

Dynamische Formulare

Konzept

Dynamische Formulare dienen dazu, GECAMed zu erweitern und an die Bedürfnisse von individuellen Praxen bzw. an verschiedene medizinischen Spezialisierungen (vom Allgemeinmediziner über den Kinderarzt bis hin zum Zahnarzt, siehe [Abbildung 1](#)) anzupassen. In den Formularen werden Datensätze von Patientendaten gespeichert, für die es ansonsten keine geeignete Eingabemöglichkeit in GECAMed gibt. Außerdem können Berechnungen auf bereits vorhandenen Daten durchgeführt werden sowie Datenserien aus mehreren Datensätzen in Tabellen und Graphen dargestellt werden. Das alles unterscheidet die „dynamischen Formulare“ von den „statischen Formularen“, die in GECAMed fest eingebaut sind und vom Benutzer nicht oder nur begrenzt angepasst werden können.

Im **Formular-Editor** können Formularvorlagen neu erstellt, editiert, ex- und importiert werden. Ein Benutzer der mit den entsprechenden Rechten ausgestattet ist kann neue **Formularvorlagen** erstellen, von denen ausgehend beliebig viele **Formulare** (= Datensätze) erstellt und zu den Patientendossiers hinzugefügt werden können. Zur Verbreitung der Formularvorlagen auf verschiedenen GECAMed Installationen lassen diese sich als XML-Dateien ex- und importieren.

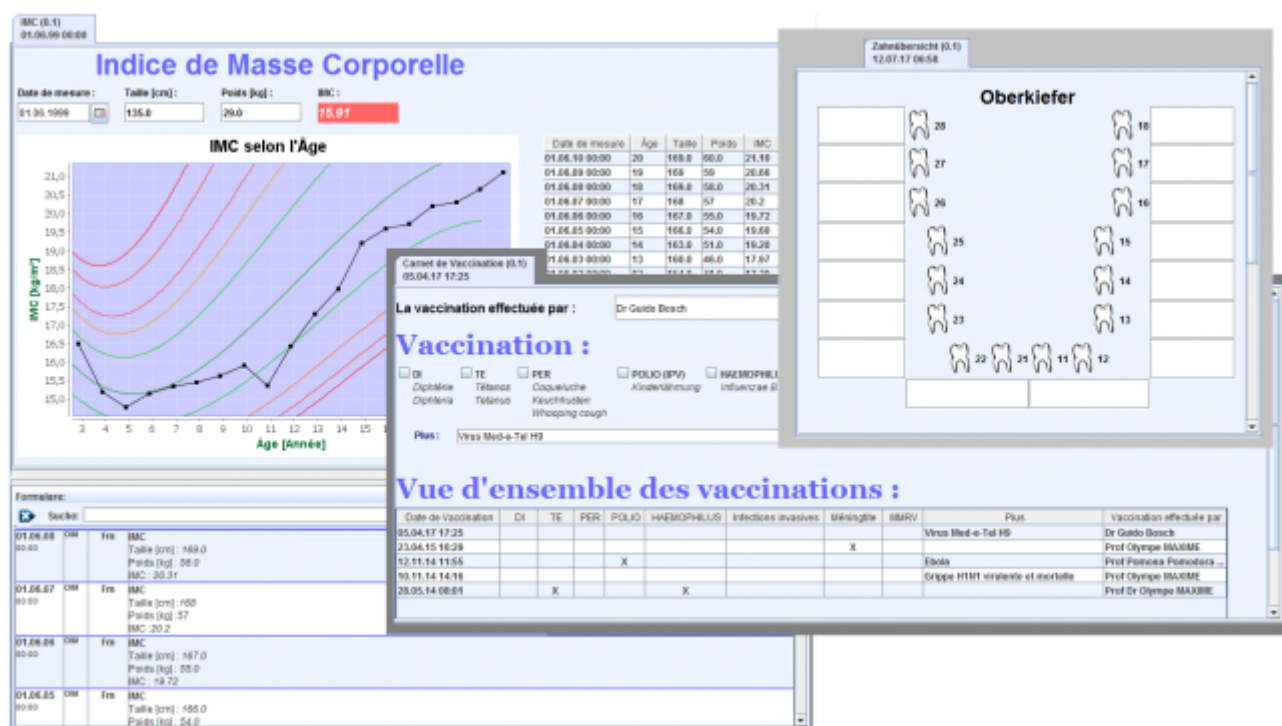


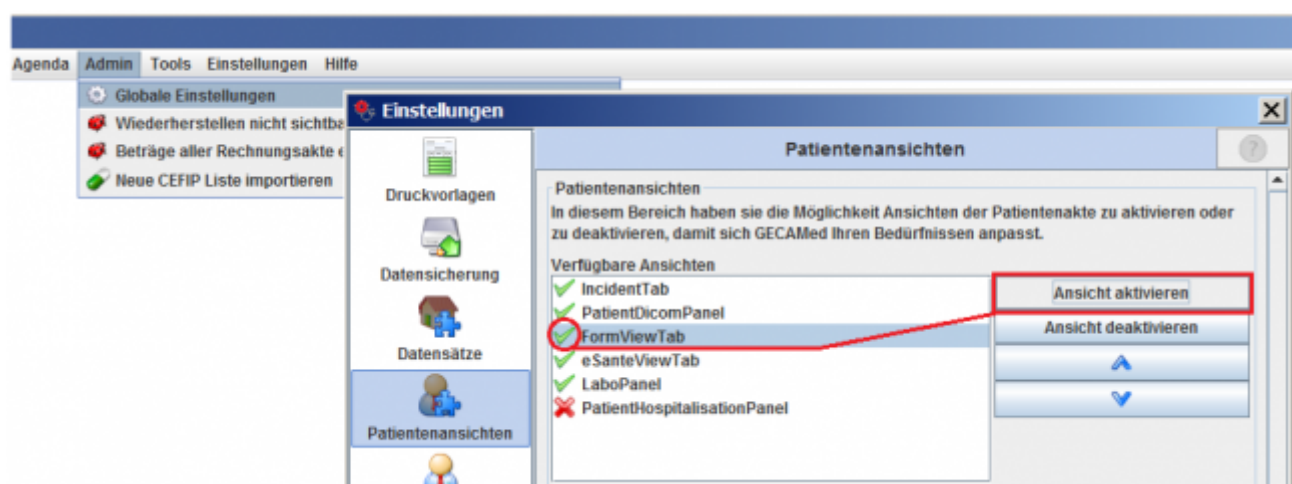
Abb. 1: Einige Beispiele für Formulare



Die für einen Patienten bereits erstellten Formulare finden Sie in der [Historie des Patienten Moduls](#). Eine spezielle [Formularansicht](#), die nur die Formulare eines Patienten anzeigt, kann über die - Schaltfläche (1) ausgewählt werden. Über die Schaltfläche „Neues Formular“ (2) können Sie für den aktuell ausgewählten Patienten ein neues Formular nach einer vorhandenen Formularvorlage erstellen.

Einrichten

Um die dynamischen Formulare in GECAMed nutzen zu können muss zunächst die Ansicht **FormViewTab** in den globalen Einstellungen für Patientenansichten aktiviert werden. Die Ansicht ist standardmässig inaktiv.



Formular-Editor



Den Formular-Editor erreichen Sie mit der **FormEditor**-Schaltfläche in der Modulleiste. Um diese sehen zu können muss das **"FormEditorModule"** aktiviert sein, und der Benutzer muss die **Rolle Formular-Editor-Administrator** haben, ansonsten ist das Modul für Ihn nicht benutzbar.



Diese Rolle ist nur für einen Formular-Administrator wichtig, also einen Benutzer, der Formulare erstellt, verändert, importiert, exportiert usw. Ein Benutzer der nur die bestehenden Formulare benutzt braucht diese Rolle nicht.

Vorlagenverwaltung

Beim Anklicken der FormEditor Schaltfläche in der Modulleiste öffnet sich der der Formular-Editor (siehe [Abbildung 2](#)).

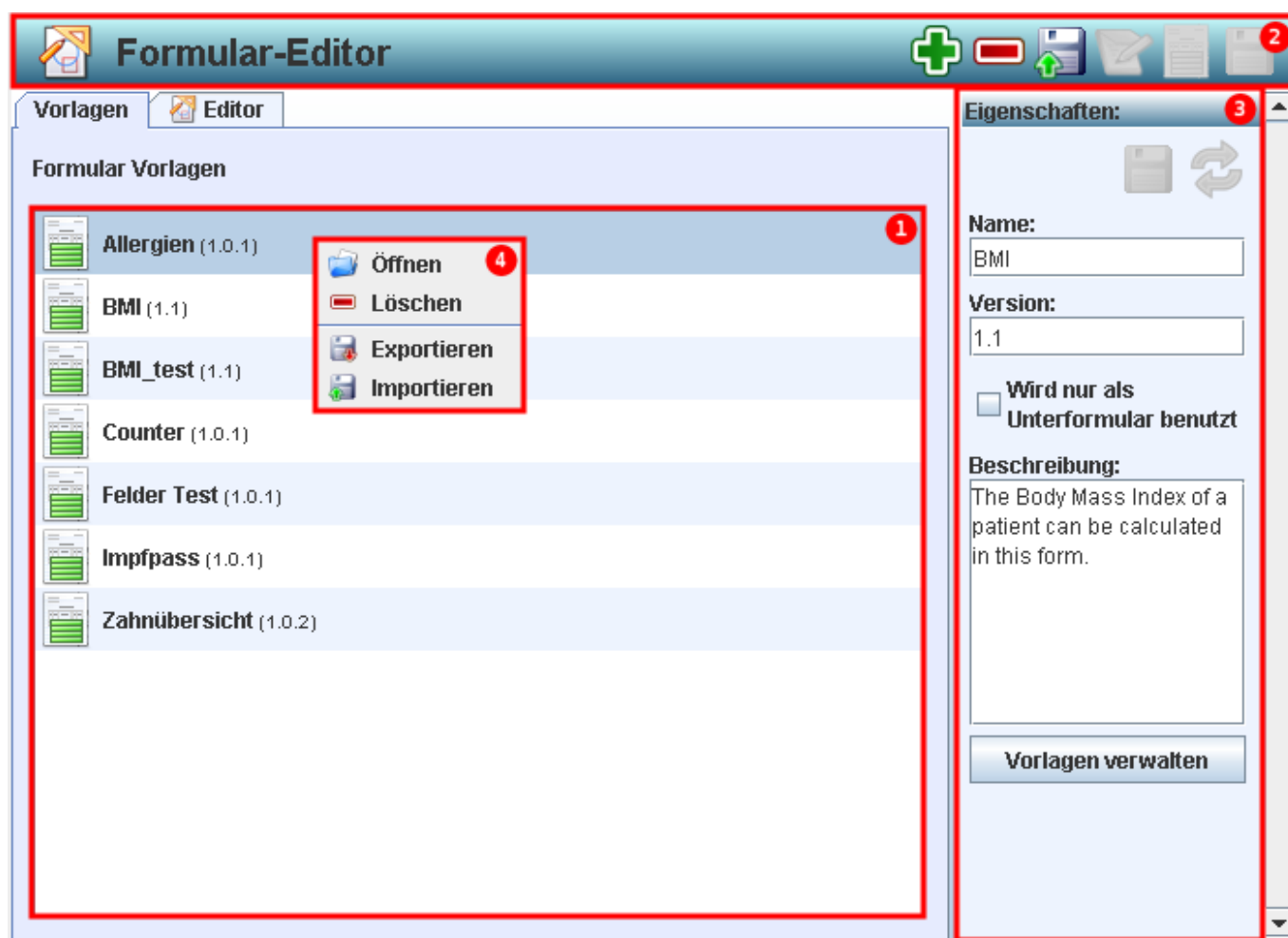
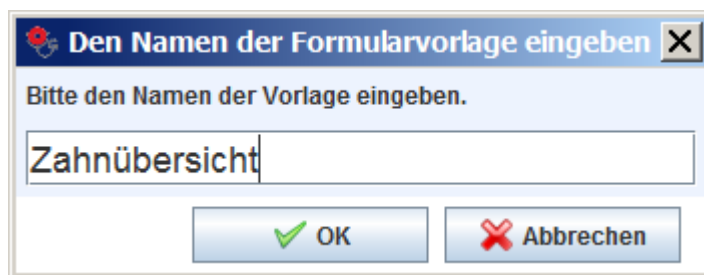


Abb. 2: Formular-Editor Vorlagenverwaltung

Zunächst wird der Vorlagen-Karteireiter angezeigt, in dem die Formularvorlagen verwaltet werden. Die Vorlagenliste (1) ist die Liste aller vorhandenen Formularvorlagen. Die Funktionen zur Verwaltung der Vorlagen befinden sich einerseits in der Titelleiste (2) und andererseits in Kontextmenü (4).

