

Zweck von SETEI

SETEI bedeutet Satzergänzungstest-Erstellerinterface. Mit dieser Anwendung werden Satzergänzungstests entworfen. Dies sind interaktive Tests, bei denen es gilt, einen angefangenen Satz so zu vervollständigen, dass er möglichst viel richtige Information enthält. Die Möglichkeiten zur Weiterführung des Satzes werden vom Ersteller der Übung bereitgestellt. Das Vervollständigen kann über beliebig viele Schritte geschehen, und je nachdem, was vorher ausgewählt wurde, können unterschiedliche Auswahlen bereitgestellt werden.

Das Prinzip von SETEI

Die Grundbausteine von SETEI sind Boxen, welche eine Liste mit Texteinträgen beinhalten. Es gibt vier Unterarten von Boxen: die Startbox (rot), die Stepbox (grau), die Eingabebox (grün) und die Kommentarbox (blau). Der Anfang des Satzes wird in die Startbox geschrieben; dies ist die Box, die Sie sehen, wenn Sie ein leeres Dokument vor sich haben. Jede Übung muss eine Startbox mit genau einem Eintrag haben. Die Liste der Möglichkeiten, die der Student jeweils pro Entscheidungsschritt zu Gesicht bekommt, sind in SETEI in einer Stepbox dargestellt. Eine Stepbox symbolisiert also einen Entscheidungsschritt. Die Reihenfolge der Schritte legen Sie fest, indem Sie Pfeile zwischen den Stepboxen anlegen. Dabei sind Verzweigungen möglich.

Eine Eingabebox funktioniert gleich wie eine Stepbox, ausser dass nach dem Auswählen eine Eingabe vom Benutzer verlangt wird. Das ist sinnvoll beispielsweise bei numerischen Angaben, da nicht für jede Zahl ein eigenes Auswahllement angeboten werden kann. Wenn der letzte Schritt erreicht ist, folgt in der Regel eine Kommentarbox. Die Einträge dieser Liste kann der Student nicht auswählen. Je nachdem, was er für einen Satz kreiert hat, wird ein bestimmtes Element dieser Liste

ausgegeben und damit einen spezifischen Kommentar zu seiner Aussage gemacht. Der Kommentar kann aus mehreren Elementen (Sätzen oder Satzteilen) bestehen. Dazu hängt man mehrere Kommentarboxen aneinander.

Neben dem Kommentar dient zur Bewertung des erstellten Satzes auch die Vergabe von Punkten. Jeder Eintrag einer Stepbox kann mit einer Punktzahl versehen werden.

Es ist möglich, in einem Kommentar eine Anschlussfrage zu stellen. Die Antwort auf diese Frage ist ein zweites Netz von Stepboxen. Die «Startbox» (es handelt sich dabei um eine graue Stepbox, welche keine eingehenden Pfeile hat) dieses Netzes wird «Followup-Box» genannt und muss bei der Frage, also dem Kommentarelement angegeben werden. Beachten Sie, dass es nicht möglich ist, einen Pfeil von einer Kommentarbox zu einer Stepbox zu machen, weil die Followupbox sich nicht auf die gesamte Kommentarbox, sondern auf einen bestimmten Eintrag daraus bezieht.

Auch bei den Stepboxen gibt es die Möglichkeit, einzelne Einträge nur dann anzuzeigen, wenn vorher der Student eine bestimmte Auswahl getroffen hat. Man kann also sowohl bei Stepboxen wie auch Kommentarboxen Bedingungen für einzelne Listeneinträge machen.

Übersicht über die Menüfunktionen

Das Menü «Datei»

Mit den hier abgebildeten Befehlen lassen sich mehrere Dokumente gleichzeitig offenhalten. «Sichern unter...» erzeugt jedesmal noch eine zusätzliche Backup-Datei mit der Endung «.BAK».

«Prolog exportieren» öffnet ein Fenster mit dem Prolog-Code, der dann in einem File abgespeichert werden kann.

