

The Mayday Processing Framework

*A Graphical Processing
Framework for Mayday*

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

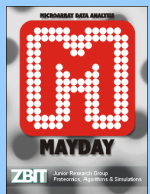
Applicator

Pipelines

MDOs

Übersicht

Vorführung



Motivation

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

- Wiederkehrende Aufgaben automatisieren
 - „Preprocessing pipeline“
Normalisierung, Imputation, Log transformation,
Herausfiltern von uninteressanten Genen
 - Andere Pipelines je nach Fragestellung
- Mayday bietet bislang keine einheitliche
Möglichkeit, diese Aufgaben durchzuführen
und zu automatisieren.

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

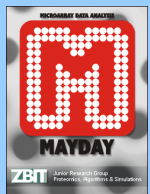
Applicator

Pipelines

MDOs

Übersicht

Vorführung



Anforderungen

- Möglichst allgemeines Framework
 - Erweiterbar durch Module
 - Maydays Komplexität vor Programmierern neuer Module verbergen
 - Möglichkeiten der Module nicht einschränken
- Anwendung auf mehrere ProbeLists (batch)
- Zusammenstellen von Modulen zu Pipelines
 - Speichern von Pipelines und Einstellungen
 - Auch für Nicht-Programmierer verständlich

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

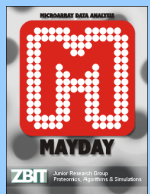
Applicator

Pipelines

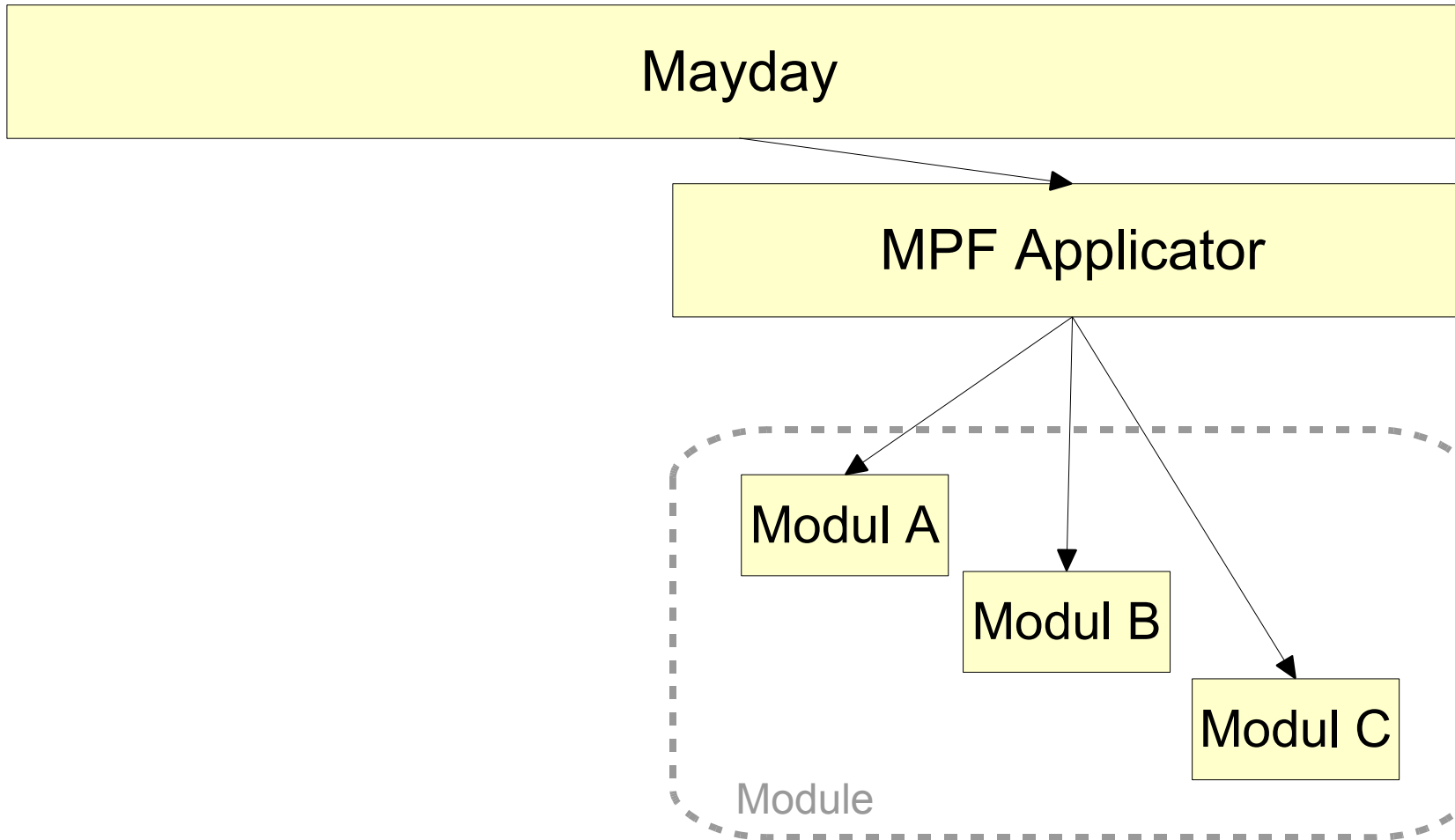
MDOs

Übersicht

Vorführung



Aufbau



MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

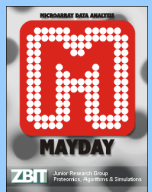
Applicator

Pipelines

MDOs

Übersicht

Vorführung



Aufbau

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

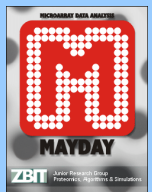
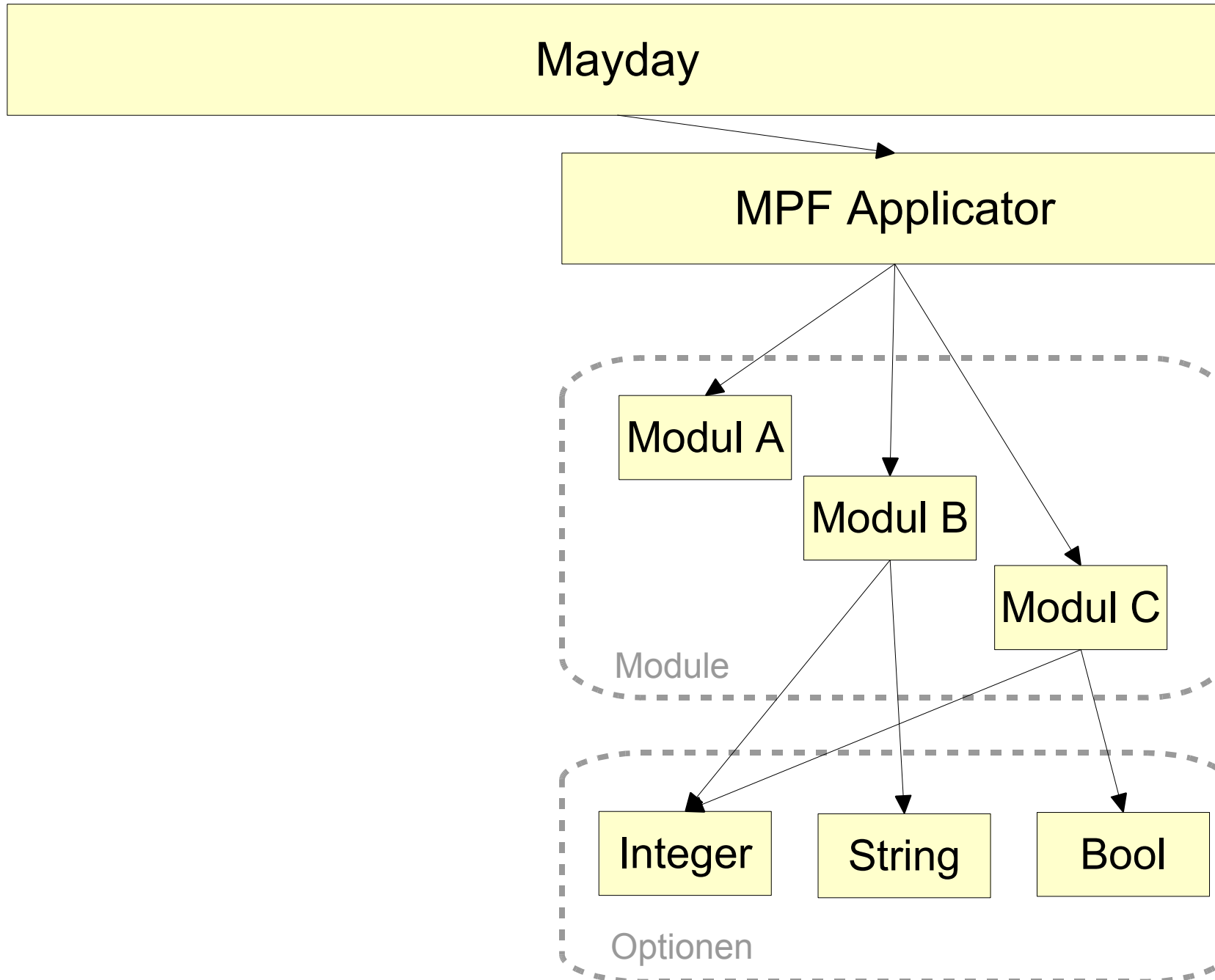
Applicator

