UNIVERSITÄT LEIPZIG

Institut für Medizinische Informatik, Statistik und Epidemiologie (IMISE)

Realisierung eines CRF Versionen Mapping Tool

Leipzig, 5. Januar 2014

vorgelegt von:
Victor Christen
Betreuer:
Matthias Löbe

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Die Resultate einer Studie basieren auf der Auswertung der in jeder Phase erfassten Daten bzlg. eines Patienten oder Probanden. Diese Daten werden mittels eines CRF (Check Report Form) Bogen erfasst. Diesbezüglich enthält ein solcher Bogen eine Menge von Datenfeldern(Items), die für die spätere Auswertung relevant sind. Zu Beginn einer jeden Studie, werden deshalb für jede Phase einer Studie die relevanten Items inklusive ihrer Eigenschaften, Regeln und inhaltlichen Strukturierung ermittelt. Die Menge an CRF Bögen für die verschiedenen Studien können in einem klinischen Datenmanagmentsystem verwaltet werden, wie z.B. OpenClinica. Im Allgemeinen ist die Entwicklung eines CRF Bogens ein fortlaufender Prozeß, da im Verlauf der Studie fehlende oder unzureichend definierte Items identifiziert werden. Diese müssen in einer aktualisierten Version ergänzt bzw. geändert werden. Sei nun folgendes Szenario gegeben: Es wurden mittels eines CRF Bogens die relevanten Daten für die Phase der Aufnahme eines Patienten erfasst. Es wurde festgestellt, dass die Auswertung der Daten umständlich ist, so dass eine Anpassung des CRF Bogens erfolgen muss. Dennoch sollen die erfassten Daten weiter verwendet werden. Hierfür ist ein Mapping, eine Abbildung der alten Items auf die neuen Items, notwendig, um die alten Daten in das Schema des neuen CRF Bogens zu transformieren. Des Weiteren ist es für die Weiterentwicklung eines CRF Bogens hilfreich zu wissen, welche Items bisher geändert wurden, so dass eine gezielte Entwicklung fortgeführt werden kann. Da ein CRF Bogen potentiell eine hohe Anzahl an Datenfeldern beinhaltet und mehrere Versionen umfasst, ist eine manuelle Generierung eines Mappings für alle Versionen nicht effizient realisierbar. Eine weitere Problematik bzgl. der Änderungen von Items ist die Relevanz der Änderung. Die Änderungen von Eigenschaften eines Items,die sich auf das Layout bzgl. des CRF Bogens beschränken, verursachen keine Änderung des medizinischen Konzepts, sodass existierende Daten auf das neue Item gemappt werden können. Die Transformation der Daten eines Items, deren medizinisches Konzept sich geändert hat, ist nicht ohne weiteres möglich.

1.2 Zielsetzung

Das Ziel dieser Arbeit ist die Entwicklung einer Applikation für die Generierung eines Mappings für eine Menge von CRF Versionen und eine entsprechende Visualisierung, sodass eine effiziente Identifikation der Änderungen und deren Ursache möglich ist. Ein Mapping repräsentiert eine Abbildung von Items,die eine effiziente Identifikation von gelöschten, hinzugefügten und geänderten Items ermöglicht.
Des Weiteren werden die geänderten Items bzgl. der Relevanz der Änderung klassifiziert, sodass eine Filterung bzgl. der Art der Änderung möglich ist.

2 Grundlagen

2.1 CRF Excel Spezifikation

Der Import eines CRF Bogens in das OpenClinica System erfolgt durch eine Excel Datei, die die enthaltenden Datenfelder mit ihren Eigenschaften, Abschnitten und potentiellen Regeln für die Items in Form von einzelnen Tabellen enthält. Das Schema der Excel Mappe entspricht dem Klassendiagramm in Abbildung 1.
Die Klassen entsprechen den einzelnen Tabellen und die Attribute einer Klasse repräsentieren die

