Dynamische Formulare

Présentation

Les formulaires dynamiques sont des extensions du logiciel GECAMed qui proposent diverses fonctionnalités spécifiques pour les médecins généralistes ou spécialistes (tel qu'on peut le voir sur figure 1 pour les pédiatres et les dentistes). Le but des formulaires dynamiques est de stocker des jeux de données pour patient pour lesquelles il n'y a pas d'autre possibilité de saisie appropriée dans GECAMed. En outre, il est possible de mettre en oeuvre des calculs sur des données déjà existantes, et de représenter plusieurs données du même type dans des tableaux et des graphiques. Contrairement aux zones de saisies “statiques” qui ne sont adaptables que de façon limitée, les “formulaires dynamiques” offrent à l'utilisateur la possibilité de mettre à jour des données de patients en insérant en base de donnée des valeurs nouvelles.

L'éditeur de formulaires (**FormEditor**) permet de créeret de modifier des **modèles de formulaires**. Pour cela, l'utilisateur doit disposer de privilèges appropriés. A partir d'un tel modèle on peut créer un nombre quelconque de **formulaires** (= jeux de données) et les ajouter aux dossiers patients. Pour distribuer les modèles de formulaires sur les installations de GECAMed il est également possible de les exporte et importer dans des fichiers au format XML.

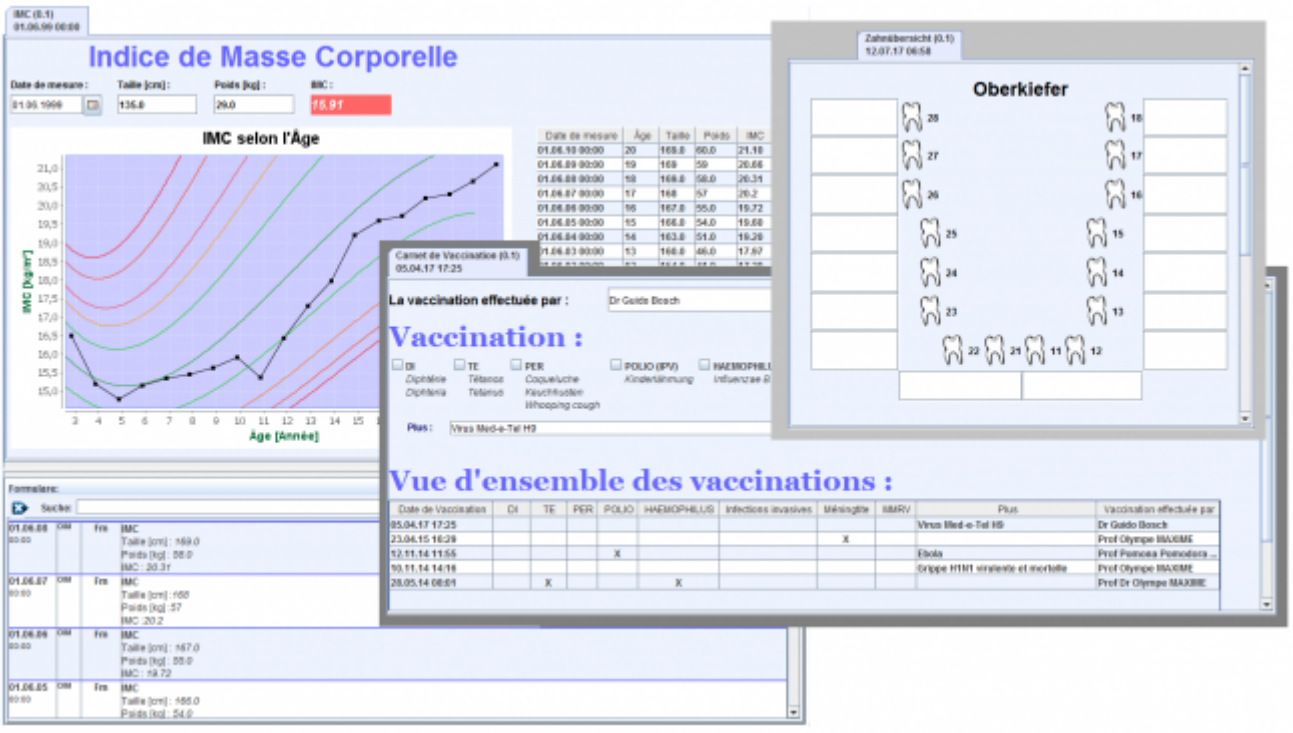
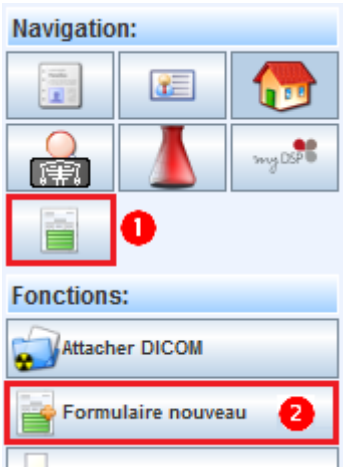