Pipelines

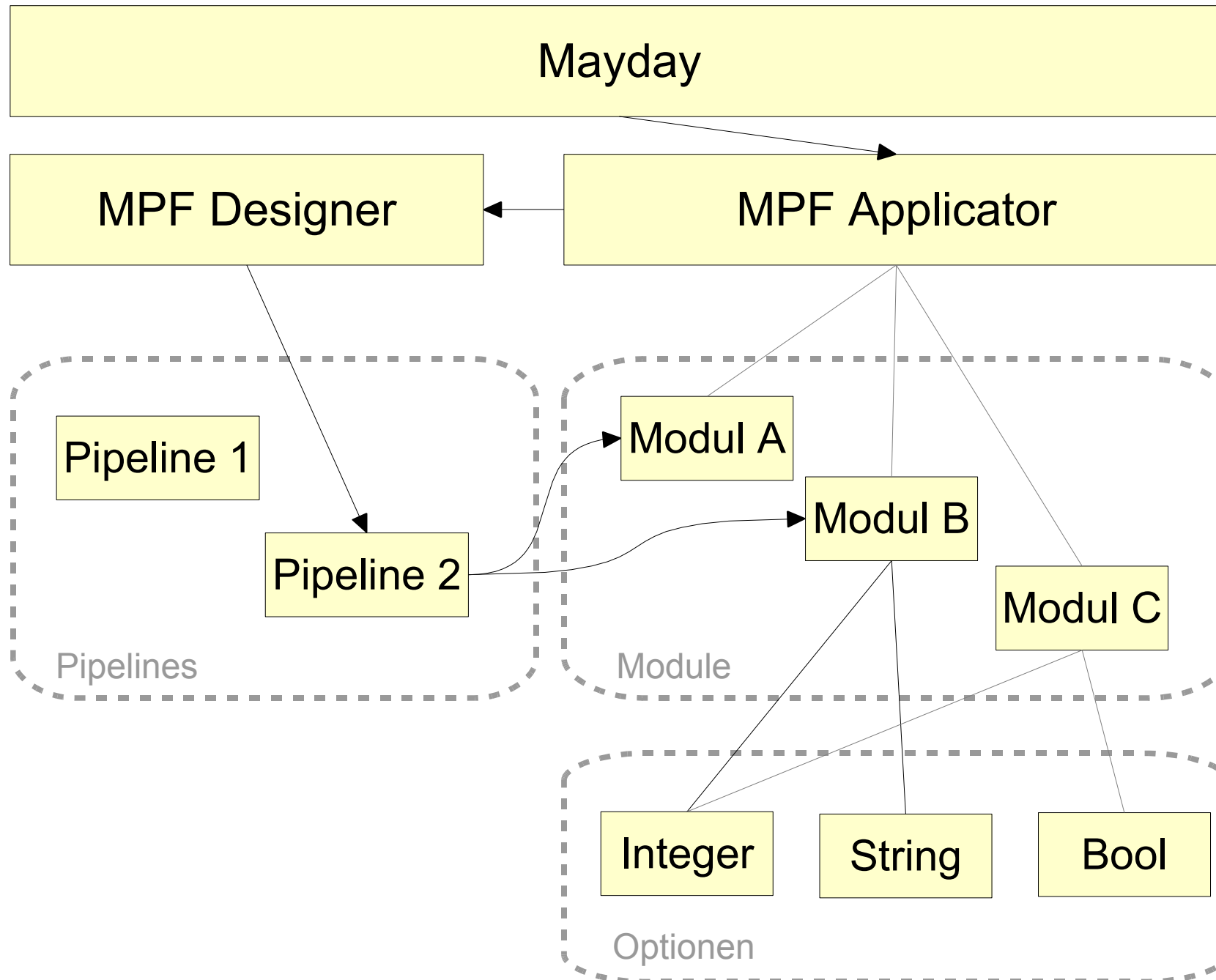
MDOs

Übersicht

Vorführung



Aufbau



MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

Applicator

Pipelines

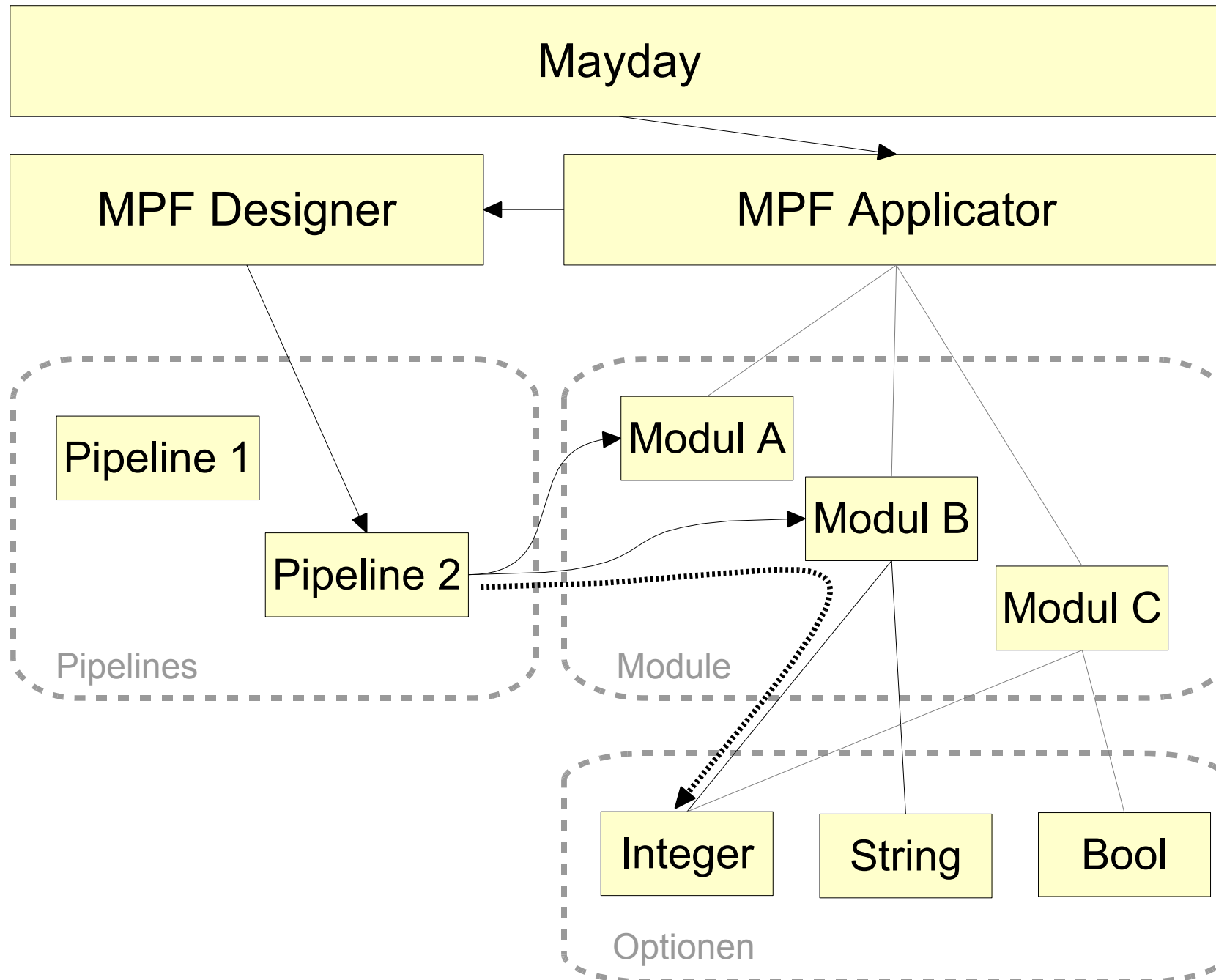
MDOs

Übersicht

Vorführung



Aufbau



MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

Applicator

Pipelines

MDOs

Übersicht

Vorführung



Aufbau

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