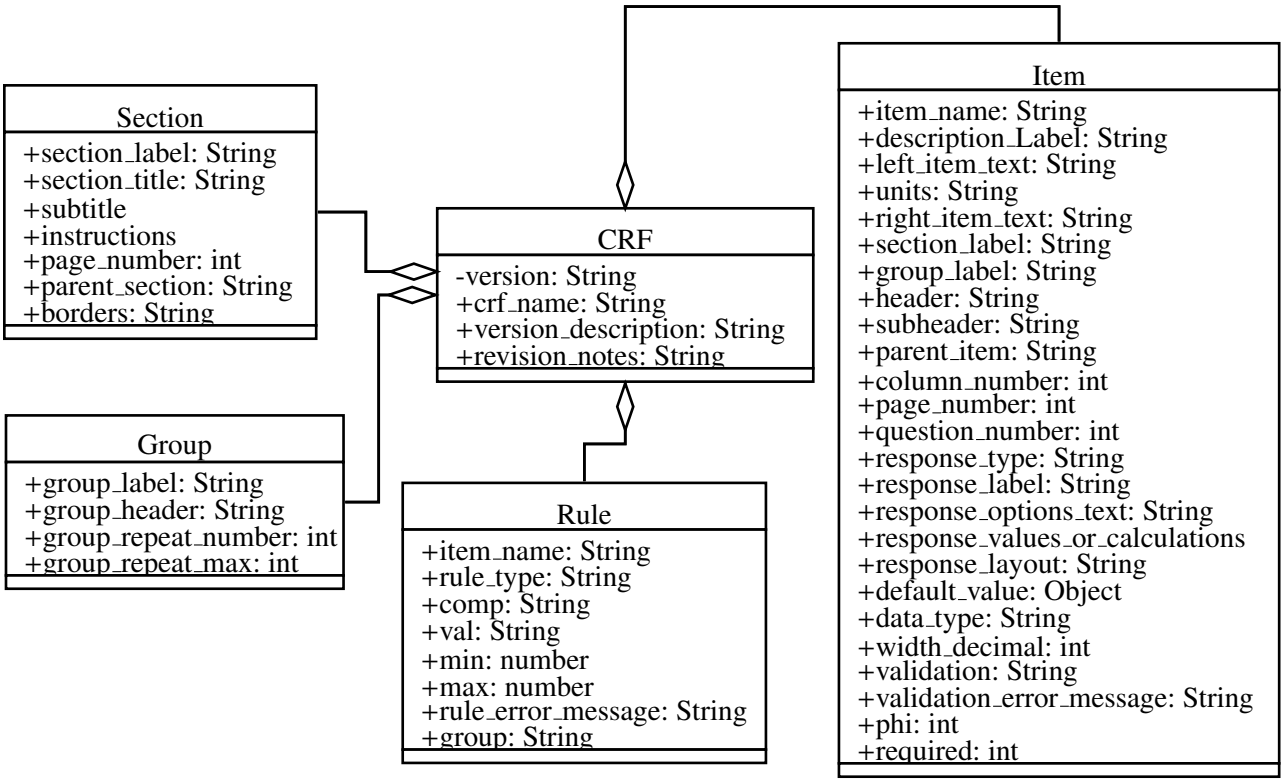

Abbildung 1: Datenschema der Excel Arbeitsmappe für einen CRF Bogen

Spalten der Tabelle. Die `CRF` Tabelle beinhaltet Informationen bzgl. des Namens des CRF Bogens, der Version, einer Beschreibung sowie Bemerkungen bzgl. der Erstellung.
Die `Sections` Tabelle definiert die layoutspezifische Struktur der Items des CRF's. Items, die das gleiche `section_label` Attribut aufweisen, werden auf der gleichen Seite aufgeführt. Jede Section wird durch das `section_label` Attribut identifiziert. Visuell wird jeder Abschnitt durch eine Überschrift getrennt, die durch den `section_title` spezifiziert wird.
Die `Groups` Tabelle ermöglicht die Gruppierung von Items. Items, die das gleiche `group_label` besitzen, erscheinen zusammen im CRF. Die Daten, die für die Auswertung notwendig sind, werden durch die Items eines CRF Bogens definiert. Die Spezifikation eines Items beinhaltet die Ei-

genschaften die in der Tabelle 1 aufgelistet sind.