Neue Vorlage erstellen

Mit der -Schaltfläche wird eine neue, leere Vorlage erzeugt. Ein Dialog öffnet sich mit dem der Name der neuen Vorlage erfasst wird.

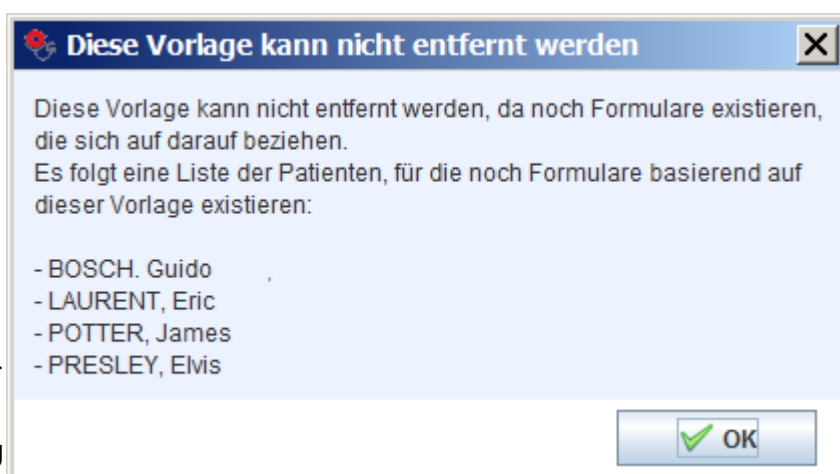


In den Eigenschaften der Vorlage (angezeigt in der Vorlagenverwaltung - siehe [Abbildung 2](#) ³) können rechts im Eigenschaften-Menü (siehe [Abbildung 2](#) ³) der Name, eine Versionsnummer und die Beschreibung der Vorlage definiert werden. Zudem kann bestimmt werden, ob diese Formularvorlage nur als Unterformular (in einer anderen Formularvorlage) Verwendung findet. Falls dem so ist, taucht diese Vorlage nicht in der Liste der Formularvorlagen auf, aus denen neue Formulare erstellt werden kann. Desweiteren können mit der Schaltfläche „Vorlagen verwalten“ [die Druckvorlagen eine Formularvorlage](#) definiert werden.

Die neue, noch leere Vorlage kann dann durch [öffnen](#) editiert werden.

Vorlage entfernen

Mit der -Schaltfläche bzw. der entsprechenden Funktion im Kontext Menü (siehe [Abbildung 2](#) ⁴) wird die ausgewählte Vorlage **entfernt**. Gelöscht werden kann eine Vorlage nur dann, wenn noch keine Formulare basierend auf der Vorlage erstellt wurden, bzw. wenn alle auf dieser Vorlage basierenden Formulare vorher gelöscht wurden. Ist dies nicht der Fall, wird der Benutzer in einem Dialog darauf hingewiesen. Dieser Dialog enthält auch die Liste der Patienten, deren Historie ein oder mehrere auf dieser Vorlage basierendes Formulare enthält.



Auf diese Weise wird verhindert, dass benutzte Vorlagen nicht aus Versehen gelöscht werden.

Vorlage importieren

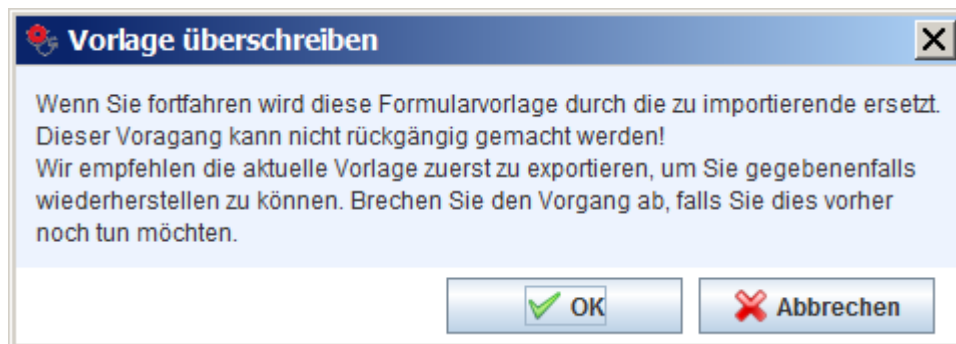
Mit der -Schaltfläche wird eine Vorlage aus einer XML-Datei importiert. Sollte bereits eine Vorlage gleichen Namens existieren wird nach einem anderen Namen verlangt.

Wird die entsprechenden Funktion aus dem Kontext

Menü (siehe [Abbildung 2](#) ⁴)

aufgerufen verhält sie sich etwas anders: Anstatt eine neue Vorlage anzulegen wird die aktuell ausgewählte überschrieben. Die Import-Funktion in der Titelleiste ist

also dazu da, neue Vorlagen hinzuzufügen, während die Import-Funktion im Kontextmenü dazu gedacht ist, eine neue Version der Vorlage einzuspielen. Allerdings überprüft GECAMed nicht, ob die neue Vorlage tatsächlich nur eine andere Version der bereits existierenden ist und überschreibt diese einfach. Zur Sicherheit wird in einem Dialog auf das Risiko eines möglichen Datenverlustes durch Überschreiben hingewiesen.



Vorlage exportieren

Vorlagen können als XML-Dateien exportiert und abgespeichert werden. Dazu benutzt man die Exportieren-Funktion aus dem Kontext-Menü der aktuell ausgewählten Vorlage (siehe [Abbildung 2](#) ⁴).

Beim Exportieren wird eine komplette Definition der Formularvorlage erstellt und gespeichert. Der Inhalt der Vorlage (Datenfelder, Layout, Scripts, ...) wird zusammen mit allen Meta-Daten aus dem Eigenschafts-Bereich (Name, Version, ...) in einer XML-Datei abgespeichert. Diese kann dann durch Importieren in anderen GECAMed Installationen wiederverwendet werden.

Vorlage öffnen

Mit der -Schaltfläche wird die aktuell ausgewählte Vorlage in den Editor-Reiter geladen, wo sie verändert werden kann. Die gleiche Funktion wird durch Doppel-Klicken auf einer Vorlage ausgeführt.

Formulare zum Herunterladen und Importieren

Die folgenden Formulare wurden als Beispiele der Möglichkeiten des Formeditors entwickelt und können jederzeit in GECAMed als „Add-Ons“ installiert werden. Dazu laden Sie zunächst die benötigten Dateien im ZIP Format herunter und entpacken die darin enthaltene XML-Datei.

Download Links:

- [Zahnübersicht](#)
- [BMI mit Wachstumskurven](#)
- [Carnet de Vaccination \(Impfpass\)](#)

Installieren Sie diese dann mit der Funktion [Vorlage importieren](#) des [Formular-Editors](#).

Zahnübersicht

Ein sehr einfaches Formular für den Zahnarzt, das die Erfassung von Anmerkungen pro Zahn in einem FDI-konformen Zahnschema ermöglicht.

Oberkiefer					
		28		18	
		27		17	
plombée		26		16	
		25		15	
manque		24		14	
		23		13	
		22		12	

Unterkiefer					
		38		48	manque
		37		47	
		36		46	cassé
		35		45	
		34		44	
		33		43	
		32		42	

détaillée

Impfbuch / Carnet de vaccination

Ein den luxemburger Anforderungen entsprechendes Impfbuch. Es werden einzelne Impftermine mit den jeweiligen Impfungen erfasst und in chronologischer Reihenfolge in der Patienten-Historie hinterlegt. Als Besonderheit werden bei der Anzeige eines solchen Eintrages in der Formular-Ansicht alle vorangegangenen Impfungen in einer Tafel zusammengefasst, wodurch der Eindruck des Impfbuches entsteht.

La vaccination effectuée par : Prof Pomona Pomodora SPROUT

Vaccination :

☐ DI Diphthérie Diphtheria
☐ TE Tétanos Tetanus
☐ PER Coqueluche Keuchhusten Whooping cough
☒ POLIO (IPV) Kinderlähmung
☐ HAEMOPHILUS Influenzae B
☐ HEPATITE B Hépatite B
☐ Infections invasives à streptococcus pneumoniae invasive infectios
☐ Méningite méningocoques C Hirnhautentzündung
☐ MMRV / RORV ROUGEOLE / Masern OREILLONS / Mumps RUBEOLE / Roteln VARICELLE / Varizellen