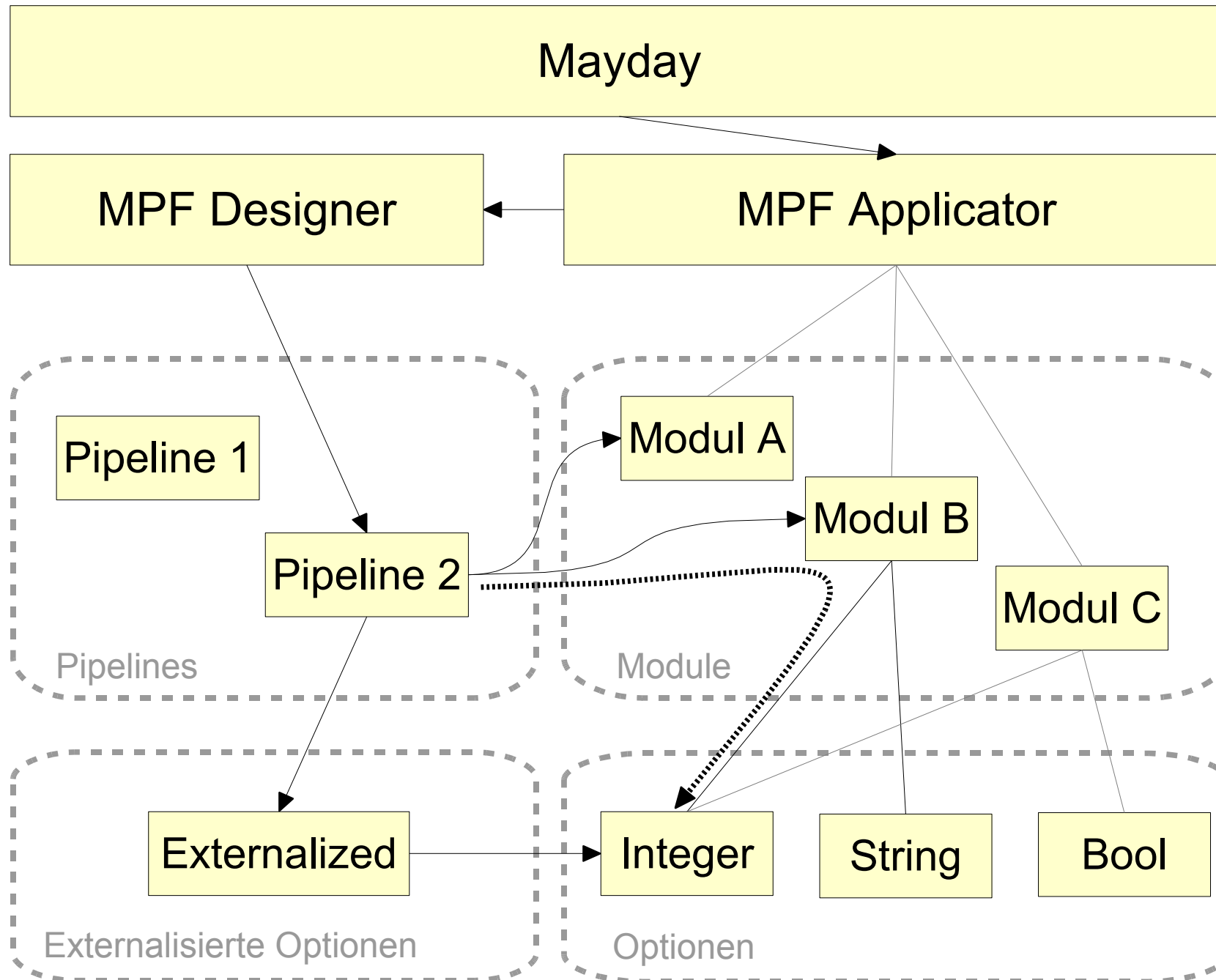
Applicator

Pipelines

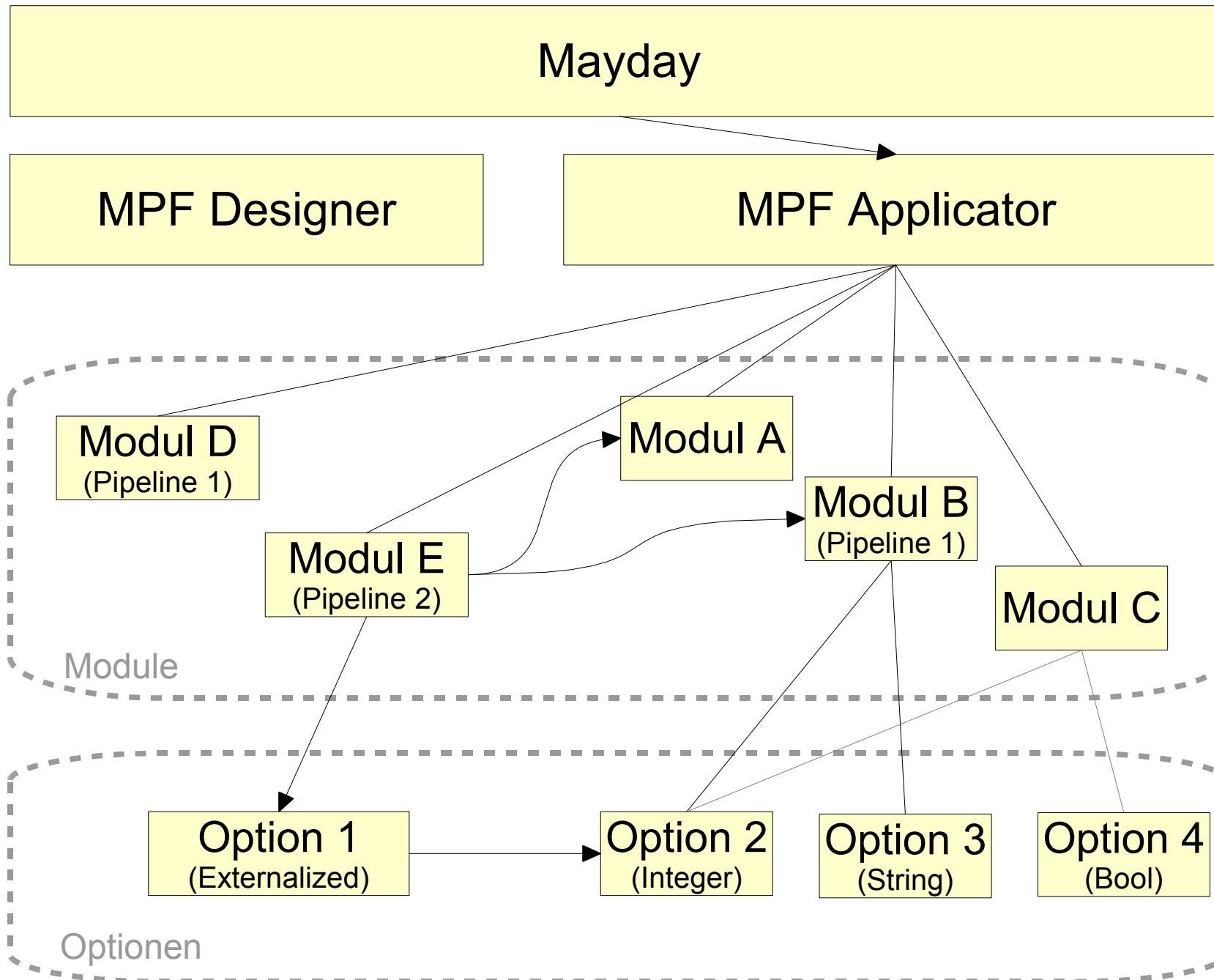
MDOs

Übersicht

Vorführung



Aufbau



MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

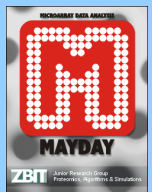
Applicator

Pipelines

MDOs

Übersicht

Vorführung



Einfache Module

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

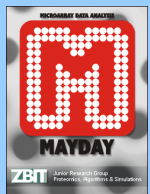
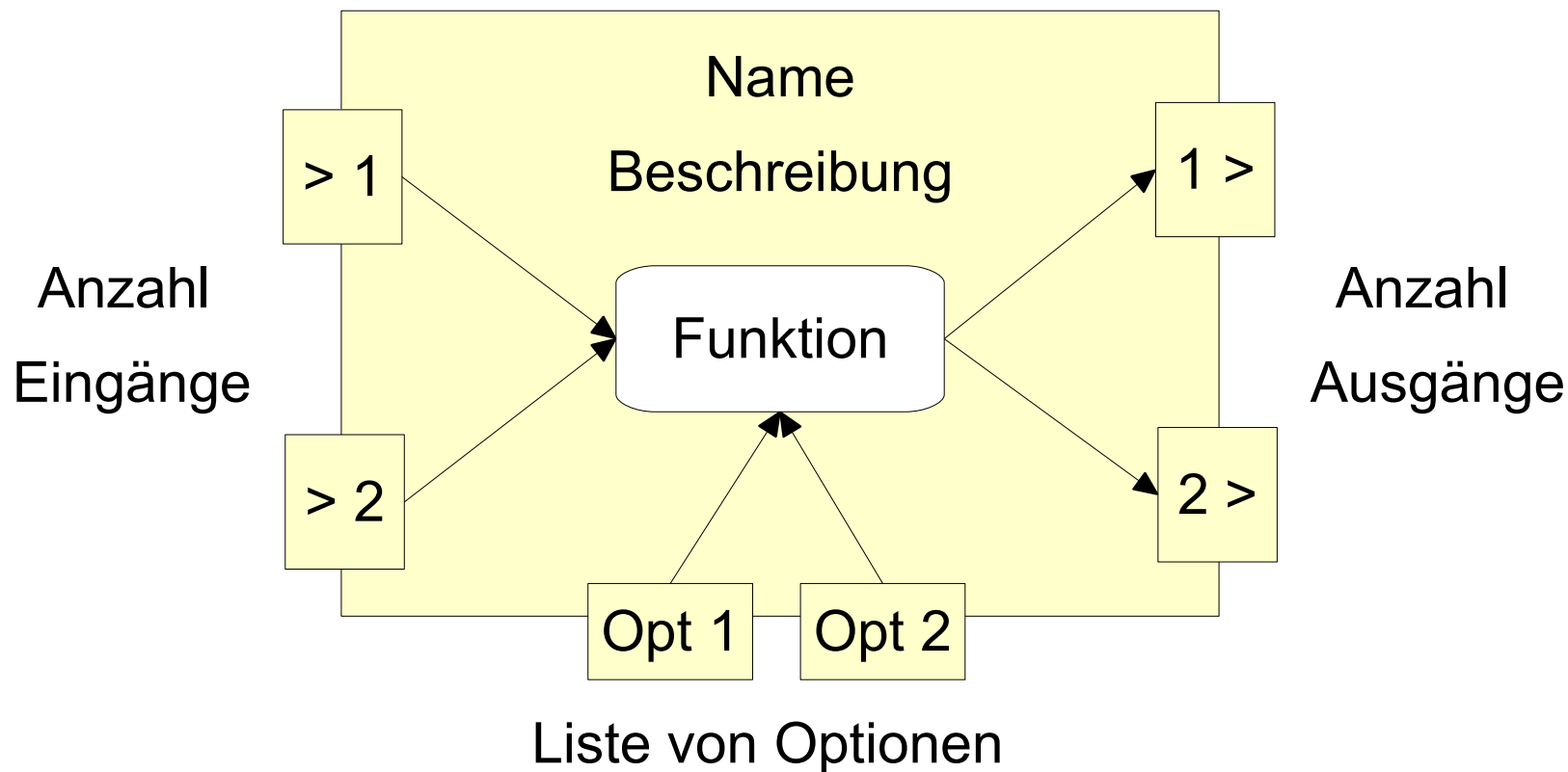
Applicator