Spaltenname	Beschreibung
ITEM_NAME	Identifier für ein Item
DESCRIPTION_LABEL	Beschreibung
LEFT_ITEM_TEXT	Text, der im CRF auf der linken Seite dargestellt werden soll
UNITS	Definition des Typs der Antwort
RIGHT_ITEM_TEXT	Text, der im CRF auf der rechten Seite dargestellt wird
SECTION_LABEL	Zuordnung zu einer SECTION
GROUP_LABEL	Zuordnung zu einer Gruppe
HEADER	Überschrift für das Item
SUBHEADER	Unterüberschrift für das Item
PARENT_ITEM	vorangeganges Item referenziert mittels ITEM_NAME
COLUMN_NUMBER	horizontale Anordnung des Items, Reihenfolge gem. Nr.
PAGE_NUMBER	Seite des Items bei Papierform des CRF's
QUESTION_NUMBER	Nr. der Frage, erscheint vor LEFT_ITEM_TEXT
RESPONSE_TYPE	Antworttyp basiert auf den Standard HTML Elementen - text, textarea, single-select, radio, multi-select, checkbox, calculation, group-calculation, file, instant-calculation
RESPONSE_LABEL	Label, das bei einmaliger Definition von anderen Items referenzierbar ist, welche die gleichen Antwortmgl. besitzen referenzierbar durch andere Items
RESPONSE_OPTIONS_TEXT	kommaseparierter Liste, die die mgl. Anwortwerte beinhaltet, die für den Einzutragenden interpretierbar sind
RESPONSE_VALUES_OR_CALCULATIONS	Wenn der RESPONSE_TYPE keine Berechnung ist, beinhaltet dieses Feld die Werte, die in der Datenbank gespeichert werden können. Bei einer Berechnung ist ein Ausdruck anzugeben, der Werte anderer Items des CRF's verwendet, um einen Wert zu ermitteln.
RESPONSE_LAYOUT	Ausrichtung der Antwortoptionen (Blank, Horizontal, Vertical)
DEFAULT_VALUE	Default Wert für das RESPONSE_OPTIONS_TEXT Feld
DATA_TYPE	Datentyp (ST, INT, REAL, DATE, PDATE, FILE)
WIDTH_DECIMAL	max. Anzahl der Zeichen und max. Anzahl der Dezimalstellen
VALIDATION	Validierungsausdruck
VALIDATION_ERROR_MESSAGE	Meldung bei Validierungsfehler
PHI	Signierung, ob Anzeige von geschützten Gesundheitsdaten (blank,0,1)
REQUIRED	Signierung, ob Wert vorhanden sein muss (blank,0,1)

Tabelle 1: Übersicht der Spalten der `Items` Tabelle

3 Realisierung

Die Applikation ist in Java implementiert und verwendet für das Auslesen der Excel Dateien die Apache POI [1] Bibliothek.
Der Unterabschnitt 3.1 beschreibt das zugrundeliegende Datenschema der Applikation. Im Unterabschnitt 3.2 wird der prinzipielle Ablauf der Applikation beschrieben. Abschließend wird detailliert auf die Berechnung der Unterschiede bzgl. einer Menge von CRF Versionen eingegangen.

3.1 Datenschema

Um einer potentiellen Veränderung der Excel Datei Spezifikation entgegenzuwirken, wurde ein eigenes generisches Datenschema für eine CRF Version konzipiert. Das Schema ist in Abbildung 2 dargestellt.
Die `CRFVersion` Klasse aggregiert die `Sections`, `Groups` und `Items` Tabelle in Form einer Hashmap mit dem jeweiligen `label` Attribut als Schlüssel und dem dazugehörigen Objekt der Klasse `Section`, `Group` bzw. `Item` als Wert. Die Klassen `Section`, `Group` und `Item` sind prinzipiell gleich aufgebaut. Als Identifier besitzen sie ein `(item|section|group)_label` Attribut und eine Hashmap `propertyMap`, die die Werte aller Spalten einer `Section`, einer `Group`, bzw. eines `Item` beinhaltet. Die Namen der Spalten fungieren als Schlüssel für die `propertyMap` und werden in der jeweiligen `(Item|Section|Group)Constants` Klasse gespeichert. Somit ist durch die lose Koppelung der Spalten zu einem Objekt mittels der `propertyMap` und der Konstanten, eine Veränderung der CRF Excelspezifikation trivial realisierbar.