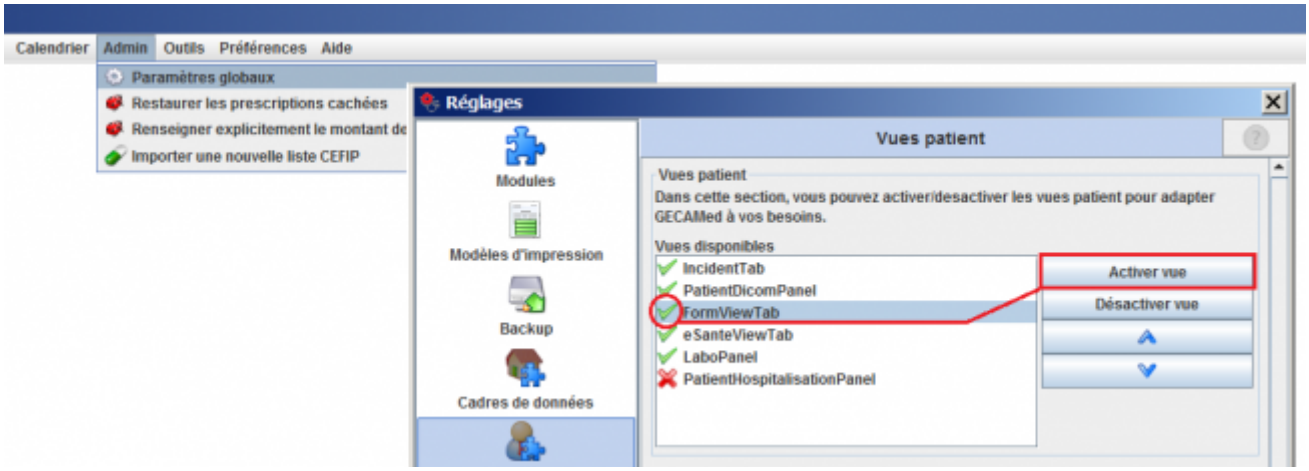
Fig. 1: Einige Beispiele für Formulare



L'utilisateur peut trouver les formulaires remplis pour un patient dans l'historique du patient. Une vue spéciale pour formulaires contenant que les formulaires d'un patient peut être activée en cliquant sur le bouton  (1). L'utilisateur peut également choisir un modèle de formulaire existant et le remplir avec des données en cliquant le bouton “Formulaire nouveau” (2).

Activer la vue formulaire

Pour pouvoir utiliser des formulaires dynamiques dans GECAMed il faut au préalable activer la vue **FormViewTab** dans les paramètres globaux du menu Admin, comme cela est montré sur l'illustration ci-dessous. Cette vue est par défaut désactivée.