Plus : Ebola

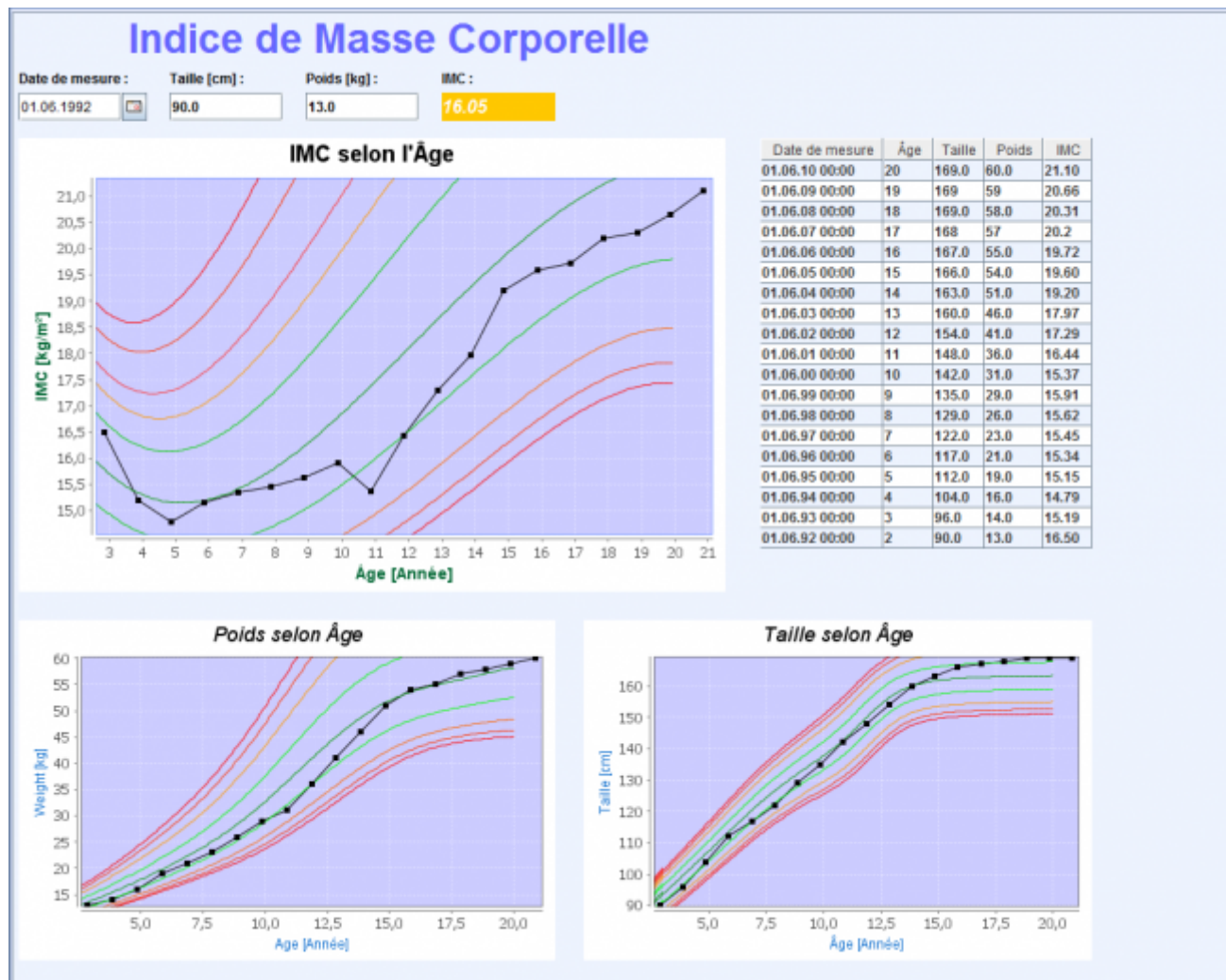
Vue d'ensemble des vaccinations :

Date de Vaccination	DI	TE	PER	POLIO	HAEMOPHILUS	Infections invasives	Méningite	MMRV	Plus	Vaccination effectuée par
05.04.17 17:25									Virus Med.e-Tel H9	Dr Guido Bosch
23.04.15 16:29							X			Prof Olympe MAXIME
12.11.14 11:55				X					Ebola	Prof Pomona Pomodora ...
10.11.14 14:16									Grippe H1N1 virulente et mortelle	Prof Olympe MAXIME
28.05.14 08:01		X			X					Prof Dr Olympe MAXIME

Zur Benutzung kreuzen Sie einfach alle am heutigen Datum vorgenommenen Impfungen an (Diphterie, Thetanus, ...), und tragen Sie im Textfeld „Plus“ weitere Impfungen ein, falls diese nicht zur Auswahl stehen. Beim Abspeichern wird unter dem heutigen Datum eine neue Zeile in der Impfpass-Tabelle mit den eingetragenen Impfungen und dem aktuellen Arzt angelegt.

BMI / IMC

Das Formular zur Erfassung des Body Mass Index (BMI) resp. *Indice de masse Corporelle (IMC)* stellt bereits eine komplexe Anwendung des Formular-Editors da, für deren Programmierung einige Kenntnisse in Java und JavaScript nötig sind. Trotz der sehr einfachen Erfassung von Körpergröße und Gewicht pro Arztbesuch (Date de mesure, Taille, Poids - IMC wird berechnet) wird durch Kummulieren und Anzeigen der vergangenen Werte in 3 verschiedenen Diagrammen ein analytisches Werkzeug geschaffen, in dem die erfassten Wachstumswerte des Patienten auf einen Blick mit der „Normalentwicklung“ verglichen werden können. Ein sehr nützliches Werkzeug für Kinderärzte.



Ähnlich wie beim Impfbuch werden auch in diesem Formular die vorherigen Werte zur Erstellung einer Historie benutzt, sowohl in Tabellenform wie auch in Diagrammen (charts), auf denen das Alter des Patienten auf der X-Achse dargestellt wird. Das Formular eignet sich auch sehr gut zur Demonstration der [Import-Funktion einer Messreihe](#). Falls bereits vorhandene Aufzeichnungen über das Wachstum eines Patienten vorhanden sind können diese aufbereitet und importiert werden. Dazu muss zunächst eine Datei im CSV-Format mit allen zu importierenden Messungen erstellt werden.

Beispiel einer Import-Datei mit mehreren Messungen:

date_measurement; date_create; height; weight; BMI

```
„2006-01-01 10:00“; „2006-01-01 10:00“; „160“; „46“; „17.97“
„2007-01-01 10:00“; „2007-01-01 10:00“; „163“; „51“; „19.20“
„2008-01-01 10:00“; „2008-01-01 10:00“; „166“; „54“; „19.60“
„2009-01-01 10:00“; „2009-01-01 10:00“; „167“; „55“; „19.72“
„2010-01-01 10:00“; „2010-01-01 10:00“; „168“; „57“; „20.20“
```

zum Importieren der Daten wählt man beim Anlegen eines neuen Eintrages des IMC-Formulares die Option **Importieren**.



Der folgende Dialog zum Importieren öffnet sich, und es werden nacheinander folgende Informationen erfasst:

Wählen Sie eine CSV-Datei aus und stellen Sie die Parameter ein, um die Datei einzulesen. Klicken Sie unten in der Vorschau auf eine Spalte und wählen Sie in der Auswahlbox den Namen der dazu passenden Komponente aus.
Wenn alles korrekt eingestellt ist, starten Sie den Import mit "OK".

ichtumskurve_Beispiel.csv

Starte ab Zeile: Set to 2 to exclude header in input table

Spalten Trennzeichen:

Texttrennzeichen:

Ordne die ausgewählte Spalte zu:

#creationDate
measuringDate
lblBmi
tfHeight
tfWeight

1. Select input column
2. Choose target data field in form

1 - #creationDate	2 - measuringDate	3 - tfHeight	4 - tfWeight	5
2006-01-01 10:00	2006-01-01 10:00	160	46	17.97
2007-01-01 10:00	2007-01-01 10:00	163	51	19.20
2008-01-01 10:00	2008-01-01 10:00	166	54	19.60
2009-01-01 10:00	2009-01-01 10:00	167	55	19.72
2010-01-01 10:00	2010-01-01 10:00	168	57	20.20

- **Suchen:** Pfad der CSV-Datei mit den zu importierenden Messungen.
- **Starte ab Zeile:** Falls die Import-Datei wie in diesem Beispiel eine Kopfzeile enthält, muss diese noch beim Importieren ausgeblendet werden. Dazu die Startzeile auf 2 setzen.
- **Spalten Trennzeichen:** Das Zeichen das aufeinanderfolgende Spalten in der Import-Datei voneinander trennt. Kann zwischen Semikolon, Komma, Tabulator und Leerzeichen variiert werden.
- **Texttrennzeichen:** Das Zeichen mit dem ein Datenelement umschlossen ist (einfache oder doppelte Hochkomma).
- **Ordne die ausgewählte Spalte zu:** Als letztes müssen die in der Import-Datei vorhandenen Spalten den Datenelementen des Formulars zugeordnet werden. Beim Öffnen des Dialoges wurde bereits der Inhalt der Import-Datei gelesen und in Tabellenform dargestellt. Durch Klicken auf den Inhalt einer Spalte wird diese aktiviert und kann nun einem Datenelement des Formulars zugeordnet werden. Alle Datenelemente des Formulars müssen einer Spalte aus der Import-Tabelle zugeordnet werden.

Auf **OK** klicken um den Importvorgang zu starten. Jetzt sollte man noch manuell einen letzten, aktuellen Messpunkt erstellen, damit die Diagramme und die Tabelle einmal berechnet werden. Das Endergebnis sieht dann in etwa so aus:



Vorlagen-Editor

Der Editor wird beim Öffnen einer Vorlage aus der Listenansicht im Vorlagen-Reiter aktiviert. In der Titelleiste des Moduls werden jetzt die rechten Schaltflächen aktiviert, mit denen der [Skript-Editor](#) aufgerufen (📝), eine [Vorschau](#) angezeigt (🖨️) und die Vorlage gespeichert werden können (💾).

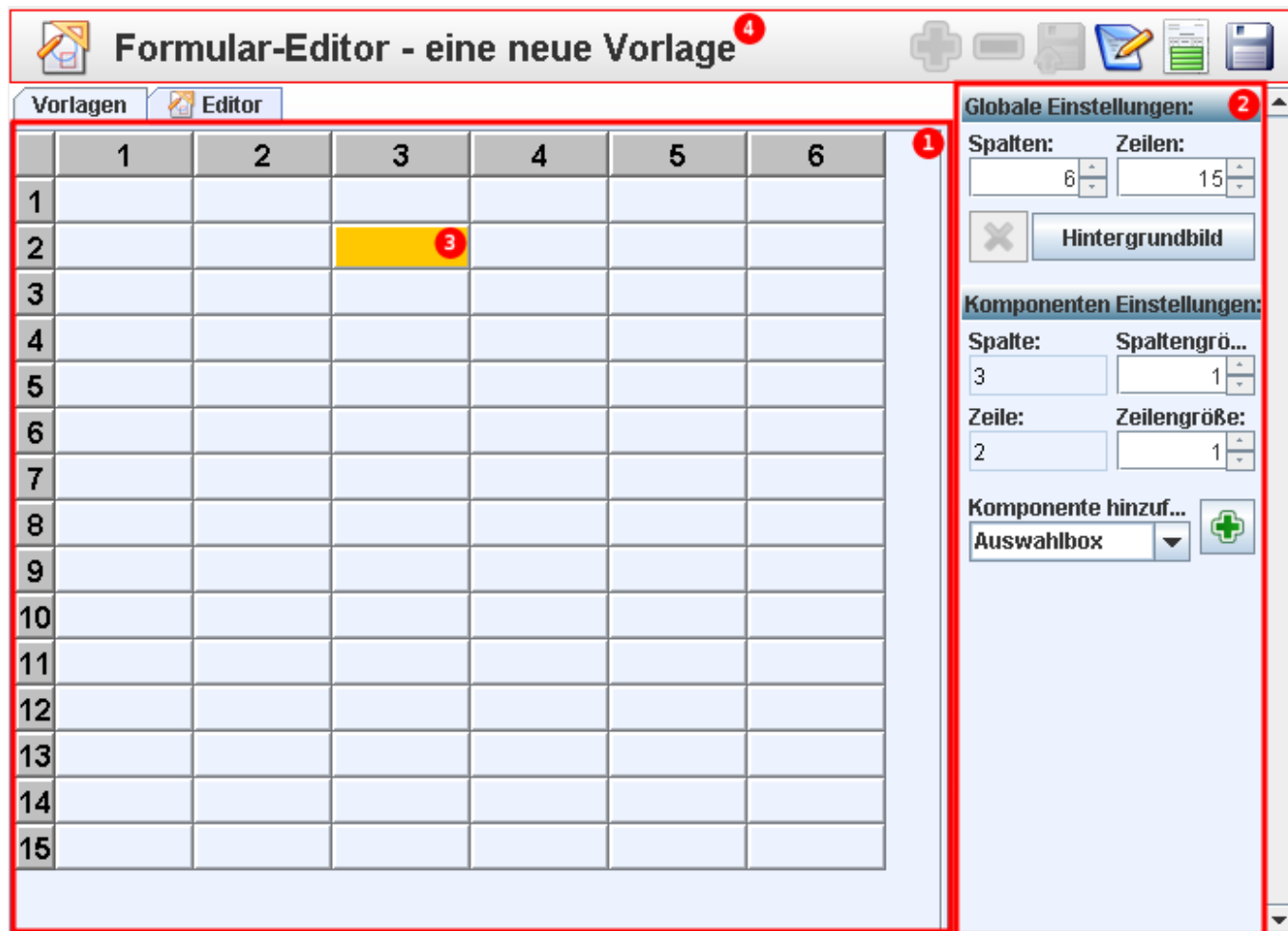


Abb. 3: Formular-Vorlagen-Editor

Im mittleren Bereich des Vorlagen-Editors befindet sich eine Gitternetzstruktur (siehe [Abbildung 3](#) ¹). Hier werden die Komponenten dargestellt, wie sie später erscheinen. Die Zellen des Editors können durch Anklicken ausgewählt werden. Die Zelle, die zur Zeit angeklickt ist, ist orange markiert (siehe [Abbildung 3](#) ³). Je nachdem welcher Zellentyp angeklickt ist - ein Leeres Feld, ein Randfeld oder eine Komponente - sehen die Einstellungen anders aus (siehe [Abbildung 3](#) ² und [Abbildung 4](#)). Die verschiedenen Arten von Zellen haben jeweils einen anderen Nutzen ...