[1] http://poi.apache.org/

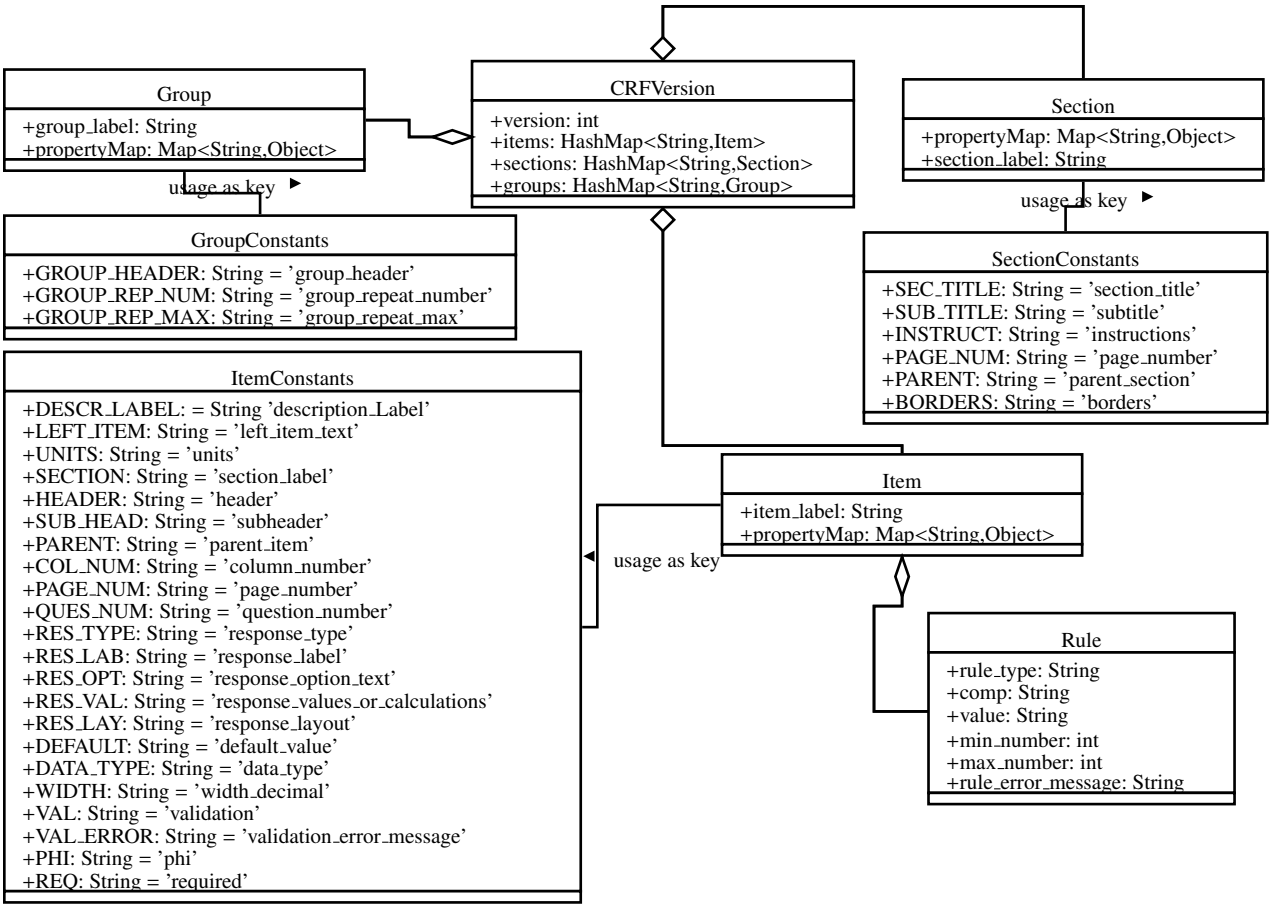

Abbildung 2: Klassendiagramm für eine CRFVersion

3.2 Allgemeiner Ablauf

Der allgemeine Ablauf der Applikationslogik ist in Abbildung 3 dargestellt. Zu Beginn spezifiziert der Anwender eine Menge von CRF Versionen. Die Dateien werden mittels der POI Bibliothek ausgelesen und zu Objekten des eigenen Datenschemas transformiert. Anschließend werden iterativ die Änderungen der CRF Versionen ermittelt und die Änderungen nach ihrer Relevanz klassifiziert. Bei jeder Iteration werden zwei aufeinander folgend Versionen miteinander verglichen und in der nächsten Iteration werden die alte zweite und die darauffolgende Version miteinander verglichen. Des Weiteren werden in jeder Iteration die Änderungen der Eigenschaften der Items bzlg. der Intensität ihres Ausmaßes klassifiziert. Die Änderungen werden in Form eines Baumes visualisiert, indem die aktuelle Version als Bezugspunkt für alle Änderungen der CRF Versionen dient.

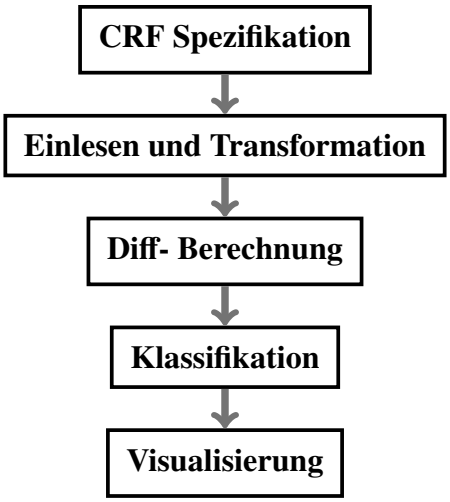

Abbildung 3: Workflow der Applikation

3.3 Diff-Berechnung

Die Menge der Änderungsoperationen wird als Diff bezeichnet. Die Ermittlung des Diff für eine Menge von CRF Versionen basiert auf dem fortlaufenden Vergleich von zwei CRF Versionen. Im Folgenden wird das Verfahren für die Ermittlung näher beschrieben.
Die Eingabe der Methode sind zwei CRF Versionen. Das Ergebnis ist eine Menge von gleichen, gelöschten, hinzugefügten und geänderten Items. Die geänderten Items werden als Mapping von altem Item zu neuem Item repräsentiert. Des Weiteren wird für jedes Mapping die Menge der