Editeur de formulaire

Pour accéder au module d'édition de formulaire, l'utilisateur doit cliquer le bouton **FormEditor** situé dans la barre de module de l'écran principal. Ce bouton n'est accessible qu'à la condition que le module **FormEditorModule** a été au préalable activée par l'administrateur; de plus, l'utilisateur doit disposer du rôle **Administrateur de l'éditeur de formulaire** que l'administrateur peut lui accorder.

Ce rôle n'est important que pour un administrateur de formulaire, c.à.d. un utilisateur qui crée, modifie, importe et exporte des modèles de formulaires. Un utilisateur “normal” se servant uniquement de modèles déjà établies n'en a pas besoin.

Gestion des modèles

Lorsque l'utilisateur clique le bouton FormEditor dans la barre de modules, l'écran ci-dessous (voir figure 2) s'affiche.

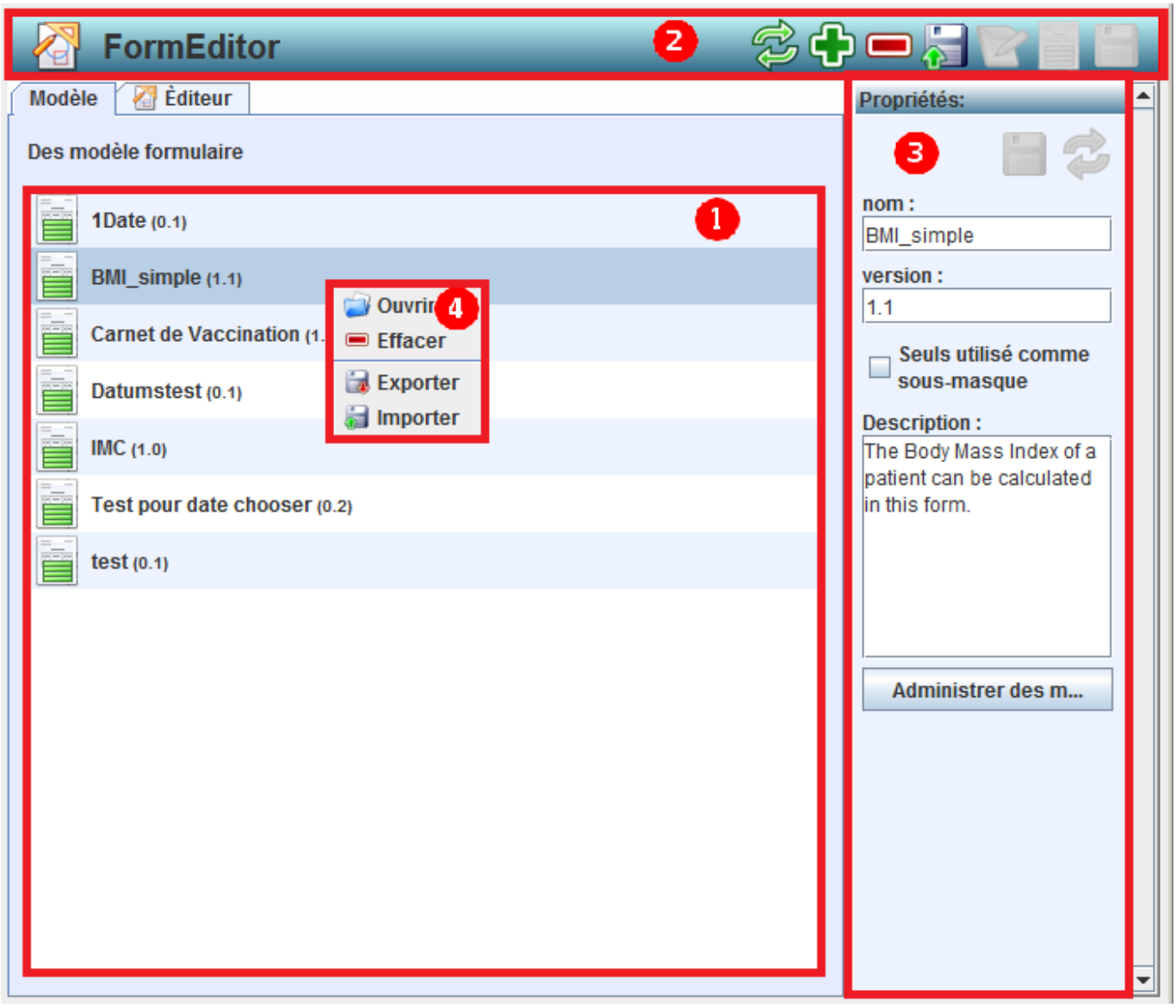
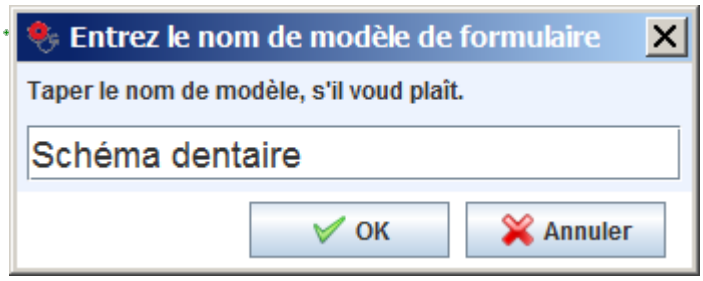


