AID und Filter

Treiber (Filter als Komponenten)

Jan Sebastian Siwy

Annahme

- Filter: Programm `ftp`
- Nutzen: FTP-Bibliothek
- Treiber für `ftp` gestartet
  - `AID_System` am ORB angemeldet
  - Methoden `FTP_NEWx` (Initialisierer in AID)

Initialisierung

- Vertreter ruft `FTP_NEWx` auf
- Prozess erzeugen
  - aktuelles Verzeichnis
  - ausführbare Datei
  - Argumente
  - Umgebungsvariablen

Initialisierung

- Beispiel in AID

```
INIT string dir, string user, string host
    | dir dir
    | cmd "ftp"
    | arg user "@" host
    | arg ...
    | env "foo" = "bar"
```

  - Standardwerte als Annotation bei **INTERFACE**

`IN`-Ereignisse

- Vertreter ruft Methode in `FTP_IN` auf
- Schreiben nach Kanal 0 (`stdin`)
  - Ausgabe muss formatiert werden

IN-Ereignisse

- Beispiel in AID

```
IN changeDirectory string dir
    | stdin "cd " dir
```

OUT-Ereignisse

- Lesen von Kanal 1 (`stdout`) und 2 (`stderr`)
- Interpretieren, welches Ereignis gemeint ist
  - **`OUT`**- und **`OUTIN`**-Ereignisse
  - Antwort eines **`INOUT`**-Ereignisses
  - gar keins
- Treiber ruft Methode in `FTP_OUT` auf

OUT-Ereignisse

- Beispiel in AID

```
OUT connectionClosed
    | stdout "421 Timeout"
    | stdout "Foo", stderr "Bar"
```

`INOUT`-Ereignisse

- Vertreter ruft Methode in `FTP_IN` auf
- Schreiben nach Kanal 0 (`stdin`)
- Lesen von Kanal 1 (`stdout`) und 2 (`stderr`)
- Interpretieren, was gemeint ist
- Antwort füllen oder Ausnahme melden

INOUT-Ereignisse

- Beispiel in AID

```
INOUT changeDirectory string dir
    RAISES NotFound
    | stdin "cd " dir
    | stdout "250 CWD"
    | NotFound stdout "550 " message
```

`OUTIN`-Ereignisse

- Lesen von Kanal 1 (`stdout`) und 2 (`stderr`)
- Interpretieren, welches Ereignis gemeint ist
- Treiber ruft Methode in `FTP_OUT` auf
- Schreiben nach Kanal 0 (`stdin`)
  - reguläre Antwort oder Ausnahme

OUTIN-Ereignisse

- Beispiel in AID

```
OUTIN doSomething
    RAISES Failed
    | stdout "doSomething"
    | stdin "done"
    | Failed stdin "forget it"
```

AID und Filter

Vertreter (Komponenten als Filter)

Jan Sebastian Siwy

Annahme

- Komponente: FTP-Bibliothek
- Nutzen: Programm `ftp`
- Treiber für FTP-Bibliothek gestartet
  - `AID_System` am ORB angemeldet
  - Methoden `FTP_NEWx`

Initialisierung

- Umgebung startet neuen Vertreter
- Vertreter bestimmt `AID_System`
  - Adresse
  - Port
- Vertreter ruft `FTP_NEWx` auf
- Verweis auf Treiber vorhanden

Initialisierung

- Beispiel in AID

```
INIT string dir, string user, string host
    | arg user "@" host
    | arg ...
    | dir dir
    | "localhost" _host
    | "12345" _port
    | "FTPlib" _component
    | "FTP" _interface
```

  - Standardwerte als Annotation bei **INTERFACE**

Ereignisse

- Annotationen bleiben gleich
  - `stdin` gibt keine Formatierung an,
    sondern entscheidet welches Ereignis ausgelöst wird
  - Interpretation der Umgebung (Filterwelt)

  - `stdout` und `stderr` bestimmen nicht das Ereignis,
    sondern geben die Formatierung an
  - Ausgabe für Umgebung (Filterwelt)

AID und Filter

Typisierte Ereignisse bei Zeichenkettenströmen

Jan Sebastian Siwy

Trennung der Ereignisse

- Erkennung, wann Ereignis abgeschlossen ist
  - Trennzeichen explizit angeben

```
| ..., delimiter "\n"
```

  - Vorgaben bei **`component`** und **`interface`**

```
| delimiter "\n"
```

Datentypen

- Filter arbeiten nur mit Zeichenketten
- Typanpassung gemäß Typ der AID-Variable

Datentypen

- Typanpassung
  - primitive Datentypen
  - Felder

```
| ..., [var].delimiter ", "
```

  - Tupel mit Punktschreibweise

```
| ... var.member ...
```

  - Tupel mit Vorlage

```
TUPLE socket
    | host ":" port
    string host, int port
```

Danke