Pipelines

MDOs

Übersicht

Vorführung



Optionen

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

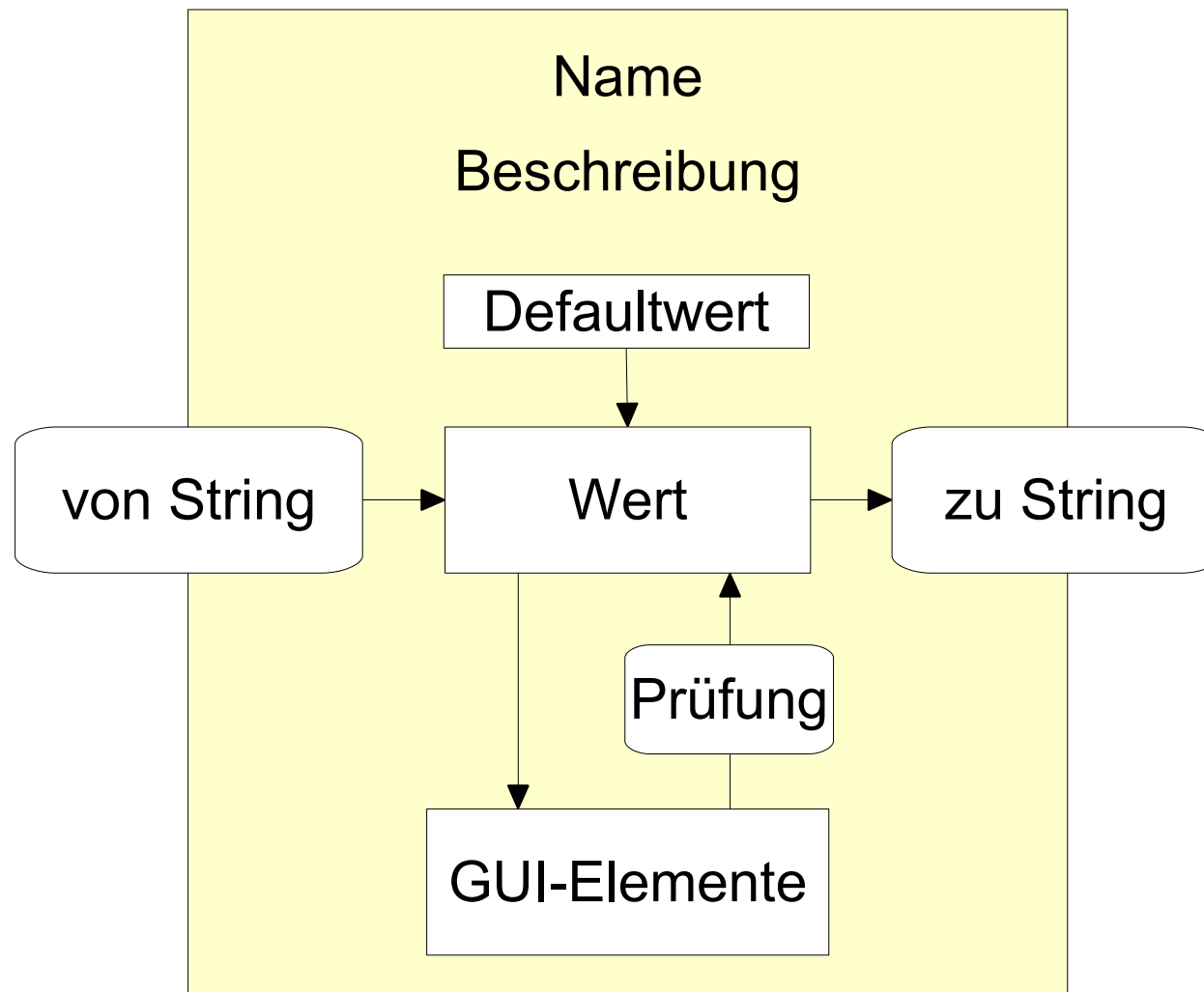
Applicator

Pipelines

MDOs

Übersicht

Vorführung



Module und Optionen

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

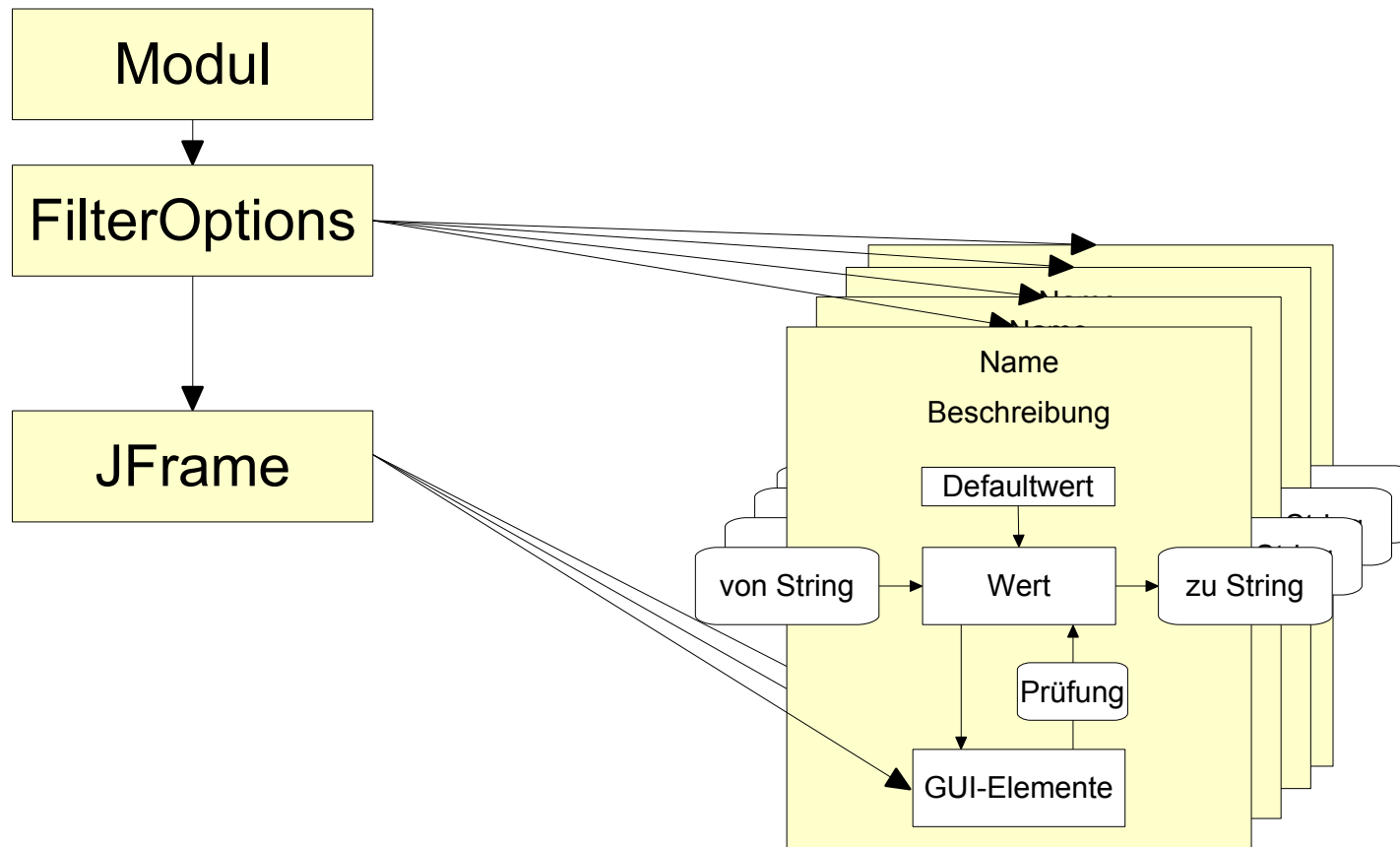
Applicator

Pipelines

MDOs

Übersicht

Vorführung



Applicator

- Wird vom Mayday PluginManager gestartet
- Drei Schritte:
 - Auswahl des anzuwendenden Moduls
 - Zuordnung von Eingabedaten auf die Eingänge des Moduls (wenn mehr als ein Eingang vorhanden ist)
 - Setzen von Optionen
- Ruft das gewählte Modul *n mal* auf
- Gibt Feedback über aktuellen Stand (%, Fehler)
- Räumt bei Fehlern im Speicher auf
- Bietet Zugang zum Designer