Fig. 2: Formular-Editor Vorlagenverwaltung

La zone principale comporte deux onglets, dont le premier contient la liste des modèles de formulaires disponibles (1). Les principales fonctionnalités de création, import/export et autres sont disponibles, soit dans la barre de titre (2), soit dans le menu contextuel, qui s'affiche en cliquant le bouton droit de la souris (4).

Créer un nouveau modèle

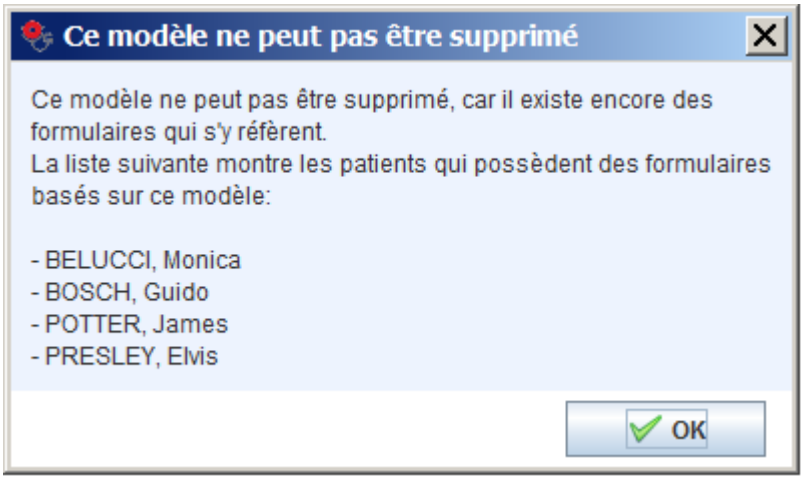
Pour créer un nouveau modèle, cliquer le bouton . Une boite de dialogue s'ouvre et l'utilisateur peut enrer le nom qu'il souhaite donner à ce modèle de formulaire.



Dans les **Propriétés** du modèle à droite (voire figure 2 3) on peut définir les propriétés générales du modèle, à savoir son nom, un numéro de version ainsi qu'une description associée à ce formulaire; de plus, l'utilisateur peut cocher la case 'seuls utiliser comme sous-masque' pour signifier que ce modèle est utilisable comme sous-formulaire (c.à.d. comme partie intégrante d'un autre formulaire: il est possible de structurer un formulaire en utilisant un ou plusieurs autres sous-formulaires); si c'est le cas, alors il n'apparaitra pas dans la liste de choix des modèles pour créer un nouveau formulaire. Avec le bouton “Administrer les modèles” on accède au dialogue de gestion des modèles d'impression pour formulaire.

