Pflichtenheft

Grafischer Simulator für BPMN-Modelle

Stefan Schweitzer

Inhaltsverzeichnis

Zielbestimmung

Ziel des Projektes ist es, einen grafischen Simulator für BPMN-Modelle zu entwickeln. Hierfür sollen BPMN-Modelle aus einem existierenden Prozessmodellierungstool exportiert werden (im XML-basierten BPMN 2.0-Format). Die zu entwickelnde Software soll das betreffende Model einlesen, grafisch anzeigen und den Markenfluss für das Modell grafisch animieren.

Muss Anforderungen

- ✔ Einlesen von Modellen im XML-basierten BPMN 2.0-Format
- ✔ Grafische Darstellung des BPMN Modells
- ✔ Animation des Diagrammablaufs
- ✔ Unterstützung der folgenden BPMN-Elemente:
  - ✔ Ereignisse
    - ✔ Start
    - ✔ Ende
    - ✔ Terminierung
  - ✔ Aktivitäten
    - ✔ Tasks
    - ✔ Unterprozesse
      - ✔ Zusammengeklappt
      - ✔ Aufgeklappt
  - ✔ Gateways
    - ✔ Exklusiv
    - ✔ Inklusive
    - ✔ Parallel
  - ✔ Sequenzflüsse
    - ✔ Unkontrollierter Fluss
    - ✔ Bedingter Fluss
    - ✔ Standardfluss

Kann Anforderungen

Unterstützung weiterer BPMN-Elemente. z.B.:

- ✔ Assoziationen
- ✗ Datenobjekte
- ✔ Kommentare

- ✔ Gruppen
- ✔ Pools / Lanes

Abgrenzungskriterien

- Es soll nicht möglich sein das Modell zu ändern.
- Es soll muss nicht das exakte Datenmodell mit allen Eigenschaften der Spezifikation in der Software abgebildet sein.
- Es soll keine Überprüfung des BPMN-Modells auf Korrektheit erfolgen.

Produkteinsatz

Der grafische BPMN-Simulator ist zu Lehrzwecken gedacht. Als Referenzsoftware zum erstellen und exportieren der Diagramme dient Signavio.

