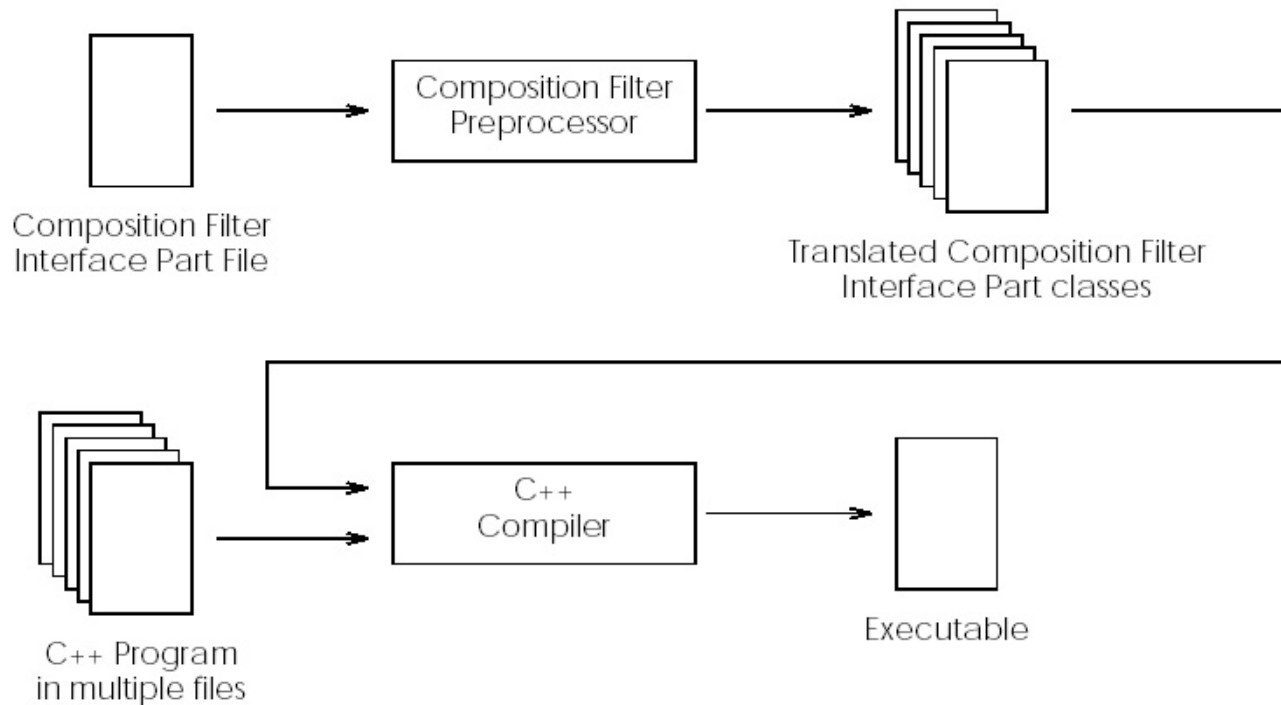
- Client
- Server
- Outer
- Inner
- Message



Aufbau von C++/CF-Programmen

- *.h
- *.C
- *.CF

Compile-Vorgang





Performanz von C++/CF- Programmen

$T(C++) < T(C++/CF) :$

- Filtern der Nachrichten
- Speicherplatz für Nachrichten
- Erzeugen eines Objekts => Erzeugen des C++, CF, Objekt-Manager, Filter-Objekts