**MVC-Konzept beim FlowDiagram** – zeigt die Zusammenhänge des Ablaufdiagramms zum grafischen erstellen des c't-Bot-Programms...

View/Controller (Komponenten für Oberfläche) | Model

GUI

FlowDiagram (FD)

**Enthält** ein FDM und CECs,
FlowDiagramModelListener, der beim FDM registriert ist
**Verantwortlich für** deselektieren aller CECs bei Klick auf die FD-Fläche. Vergabe der Instanz-UIDs der Ces – dadurch wird's eindeutig in diesem FD/FDM. Zeichnen und überwachen/ändern der Pfade.

**Selection (Ausgewählte CECs oder auch Pfeile)**
SelectAll(), clearSelection(), addTo- und removeFromSelection(CEC), isSelected(CEC)
**Kernfunktionen**
createControlElementComponent(CEC, Point), deleteSelectedElements(), nextControlElementUID(), setSelectedAsRootControlElement(), userChangeParameter()

Enthält mehrere

ControlElementComponent (CEC)

**Enthält** zugehöriges CE und kann über getParentFlowDiagram() auf sein übergeordnetes FD zugreifen (null wenn „parentless")
**Verantwortlich für** die meisten Benutzereingaben, wie doppelklick (Parameter ändern), Drag (verschieben/bewegen), DnD von einem CEC zum anderen um bei CE das nextControlElement zu setzen, klicken zum Auswählen (Auswahl wird tatsächlich in FD festgelegt), Strg oder Shift + klicken zur Mehrfachauswahl (Path kümmert sich nicht selbst darum, weil es kein Component ist).
**Kernfunktionen**
Konstruktor der CE erwartet, Funktionen, die Start-/Endpunkte der Pfade zurückgeben, userChangeParameter-Behandlung

Informiert über die Änderungen im CE und zusätzlich über: CE added/removed, rootCE changed

FlowDiagramModel (FDM)

**Enthält** CEs,
ControlElementListener, der beim jedem CE registriert ist.

Legt Start-Steuerelement fest. Beinhaltet CEs die zum Model gehören.
**Kernfunktionen:**
makePath(CE,CE, path), setRootCE(CE), add-/removeCE(CE)
autoAdd gibt an, ob CEs, die als Root gesetzt werden oder zw. denen ein Pfad gemacht wird, automatisch zum Model hinzugefügt werden, falls sie noch nicht drin sind.

Path

Ist Model für Pfad zwischen zwei Steuerelementen. Referenziert aber nicht CEs, sondern CECs! Ist für nichts verantwortlich!

Referenziert das zugehörige CE

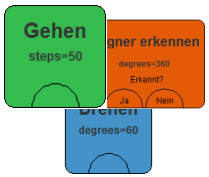
Informiert über Änderungen: nextCE changed, parameter-list changed, parameter-value changed

ControlElement (CE)

Model für ein Steuerelement im Ablaufdiagramm.
Attribute:
Id, Id für den Typ, Anzahl der Pfade zu folgenden Steuerelementen, Name, Parameter
Kernfunktionen:
setParameterValue(String), setNextCE(path, CE) und die ganzen Getter

**Begriffe/Abkürzungen**

- **ControlElement, (Model für) Steuerelement, Abk.: CE**
  Das Model für ein Steuerelement; enthält nur die Daten, die für das Programm/FDM nötig sind!




- **ControlElementComponent, Steuerelement, Abk.: CEC**
  Die View für ein zugehöriges CE – enthält alle Daten die eben für die View notwendig sind und referenziert zusätzlich das zugehörige CE (Referenz kann nicht geändert werden).

- **FlowDiagramModel, Model, Abk.: FDM**
  Das gesamte Model für das Ablaufdiagramm. Das Model *ist gleich* das Programm für den Roboter. Es definiert alle Sachen die nötig sind um das Programm abzuarbeiten und enthält die zugehörigen CEs.

- **FlowDiagram, Ablaufdiagramm, Abk.: FD**
  Die View („obereste“ Komponente), die *alles* enthält.

**Außerdem:**

- **Root CE, Start-Steuerelement**
  ist der Anfangspunkt/Start-Steuerelement, da, wo das Programm beginnt.
- **Path**
  ist ein Pfad (=Pfeil) zwischen zwei Steuerelementen.

Hinweis: Man lege sich für Folgendes am besten die erste Seite des Dokuments zur Übersicht über Zusammenhänge zwischen FD, FDM, CEC und CE neben dieses Blatt!

## Kurze Erklärung des MVC (Model-View-Controller)-Konzepts

Das läuft so: Wenn der User Änderungen macht, gibt er diese mit der Maus oder Tastatur in der View (FD und CEC) ein – wo auch sonst, er sieht ja nur die View.

Die View/Controller macht aber erst mal gar nichts, außer die Änderungen auf direktem Weg ins Model (v.a. FDM – FDM kapselt CEs) zu übertragen. D.h. die View wird nicht upgedated, zeigt keine Änderungen an.
Das Model (FDM) stellt für solche Model-Änderungen eine schöne API bereit.
Hier werden also die Änderungen gemacht. Jetzt ist es die Aufgabe des Models, die View darüber zu informieren, so dass die View sich eben updaten kann. (Wer denkt „Da hätte die View ja auch gleich updaten können...“ dem sage ich: „Stell die MVC-Architektur nicht in Frage.“)

Die View wird also informiert, dass sich das Model geändert hat, und tut sich daraufhin updaten.

Das läuft hier bei allen Aktionen so.

## Erklärung des Konzepts anhand eines Beispiels

Am Anfang haben wir ein leeres FlowDiagram.

Neues Steuerelement hinzufügen
Jetzt wird aus der Liste rechts ein Steuerelement auf das Ablaufdiagramm gezogen. Hier passiert folgendes: Beim Drop wird beim FlowDiagram die Methode createControlElement() aufgerufen. Diese fügt das, im Argument angegebene, ControlElement (kein ControlElementComponent!) dem FDM mit der Methode addControlElement() hinzu. Das führt dazu, dass das Model alle FlowDiagramListener informiert, dass eben gerade ein CE hinzugefügt wurde. Der Listener im FD reagiert darauf, indem er dem FD ein CEC (korrespondierend zum neuen CE) hinzugefügt – falls so eines nicht schon existiert, was nicht der Fall sein sollte/dürfte.
Anschließend wird die Position des Steuerelement, das zum neuen CE korrespondiert, gesetzt. Dies geschieht wieder in der ersten Methode createControlElement().

Pfad zu sich selbst erstellen
Bevor der Drag gestartet wird, wird geschaut, ob überhaupt von einem der Halbkreise aus gezogen wurde (das sind die Pfad-Startpunkte). Wenn ja, wird sich das „Source“-CE gemerkt (zu Drag & Drop siehe „Drag & Drop und Verschieben von Steuerelementen.txt“). Beim Drop (hier im Beispiel auf sich selbst) wird im FDM die Methode makePath() aufgerufen, die wiederum die Methode setNextControlElement() vom „Source“-CE aufruft, welche die eigentliche Verbindung herstellt/speichert. Dabei wird der ControlElementListener vom FDM informiert, dieser informiert den FlowDiagramListener vom FD, und dieser (FD ist ja die View) erstellt im FD ein Path-Objekt, fügt es der Liste der Path-Objekte hinzu und zeichnet neu. Beim Zeichnen werden CECs und Pfad-Objekte berücksichtigt!