«Übung ausprobieren» startet die Java-Anwendung «setX», sofern diese zur Verfügung steht. Diese Anwendung simuliert das HTML-

Interface für die Benutzer der Übung. «setX» erfordert übrigens eine höhere Java-Version als SETEI. Auf einem Macintosh mit System 9 läuft SETEI, aber nicht setX.

Datei	Bearbeiten	Box
Neu		⌘N
Öffnen...		⌘O
Schliessen		⌘W
Sichern		⌘S
Sichern unter...		
Prolog exportieren...		
Übung ausprobieren		
SETEI Hilfe		
Beenden		⌘Q

Das Menü «Bearbeiten»

Rückgängig und Wiederherstellen lässt sich auf die meisten Operationen anwenden. Die Befehle Ausschneiden, Kopieren, Einsetzen und Löschen beziehen sich auf die ausgewählten Boxen. Die Textelemente in den Boxen werden mittels der Buttons unten in jeder Box gelöscht oder neu angelegt.

Bearbeiten	Box	AutoKomment
Rückgängig		⌘Z
Wiederherstellen		⇧⌘Z
Russchneiden		⌘H
Kopieren		⌘C
Einsetzen		⌘V
Löschen		
Boxen im Raster anordnen...		
Suchen/Ersetzen...		⌘F
Weitersuchen		⌘G
Zeilennummer suchen...		
Konsistenz überprüfen		
Sätze generieren...		
Optionen...		
Prädikate definieren...		

Boxen im Raster anordnen

Mit dieser Funktion lassen sich die Boxen regelmässig anordnen. Man gibt die Anzahl Zeilen und Spalten sowie den Abstand zwischen den Boxen an. Die Boxen werden nach ihren IDs sortiert.

Suchen und Ersetzen

Suchen und Ersetzen kann man im Boxbrowser über den entsprechenden Button oder im normalen Display über den entsprechenden Menüpunkt. Sucht man im Feld «Bedingungen» oder «ID» nach einer Zahl, so wird nur ein Treffer für die Zahl und nicht für die Ziffernfolge gefunden. Beispielsweise ergibt ein Eintrag «13» keinen Treffer für die Suche nach «1» oder nach «3». Im Text gäbe es jedoch einen Treffer dafür.

Konsistenz überprüfen

Das Programm führt in der jetzigen Version keine Kontrolle darüber, ob die Bedingungen theoretisch überhaupt erfüllbar sind. Allerdings merkt der Konsistenztest, wenn eine Zeilennummer angegeben wird, die gar nicht existiert. Die Vergabe einer bestimmten Zeilennummer ist einmalig, d.h. wenn Zeile Nr. 7 gelöscht wird, tritt später nie mehr eine Zeile mit dieser Nummer auf. Im weiteren entfernt der Test unerlaubte Zeichen sowie Operatoren, die an einem unzulässigen Ort stehen (wenn z.B. die Bedingungen mit einem Operator oder einer öffnenden Klammer aufhören).

Mit dem Button «Reparieren» werden die im Fenster angezeigten Reparaturen durchgeführt. Vor dem Reparieren empfiehlt es sich, diese Veränderungen durchzusehen, da in gewissen Situationen die automatische Korrektur unerwünscht oder fehlerhaft sein kann.

Sätze generieren

Wie die Bedingungen sich auswirken, kann mit der Funktion «Sätze generieren» getestet werden. Es werden alle nach den Bedingungen möglichen Zeichenketten mit den entsprechenden Kommentaren und Punktzahlen ausgegeben.

Optionen

Hier können Sie Ihre bevorzugte Schriftgrösse, Dokumentgrösse und die Anzahl Widerrufenschritte angeben. Die Arbeitsfläche ist standardmässig auf 2000 x 2000 Pixel gesetzt.