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

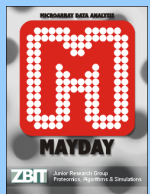
Applicator

Pipelines

MDOs

Übersicht

Vorführung



Pipelines

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

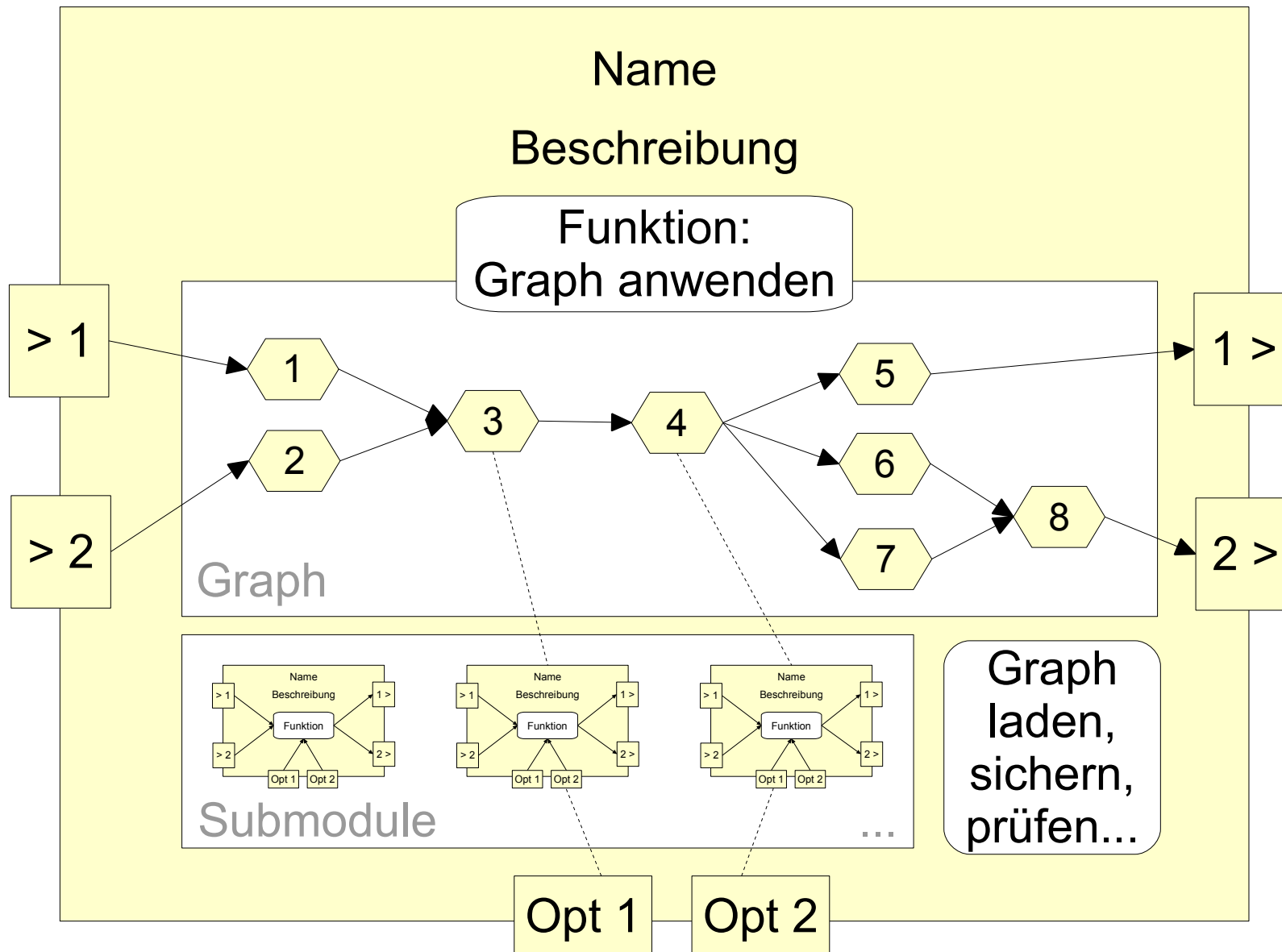
Applicator

Pipelines

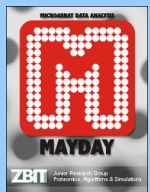
MDOs

Übersicht

Vorführung



Liste von externalisierten Optionen



Pipelines

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

- Laden und Speichern
 - Struktur des Graphen
 - Optionen der Submodule
 - Textformat beschreibt Daten in Baumstruktur
- Graph anwenden
 - Ausführungsreihenfolge berechnen
 - Submodul anwenden,
Ausgabe an nächstes Submodul geben
- Graph prüfen
 - Direkte Zyklen
 - Verschachtelte Zyklen (nested pipelines)