Abb. 4: Einstellungsmenüs

Randfelder (siehe [Abbildung 4](#) ¹) dienen zur Positionierung der Leerfelder und Komponenten nach Zeile und Spalte. Sie bestimmen deren Lage im Gitternetz, sowie deren allgemeines Positionierungsverhalten. Sie befinden sich am oberen und linken Rand des Gitternetzes und sind mit 1 startend durchnummeriert. In ein Randfeld kann keine Komponente eingefügt werden. Randfelder sind im fertigen Formular nicht sichtbar.

Leerfelder (siehe [Abbildung 4](#) ² & ³) werden zwar, ähnlich wie die Randfelder, im fertigen Formular nicht sichtbar sein, nehmen aber dennoch eine bestimmte Fläche ein. Anstelle eines Leerfeldes kann auch eine Komponente eingeführt werden. Sie dienen somit gewissermaßen als Platzhalter für Komponenten. Um neue Komponenten in das aktuelle leere Feld einzufügen kann entweder die Drop-Down-Box-Ansicht ² oder die Listen-Ansicht ³ genutzt werden.

Komponentenfelder (siehe [Abbildung 4](#) ⁴) sind Elemente wie Beschriftungen (Label), Schaltflächen, Textfelder, usw., die später im Formular dargestellt werden.

Formularvorlage einstellen

Die Grundstruktur eines Formulars ist ein Gitternetz. Mithilfe dieser Gitternetzstruktur wird zunächst die matrizielle Größe des Formulars bestimmt. Dazu befinden sich am rechten Rand im oberen Teil des

Menüs zwei Zähler-Schalter, mit denen die Spalten- und Zeilenanzahl festgelegt werden. Bei deren Veränderung werden Spalten / Zeilen jeweils am rechten / unteren Rand des Gitters hinzugefügt bzw. entfernt. Wenn eine Komponente sich über eine zu entfernende Spalte, bzw. Zeile erstreckt, kann die Spalte, bzw. Zeile nicht entfernt werden. Um eine Spalte bzw. Zeile zu entfernen darf sich in ihr keine Komponente befinden.

Ist ein Randfeld ausgewählt können im Einstellungsmenü (siehe [Abbildung 3](#) ²) Einstellungen zu der Spalte bzw. Zeile vorgenommen werden. Dazu gehören die Breite bzw. Höhe, die Ausrichtung (links, rechts, oben, unten, zentriert, ausfüllen) und das Änderungsverhalten (nichts, wachsen) der Komponenten. Diese Einstellungen sind in [Abbildung 4](#) ¹ zu sehen.

Komponenten erstellen

Um eine neue Komponente zu erstellen, wählen Sie zunächst die Stelle aus, an der die Komponente eingefügt werden soll. Klicken Sie dazu im Vorlagen-Editor auf ein Leerfeld. Das ausgewählte Feld erscheint in orangener Farbe (siehe [Abbildung 3](#) ³). Am rechten Rand befinden sich nun Optionen, um eine neue Komponente zu erstellen (siehe [Abbildung 4](#) ²).

Spalte und Zeile geben an, in welcher Spalte und in welcher Zeile sich die ausgewählte Komponente befindet. Diese Felder sind rein informativ, hier kann nichts eingestellt werden. Die Spalten- und Zeilengröße gibt an, über wie viele Spalten, bzw. Zeilen sich die Komponente erstrecken wird. Eine Komponente ist also nicht auf *ein* Leerfeld beschränkt, sondern kann *mehrere* Leerfelder einnehmen. Die Größe der Komponente kann aber auch noch im Nachhinein bestimmt werden.

Neben der -Schaltfläche, die eine neue Komponente erstellt und dem Editor hinzufügt, kann ausgewählt werden, welche Komponente erstellt werden soll. Eine vollständige Liste der zur Verfügung stehenden Komponenten finden Sie [hier](#).

Nachdem die Komponente hinzugefügt wurde, ist sie direkt ausgewählt. Dabei passen sich die Optionen am rechten Rand an, da jetzt ein Komponentenfeld ausgewählt ist (siehe [Abbildung 4](#) ³).

Komponenten einstellen

Um eine Komponente einzustellen muss sie ausgewählt sein. Das passiert, genau wie bei einem Leerfeld, mit einem Klick auf die Komponente. Eine Komponente ist (normalerweise) orange hinterlegt, wenn sie ausgewählt ist. Einstellungen können dann im Menü am rechten Rand ([Abbildung 4](#) ³) vorgenommen werden.

Im oberen Teil der Einstellungszone sind wieder die Spalten- und Zeilennummer zu sehen. Daneben steht die Spalten- und Zeilengröße, die angeben, über wie viele Spalten und Zeilen sich die Komponente ausdehnt. Diese können vergrößert oder verkleinert werden. Allerdings kann eine Komponente nicht kleiner sein als eine Zelle und nicht so groß werden, dass sie eine andere Komponente überlappt oder über den Rand des Formulars hinaus ragt.

Wo bei einem Leerfeld eine Auswahlliste mit den verfügbaren Komponenten ist, wird nun angezeigt, welche Art von Komponente ausgewählt wurde (Beschriftung, Schaltfläche, Textfeld, ...). Daneben befindet sich die -Schaltfläche, mit der die Komponente entfernt werden kann.

Das Schlüsselwort definiert die Komponente eindeutig. Ein Name kann deshalb in einer Vorlage nur einmal vergeben werden. Er ist der Objektname dieser Komponente im [Skript-Editor](#). Das Schlüsselwort muss deshalb ein gültiger Java-Object Name sein und darf somit nur aus Buchstaben, Zahlen und einem Unterstrich (,_) bestehen, wobei der Name nicht mit einer Zahl beginnen darf. Ein ungültig eingegebenes Zeichen wird sofort wieder entfernt. Ein leerer oder doppelter und damit ungültiger Eintrag wird durch ein rot hinterlegtes Textfeld signalisiert. Ist der Name gültig, ist das Textfeld grün hinterlegt.

Das Kontrollkästchen „Diese Komponente speichern“ gibt an, ob der Wert dieser Komponente in der Datenbank gespeichert werden soll. Bei Eingabefeldern wie Textfelder, Textareas oder Kontrollkästchen ist dieses Häkchen standardmäßig gesetzt, bei anderen Komponenten, wie Datentabellen oder Diagrammen kann dieses Kästchen nicht gesetzt werden, da diese ihre Informationen von einer Datenbankabfrage erhalten und diese immer aktualisiert werden müssen. Außerdem gibt es in dem für das Formular generierten XML-Dokument, das von der XSL-Vorlag benötigt wird, für jede Komponente ein Attribut mit einem Wahrheitswert, der angibt, ob dieser Haken gesetzt ist.

Das Kontrollkästchen „In der Historie anzeigen“ gibt an, ob der Wert dieser Komponente in der [Historienansicht](#) von GECAMed angezeigt werden soll, wenn keine XSL Druckvorlage für die Historie definiert ist. Außerdem gibt es in dem für das Formular generierten XML-Dokument, das von der XSL-Vorlag benötigt wird, für jede Komponente ein Attribut mit einem Wahrheitswert, der angibt, ob dieser Haken gesetzt ist. Der Historien Index definiert die Reihenfolge, in der die Werte der Komponenten dann in der Historie angezeigt werden. Die Werte werden in der folgenden Form angegeben:

FORMULARNAME

BESCHREIBUNG 1: WERT 1

BESCHREIBUNG 2: WERT 2

...

BESCHREIBUNG n: WERT n

Der Text der Beschreibung kann im Textfeld unter „Beschreibung“ eingegeben werden. Die Position der Beschreibung kann unter „Beschreibungsposition“ eingestellt werden. Die Position definiert, ob die Beschreibung oberhalb, unterhalb, links oder rechts der Komponente angezeigt wird. Wird „nichts“ gewählt, wird die Beschreibung zwar in der Historie angezeigt, aber nicht im Formular.

Der Startwert einer Komponente wird auf unterschiedliche Weise definiert. Der Startwert gibt an, wie die Komponente beim Erzeugen eines neuen Formulars aussieht. Bei einem Label entspricht dies der Beschriftung, bei einem Kontrollkästchen, zum einen der Beschriftung, zum anderen, ob das Kästchen angewählt ist oder nicht und bei einer Auswahlbox, werden dort die Werte bestimmt, die ausgewählt werden können und der Wert, der zu Beginn ausgewählt ist. Manchen Komponenten kann auch ein Bild übergeben werden, dessen Skalierung angegeben werden kann.

Das Diagramm

Die Einstellung eines Diagramms ist komplexer als die anderer Komponenten. Deshalb wird das Konfigurieren des Diagramms im Folgenden genauer erklärt.

Zunächst kann unter „Diagrammtyp“ ausgewählt werden, um welche Art von Diagramm es sich handelt. Unter „Datensatzeintrag“ können Graphen (Kurven) ausgewählt, hinzugefügt, entfernt und bearbeitet werden. Zum Auswählen, welcher Datensatz bearbeitet oder entfernt werden soll, wird die

Auswahlbox benutzt. Zum hinzufügen und entfernen können die  - und  -Schaltfläche unterhalb der Auswahlbox genutzt werden.

Der Name des Datensatzes, der zum einen in der Auswahlbox erscheint und zum anderen in der Legende, kann im Textfeld „Datensatzname“ bearbeitet werden. Die Farbe des Graphen wird unter Linienfarbe geändert.

Ein Graph eines Diagramms zeigt den Verlauf eingegebener Werte eines Formulartyps. D.h. es können zwei Komponenten eines Formulartyps gewählt werden, die in der Datenbank gespeichert wurden. Von diesen Komponenten wird versucht ihre Werte in Zahlen (oder ein Datum) zu konvertieren, um diese der X- bzw. Y-Achse zuzuordnen. So erhält man einen Überblick über alle wichtigen Werte aller erstellten Formulare dieses Typs.

„Linienfarbe“ gibt die Farbe für diesen Graphen an. „Markierungen anzeigen“ definiert, ob für jeden eingetragenen Wert ein kleines Viereck an den Graphen gezeichnet werden soll oder ob die Linie des Graphen ausreicht zur Darstellung.