Une fois créé, le nouveau modèle vierge peut alors être édité.

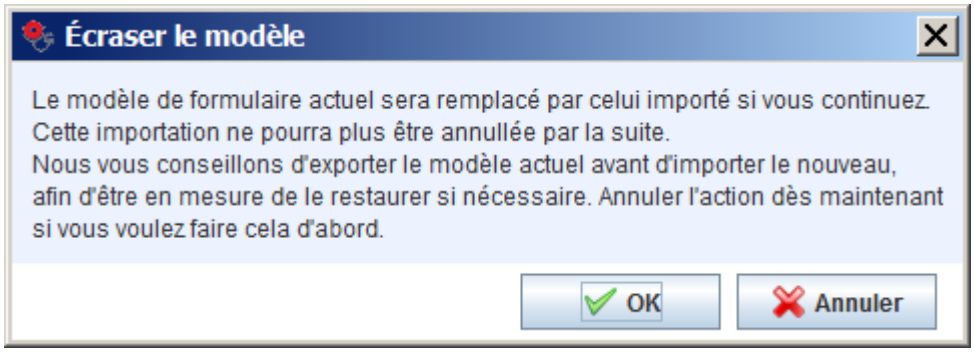
Supprimer un modèle



Avec le bouton , aussi disponible dans le menu contextuel (voir figure 2 4), l'utilisateur peut **supprimer** un modèle. Ceci n'est possible que s'il n'est pas encore utilisée comme le modèle d'un formulaire remplie dans un dossier patient; dans le cas contraire, une boite de dialogue apparait à l'écran, indiquant la liste des dossiers patients utilisant un formulaire tiré de ce modèle. Cela évite la supression accidentelle d'un modèle encore “en usage”.

Importer un modèle

En cliquant le bouton , l'utilisateur peut importer un modèle de formulaire définie par un fichier au format XML. Si un modèle du même nom est déjà présent un nom différent est demandé.



Si l'importation est appellée à partir du menu contextuel (voir figure 2 4), le fonctionnement est légèrement différent: au lieu de créer un nouveau modèle vierge, celui qui est actuellement sélectionné lors de lâppel contextuel est écrasé et remplacé par le modèle importé. GECAMed ne réexamine pas si le nouveau modèle est en effet une autre version du modèle existant, ou compatible avec celui-ci, et le remplace simplement. Par précaution un dialogue est affiché, qui explique le risque d'une perte de données par écrasement.

A noter également que l'écrasement d'un modèle par un autre peut causer des problèmes lors de l'affichage des données saisies avec le modèle précédent, si celui n'est plus compatible avec le nouveau modèle.

Il es d'autant plus important de faire une copie de sécurité par exportation d'un modèle AVANT de l'écraser par une autre version.

Exporter un modèle

Un modèle de formulaire peut être exporté et stocké dans un fichier au format XML. Cette fonction d'exportation est accessible via le menu contextuel comme il est indiqué sur figure 2 4).

La fonctionnalité d'exportation permet de stocker une définition complète de du modèle de formulaire. Le contenu du modèle (zones de données, layout , scripts, ...) avec toutes les métadonnées (nom, version, ...) est stocké dans un fichier XML. Celui-ci peut alors être réutilisé par importation dans d'autres installations de GECAMed. Nous proposons d'ailleurs un peu plus loin quelques exports de modèles de formulaire pour téléchargement.

Ouvrir un modèle

Pour ouvrir un modèle sélectionné dans la liste, l'utilisateur peut soit choisir la fonction  dans le menu contextuel, soit double-cliquer sur un modèle.

Télécharger et importer des modèles

Nous donnons ci-dessous des modèles de formulaires construits spécialemant à l'usage de mèdecins spécialistes, pédiatres et dentistes: une fois les fichiers téléchargés, l'utilisateur peut les décompresser puis importer (comme indiqué précedemment) les fichiers XML obtenus.