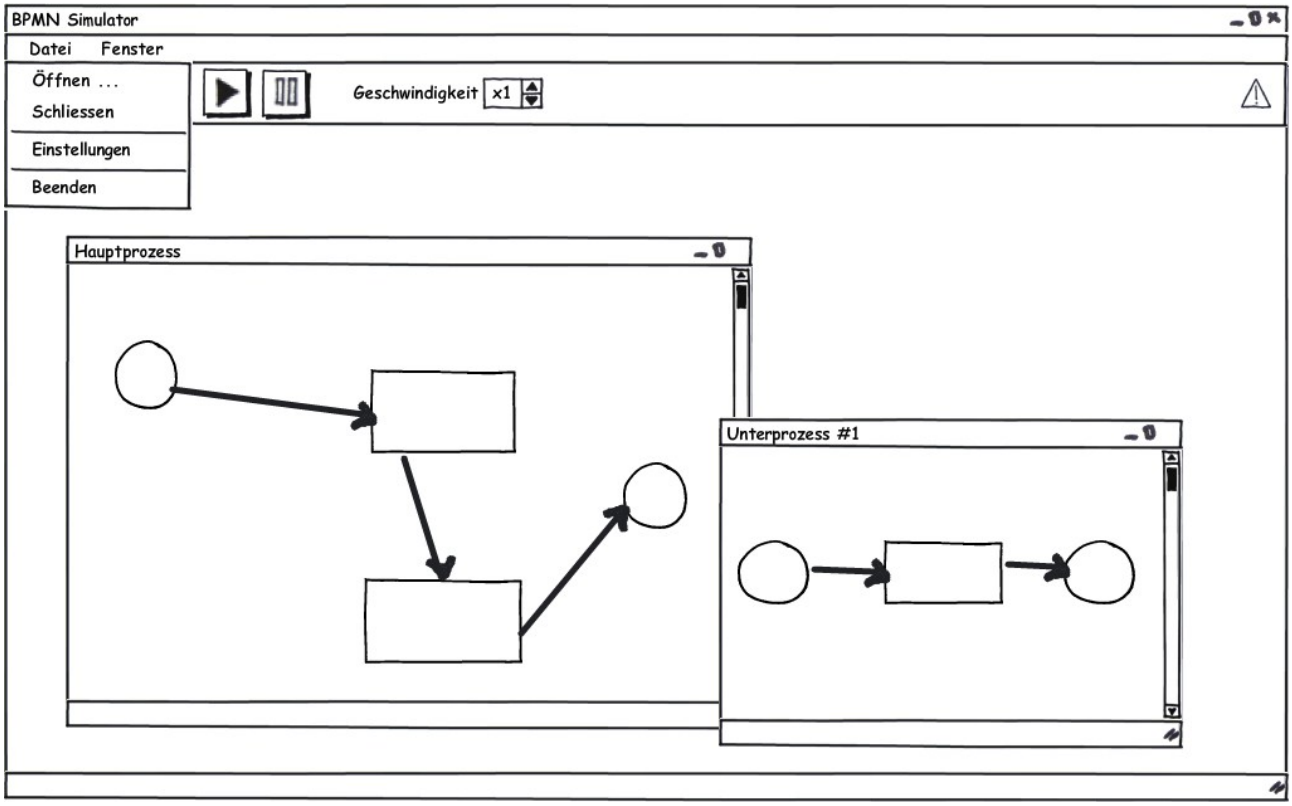
Nichtfunktionale Anforderungen

- Wartbarkeit
- Änderbarkeit
- Erweiterbarkeit
- Korrektheit

Produktumgebung

Desktop-Betriebssystem mit Unterstützung von Java Version 7 oder höher.

Benutzeroberfläche



Die Software kann nur jeweils eine Modell gleichzeitig öffnen. Für zusammengeklappte Unterprozesse gibt es je ein extra Fenster indem der komplette Unterprozess angezeigt wird.

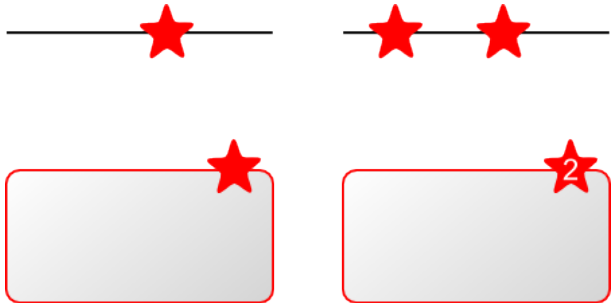
Durch das Klicken auf ein Startevent beginnt der Markendurchlauf und kann pausiert oder abgebrochen werden. Es ist möglich das mehrere Marken gleichzeitig das Modell durchlaufen.

Darstellung des Markendurchlaufs

Wenn die Marke in einen zusammengeklappten Unterprozess läuft, wird dieser markiert und die Marke durchläuft den Unterprozess in dem extra Prozessfenster. Erst danach läuft die Marke in dem ursprünglichen Fenster ab dem zusammengeklappten Unterprozess weiter.