Prolog-Prädikate definieren

Die selbst definierten cl/2-klauseln ermöglichen dem Ersteller, Tests über dem Pfad zu definieren, die er zusätzlich zu den ho-Tests, die schon implementiert sind, in den Bedingungen benutzen möchte.

Beispiel: ein Ersteller will immer, wenn der Benutzer 5 schritte gemacht hat, eine Aufmunterung an den Endbenutzer ausgeben. Er definiert dazu ein cl/2-Prädikat:

```
cl(aufmunterung(Pfad),
(length(Pfad, Laenge), 0 is
Laenge rem 5)).
```

(rem ist vordefiniert in sicstus als Rest, der bei der Integerteilung übrigbleibt.)

Verwendung in der Kommentarzeile:

Text: 'Weiter so!'

Bedingung:

```
ho(Pfad,aufmunterung(Pfad))
```

Das Menü «Box»

In diesem Menü sind die Befehle zum Erstellen von neuen Boxen zu finden.



Enthält eine Box mehrere Elemente, so kann die Box geteilt werden. Der Button «Teilen» oder das Menü «Box teilen» öffnet einen Dialog, der die Listenelemente einzeln auswählen lässt. Die selektierten Elemente werden aus der aktuellen Box entfernt und in eine neue eingefügt.

Die Elemente von zwei oder mehreren Boxen können in einer Box vereinigt werden. Dazu steht im Box-Menü «Boxen vereinen» zur Verfügung. Es lassen sich nur Boxen vom gleichen Typ (Kommentar- oder Stepboxen) ver-

schmelzen. Der Befehl vereint die ausgewählten Boxen; wenn keine oder nur eine ausgewählt ist, passiert nichts.

Die untersten vier Menüpunkte betreffen die Ansicht der Boxen. Der Boxbrowser ist ein Fenster, welches die jeweils angewählte Box gross darstellt und ein bequemes editieren der Einträge zulässt. In Kombination mit dem Boxbrowser ist die «Miniaturansicht» nützlich. Sie stellt die Boxen nur als kleines Viereck mit Nummer dar. Dasselbe tut auch die «Verbindungsansicht». Der Unterschied zwischen den Ansichten besteht darin, dass die Miniaturansicht die ganze Arbeitsfläche verkleinert, also auch der Abstand zwischen den Boxen. In der Verbindungsansicht ist dieser Abstand genau gleich wie in der normalen Ansicht, aber die Boxen sind verkleinert. Dadurch werden die Verbindungen besser sichtbar, die manchmal durch die Boxen allzusehr verdeckt werden.

Das Menü «AutoKommentar»



Die Bedeutung dieser Menüpunkte ist in der separaten Dokumentation zum Programmierprojekt «Automatische Kommentare in SETEI» nachzulesen.

Arbeiten mit Boxen

Auf der Macintosh-Plattform sieht eine Box so aus:



Das dunkelgraue Feld oben links ist ein Ziehfeld: in diesem Bereich kann eine Box angefasst und an einen anderen Ort geschoben werden. Ein Doppelklick auf dieses Feld öffnet den Boxbrowser.

Die Tasten «Löschen» und «Teilen» oben in der Mitte beziehen sich auf die Box, während die unteren Buttons sich auf das gewählte Element in der Liste beziehen. «Teilen» ist hier nicht aktiv, da mindestens zwei Elemente in der Box sein müssen, um sie teilen zu können. Die Tasten «Aufwärts» und «Abwärts» verschieben das selektierte Element in der Liste um einen Schritt. Bei nur einem Element sind diese Tasten ebenfalls nicht aktiv.

Das gelbe Feld rechts oben zeigt die Nummer der Box und dient gleichzeitig dazu, Verbindungen zu legen. Klicken Sie auf dieses Feld, so wird es blau. Klicken Sie anschliessend in das Nummernfeld einer anderen Box, haben Sie eine Verbindung gelegt. Bei einem Doppelklick wird das Nummernfeld rot; dies bedeutet, dass man durch Klick auf das Nummernfeld einer anderen Box eine Verbindung löscht, sofern diese vorhanden ist.