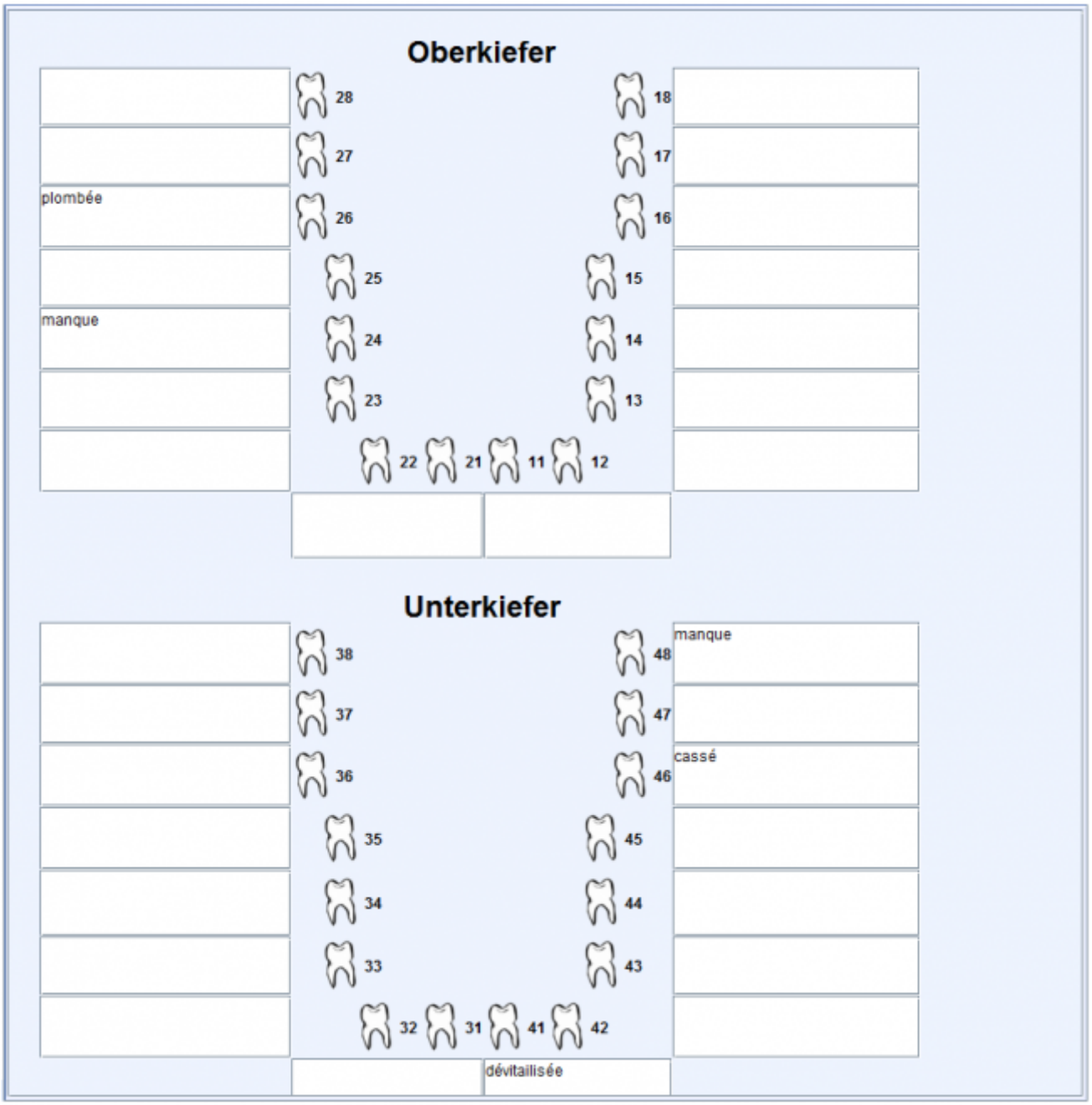
Liens de téléchargement :

- Aperçu dentaire
- IMC avec courbes de croissances
- Carnet de Vaccination

Vous pouvez les installer en utilisant la fonctionnalité import de l'éditeur de formulaires.