Darstellung der Marken

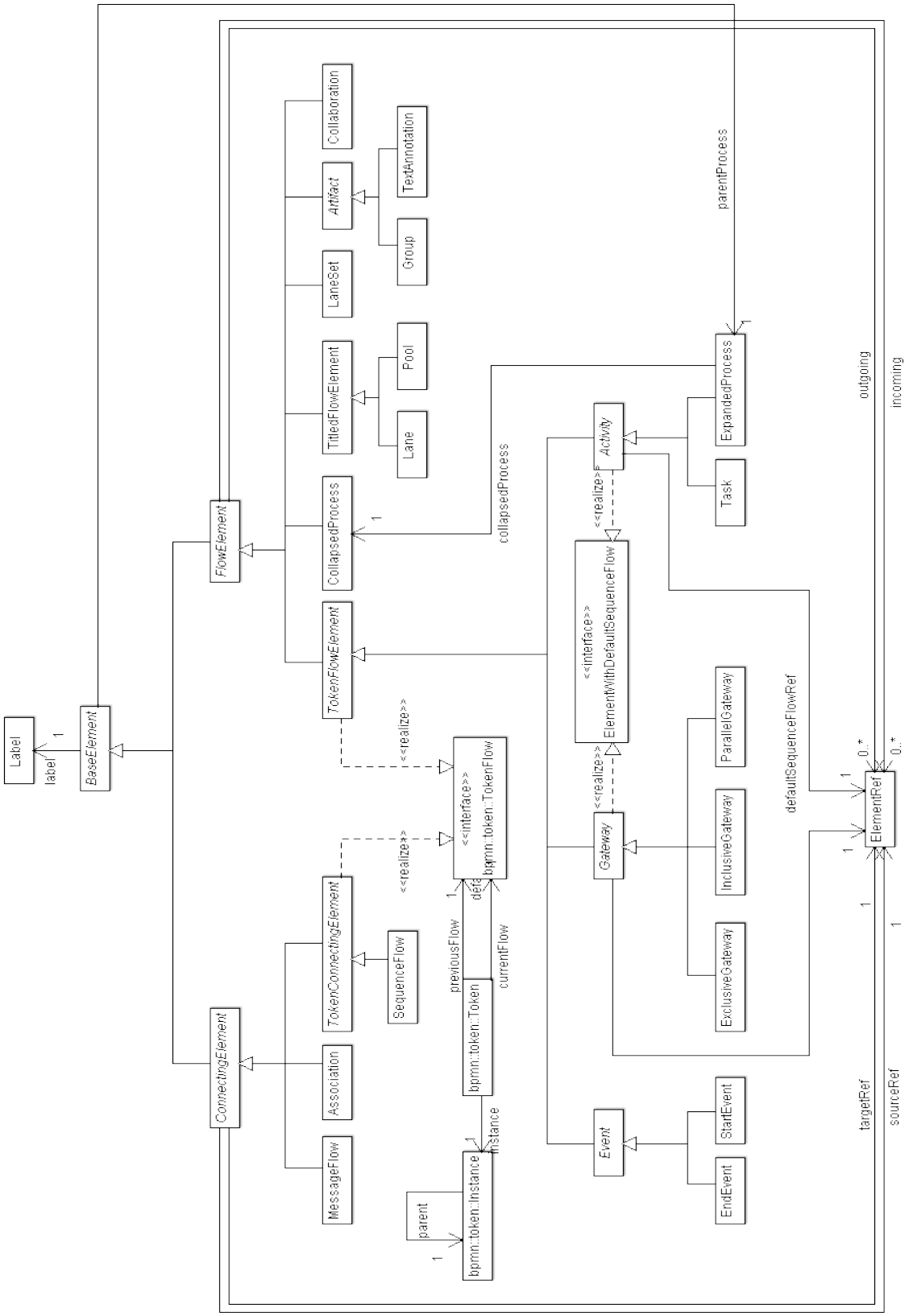
Marken werden als farbiger Stern dargestellt um sich von den anderen Elementen des Diagramms abzuheben. Jedes Token das das Diagramm durchläuft bekommt eine eigene Farbe. Befinden sich mehrere Token in einer Aktivität wird die Anzahl dazu im Token angezeigt.



Implementierung

Diagramm

Klassendiagramm

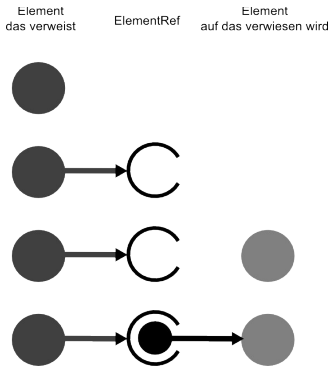


Einlesen der BPMN 2.0 XML Dateien

Zum einlesen der XML-Dateien wird die DOM Schnittstelle der JAXP verwendet der die XML-Daten gegen die von der OMG bereitgestellten XML Schema Definition validiert. Die lokale Kopie der XSD-Dateien die dazu verwendet werden befinden sich im Package 'bpmn'. Die Validierung gegen das Schema dient zum einen um die Gültigkeit der Daten zu prüfen und zum anderen um nicht benötigte Elemente (Text-Knoten die nur Leerzeichen beinhalten) zu entfernen.

Referenzen zwischen Elementen

Beim laden der Elemente kann es vorkommen das, das zu erstellende Element auf Elemente verweist, die noch nicht existieren. Deshalb hat jedes Element eine Referenz auf eine Referenz. Existiert das Element noch nicht wird zuerst eine leere Referenz zu der Element-Id erstellt, auf die das Quellelement verweist, später wenn das Zielelement geladen wird, wird diese gesetzt.



Anzeige und Positionierung der Elemente

Jedes Element ist eine eigenständige Komponente, die sich selbst und die dazugehörige Token zeichnet. Jede Komponente besitzt eine unsichtbaren Rand der benötigt wird um die Token des Elements zu zeichnen, da nur innerhalb der Komponente gezeichnet werden kann.