Es ist ein Unterschied, ob zuerst Box 1 und dann Box 2 angeklickt wird oder umgekehrt: Der Pfeil der Verbindung zeigt in die andere Richtung. Daher kann man eine Verbindung 1 -> 2 nicht löschen, indem man zuerst auf Box 2 doppelklickt.

Boxen auswählen

Eine Box wird ausgewählt, wenn auf sie geklickt wird. Mehrere Boxen wählt man aus, indem man ein Rechteck aufzieht, indem die Boxen enthalten sind, oder indem man in die «Auswählen»-Checkboxes der verschiedenen Boxen klickt.

Um alle Boxen auszuwählen, wählt man das dafür vorgesehene Menü.

Alle Boxen auswählen ausser eine: Menü «Alle auswählen» und dann in die Checkbox der einen Box klicken.

Die Checkbox ist eigentlich eine Übergangslösung. Ziel ist es, das Selektieren von mehreren Boxen

durch Drücken der Shift-Taste zu ermöglichen, wie man es aus anderen GUIs kennt. Das ist bereits programmiert, funktioniert jedoch noch nicht immer zuverlässig.

Eine Auswahl aufheben kann man, indem man irgendwo in die Arbeitsfläche klickt.

Boxen verschieben

Die Box mit der Maus am grauen Feld links oben (dragLabel) anfassen und ziehen. Alle anderen selektierten Boxen verschieben sich mit.

Das Verschieben der Boxen kann nicht rückgängig gemacht werden.

Zeilen erzeugen, löschen und bearbeiten

Erzeugen: Button «Neu» auf der Box anklicken. Bei der Startbox ist dieser Button deaktiviert, wenn sie nicht leer ist.

Wenn auf die Liste doppelgeklickt wird, so wird entweder ein Dialog zum Editieren des angeklickten Elements geöffnet, oder, falls in einen leeren Bereich der Liste geklickt wurde, ein Dialog zum Erstellen eines neuen Elements. Eine markierte Zeile kann gelöscht werden, in dem die Backspace-Taste gedrückt wird.

Arbeiten in der Verbindungs- und Miniaturansicht

Die Boxminiaturen werden einfach herumgeschoben, indem man sie irgendwo anfassen und daran zieht. Ein Klick wählt die Box aus. Mit gedrückter Shift-Taste können Sie mehrere Boxen auswählen.

Die ctrl-Taste bewirkt in der Verbindungsansicht dasselbe wie in der normalen Ansicht ein Klick ins Nummernfeld der Box: Es können so Verbindungen gelegt oder entfernt werden. Ein Doppelklick auf die Boxminiatur öffnet den Boxbrowser. Ist der Boxbrowser bereits geöffnet, zeigt er die jeweils ausgewählte Box an – es genügt also ein Klick.

Sie können Boxelemente nur in der Normalansicht sehen und editieren. Sie brauchen jedoch dafür nicht in die Normalansicht zu wechseln, wenn sie mit dem Boxbrowser arbeiten.

Arbeiten mit dem Boxbrowser

Der Boxbrowser (Bild unten) zeigt jeweils die Elemente einer Box gross an. In die Liste des Browsers kann direkt hineingeschrieben werden. Änderungen werden

ID	Text	Constraints	Punkte
2	von den formalen Wortausprägungen	1	5
4	vom Bau	1	4
8	von der Zusammenstellung	1	2
10	von der Kombination	1	2
18	von den Wortklassen	1	0
20	von den syntaktischen Verwendungen	1	0
24	von der Bedeutung	1	0
26	vom Wort	1	3

laufen synchronisiert.