Die Auswahlbox „X-Achse“ definiert die Komponente, deren Werte an der Rubrikenachse angezeigt werden, die Auswahlbox „Y-Achse“ die, deren Werte an der Werteachse des Graphen angezeigt werden. Ist als Diagrammtyp Zeit-Diagramm gewählt, ist als X-Achse automatisch das Erstellungsdatum des Formulars gewählt. „Startdatum“ gibt an, ab welchem Erstellungsdatum ein Formular berücksichtigt wird. Der Wert „Anzahl an Formularen“ gibt an, die wie viel zuletzt erstellten Formulare im Graphen angezeigt werden sollen. Ist der Wert für das Startdatum gelöscht oder die Anzahl an Formularen gleich 0, wird das entsprechende Feld bei der Einschränkung zur Auswahl der Formulare nicht berücksichtigt.

Die Auswahlbox „Legende anzeigen“ gibt an, ob eine Legende mit den Namen der Datensätze angezeigt werden soll.

Jeder Graph kann auch statisch sein, d.h. seine Werte werden nicht dynamisch aus der Datenbank geladen, sondern vorab einmalig definiert. Diese Funktionalität kann genutzt werden, um beispielsweise Sollwerte darzustellen, wodurch Abweichungen von diesen leichter festzustellen sind.

CSV-Datei importieren

Wähle Sie zunächst eine CSV-Datei aus.
 Wählen Sie danach aus ab welcher Zeile die Daten beginnen, die Sie in den Graphen einbinden möchten.
 Anschließend wählen Sie bitte aus, mit welchem Zeichen die Spalten getrennt werden sollen (es kann nur ein Zeichen als Trennzeichen gewählt werden).
 Nachdem Sie das Texttrennzeichen ausgewählt haben, Ordnen Sie bitte jeweils eine Spalte der X-Achse und eine Spalte der

dateien\boys_bmi_10th.csv **Suchen...**

Starte ab Zeile: 1

Spalten Trennzeichen: Semikolon

Texttrennzeichen: "

Ordne die ausgewählte Spalte zu: x

1 - x	2
2,00	15,0712
2,08	15,0334
2,17	14,9962
2,25	14,9597

X und Y müssen einer Spalte zugeordnet werden.

OK **abbrechen**

Abb. 5: CSV-Import Dialog

Um einen statischen Graphen zu erstellen muss das Kontrollkästchen „statisch“ angeklickt werden. Wo vorher die Komponenten für X- und Y-Achse definiert wurden, erscheint nun eine Schaltfläche mit der Aufschrift „Daten aus CSV-Datei laden“. Wird darauf geklickt erscheint der CSV-Import-Dialog (siehe [Abbildung 5](#)).

Die Beschreibung (siehe [Abbildung 5](#) ¹) erklärt was hier getan werden muss. Mit der Suchen-Schaltfläche kann die gewünschte CSV-Datei ausgewählt werden (siehe [Abbildung 5](#) ²). Danach kann eingestellt werden, ab welcher Zeile die Datei genutzt werden soll, welches Zeichen die Spalten von einander trennt und welches Zeichen den Text umschließt, damit Spalten genauer angegeben werden können und das Trennzeichen auch in einer Spalte verwendet werden kann (siehe [Abbildung 5](#) ³).

Sind diese Einstellungen vorgenommen wird unten ⁵ eine Tabelle angezeigt, die die CSV-Datei widerspiegelt. Klicken Sie auf eine Spalte der Tabelle und wählen Sie in der Auswahlbox darüber (siehe [Abbildung 5](#) ⁴) eine Referenz aus, um die Werte der Spalte dieser Referenz zuzuordnen. Sollten bei der Zuordnung Fehler entstehen, so dass kein Import der Datei durchgeführt werden kann, werden diese unter der Tabelle angezeigt (siehe [Abbildung 5](#) ⁶). Solange ein Fehler besteht, bleibt die OK-Schaltfläche deaktiviert. Sobald eine Datei geladen und alle Fehler behoben wurden, kann sie angeklickt werden.

Komponenten Layout einstellen

Layout Eigenschaften:

☐ Rahmen zeichnen

Rahmenfarbe: 

☐ Hintergrund zeichnen

Hintergrundfarbe: 

Schriftstil:

☒ B ☒ I ☒ U

Schriftgröße:

Schriftfamilie:

Schriftfarbe: 

Horizontale Ausrichtung:

Vertikale Ausrichtung:

Abb. 6: Eigenschaften

Die verfügbaren Optionen zum Einstellen des Layouts können von Komponententyp zu Komponententyp variieren. Es gibt jedoch Standard-Layoutoptionen, die von den meisten Komponenten verwendet werden (siehe [Abbildung 6](#)). Die Layoutoptionen werden unterhalb der Einstellungen unter dem Menüpunkt „Eigenschaften“ vorgenommen.

Das Diagramm

Für Diagramme wird ein JFreeChart genutzt. Da das JFreeChart bereits ein Menü zum Konfigurieren des Layouts besitzt, wird kein Eigenschaften-Menü im Formular-Editor angezeigt. Stattdessen kann das bereits bestehende Menü mit einem Rechtsklick auf das Diagramm und einem Klick auf Eigenschaften verwendet werden.

Druckvorlagen Verwalten



Abb. 7: Druckvorlagen Verwaltung

Durch Klicken auf die Schaltfläche „Vorlagen verwalten“ öffnet sich die Formular-Druckvorlagen-Verwaltung (siehe [Abbildung 7](#)).

Unter dem Punkt „iReport-Druckvorlage“ kann eine neue iReport Druckvorlage geladen , die aktuelle heruntergeladen  oder entfernt  werden. Es kann pro Formular nur eine iReport-Vorlage verwendet werden.

Um die Werte von Komponenten in der Vorlage zu drucken, können die Schlüsselwörter der Komponenten als Referenz genutzt werden. Eine Einleitung zu iReport-Vorlagen erhalten Sie im [iReport Tutorial](#). Ist keine iReport-Vorlage geladen, kann iReport nicht zum Drucken des Formulars verwendet werden.

Unter dem Punkt „XSL-Druckvorlagen“ können mehrere Druckvorlagen im XSL-Format (**EX**tensible **S**tylesheet **L**anguage) angegeben werden. Mittels XSL-Dateien können Daten aus XML-Dateien in Verschiedene andere Formate konvertiert werden. In XSL wird im Prinzip ein Textdokument geschrieben, wobei Daten aus einer XML-Datei ausgelesen werden. Abhängig von der XML-Datei sieht das Textdokument immer anders aus. Dabei können Bedingte Anweisungen, Schleife und andere Programmieretechniken verwendet werden.

Man unterscheidet zwischen XSLT (**EX**tensible **S**tylesheet **L**anguage **T**ransformation) und XSL-FO (**EX**tensible **S**tylesheet **L**anguage - **F**ormating **O**bjects).

XSLT beschreibt die Transformation von Daten. Mehr über XSLT und ein gutes Tutorial finden Sie unter www.w3schools.com/xsl.

XSL-FO ist ein Ausgabeformat und kann einfach in PDF, RTF (für MS Word und OO Writer) oder andere Ausgabeformate umgewandelt werden. Mehr über XSL-FO und ein gutes Tutorial finden Sie unter www.w3schools.com/xslfo.

Zunächst wird der Name der XSL-Druckvorlage bestimmt ¹. Unter Name der Vorlage geben Sie den Namen der Druckvorlage an, so wie er später bei der Druckvorlagenauswahl angezeigt wird. „Endung“ bestimmt die Dateierweiterung der aus XML mittels XSL erstellten Datei. Anhand der Dateierweiterung entscheidet ihr Betriebssystem, mit welchem Programm die Datei geöffnet wird. Ist es eine XSL-FO Druckvorlage entscheidet GECAMed außerdem anhand der Endung, in welches Format die Datei konvertiert wird.

Mittels den Schaltflächen unter  kann die aktuelle Druckvorlage editiert , gelöscht  oder eine neue geladen  werden.

Im Bereich  kann angegeben werden, für was die Druckvorlage verwendet wird. Maximal eine der Druckvorlagen kann zum anzeigen in der Historie verwendet werden. Diese muss im HTML Format sein. Ist keine ausgewählt, wird aus dem Formular automatisch eine HTML-Datei erzeugt. Die XSL Vorlage kann zum direkten Drucken verwendet werden oder, falls die Datei vor dem Drucken noch bearbeitet werden soll, weil z.B. der Arzt noch etwas hinzufügen möchte, kann die Datei bearbeitet werden. Ein RTF-Dokument kann z.B. sehr gut mit MS Word oder OO Writer geöffnet werden.

Damit man eine Vorstellung hat, wie die XML-Datei eines Formulars aussehen kann, die von der XSL-Datei verwendet wird, um die Ausgabedatei zu erstellen, kann man sich eine Beispiel XML-Datei erstellen lassen. Dazu steht die Schaltfläche „Daten-XML Beispiel“ zur Verfügung. Dabei wird ein neues, leeres Formular von der ausgewählten Formularvorlage erstellt und davon eine XML-Datei erzeugt.

Formularvorschau

Um eine Formularvorlage zu testen, kann eine Vorschau mit einem Klick auf die -Schaltfläche erstellt werden. Dabei wird aus der aktuellen Vorlage ein Formular erstellt, ohne dabei die Vorlage in der GECAMed-Datenbank zu speichern. Als Patient wird der aktuell gewählte Patient, als Arzt, **der aktuell gewählte Arzt** verwendet. Dabei wird nichts für den Patienten abgespeichert. Es werden lediglich Daten ausgelesen, die auch dann ausgelesen werden würden, falls aktuelle Arzt für den aktuellen Patienten ein Formular dieser Vorlage erstellen würde. Es kann jedoch aus bestimmten Gründen nicht auf alle Daten zugegriffen werden, weshalb beispielsweise ein Diagramm in der Vorschau nicht richtig angezeigt werden wird.

Skript-Editor

Mit Hilfe des Skript-Editors kann das Verhalten eines Formulars wesentlich beeinflusst werden. So kann z.B. auf Benutzeraktionen reagiert, Berechnungen ausgeführt und sogar Komponenten verändert werden. Wer Skripte selber erstellen möchte, sollte über Java und JavaScript Kenntnisse verfügen. Das ist notwendig, da es sich bei der Skriptsprache um JavaScript handelt und im Skript auf vordefinierte Java Objekte zugegriffen werden kann und neue Java Objekte erstellt werden können. Mit diesen Objekten kann dann wie in Java gearbeitet werden. Falls keine Java Kenntnisse vorhanden sind, sollten zumindest Grundkenntnisse in der objektorientierten Programmierung vorhanden sein.

Nützliche Links zur Einführung in diese Themen:

[Galileo Computing - Java ist auch eine Insel \(Ereignisse beim AWT\)](#)

[Galileo Computing - JavaScript: Browser übergreifende Lösungen](#)

[Java Scripting Programmer's Guide](#)

Der Script-Editor wird durch einen Klick auf die -Schaltfläche in der Titelleiste des Formular-Editors geöffnet.

Grundlegender Aufbau des Skript-Editors

Der Skript-Editor ist ein eigenes Fenster, das zusätzlich zu GECAMed geöffnet wird und anders als ein Dialogfenster, gleichzeitig benutzt werden kann. Er stellt eine kleine Entwicklungsumgebung für die Skripte der dynamischen Formulare dar.

In der Mitte befindet sich ein Texteditor, in der die Skripte editiert werden. Er verfügt über Syntax-Hervorhebung für JavaScript.