geänderten Eigenschaften sowie der alte und der neue Wert gespeichert. Das Vorgehen ist im Algorithmus 1 dargestellt. Jedes Item der alten Version wird mit jedem Item der neuen Version verglichen, indem die `item_labels` und die`propertyMap` Objekte verglichen werden (Zeile 3). Wenn diese gleich sind, sind die Items gleich und ein Repräsentant wird zu der Menge *eqItemSet*, die die Menge der gleichen Items repräsentiert, hinzugefügt (Zeile 4). Wenn keine Gleichheit besteht, ist das item der alten Version gelöscht(Zeile 5-8) bzw. ist das Item der neuen Version hinzugefügt (Zeile 9-12). Da geänderte Items sich durch einen anderen Suffix beim `item_label` unterscheiden, befinden sich in den Mengen *delItemSet* und *addItemSet* eventuell geänderte Items. Um die Menge der geänderten Items zu ermitteln, wird mithilfe eines Fuzzy- Matchers die Ähnlichkeit für jedes Item der *delItemSet* Menge und jedem Item der *addItemSet* Menge bestimmt (Zeile 13-23). Der Fuzzy- Matcher verwendet für die Berechnung der Ähnlichkeit eine Kombination der Editierdistanz für die `item_labels` und die `description_labels`. Als geändertes Item wird das Paar angesehen, dessen Ähnlichkeit größer als ein Threshold $\delta(0.6)$ ist und das die höchste Ähnlichkeit aufweist. Anschließend wird für jedes Paar die Menge der geänderten Eigenschaften, die die Spalten repräsentieren, ermittelt.
Diese Methode wird iterativ für die nächsten beiden Versionen fortgeführt, wobei die aktuelle neue Version, die alte Version ist und die darauffolgende die neue. Die Berechnung des Diff's ist abgeschlossen, wenn alle Versionen verarbeitet sind.