Die Elemente des Diagramms werden auf Pools oder Prozessen positioniert. Die Z-Reihenfolge richtet sich nach der Reihenfolge aus der XML-Datei. Ansonsten gibt es keine Hierarchie der Komponenten.

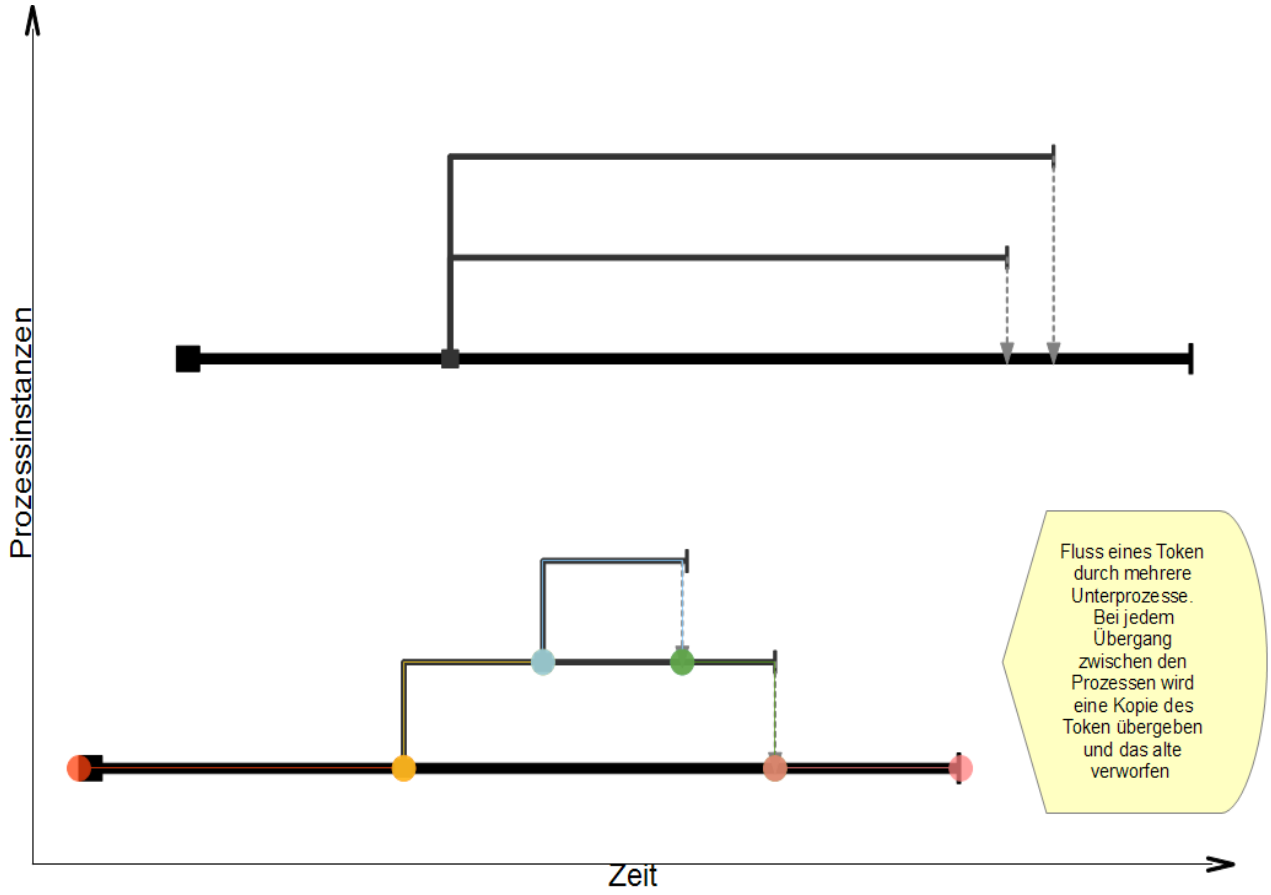
Beschriftung der Elemente

Jedes Element erzeugt und positioniert ein eigenes Label als eigenständige Componente. Diese wird auf der gleichen Elterncomponente wie das Element selbst angeordnet.

Token

Jedes Token besitzt eine Referenz auf ein Element indem es sich momentan befindet und auf eine Instanz für dieses Element.