Im oberen Teil befinden sich Schaltflächen zum Speichern, Zurücksetzen auf den zuletzt gespeicherten Punkt, Rückgängig machen, Wiederherstellen, Suchen und Ersetzen. Dabei erfolgt das Speichern einer Änderungen nicht direkt in der GECAMed Datenbank oder einer XML-Datei, sondern gibt nur die Änderungen an die aktuell editierte Formular-Vorlage weiter. Diese verwendet also erst die neuen Skripte, wenn sie im Skript-Editor gespeichert wurden. Mit der -Schaltfläche wird nur das momentan geöffnete Skript gespeichert. Erst mit der -Schaltfläche werden alle Skripte dieser Komponente gespeichert.

Am linken Rand befindet sich die Auswahl der zu editierenden Skripte. In der Auswahlbox kann die Komponente ausgewählt werden, deren Skripte editiert werden sollen. Dabei wird jeweils der Komponentenschlüssel angezeigt. Das erste Skript ist die Initialisierungsmethode, die global für alle Komponenten gleich ist und nur einmal, nämlich direkt nach dem öffnen des Formulars aufgerufen wird.

Darunter befinden sich die für diese Komponente möglichen Ereignisse. Standardmäßig werden nur die am häufigsten benötigten Ereignisse dargestellt. Wird das Kontrollkästchen „erweiterter Modus“ aktiviert, werden alle verfügbaren Ereignisse angezeigt. Die Ereignisse entsprechen den Methoden, der Listener, die an die entsprechende Komponente angemeldet werden können (eine Schaltfläche entspricht einem `javax.swing.JButton` – entsprechend wird hier unter anderem das Ereignis „actionPerformed“ zu finden sein).

Als drittes werden die für diese Formularvorlage definierten Funktionen angezeigt. Funktionen werden vom Benutzer selbst definiert. Es können jeder Zeit Funktionen hinzugefügt, entfernt und umbenannt werden. Funktionen sind genau wie die Initialisierungsmethode global für alle Komponenten eines Formulars verfügbar, werden jedoch nicht automatisch aufgerufen, sondern können innerhalb eines Skripts (also in der Initialisierungsmethode, in einem Ereignis oder in einer Funktion) aufgerufen werden.

Ein Skript schreiben

In diesem Kapitel werden JavaScript Kenntnisse, ein grundlegendes Verständnis von objektorientierter Programmierung und Kenntnisse über das Event-Handling von `javax.swing`-Komponenten vorausgesetzt. Es werden nicht die Grundlagen der Java Scripting API erläutert, sondern nur die Besonderheiten in der Verwendung von Java Scripting für den Formular-Editor. Eine detailliertere Anleitung über die Java Scripting API ist im weiter oben erwähnten [Java Scripting API Programmer's Guide](#) zu finden.

Jede im Formular-Editor eingefügte Komponente entspricht einer `javax.swing`-Komponente, also einer `JComponent`. An jede dieser `JComponents` sind alle von ihr unterstützten Listener angemeldet. Die Ereignisse, die an der linken Seite angezeigt werden entsprechen den Methoden, die von einer Klasse implementiert werden müssen, die diese Listener implementiert.



Da an einen JButton ein ActionListener angemeldet werden kann, existiert das Ereignis „actionPerformed“, das beim drücken des JButtons aufgerufen wird. Das Skript, das dem actionPerformed-Ereignis hinterlegt ist, wird dann ausgeführt.

Auf jede Komponente im Formular kann mit ihrem Schlüsselwort als Objektname zugegriffen werden und sie kann wie ein Java Objekt verwendet werden. Es gibt also zum einen die Komponenten, die vom Benutzer eingefügt werden, zum anderen aber auch fest definierte Objekte auf die innerhalb des Skriptes zugegriffen wird.

Eine Liste der Komponententypen und der vorgegebenen Objekte sind in in den beiden darunterstehenden Tafeln zu sehen.

Einfügbare Komponententypen

Komponententyp	Klasse	Beschreibung
Auswahlbox	javax.swing.JComboBox	Eine Box, in der mehrere Werte gespeichert werden können, von denen einer ausgewählt ist. Das gespeicherte Objekt, das der Auswahlbox übergeben wird, hat eine Übersetzung und einen Wert. Als Model enthält die Auswahlbox einen Vektor von ComboBoxElement-Elementen. Mit getTranslation kann die Übersetzung zurückgegeben werden, mit getValue der Wert.
Datentabelle	javax.swing.JTable	Eine nicht editierbare Tabelle, deren Einträge aus der Datenbank gelesen werden. Diese Tabelle ist die Schriftliche Darstellung des Diagramms, nur das hier nicht nur Zahlen- und Datumswerte eingetragen werden können, sondern alles
Datumswählbox	com.toedter.calendar.JDateChooser	Eine Box zum auswählen des Datums. Einführung, JDateChooser API
Diagramm	org.jfree.chart.ChartPanel	Ein durch ein JFreeChart erzeugter Panel, der eine graphische Darstellung von Daten repräsentiert. ChartPanel API
Kontrollkästchen	javax.swing.JCheckBox	Ein Label mit einem Kontrollkästchen, dass einen booleschen Wert speichert.
Label	javax.swing.JLabel	Eine Beschriftung, die einen Text aufnehmen kann, der über das Skript verändert werden kann.
Numerisches Textfeld	javax.swing.JTextField	Ein JTextField, in das nur Zahlen eingegeben werden können. Außerdem besitzt es noch die Methoden getValue, die einen double-Wert zurückgibt und setValue, die einen double-Wert oder einen Wert vom Typ Numeric annimmt.
Schaltfläche	javax.swing.JButton	Eine Schaltfläche, die gedrückt werden kann, um ein Ereignis auszulösen.
Tabelle	javax.swing.JTable	Eine editierbare Tabelle

Komponententyp	Klasse	Beschreibung
Textarea	javax.swing.TextArea	Ein Texteingabefeld, dessen Text über mehrere Zeilen gehen kann und einen Scrollbalken anzeigt, wenn der Text zu groß für die Komponente ist.
Textfeld	javax.swing.TextField	Ein einzeliges Texteingabefeld.

Verfügbare fixe Objekte

Objektname	Klasse	Beschreibung
componentList	java.util.ArrayList<JComponent>	Aller eingefügten Komponenten in einer Liste.
event	java.util.EventObject	Das EventObject, dass jedem Event, bzw. Ereignis übergeben wird.
formContext	java.util.HashMap<String, Object>	Dieses ist das einzige Objekt, auf das von allen Events, Funktionen und der initialisierungs Methode eines Formulars aus zugegriffen werden kann. Hier können Daten zwischen gespeichert werden um sie innerhalb des Formulars von jeder Stelle aus zu benutzen.
mainPanel	javax.swing.JPanel	Das Panel, auf dem die Komponenten liegen. Als Layout besitzt dieses Panel ein FormLayout.
patient	eine Java-Bean	Der Patient, für den dieses Formular erstellt wurde, bzw. der zurzeit ausgewählt ist.
physician	eine Java-Bean	Der zurzeit ausgewählte Arzt (unten rechts).



Bei nachträglichen Änderungen an einer Formularvorlage, d.h. nachdem diese bereits in Patientendossiers benutzt wurde, sollten die Name von Komponenten, deren Wert in der Datenbank gespeichert wird, unbedingt gleich bleiben, da es sonst zu inkonsistenten Daten kommen kann.

Eine Übersicht über die verfügbaren benutzerdefinierten Komponenten, fest definierten Objekte und Funktionen kann mit einem Rechtsklick auf den Texteditor des Skript-Editors aufgerufen werden. Per klick auf eines der Elemente wird es an die aktuelle Cursorposition des Editors eingefügt.

BMI berechnen - Ein Beispielskript

	1	2	3	4	5	6	7
1		Body Mass Index					
2		Height [cm]:		Weight [kg]:		BMI:	
3		0	1	0	2		3
4							

Abb. 8: Struktur der

BMI Berechnung

Für ein besseres Verständnis, wie Skripte hinterlegt werden, soll im Folgenden ein Formular erstellt

werden, das den Body Mass Index (BMI) eines Patienten berechnet.

Gehen wir davon aus, dass im Formular-Editor die Struktur bereits erstellt wurde (wie in [Abbildung 8](#) zu sehen). Dabei besteht ein numerisches Textfeld mit dem Schlüssel „tfHeight“ ¹ zum eintragen der Größe des Patienten, eines mit dem Schlüssel „tfWeight“ ² zum eintragen des Gewicht des Patienten und ein Label mit dem Schlüssel „lblBmi“ ³ in das das BMI hineingeschrieben wird.

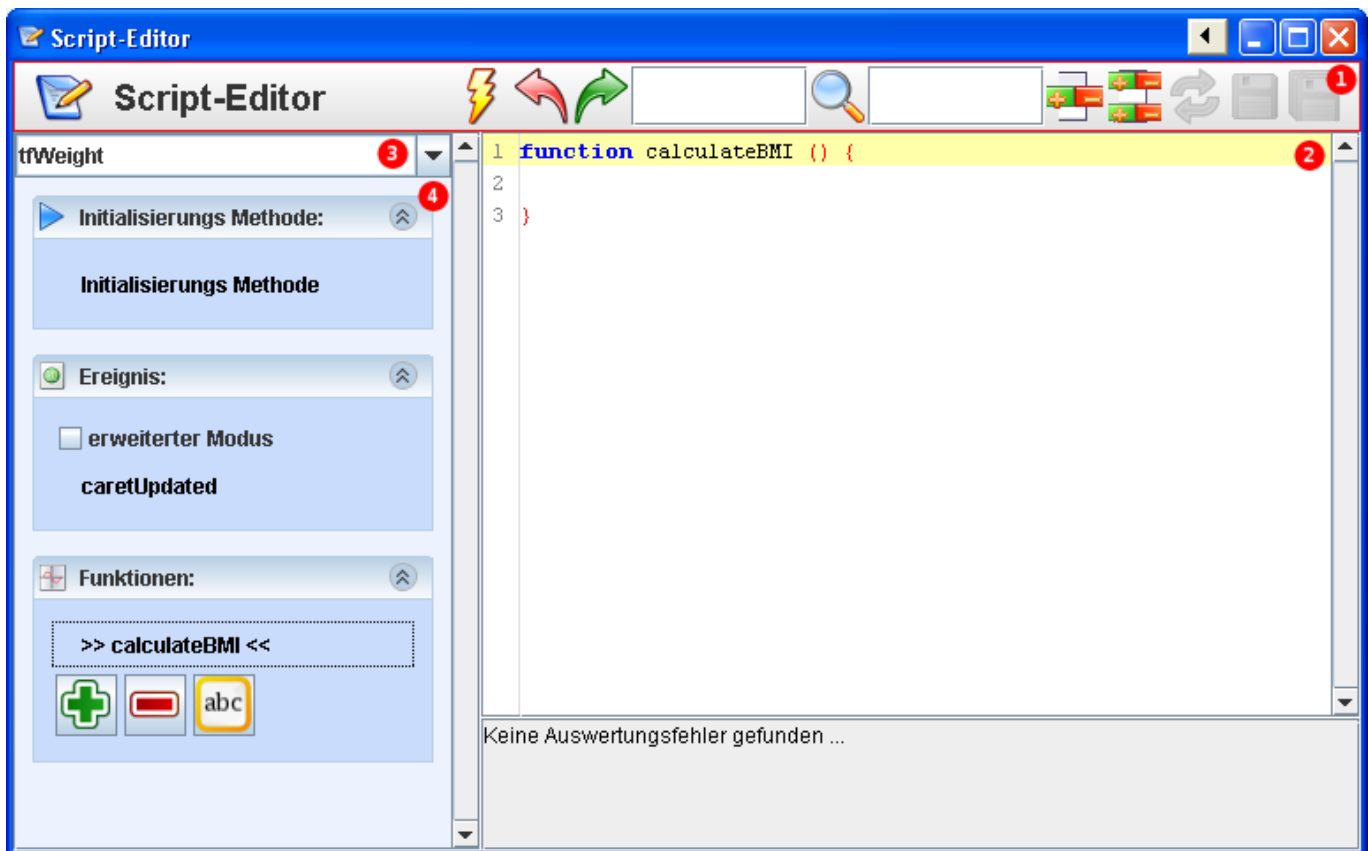


Abb. 9: Skript-Editor

Öffnen Sie nun den Skript-Editor und erstellen Sie eine neue Funktion, indem auf die -Schaltfläche links unterhalb von „Funktionen“ in Bereich ⁴ klicken. Es öffnet sich ein Dialog, indem Sie den Namen der Funktion eingeben, in unserem Fall „calculateBMI“. Klicken Sie nun auf den neu erschienenen Eintrag in der Liste der Funktionen namens „calculateBMI“, um das Skript zu öffnen.