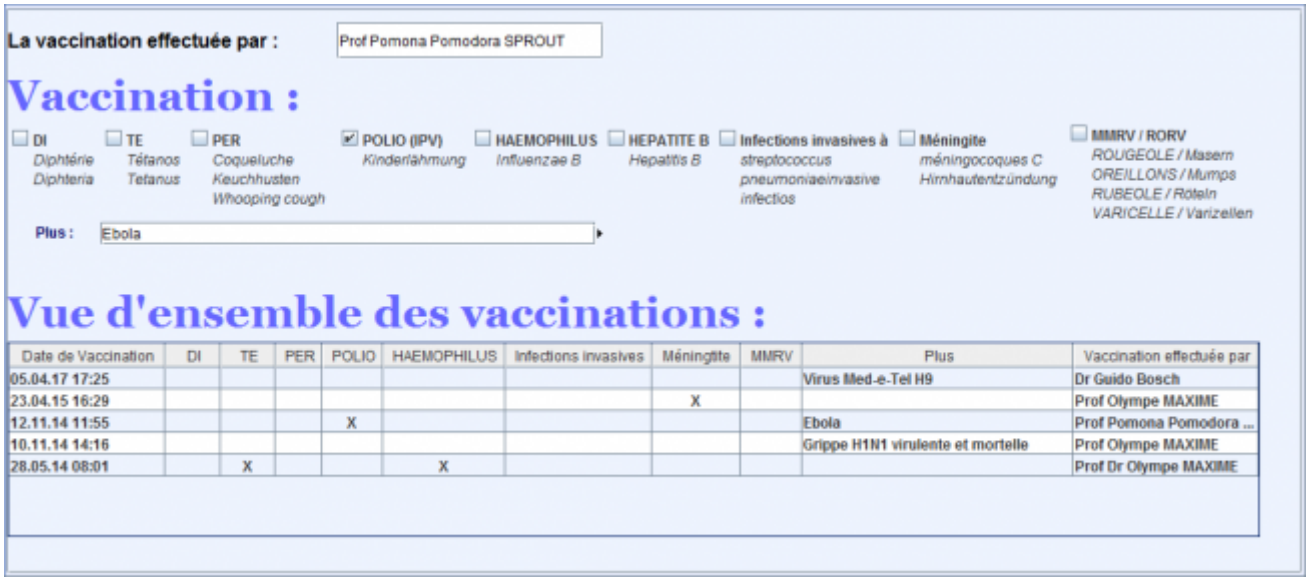
Aperçu dentaire

Un modèle de formulaire trés simple, permettant la saisie de remarques par dent dans un schéma conforme FDI, à utiliser par un chirurgien dentiste.