Algorithm 1: Diff Berechnung für zwei CRF Versionen

input : $oldVersion$, $newVersion$
output: $eqItemSet$, $addItemSet$, $delItemSet$, $modItemMap$, $propertyMap$

```
1  for oldItem ∈ oldVersion do
2      for newItem ∈ newVersion do
3          if oldItem.equals(newItem) then
4              eqItemSet ← eqItemSet ∪newItem
5  for oldItem ∈ oldVersion do
6      for equalItem ∈ eqItemSet do
7          if oldItem.item_label!=equalItem.item_label then
8              delItemSet ← delItemSet ∪ oldItem
9  for newItem ∈ newVersion do
10     for equalItem ∈ eqItemSet do
11         if newItem.item_label!=equalItem.item_label then
12             delItemSet ← delItemSet ∪ newItem
13 for addItem ∈ addItemSet do
14     topMapping ← ∅
15     for delItem ∈ delItemSet do
16         similarity = fuzzyMatch(addItem, delItem)
17         if similarity ≥ δ then
18             topMapping ← add(similarity, (delItem, addItem))
19         if topMapping ≠ ∅ then
20             (delItem,addItem) = bestMapping(topMapping)
21             modItemMap ← add(delItem,addItem)
22             addItemSet = addItemSet \ { addItem }
23             delItemSet = delItemSet \ { delItem }
24 for (oldItem, newItem) ∈ modItemMap do
25     oldPropertySet = oldItem.properties
26     newPropertySet = newItem.properties
27     modProperties = diff(oldPropertySet,newPropertySet)
28     propertyMap ← add(modProperties)
```

3.4 Klassifikation

Um die Auswirkungen der geänderten Datenfelder abzuschätzen, ist eine Einordnung dieser nach spezifizierten Kategorien erforderlich. Die Kategorien sind vordefiniert und orientieren sich an der möglichen Ausprägung der Semantik eines Items. Es existieren folgende Klassen:

- *semantic*
  Diese Kategorie umfasst Änderungen, die durch die Modifikation der Beschreibung, der Header oder der linken bzw. rechten Textlabels hervor gerufen werden. Es wird angenommen, dass eine solche Änderung eine Veränderung des semantischen Konzepts als Ursache hat.

- *semantic specialization*
  Diese Kategorie ist eine Erweiterung der *semantic* Kategorie. Wenn der Inhalt durch einen Suffix erweitert wurde, wird angenommen, dass es sich um eine Spezialisierung handelt.

- *response range*
  Diese Kategorie repräsentiert alle Änderungen, die einen Einfluss auf die Spezifikation der möglichen Antworten des Datenfelds haben.
  Die Eigenschaften, die davon betroffen sind:
  DATA_TYPE, DEFAULT_VALUE, RESPONSE_LABEL, RESPONSE_OPTIONS_TEXT, RESPONSE_TYPE,REQUIRED, RESPONSE_VALUES_OR_CALCULATIONS und UNITS.

- *item interrelation*
  Eine Änderungen, die die Beziehung bzgl. eines anderen Items ändert, wird zu dieser Kategorie zugeordnet.
  Die Eigenschaften, die davon betroffen sind:
  GROUP_LABEL und PARENT_ITEM.