Auf dem Bild ist Box 2 zu sehen, welche in der Miniaturansicht selektiert ist. Links und rechts der Liste sind die mit dieser Box verbundenen Boxen aufgelistet. Beim Klick auf diese Boxen wechselt der Boxbrowser zu der geklickten Box. Er merkt sich ausserdem den Weg, den der Benutzer gegangen ist, so dass dieser mit den Pfeiltasten alles zurücknavigieren kann.

In der Liste können mit der Shifttaste mehrere Elemente blockweise ausgewählt werden und mit der ctrl-Taste mehrere nicht zusammenhängende Elemente. Die Buttons «Aufwärts», «Abwärts» und «Löschen» lassen sich dann auf die Auswahl auch von mehreren Zeilen anwenden. Beim Schreiben in der Liste muss Copy/Paste mit den Tastenkombinationen ctrl-X/V/C (und nicht Command-X/V/C) erledigt werden. Undo (Command-Z) funktioniert, wenn das Hauptfenster angeklickt wird oder wenn im Boxbrowser kein Feld zum Schreiben aktiviert ist. Um das zu erreichen, kann ins ID-Feld geklickt werden, welches nicht editierbar ist.

Interne Repräsentation

Wird in SETEI ein neues Dokument erzeugt oder eines geöffnet, so erscheint ein neues Dokumentfenster. Dies ist die Klasse Doc, welche eine Unterklasse von awt.Frame ist. Jedes Doc enthält je eine Instanz der Klasse

- DataHolder, welche die gesamten Daten zusammenfasst
- Display, welche die Arbeitsfläche repräsentiert
- MenuManager, welche die Menuleiste verwaltet
- Protocol, welche eine Liste mit Benutzeraktionen (UserActions) führt (für die Widerruffunktion).
- Boxbrowser, ein Fenster, welches die gerade gewählte Box im Detail anzeigt
- ConstraintsManager, welcher die Funktion der Buttons zum Erstellen von Bedingungen regelt. Die Buttons selbst, oberhalb der Arbeitsoberfläche ange-

siedelt, sind allerdings Teil von Doc.

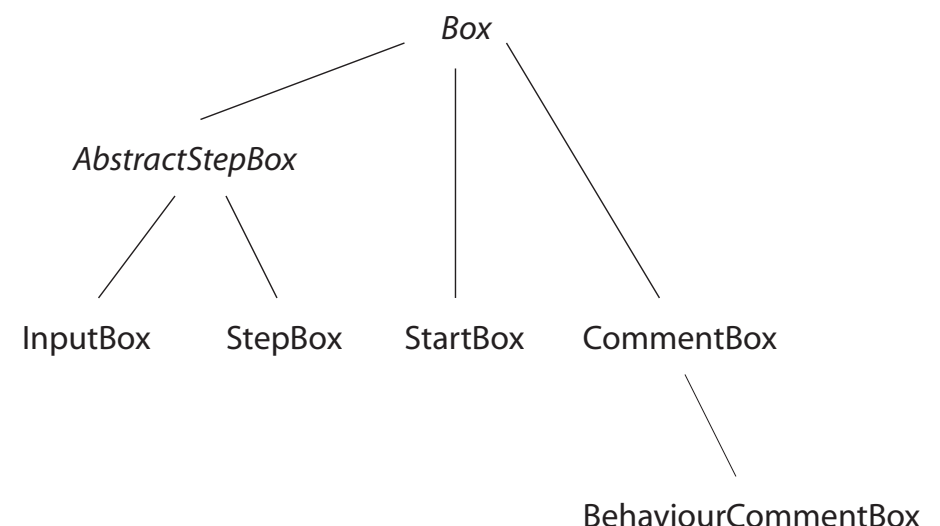
Das Doc enthält zudem das Anzeigefeld für die EdgeDatas ganz unten im Fenster sowie eine Reihe von Feldern, welche fürs Dokument global abgespeichert werden.