Der Rumpf der Funktion existiert bereits, der Skript-Editor sollte so aussehen, wie in [Abbildung 9](#) ² zu sehen. Wir füllen diesen Rumpf nun mit Anweisungen.

Zunächst sollen die Werte für Größe und Gewicht des Patienten aus den Textfeldern ausgelesen werden:

```
var height = new java.lang.Float(tfHeight.getValue());
var weight = new java.lang.Float(tfWeight.getValue());
```

Die Variablen *height* und *weight* enthalten jetzt die Größe und das Gewicht des Patienten. Der BMI wird berechnet, indem das Gewicht (in Kilogramm) durch die Größe (in Meter) zum Quadrat geteilt wird, also:

$$\text{BMI} = \text{Gewicht [kg]} / \text{Größe [m]}^2$$

Da wir die Größe des Patienten in cm angeben lassen, müssen wir sie vorher noch in Meter umrechnen. Das Skript zum berechnen des BMIs würde also aussehen wie folgt:

```
height = height / 100;  
  
var bmiValue = weight / (height * height);
```

Da wir aber keine Division durch 0 wollen, prüfen wir zunächst noch ab, ob die Größe des Patienten ein sinnvoller Wert ist. Außerdem wollen wir keine Zahl mit vielen Nachkommastellen, sondern nur zwei Nachkommastelle. Anschließend setzen wir den berechneten Wert als Text des Labels *lblBmi*.

```
if (height <= 0) {  
    return;  
}  
  
/* Berechneter Wert mal 100, auf ganze Zahl runden und danach durch 100  
rechnen,  
* um genau zwei Nachkommastelle zu erhalten.  
*/  
var bmiValue = java.lang.Math.round(100 * (weight / (height * height))) /  
100;  
  
height = height / 100;  
  
var bmiValue = weight / (height * height);  
  
lblBmi.setText(bmiValue);
```

Um das ganze etwas anschaulicher zu gestalten, könnte man die Hintergrundfarbe des Ergebnisses je nach Wert des BMIs verändern.

Normalgewicht hat man beispielsweise mit einem BMI von 18,5 bis 25. Untergewicht ist also alles unter 18,5, Übergewicht alles über 25 (stark vereinfacht dargestellt). Wir färben also das BMI grün, für Normalgewicht und rot für Über- und Untergewicht.

```
if (bmiValue < 18.5 || bmiValue > 25) {  
    lblBmi.setBackground(java.awt.Color.RED);  
} else {  
    lblBmi.setBackground(java.awt.Color.GREEN);  
}
```

Um nicht immer den Packagenamen vor dem Klassennamen zu schreiben, kann ein Package auch importiert werden. Für die hier verwendeten Packages sähe das aus wie folgt.

```
importPackage(java.lang);  
importPackage(java.awt);
```

Die gesamte Funktion könnte folglich so aussehen:

```
/* Calculates the BMI out of the given values  
*/  
function calculateBMI () {  
    // IMPORTS  
    importPackage(java.awt);
```

```

importPackage(java.lang);

/* Get the patient's size and weight and
 * allocate it to the variables height and weight.
 */
var height = new Float(tfHeight.getValue());
var weight = new Float(tfWeight.getValue());

/* The patient's size must be > 0,
 * else: exit the function as you cannot divide through 0
 */
if (height <= 0) {
    return 0;
}

// convert the patient's size from cm to m
height = height / 100;

// calculate the BMI
// (rounded on two decimal places)
var bmiValue = Math.round(100 * (weight / (height * height))) / 100;

// set the calculated bmi as value of the label
lblBmi.setText(bmiValue);

// set the color, depending on the BMI
if (bmiValue < 18.5 || bmiValue > 25) {
    lblBmi.setBackground(java.awt.Color.RED);
} else {
    lblBmi.setBackground(java.awt.Color.GREEN);
}
}

```

Sie können jeder Zeit die Syntax des geöffneten Skripts auf Fehler prüfen. Klicken Sie dazu auf die



-Schaltfläche. Daraufhin erscheint unterhalb des Editors eine Nachricht in rot, falls Fehler bestehen oder in schwarz, die Ihnen sagt, dass keine Fehler bestehen.

Wählen Sie nun links als Komponente das numerische Textfeld *tfHeight* und wählen Sie dann das Ereignis „caretUpdated“, das immer dann ausgelöst wird, wenn irgendetwas im Eingabebereich des Textfeldes passiert. Rufen Sie dort die Funktion *calculateBmi* auf. Das gleiche tun Sie für die Komponente *tfWeight*.

Wenn alles gespeichert ist, können Sie eine Vorschau aufrufen. Dort werden Sie sehen, wie der Wert des Labels entsprechend geändert wird, sobald etwas in eines der Textfelder eingetippt wird.

Hier können Sie sich die Formularvorlage als XML-Datei zum importieren in ihr GECAMed herunterladen: [bmi_simple.xml](#)

Hier können Sie sich eine erweiterte Version der BMI-Formularvorlage herunterladen. Sie besitzt mehrere Diagramme und eine Datentabelle zur Übersicht über alle eingetragenen Daten: [bmi_advanced.zip](#)

Beispiel Funktion

Im Folgenden finden Sie ein paar Beispiel Funktionen, die so in Ihr Formular übernommen werden können oder als Vorlage für eigene Funktionen dienen können:

[calculateAge\(\)](#)
[changeGraphDateToPatientAge\(\)](#)
[clearSeriesOfDiagram\(\)](#)
[convertColumnDateToAge\(\)](#)
[convertXValueFromYearsToMonth\(\)](#)
[getBirthdayOfCurrentPatient\(\)](#)
[getBMI\(\)](#)
[manipulateChart\(\)](#)

Formularansicht

Aus den im Formular-Editor erstellten Vorlagen können in der Formularansicht neue Formulare erstellt werden. Wird ein Formular neu erstellt oder ein bestehendes geöffnet, wird es in der Formularansicht ([Abbildung 10](#)) dargestellt. Hier kann es bearbeitet, gespeichert, gelöscht und gedruckt werden.

Patient: Schulze, Peter (19840512 654) *

BMI (1.0.1)
04.06.10 14:43 *

calculate the BMI

Height [cm]:

Weight [kg]:

BMI:

Navigation:

Funktionen:

- DICOM hinzufügen
- Neues Formular**
- Zertifikat drucken
- Datei anhängen
- Neuer Brief
- Verschreibung
- Leeres Rezept
- Behandlung erfol...

Wartezimmer:

Termine:

Abrechnung:

Patienten:

1: Schulze, Peter *

Formulare:

Von: Bis: Formulare: alle

Datum	JeF	Fm	BMI
04.06.10 14:43			Weight [kg]: 0 Height [cm]: 0

Abb. 10: Formularansicht

Im unteren Bereich der Formularansicht ist eine Historie eingeblendet (siehe [Abbildung 10](#) ¹), die nur Formulare anzeigt. Das Formular selber wird in der Mitte der Ansicht (siehe [Abbildung 10](#) ²) angezeigt. Oberhalb des geöffneten Formulars sind verschiedene Karteireiter zu sehen, mit denen die Ansicht zwischen den geöffneten Formularen gewechselt werden kann.

Formulare erstellen

Sie können ein neues Formular erstellen, indem Sie die Schaltfläche „Neues Formular“ (siehe [Abbildung 10](#) ³) betätigen. Daraufhin öffnet sich ein Dialog wie in [Abbildung 11](#) zu sehen, in dem alle in dieser GECAMed-Installation vorhandenen Formular-Vorlagen angezeigt werden (siehe [Abbildung 11](#) ¹). Wählen Sie eine davon aus und bestätigen Sie mit „OK“, damit für den momentan ausgewählten Patienten ein neues Formular nach dieser Vorlage erstellt wird.

Ein neues Formular kann aus jeder Ansicht des Patienten-Moduls heraus werden. Nach dem Erstellen wechselt GECAMed immer in die Formularansicht und zeigt das gerade erstellte Formular in einem

neuen Karteireiter an.

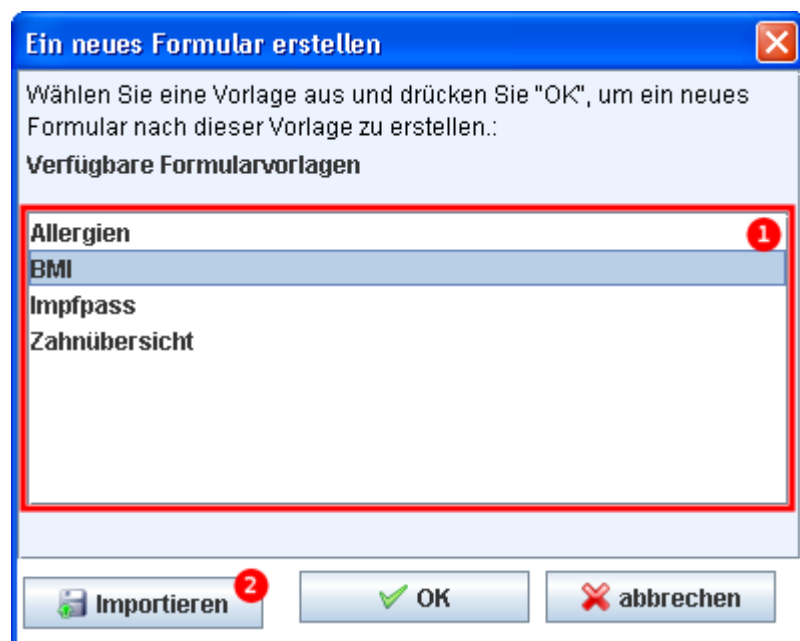


Abb. 11: neues Formular erstellen

Formulare speichern

Wird an einem Formular etwas geändert, wechselt die Titelleiste der Patientenansicht von grau nach gelb und die -Schaltfläche wird aktiviert (siehe [Abbildung 10](#) ⁴). Weil es sich hierbei um die allgemeine Speichern-Funktion des Patienten-Moduls handelt werden dadurch alle Änderungen dieses Patienten gespeichert, auch solche außerhalb der Formularansicht.

Formulare öffnen und schließen



Abb. 12: Formular Rechtsklickmenü in der Historie

Formulare können aus der Patienten-Historie heraus geöffnet werden. Dabei kann sowohl die Historie in der [Historienansicht](#), als auch die in der Formularansicht benutzt werden. Zum schnellen Öffnen eines Formulars genügt ein Doppelklick auf die entsprechende Formularzeile. Beim Rechtsklick erscheint das Kontextmenü (siehe [Abbildung 12](#) ¹). In diesem Menü kann durch Auswahl des Menüpunktes „Formular öffnen“ das Formular geöffnet werden.