Instanzen



Jedes Token ist einer Instanz zugeordnet. Beim Eintritt eines Token in einen Unterprozess wird abgehend von der Instanz eine Unterinstanz erstellt und das Token dieser zugewiesen. Beim Austritt aus einem Unterprozess wird die Unterinstanz wieder gelöscht und das Token der übergeordneten Instanz zugewiesen.

Tokendurchlauf

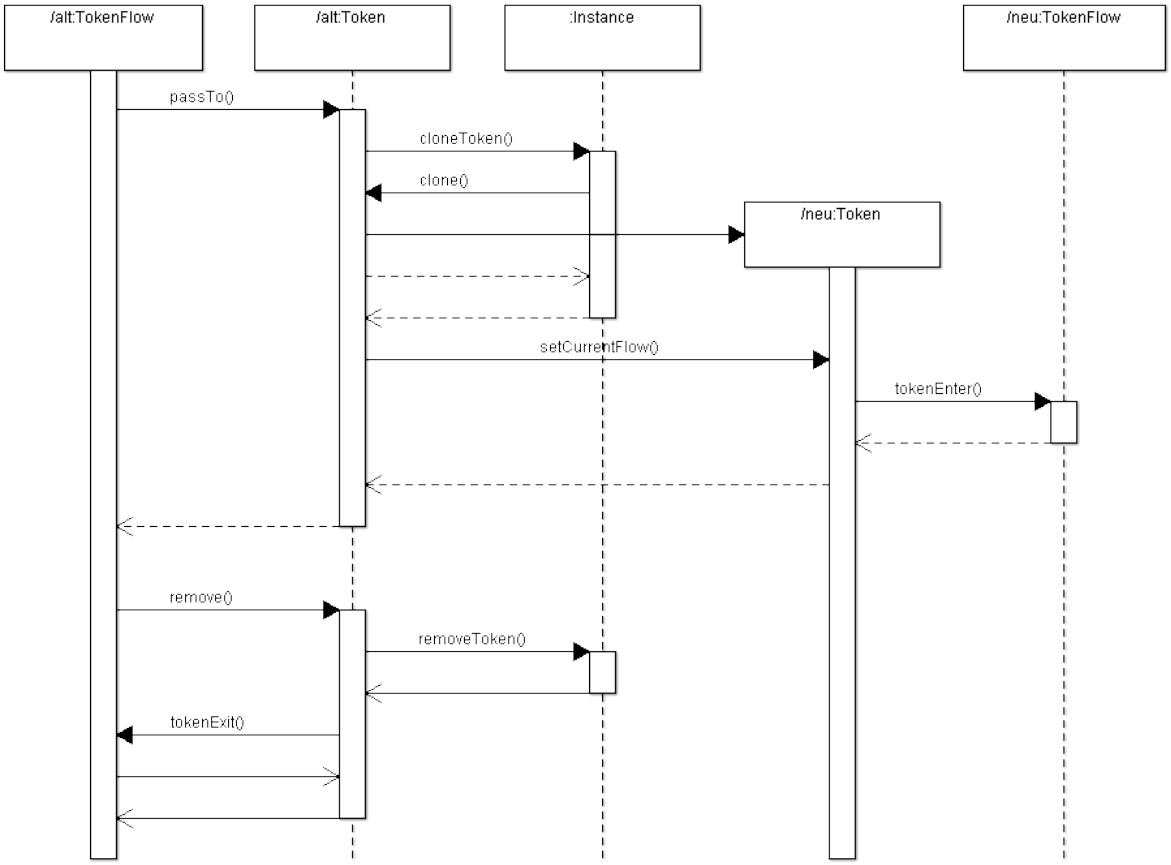


Abbildung 1: Übergabe eines Token an einen anderen Tokenflow und anschließendes entfernen aus dem alten Tokenflow.

Bei einem Animationsschritt benachrichtigt jedes Token sein aktuelles Element in dem es sich befindet um abgearbeitet zu werden. Das Element bestimmt dann für dieses Token ob nur die Position weiter gesetzt wird oder an welche Folgeelemente ein Token weitergeleitet werden muss. Jedem Folgeelement wird immer nur eine Kopie des Token übergeben, da die aktuelle Position eines Token im Element im Token selbst gespeichert ist. Jedes Element muss ein Token das weitergeleitet wurde danach selbst löschen.

Übersetzung

Für Übersetzungen werden die ResourceBundles von Java verwendet. Diese laden Dateien mit Schlüssel-Wertepaaren anhand der im Betriebssystem eingestellten Sprache. Für ein zu startendes Programm kann die Sprache auch fest gesetzt werden, indem die Systemeigenschaften 'user.language' und 'user.country' der Java VM in der Befehlszeile überschrieben werden.