Da es mehrere Docs geben kann, ist die Klasse, welche die main()-Routine enthält, nicht Doc, sondern WindowController. Der WindowController ist ebenfalls ein Frame und daher für den Benutzer sichtbar (es ist das ganz kleine Fenster mit der Programminformation, welches sich nicht schliessen lässt). In Java können keine Menüleisten angezeigt werden, ohne dass ein Fenster sichtbar und aktiv ist. Damit auch nach dem Schliessen des letzten Dokuments per Menü wieder eines geöffnet werden kann, ist der WindowController ein Frame mit einer eigenen, reduzierten Menüleiste.

Der WindowController unterhält einen Vektor mit den gegenwärtig geöffneten Dokumenten sowie einige ganz globale Informationen wie z.B. die gewünschte Textgrösse, die bei einer Änderung auf alle Dokumente angewandt wird. Alles, was auf mehrere Dokumente angewandt wird, muss über den WindowController laufen, denn die Docs haben keine Referenzen auf andere Docs.

Boxen

Alle Boxen sind Unterklassen der abstrakten Klasse «Box». Das



Layout der Boxen ist mit Ausnahme der Hintergrundfarbe in der Box-Klasse definiert. Aus der untenstehenden Grafik geht hervor, wie die verschiedenen Boxentypen miteinander verwandt sind. Kursiv bedeutet eine abstrakte Klasse. Zu beachten gilt insbesondere, dass weder die InputBox noch die StartBox eine Unterklasse der StepBox ist. Im Gegensatz zu Kommentarbox, Stepbox etc. steht der Begriff «Followup-Box» nicht für eine Boxklasse, sondern für eine zusätzliche Eigenschaft irgendeiner Stepbox, Startbox oder Inputbox. Eine Box wird dann zu einer Followup-Box, wenn eine Kommentarzeile auf diese Box referenziert.

Verbindungen

Die Verbindungen der Boxen werden durch die Klasse Connection repräsentiert. Obschon Verbindungen zu einer Kommentarbox blau statt schwarz sind und eine andere Bedeutung haben, gibt es nur eine Connection-Klasse. Die Klasse wird definiert durch die zwei Boxen, die sie verbindet. Unterscheidungen zwischen den zwei Arten von Verbindungen erfolgen über die Abfrage, ob es sich bei der zweiten Box um eine Kommentar- oder eine Stepbox handelt.

Boxelemente

Die Zeilen in den Boxen entsprechen der Java-Klasse EdgeData. In SETEI 2 war EdgeData eine abstrakte Superklasse der Klassen CommentData und Step-

Data. Zudem gab es eine Klasse namens GraphData, welche die alle Kanten in einem Vektor sammelte. In SETEI 3 gibt es nur noch nicht-abstrakte EdgeDatas. Wie bei den Connections wird die Unterscheidung zwischen Kommentarzeilen und Satzschrittzeilen durch die Referenz auf die Box vorgenommen, zu der die EdgeData gehört.

Datenverwaltung

Die Klasse DataHolder enthielt in SETEI 2 drei Vektoren für die drei Bestandteile Boxen, Connections und EdgeDatas. In SETEI 3 sind die EdgeDatas nun in der jeweiligen Box gespeichert und nicht mehr zentral im DataHolder. Das widerspricht zwar dem Grundsatz, die Daten von grafischen Komponenten zu trennen. Allerdings erweist sich diese Repräsentation als viel praktischer, weil die Box ständig auf ihre eigenen Zeilen zugreifen muss. Ausserdem waren die Daten ja schon in der alten Form in der Box enthalten, nur nicht vollständig und nicht ohne weiteres abrufbar. Denn die Liste, welche die Box enthält, besteht aus einem String, zusammengesetzt aus der ID der Zeile und dem Text.