- *validation*
  Diese Kategorie umfasst alle spezifizierenden Eigenschaften, die die Validierung der Daten festlegen oder Sicherheitseinstellungen bzgl der Sichtbarkeit der Daten spezifizieren.
  Die Eigenschaften, die davon betroffen sind:
  VALIDATION, VALIDATION_ERROR_MESSAGE und PHI.

- *small editing*
  Die Änderungen, die keinen Einfluss auf die Eigenschaft haben, sondern lediglich eine Korrektur oder eine andere Codierung realisieren, werden dieser Kategorie zugeordnet.

- *layout*
  Diese Kategorie beinhaltet alle Änderungen, die sich auf die Anordnung und die Strukturierung der Datenfelder beziehen. Die Eigenschaften, die davon betroffen sind:

4 Funktionsweise

Die Applikation besitzt drei basale Funktionalitäten, die Spezifikation der CRF Versionen, die Betrachtung der einzelnen CRF Versionen und die Betrachtung der Unterschiede der CRF Versionen im Bezug auf die aktuelle Version.
Die Spezifikation der CRF Version erfolgt im Menü File/Open. Für die Betrachtung der Unterschiede müssen mindestens zwei Versionen eines CRF Bogens selektiert werden.
Der Anwender ist in der Lage sich die einzelnen CRF Versionen einzeln zu betrachten (siehe Abb. 4), indem er eine Version in der ComboBox selektiert und hinzufügt. Die selektierte Version wird als Baum, deren Blätter die Items repräsentieren, auf der linken Seite der Applikation dargestellt. Bei der Selektion eines Items können die Eigenschaften im unteren Teil der Anwendung betrachtet werden.
Die Änderungen werden auf der rechten Seite des Programms (siehe Abb. 5). Die Items der ak-