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

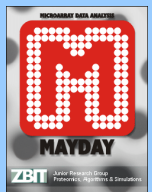
Applicator

Pipelines

MDOs

Übersicht

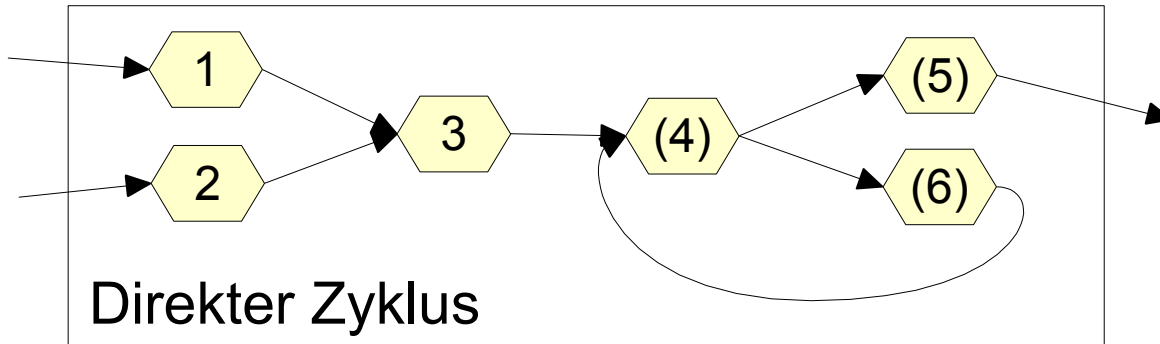
Vorführung



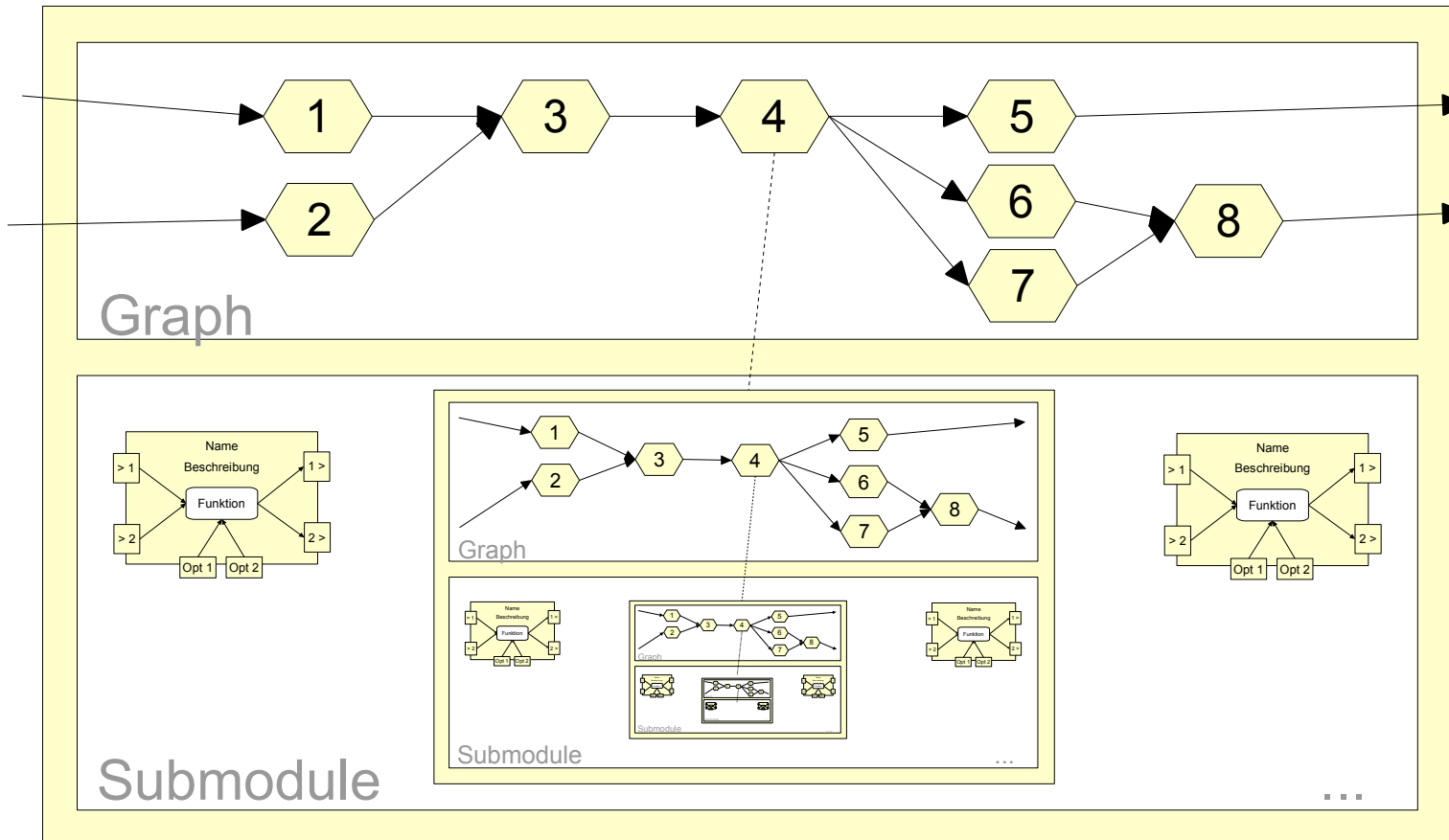
Pipelines: Zyklen im Graph

MPF
*A processing
framework
for Mayday*

Florian Battke 2006



Indirekter Zyklus



Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

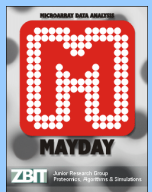
Applicator

Pipelines

MDOs

Übersicht

Vorführung



MaydayDataObject

- Kapselt eine ProbeList
- Stellt sicher, daß Veränderungen
 - nicht die ursprünglichen Daten betreffen
 - ganz oder gar nicht in Mayday auftreten
 - für Programmierer einfach durchzuführen sind
 - möglichst wenig Speicher verwenden
- Kümmert sich um
 - Einfügen und Löschen von Probes
 - Erzeugen eindeutiger Namen
 - Annotieren neuer/geänderter Probes
 - Weiterleitung von Daten zwischen Subfiltern
 - ...

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

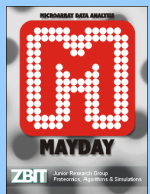
Applicator

Pipelines

MDOs

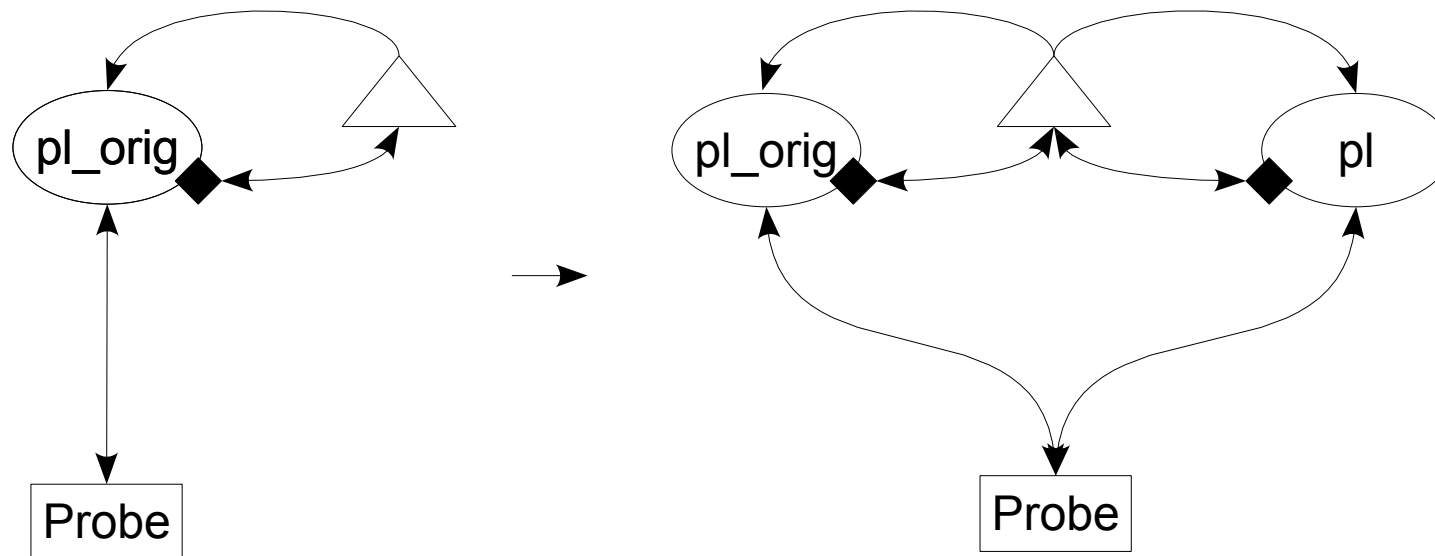
Übersicht

Vorführung



MaydayDataObject: Beispiel

- makeUniqueProbe: Bevor eine Probe verändert werden darf, muß sie geklont werden, damit das Original unverändert bleibt.
- Zunächst muß die ProbeList geklont werden



MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

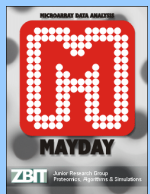
Applicator

Pipelines

MDOs

Übersicht

Vorführung



MaydayDataObject: Beispiel

MPF
A processing
framework
for Mayday

Florian Battke 2006

- makeUniqueProbe: Bevor eine Probe verändert werden darf, muß sie geklont werden, damit das Original unverändert bleibt.

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

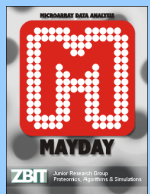
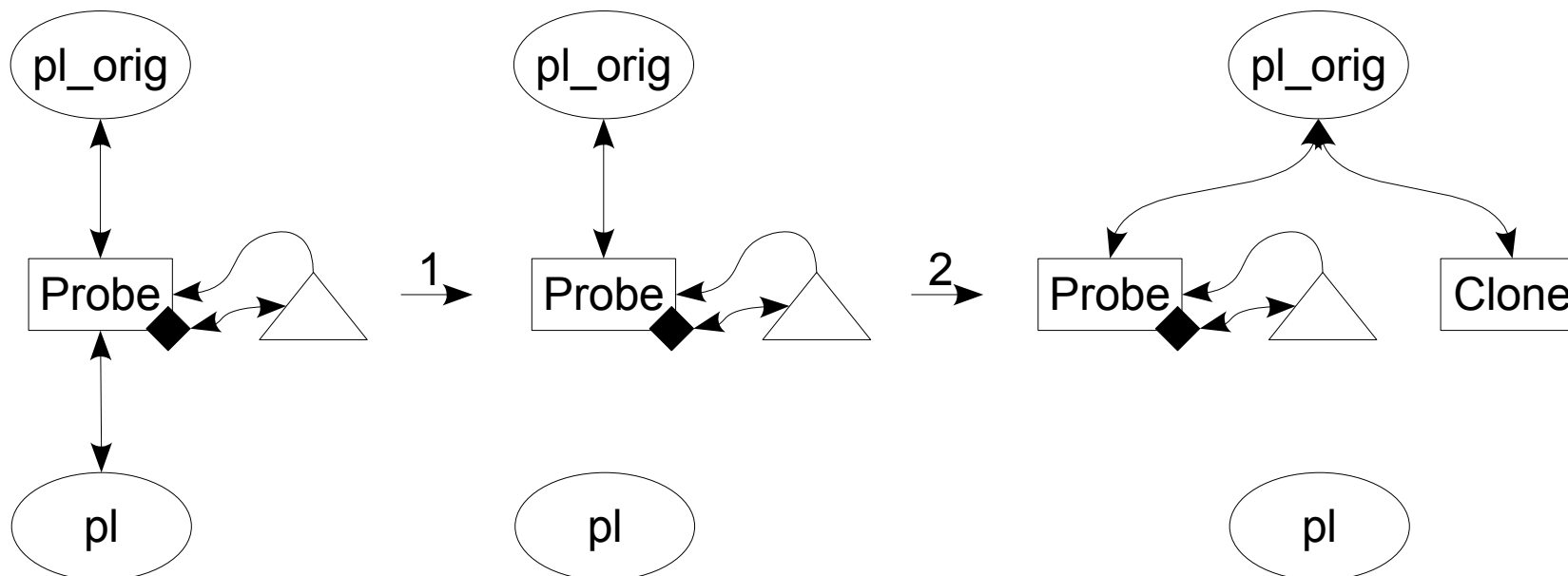
Applicator