Die anfängliche Konzeption sah eine Trennung von Daten und Grafikkomponenten nach folgendem Konzept vor: EdgeData (bzw. CommentData, StepData und GraphData) sowie Connection als Datenklassen, welche in der Klasse DataHolder zusammengefasst werden. Daneben eine Klasse Display, welche die Benutzeroberfläche regelt. Die Boxen, eigentlich eine grafische Komponente, waren aber schon immer sowohl im Display als auch im DataHolder vorhanden. Ausserdem sind auch die Verbindungen gleichzeitig eine grafische Komponente. Die Methoden, welche die Connection auf dem Display zeichnen, sind in der Connection selbst enthalten und nicht etwa im Display.

Anzeige

Im Display kann zwischen verschiedenen Ansichten gewechselt

werden. In der Verbindungsansicht und in der Miniaturansicht werden die Boxen im Display unsichtbar gemacht, die Boxlabels sichtbar. In der normalen Ansicht ist es umgekehrt. Boxlabels sind Unterklassen von java.awt.Label, welche die Box-ID als Text haben und die Hintergrundfarbe der betreffenden Box. Jedesmal, wenn mittels der Methode addBox() von Display eine Box hinzugefügt wird, wird auch ein Boxlabel erzeugt. Somit existiert immer parallel zu jeder Box ein Boxlabel. Vom Boxlabel kann auf die Box zugegriffen werden, nicht aber umgekehrt. Auf ein Boxlabel wird über den Display mit getBoxlabel(int id) zugegriffen. In der Verbindungsansicht nehmen die Boxlabels den Platz der Boxen ein. Da das Boxlabel viel kleiner ist (es ist eine Symboldarstellung der Box), werden die Verbindungen besser sichtbar. In der Miniaturansicht werden auch die Verbindungen in der Verbindungsansicht kürzer. Dann sieht man zwar die Verbindungen nicht besser, aber der gesamte Automat hat auf sehr kleinem Raum Platz. So ist der Display nur eine Art Navigationsübersicht, während im Boxbrowser editiert wird. Ist gleichzeitig der Boxbrowser geöffnet, wird in diesem jeweils die Box angezeigt, welche im Display gerade angeklickt wird.

Befehle

Grundsätzlich ruft der MenuManager bei jedem Menü, das der Benutzer wählt, eine entsprechende Methode in der abstrakten Klasse Commands auf.

Meldungen

Dialog-Meldungen, Tastenbeschriftungen, Menüpunkte etc. werden nicht im Code selbst getextet, sondern in den externen Files messages.set, menus.set und version.set. Im File messages.set steht pro Zeile eine ID, dann ein Tabulator und dann die Meldung. Die Methode SharedMethods.getMessage(ID) holt den String zur angegebenen ID.

Der Vorteil dieses Konzepts: erstens können so Meldungen, die an mehreren Orten aufgerufen werden, zentral verwaltet werden, und zweitens erlaubt es Mehrsprachigkeit. Für eine englische Version existieren ebenfalls externe Files.

Formatkonventionen

Prolog-Export

Der Prolog-Code ist das Endformat für die Übung. Das HTML-Interface, das der Übende verwendet, greift mittels PERL auf einen Prolog-Server zu, auf dem ein endlicher Automat diesen Code abarbeitet.

Im Schema dieses Automaten gibt es keine Boxen. Die Zeilen in den Boxen stellen die Kanten des Automaten dar. Als Knotenpunkte werden aber dennoch die Boxnummern verwendet. Das führt dazu, dass eine Boxzeile als Kante zwischen ihrer Box und der vorhergehenden Box repräsentiert ist:

delta(ID, vorherige Box-ID, Text, Box-ID, Bedingungen, Punkte).
Kommentarzeilen sind wie folgt notiert:

comment(Bedingungen, vorherige Box, ID, Text, Followup-Box-ID).

Bei Boxen, die mehrere eingehende Verbindungen von anderen Boxen haben, müssen mehrere delta-Prädikate für jede Boxzeile geschrieben werden. Das gilt auch für die Kommentarzeilen.

Weiter gehören zum Prologcode die Angaben darüber, welche Boxen Startzustände und Endzustände sind.