In den Packages 'gui' und 'bpmn' gibt es je eine 'messages.properties'. Für jede Sprache muss eine neue Datei mit dem Namen und einem Sprachkürzel erstellt werden. Wird keine Datei für die Sprache gefunden wird jeweils die nächst mögliche Datei verwendet.

```
Beispiel:   messages_de_DE.properties → messages_de.properties → messages.properties
```

In Eclipse ist das Hinzufügen neuer übersetzter Strings über die Funktion 'Source' → 'Externalize Strings...' möglich.

Quellcode

Format

Für Einrückung, Groß-/Kleinschreibung, Leerzeichen/-zeilen und Benennung im Quellcode wird sich nach den Code Conventions for the Java Programming Language (http://www.oracle.com/technetwork/java/codeconv-138413.html) gerichtet. Bezeichnungen für Variable, Klassen und Methoden sind in Englisch.

Struktur

- /src/bpmn/
  Enthält Klassen zum Laden/Darstellung des BPMN-Modells und zur Anzeige von Meldungen.
- /src/bpmn/element/
  Enthält die Klassen zur Darstellung der BPMN-Elemente.
- /src/bpmn/token'
  Enthält die Klassen zur Verwaltung und Darstellung der Token und Interfaces für den Tokendurchlauf.
- /src/bpmn/xsd/
  Enthält die Dateien zur Validierung der BPMN 2.0 XML Dateien.
- /src/gui/
  Enthält die Klassen für das eigentliche Programm (Fenster, Controls, Verwaltung der Einstellungen).
- /test/

Enthält die BPMN-Dateien in denen die Testfälle beschrieben sind.

Test

Zum Testen wird für jedes BPMN-Element ein Diagramm erstellt, in dem alle möglichen Testfälle vorkommen. In jedem Diagramm sind Kommentare enthalten der das erwartet Verhalten/Darstellung beschreiben.

Testprotokoll

Dateiname	Anmerkung
test_artifacts.bpmn	OK
test_event_start.bpmn	OK
test_event_termination.bpmn	Bei der Terminierung eines Unterprozesses wurde kein Token daraus weitergeleitet
test_farben.bpmn	OK
test_gateway_exclusive.bpmn	OK
test_gateway_inclusive.bpmn	OK
test_gateway_parallel.bpmn	OK
test_gleichzeitige_prozessinstanziierung.bpmn	OK
test_label.bpmn	OK
test_pool_horizontal.bpmn	OK
test_pool_vertical.bpmn	OK
test_task.bpmn	Standard-Sequenzflüsse werden von Signavio nicht exportiert.
test_token_anzeige.bpmn	OK
test_unterprozess.bpmn	OK

Verwendete Software

- Eclipse IDE Indigo http://www.eclipse.org/
- LibreOffice 3.5 http://de.libreoffice.org/
- ArgoUML 0.34 http://argouml.tigris.org/

Definitionsdateien

- XML Schema Definition http://www.omg.org/spec/BPMN/20100501/

Icons

- Developpers Icons http://www.iconfinder.com/search/?q=iconset:developperss

Fehler/Unvollständigkeiten in Signavio

- Die Rahmenfarbe der Elemente ist in Signavio einstellbar, wird aber nicht mit in die XML-Datei exportiert (nur die Hintergrundfarbe wird exportiert).
- Es werden manchmal nicht alle Darstellungsinformationen mit exportiert (z.B.: test_unterprozess.bpmn werden keine Darstellungsinformationen für "unterprozess_unterprozess_aufgeklappt" im Unterprozess "unterprozess2" exportiert). Im Definitionsbereich der XML-Datei sind diese Elemente allerdings vorhanden.
- Die Positionierung der Element-Labels wird in der XML-Datei als Extension gespeichert und nicht wie im Standard definiert als BPMNLabel.
- Positionspunkte von Linien werden nicht immer korrekt exportiert (siehe test_event_start).
- Standard-Sequenzflüsse die von Tasks ausgehen werden nicht exportiert. (Der Task besitzt kein Attribut „default“).

Ergänzungen

Referenzdokumente

OMG BPMN Reference Version 2.0
http://www.omg.org/spec/BPMN/2.0/PDF

Glossar

BPMN	Business Process Model and Notation
OMG	Object Management Group
JAXP	Java API for XML Processing