Wird ein Formular geöffnet wechselt GECAMed in die Formularansicht, falls Sie sich nicht schon in dieser befinden (siehe [Abbildung 10](#)), und zeigt das Formular dort in einem neuen Karteireiter an.



Abb. 13: Kontextmenü des Kartenreiters in der Formular-Ansicht Mit einem Rechtsklick auf den Karteireiter eines Formulars öffnet sich das Kontextmenü des Fortmulares. Folgende Funktionen stehen zur Verfügung:

- **Formular Schließen:** schließt das aktuell angezeigte Formular. Falls dieses ungespeicherte Änderungen enthält muss diese Operation bestätigt werden (siehe [Abbildung 14](#));
- **Alle Formulare schließen:** alle zurzeit geöffneten Formulare des Patienten werden geschlossen. Sollte dabei ein oder mehrere noch nicht gespeicherte Formulare geschlossen werden wird eine entsprechende Bestätigung verlangt. Die gewählte Option wird dann auf alle Formulare angewand.

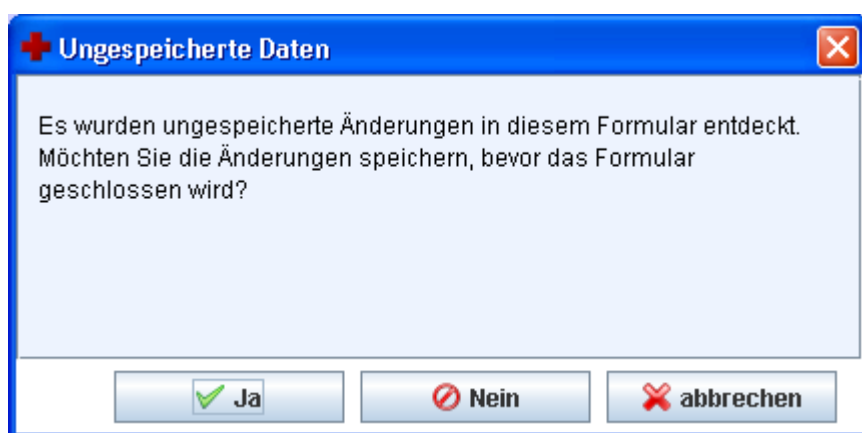


Abb. 14: Bestätigungsdialog:

Formular schließen

Formulare löschen

Formulare können direkt über die Historie durch Rechtsklick auf das entsprechende Formular und Auswahl des Menüpunktes „Formular löschen“ (siehe [Abbildung 12](#) ) gelöscht werden. Die gleiche Funktion ist bei bereits geöffnetem Formular auch über das Kontextmenü des zugehörigen Karteireiters verfügbar (siehe [Abbildung 13](#)).

Bevor ein Formular gelöscht wird, erscheint ein Bestätigungsdialog (siehe [Abbildung 15](#)). Wird der Löschvorgang mit „OK“ bestätigt, wird das Formular unwiderruflich gelöscht. Wird der Vorgang abgebrochen, bleibt das Formular bestehen.

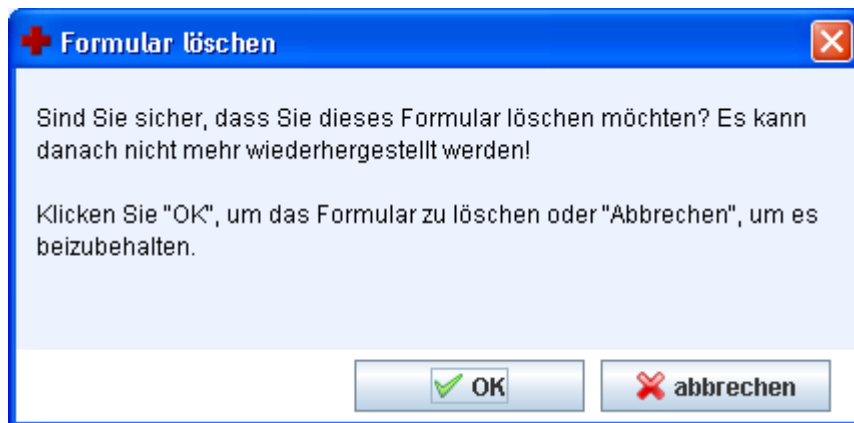


Abb. 15: Bestätigungsdialog:

Formular löschen

Datensätze importieren

Es besteht die Möglichkeit, Datensätze als Formularinhalte für eine in GECAMed bestehende Formularvorlage zu importieren. Wenn beispielsweise bereits Messungen oder Auswertungen für einen Patienten bestehen, eine Formularvorlage zum Eintragen der Daten in GECAMed aber erst nachträglich erstellt wurde, ist es eventuell mühsam für jeden Datensatz ein Formular zu erstellen und die Daten in GECAMed einzugeben. Oftmals ist es einfacher die Daten in Excel (oder ein vergleichbares Programm) einzugeben. Aus diesem kann dann ganz einfach eine CSV-Datei erstellt werden. Mittels dieser wiederum können die Werte importiert und als Formulare importiert werden.

Für jedes Schlüsselwort einer Formularvorlage, dessen Komponente in der Datenbank gespeichert wird, *kann* dabei ein Wert eingegeben werden. Für das Erstellungsdatum des Formulars *muss* hingegen ein Wert eingegeben werden. Die Erstellungsdaten können alle auf heute ausgestellt werden, genau so gut kann aber auch als Erstellungsdatum des Formulars das tatsächliche Messdatum angegeben werden. Das wäre beispielsweise Sinnvoll, falls eine Datentabelle oder ein Diagramm verwendet wird, um eine Übersicht zu geben, oder um die Werte in der Historie entsprechend anzuordnen.

Eine Spalte in der CSV-Datei stellt dabei die Werte für jeweils eine Komponente der Formularvorlage bzw. das Erstellungsdatum dar. Jede Zeile entspricht einem neuen Formular, das durch beim Import erzeugt und mit den entsprechenden Dateninhalten abgespeichert wird.

Eine CSV-Datei, für das oben gezeigte [BMI-Beispiel](#) könnte also so aussehen:

Erstellungsdatum	Größe	Gewicht	BMI
2006-01-01 10:00	160	46	17,97
2007-01-01 10:00	163	51	19,20
2008-01-01 10:00	166	54	19,60
2009-01-01 10:00	167	55	19,72
2010-01-01 10:00	168	57	20,20

Um die in einer CSV-Datei angegebenen Formulare nun automatisch erstellen zu lassen öffnen Sie den [Neues-Formular-Erstellen-Dialog](#). Wählen Sie die Vorlage aus, zu der Sie Formulare importieren möchten und klicken Sie dann auf die Importieren-Schaltfläche (siehe [Abbildung 11](#) ²). Daraufhin öffnet sich der CSV-Datei-Import-Dialog (siehe [Abbildung 5](#)). Die Bedienung ist Äquivalent zum [Importieren von statischen Graphen](#).

Ein ausführliches Beispiel für das Importieren von Datensätzen befindet sich bei der Beschreibung des [BMI / IMC Formulars](#)

formeditor#bmiimc

Formulare ausdrucken

Wie bereits unter [Formularvorlage einstellen](#) beschrieben, gibt es mehrere Möglichkeiten, ein Formular zu drucken. Via iReport oder XSL. Bei einem Rechtsklick auf das Tab eines geöffneten Formulars erscheint ein Kontextmenü (siehe [Abbildung 13](#)), welches 3 Druckoptionen zur Verfügung stellt.

Via iReport drucken

Wir „iReport Drucken“ angeklickt, öffnet sich die Druckvorschau, das Druckenmenü oder das Dokument wird umgehend gedruckt, je nachdem wie die [Druckeinstellungen](#) in GECAMed gesetzt wurden. Diese Option ist jedoch nur aktiv, wenn das Formular abgespeichert ist. Anderenfalls ist der Menüpunkt ausgegraut und kann nicht angeklickt werden.

Wie das Resultat des Druckauftrags aussieht, hängt von mehreren Faktoren ab. Zunächst natürlich vom Formular selber. Jedoch wird das Formular nicht einfach so ausgedruckt, wie es auf dem Bildschirm zu sehen ist. Um das eigentliche Formular herum können andere Informationen dargestellt werden. Dazu gibt es eine Druckvorlage, die im [Administrationsmenü für Druckvorlagen](#) eingestellt werden kann. In den PageHeader-Tag dieser iReport-Vorlage wird das Formular eingefügt. Der PageHeader-Tag sollte beim Erstellen der iReport-Vorlage deshalb frei bleiben. Diese Druckvorlage wird für jedes Formular verwendet.

Für jede Formularvorlage kann eine weitere iReport-Vorlage definiert werden. Während die erste Vorlage den Rahmen des Ausdrucks definiert, definiert diese die Darstellung des eigentlichen Formulars. Welche iReport-Vorlage verwendet wird, kann im [Formular-Editor](#) bestimmt werden (siehe [Formularvorlage einstellen](#)). Falls keine iReport-Vorlage für die Formularvorlage des Formulars gespeichert ist, wird ein Abbild (eine Art Screenshot) des gesamten Formulars gemacht und als Bild in die Druckvorlage eingefügt und gegebenenfalls herunter skaliert, damit das Formular auf eine Seite passt.

Via XSL drucken / editieren

Wenn eine der beiden Optionen ausgewählt wird, wird geprüft, wie viele XSL-Druckvorlagen für dieses Formular und diese Option zur Verfügung stehen. Sind es mehr als eins, wird ein Auswahldialog angezeigt, bevor es weiter geht. Falls keine Vorlage zur Verfügung steht ist die Option ausgegraut und kann gar nicht erst ausgewählt werden. Die Option ist außerdem ausgegraut, wenn das Formular nicht abgespeichert ist oder in den Benutzereinstellungen nicht festgelegt wurde, mit welchem Programm die zu erstellende Datei geöffnet werden soll („Einstellungen“ - „Einstellungen“ - „Formular-Editor“ - „Drucke und öffne Formulare mit ...“). Zur Auswahl stehen Microsoft Word, OpenOffice Writer und „das Standard Programm“. Letzteres bedeutet, dass der PC selber, anhand der Dateierweiterung, entscheidet, mit welchem Programm er die Datei öffnet. Dabei wird der das Standardprogramm zum Drucken genutzt, wenn „XSL drucken“ ausgewählt wurde und das

Standardprogramm zum editieren genutzt, wenn „XSL editieren“ ausgewählt wurde. Falls Word oder Writer ausgewählt wurde, wird die Datei immer mit diesem Programm geöffnet. Der Unterschied zwischen Editieren und Drucken besteht darin, dass beim Drucken die Datei zwar in Word oder Writer geladen wird, sie das aber nicht sehen, da sie nur gedruckt und direkt wieder geschlossen wird.

Hier zwei Beispiel XSL-Druckvorlagen, um am Beispiel BMI darzustellen, wie so eine Druckvorlage aussehen kann.

Zum einen eine XSLT-Datei, die eine HTML-Datei für die Historie erstellt, zum anderen eine XSL-FO-Datei, die eine HTML-, RTF- oder PDF-Datei für einen Brief erstellt: [Simple BMI XSL Druckvorlagen](#).

From:

<https://gm.apps.lu/> - **GECAMed - Gestion de Cabinets Médicaux**

Permanent link:

<https://gm.apps.lu/de/userguide/formeditor>

Last update: **2019/12/09 10:19**