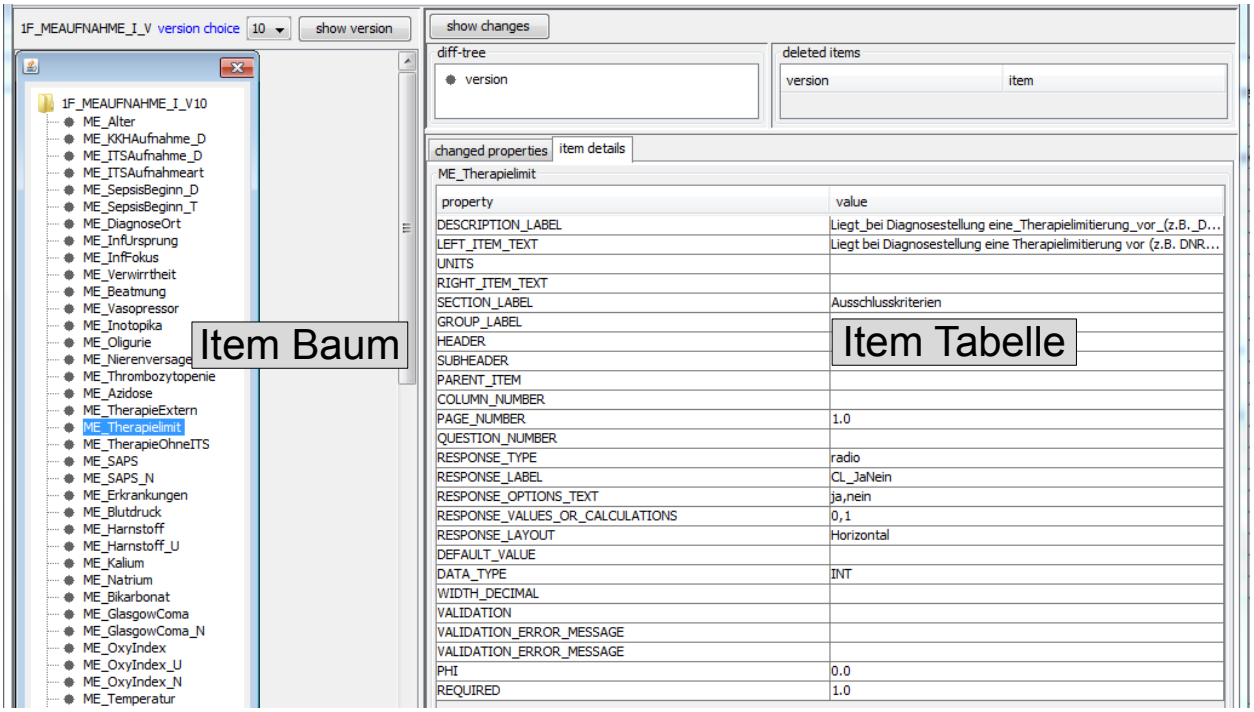

Abbildung 4: Darstellung der Items einer CRF Version in einem Baum inklusive der Ansicht der Eigenschaften eines Items in einer Tabelle

tuellen Version werden in einem Baum dargestellt. Die geänderten Items werden farblich hervor gehoben. Des Weiteren werden alle Änderungen die ein Item betreffen chronologisch in einer Tabelle als Blattknoten des betreffenden Items dargestellt. Die Tabelle enthält die Versionsnummer von der vorherigen Version, den Namen des alten Items und die Art der Änderung. Bei der Selektion eines Eintrages der Tabelle, werden die geänderten Eigenschaften im unteren Teil in einer Tabelle präsentiert. Diese Tabelle enthält den Namen der geänderten Eigenschaft, den alten und neuen Wert und die Kategorie der Änderung. Der Anwender ist in der Lage die geänderten Items

nach der Kategorie der Änderung zu filtern, indem er im category selection Menü die jeweiligen Kategorien selektiert. Des Weiteren werden die gelöschten Items in einer separaten Tabelle chronologisch nach ihrem Löschzeitpunkt präsentiert, da diese in der aktuellen Version nicht vorhanden sind.

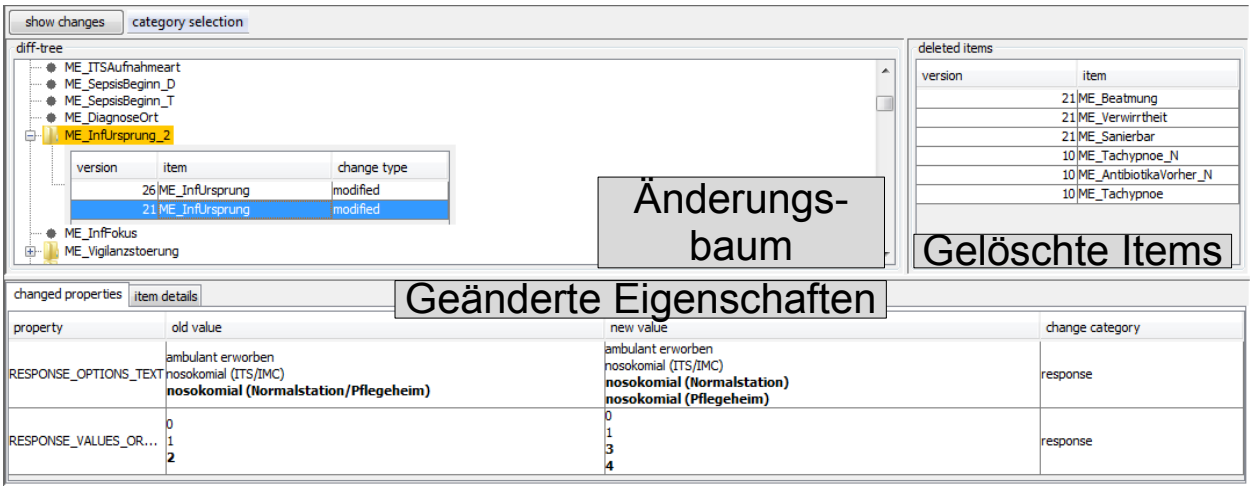

Abbildung 5: Darstellung der Änderung der geänderten Items und der geänderten Eigenschaften

5 Zusammenfassung