Pipelines

MDOs

Übersicht

Vorführung



MaydayDataObject: Beispiel

MPF
*A processing
framework
for Mayday*

Florian Battke 2006

- makeUniqueProbe: Bevor eine Probe verändert werden darf, muß sie geklont werden, damit das Original unverändert bleibt.
 - 3) Klon umbenennen, damit es keine Verwechslungen gibt
 - 4) Klon aus der ursprünglichen Probelist entfernen
 - 5) Metadaten klonen

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

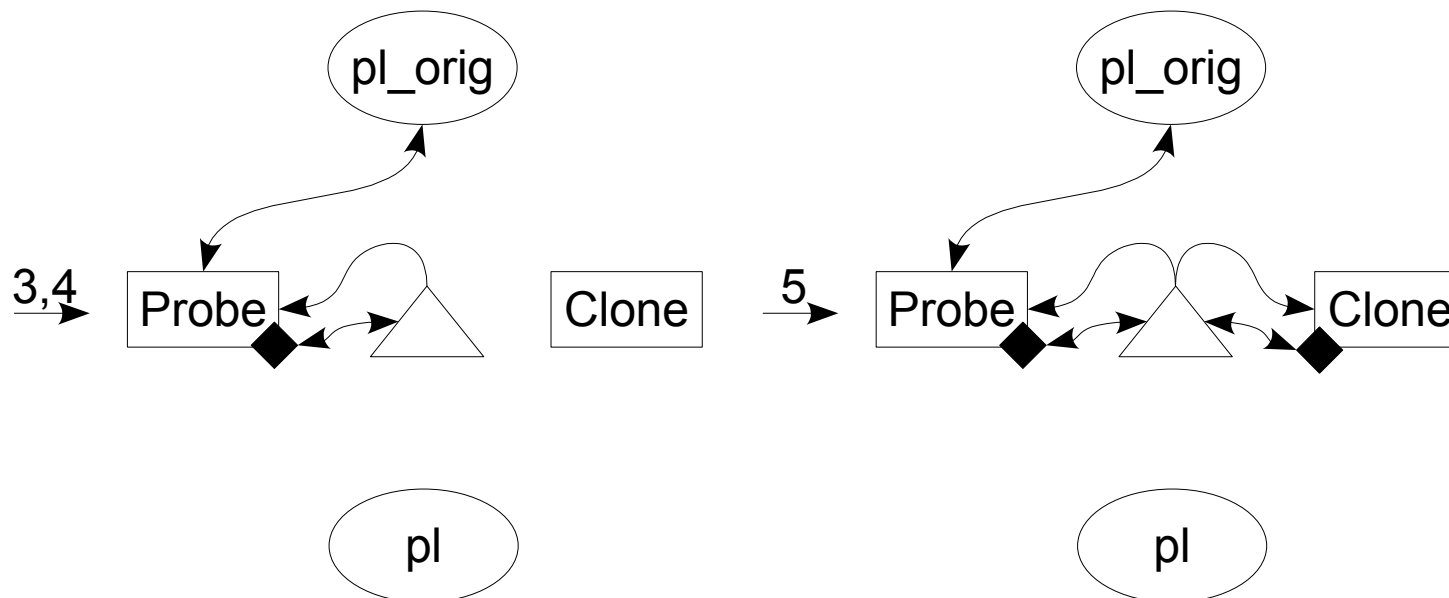
Applicator

Pipelines

MDOs

Übersicht

Vorführung



MaydayDataObject: Beispiel

MPF
*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

Applicator

Pipelines

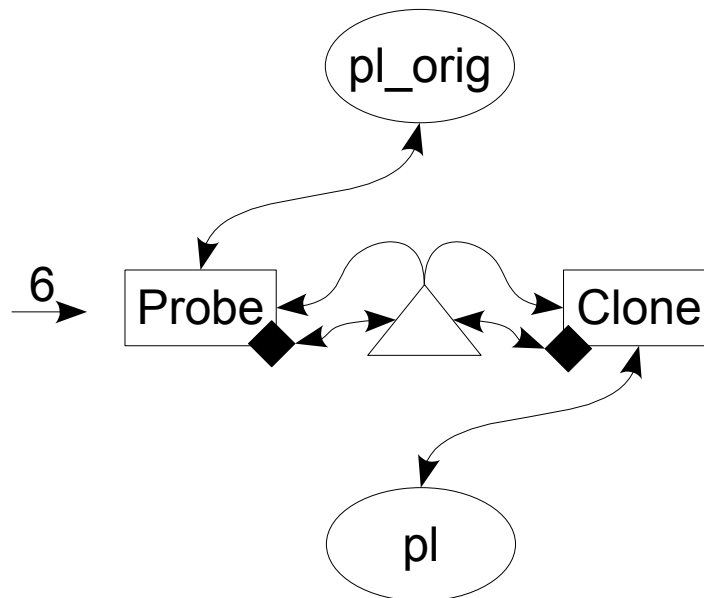
MDOs

Übersicht

Vorführung

- makeUniqueProbe: Bevor eine Probe verändert werden darf, muß sie geklont werden, damit das Original unverändert bleibt.

6) Klon in die geklonte Probelist einfügen



- Programmierer muß makeUniqueProbe in der Regeln nicht explizit aufrufen.



MaydayDataObject: Nutzen

MPF
*A processing
framework
for Mayday*

Florian Battke 2006

- Beispiel: Invertierbare Filterfunktion

```
public void execute() {  
    OutputData[0]=InputData[0];  
    for (Probe pb : OutputData[0]) {  
        boolean keep = checkCriteria(pb);  
        if (invert.Value) keep=!keep;  
        if (!keep) OutputData[0].remove(pb);  
        if (isCancelled()) return;  
    }  
}
```

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

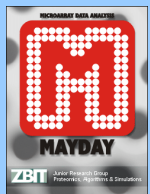
Applicator

Pipelines

MDOs

Übersicht

Vorführung



MaydayDataObject: Nutzen

MPF
*A processing
framework
for Mayday*

Florian Battke 2006

- Beispiel: Invertierbare Filterfunktion

```
public void execute() {  
    OutputData[0]=InputData[0];  
  
    for (Probe pb : OutputData[0]) {  
        boolean keep = checkCriteria(pb);  
        if (invert.Value) keep=!keep;  
        if (!keep) OutputData[0].remove(pb);  
        if (isCancelled()) return;  
    }  
}
```

Zunächst den Eingang direkt auf
den Ausgang legen.

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

Applicator

Pipelines

MDOs

Übersicht

Vorführung



MaydayDataObject: Nutzen

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

- Beispiel: Invertierbare Filterfunktion

```
public void execute() {  
    OutputData[0]=InputData[0];  
    for (Probe pb : OutputData[0]) {  
        boolean keep = checkCriteria(pb);  
        if (invert.Value) keep=!keep;  
        if (!keep) OutputData[0].remove(pb);  
        if (isCancelled()) return;  
    }  
}
```

Java 1.5 – for-Schleife iteriert
sicher über alle Probes im MDO:

Iterator übersteht Einfügen und
Entfernen von Probes, iteriert nur
über Probes, die zu Anfang der
Schleife im MDO waren.

=> Besucht jede Probe
genau einmal

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

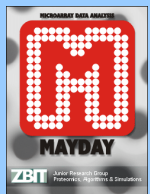
Applicator

Pipelines

MDOs

Übersicht

Vorführung



MaydayDataObject: Nutzen

MPF
*A processing
framework
for Mayday*

Florian Battke 2006

- Beispiel: Invertierbare Filterfunktion

```
public void execute() {  
    OutputData[0]=InputData[0];  
    for (Probe pb : OutputData[0]) {  
        boolean keep = checkCriteria(pb);  
        if (invert.Value) keep=!keep;  
        if (!keep) OutputData[0].remove(pb);  
        if (isCancelled()) return;  
    }  
}
```

Mit der Probe arbeiten

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

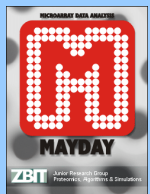
Applicator

Pipelines

MDOs

Übersicht

Vorführung



MaydayDataObject: Nutzen

MPF
*A processing
framework
for Mayday*

Florian Battke 2006

- Beispiel: Invertierbare Filterfunktion

```
public void execute() {  
    OutputData[0]=InputData[0];  
  
    for (Probe pb : OutputData[0]) {  
        boolean keep = checkCriteria(pb);  
        if (invert.Value) keep=!keep;  
        if (!keep) OutputData[0].remove(pb);  
        if (isCancelled()) return;  
    }  
}
```

Der Wert einer Option
bestimmt, welche Probes
vom Filter entfernt werden.