«final(1).» bedeutet, dass Box 1 keine weiteren Verbindungen zu anderen Stepboxen hat.

«start(s(0)).» Deklariert die Startbox (ID 0) als Startzustand.

Followup-Boxen

Followup-Boxen sind Stepboxen oder Inputboxen, die auf einen Kommentar folgen. Da sie auf eine bestimmte Kommentarzeile folgen, wird in der entsprechenden Kommentarzeile die Nummer der Box angegeben, und nicht etwa ein Pfeil von der Kommentar-

box zu Followup-Box erstellt, denn in der Kommentarbox können auch Kommentare stehen, auf die keine Followup-Box folgt.

Followup-Boxen müssen keine eingehenden Verbindungen enthalten.

Followup-Boxen werden im Prolog-Code wie folgt behandelt:

a) Boxen ohne eingehende Verbindungen

Für jede Followup-Box mit ID n wird ein Eintrag erzeugt: $\text{start}(\text{fu}(\text{s}(n)))$.

für jede Zeile z der FollowupBox: $\text{delta}(z, \text{s}(n), \langle \text{Text der Zeile} \rangle, n, \text{true}, \emptyset)$.

Das zweite Argument des Deltas ist die Box, von der die Verbindung kommt. Da hier diese Box nicht angegeben wird, muss für jede Zeile der Followup-Box nur ein delta erzeugt werden, statt wie sonst für jede eingehende Box ein delta.

b) Boxen mit eingehenden Verbindungen

Für Followup-Boxen mit eingehenden Verbindungen braucht es keine weiteren delta-Prädikate und auch kein start-Prädikat. Die Angabe der Boxnummer in der Kommentarzeile genügt als Verbindung.

Das .dat-Format

SETEI speichert die Dokumente im Textformat und erzeugt dabei eine Datei mit der Endung .dat sowie ein Backup-File mit der Endung .dat.bak (nur bei «Sichern unter...»).

Das Dateiformat repräsentiert die Informationen in folgender Reihenfolge:

1. Boxen
2. Verbindungen
3. normale Zeilen
4. Kommentarzeilen
5. Selbst definierter Prolog-Code

Die Haupt-Informationsblöcke der Datei werden jeweils mit einem Separator « $\#\#$ » getrennt. Die Dateien werden zeilenweise gelesen oder geschrieben, so dass

Zeilenumbrüche ebenfalls Separatoren darstellen: nach jeder Zeile beginnt eine neue Box/Verbindung/Boxzeile.

Die verschiedenen Objekte enthalten folgende Informationen (in dieser Reihenfolge):

Box: ID, Typ, Koordinaten

Verbindungen: ID erste Box, ID zweite Box

Zeilen: BoxID, Text, Zeilen-ID, Bedingungen, Punkte

Kommentarzeilen: BoxID, Text, Zeilen-ID, Bedingungen, Followupbox-ID

Wobei die Informationen jeweils durch ein Tabulatorzeichen getrennt werden.

Bedingungen

Steht in den Bedingungen «1», bedeutet das, dass die Zeile 1 bereits durchlaufen worden sein muss, damit dieses Element angezeigt wird. « ~ 1 » bedeutet die Negation: Zeile 1 darf (noch) nicht durchlaufen worden sein.

Die Zeilennummern können mit logischen Operatoren (# für ODER, & für UND) und Klammern verknüpft werden.

Soll ein Element auf alle Fälle erscheinen, lautet die Bedingung «true».

« $1 < 2$ » bedeutet, dass Zeile 1 vor der Zeile 2 durchlaufen worden sein muss. SETEI 3.0 akzeptiert diesen Operator im Konsistenztest, kann ihn jedoch nicht interpretieren, wenn er alle möglichen Sätze erzeugen soll.

Schliesslich gibt es noch die Bedingungen, welche mit dem Prologprädikat $\text{ho}/2$ (für Higher Order) beginnen.