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

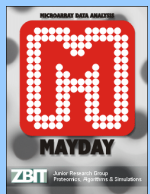
Applicator

Pipelines

MDOs

Übersicht

Vorführung



MaydayDataObject: Nutzen

MPF
*A processing
framework
for Mayday*

Florian Battke 2006

- Beispiel: Invertierbare Filterfunktion

```
public void execute() {  
    OutputData[0]=InputData[0];  
    for (Probe pb : OutputData[0]) {  
        boolean keep = checkCriteria(pb);  
        if (invert.Value) keep=!keep;  
        if (!keep) OutputData[0].remove(pb);  
        if (isCancelled()) return;  
    }  
}
```

Entfernen einer Probe

- Stört den Iterator nicht
- MDO prüft, ob pb ein Klon war, und gibt evtl. Speicher frei
- Ggf. muß MDO zunächst die ProbeList klonen

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

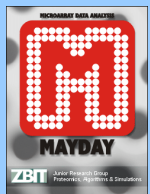
Applicator

Pipelines

MDOs

Übersicht

Vorführung



MaydayDataObject: Nutzen

MPF
*A processing
framework
for Mayday*

Florian Battke 2006

- Beispiel: Invertierbare Filterfunktion

```
public void execute() {  
    OutputData[0]=InputData[0];  
  
    for (Probe pb : OutputData[0]) {  
        boolean keep = checkCriteria(pb);  
        if (invert.Value) keep=!keep;  
        if (!keep) OutputData[0].remove(pb);  
        if (isCancelled()) return;  
    }  
}
```

Abbruch durch Benutzer

- Einfach aufhören
- Applicator und MDO
geben Speicher frei

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

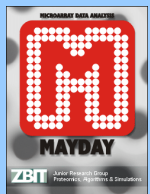
Applicator

Pipelines

MDOs

Übersicht

Vorführung



MaydayDataObject: Nutzen

MPF
*A processing
framework
for Mayday*

Florian Battke 2006

- Beispiel: Invertierbare Filterfunktion

```
public void execute() {  
    OutputData[0]=InputData[0];  
  
    for (Probe pb : OutputData[0]) {  
        boolean keep = checkCriteria(pb);  
  
        if (invert.Value) keep=!keep;  
  
        if (!keep) OutputData[0].remove(pb);  
  
        if (isCancelled()) return;  
    }  
}
```

Erfolgreiches Ende

In Pipelines: Ergebnis
wird weitergegeben an
das nächste Modul.

Sonst: Applicator fügt
das Ergebnis in die
Datenstrukturen von
Mayday ein.

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

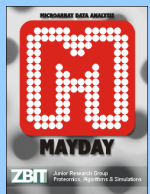
Applicator

Pipelines

MDOs

Übersicht

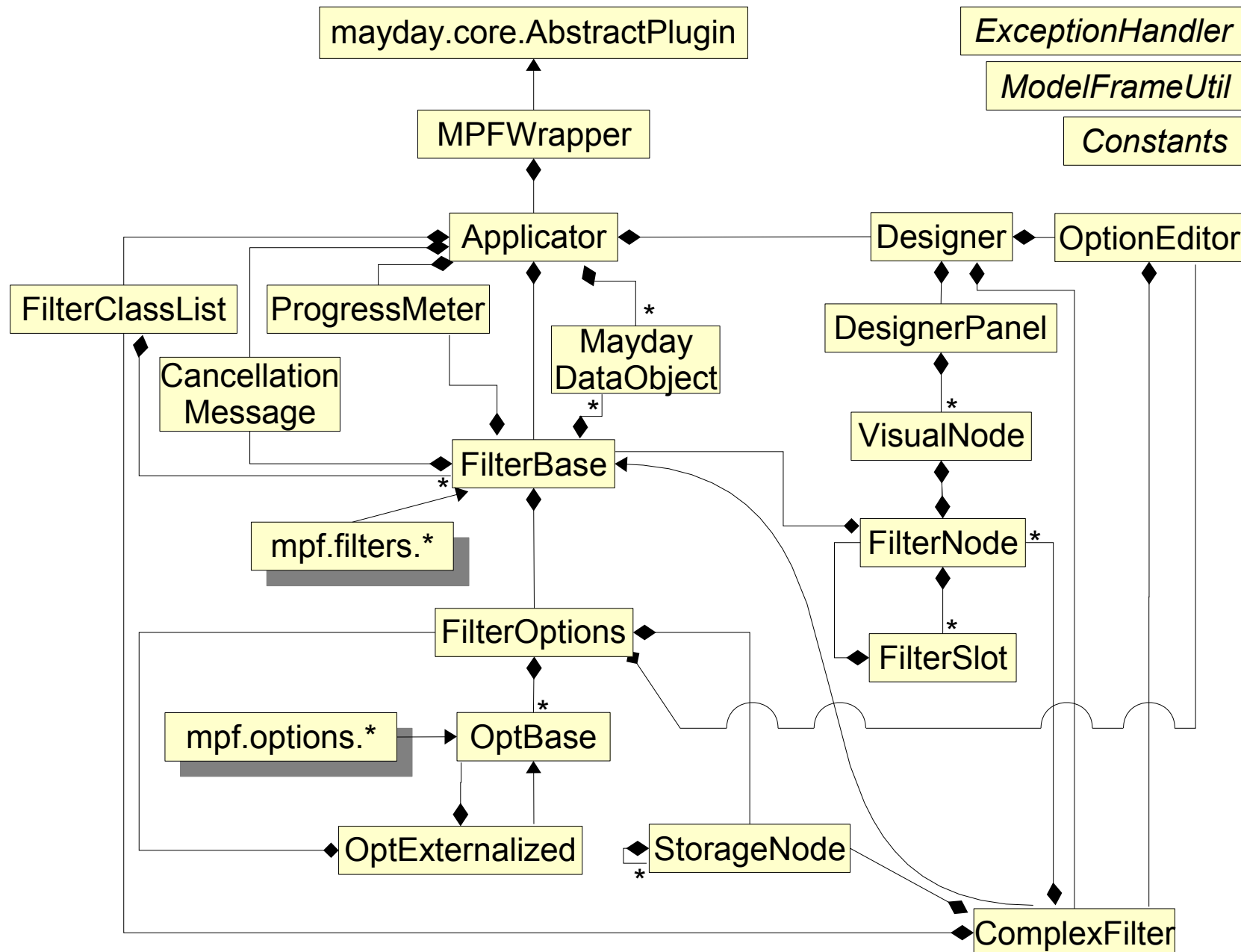
Vorführung



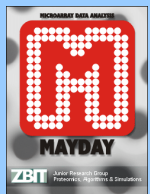
(Un-)Übersicht über alle Klassen

MPF
*A processing
framework
for Mayday*

Florian Battke 2006



Motivation
Anforderungen
Aufbau
Einfache Module
Optionen
Applicator
Pipelines
MDOs
Übersicht
Vorführung



(+weitere interne Klassen)

Übersicht

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

- Bislang vorhandene Module
 - Imputation (verschiedene Methoden)
 - Normalisierung (z-Transformation)
 - Log Transformation
 - Filtern anhand von MIOs (String, Int, Double)
 - Expression Value Corridor
 - Wrapper für das RPlugin
- Was noch schön wäre:
 - Maydays Visualisierungsplugins einbinden
 - Visualisierung durch R-Funktionen

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

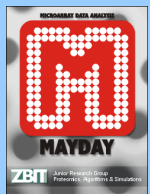
Applicator

Pipelines

MDOs

Übersicht

Vorführung



Vorführung

MPF

*A processing
framework
for Mayday*

Florian Battke 2006

Motivation

Anforderungen

Aufbau

Einfache Module

Optionen

Applicator

Pipelines

MDOs

Übersicht

Vorführung

- Spellmans cdc15-Datensatz (Hefe-Zellzyklus)
 - Große Anzahl von fehlenden Werten
 - Einige Gene sind mit Zellzyklus-Phasen annotiert
- Fragestellung
 - Zeigt sich die durch Clustering ermittelte Ähnlichkeit von Genen auch in der Annotation?
- Ablauf
 - Daten laden, Annotation als MIOs laden
 - Imputation, Normalisierung
 - Herausfiltern aller Gene ohne Annotation
 - Clustering, Darstellung der Cluster
 - Einblenden aller Gene einer Phase (z.B. G1)