Carnet de vaccination

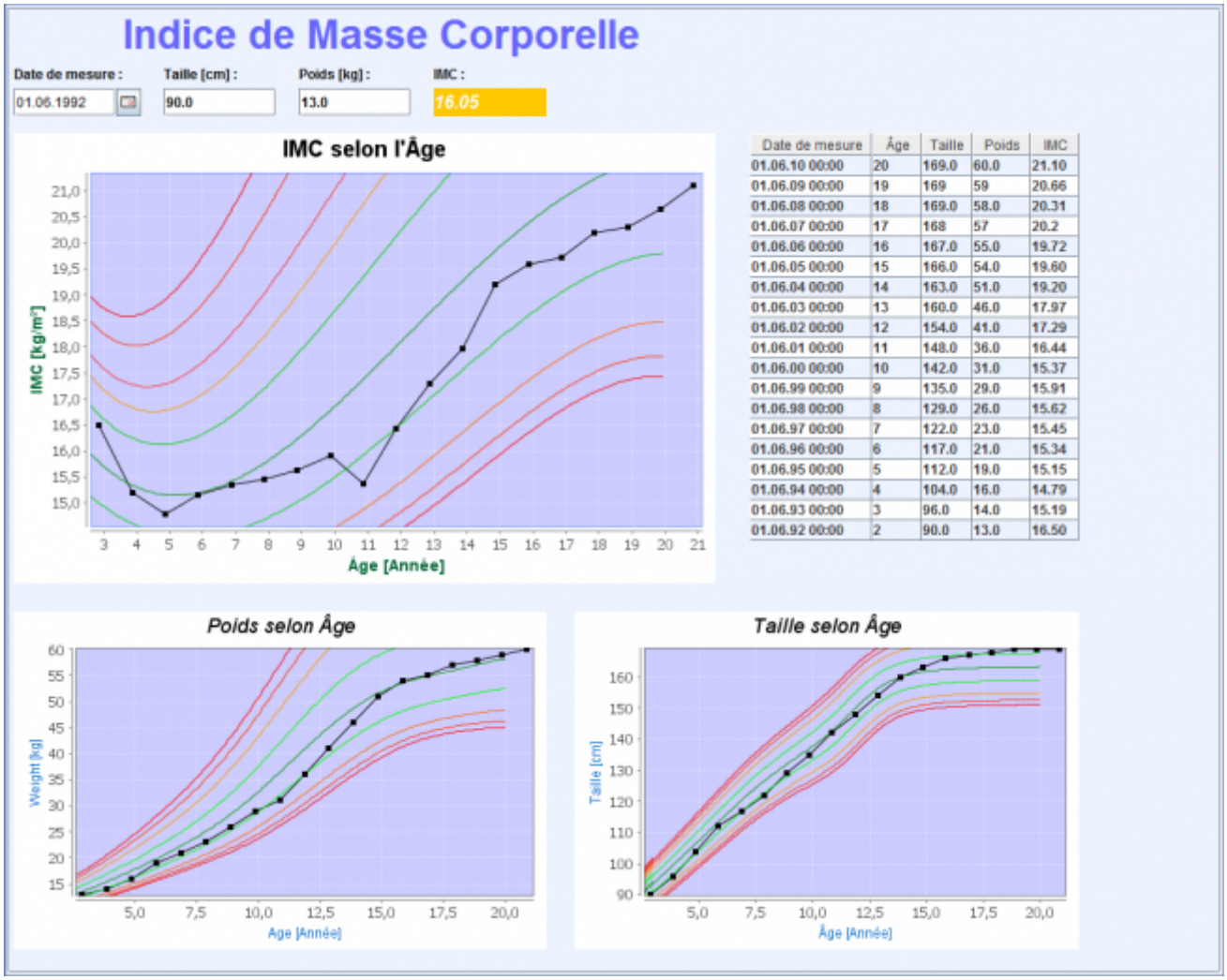
Le formulaire implémente un carnet de vaccination tel que utilisé par les médecins au Luxembourg. On saisi les vaccinations par date d'administration et les intègre ainsi dans l'historique du patient. La particularité de ce formulaire est de rassembler sous forme d'un tableau toutes les entrés précédentes par rapport à l'entré actuelle. Bien qu'il s'agisse dune suite de vaccins individuelles, cela crée l'impression d'un carnet de vaccin avec toute l'historique.



L'utilisateur peut cocher toutes les vaccins administrées aujourd'hui (Diphterie, Thetanus, ...), et s'il y a d'autres qui ne sont pas proposées il peut les complèter textuellement dans la case “Plus”. Lors de la sauvegarde le formulaire ajoute pour la date d'aujourd'hui une nouvelle ligne dans le tableau avec les vaccins renseignées et le nom du médecin courrant.

IMC / BMI

Le formulaire d'indice de masse de corporelle IMC (ou *BMI: body mass index*), particulièrement utile pour les pédiatres, est relativement simple à utiliser, mais sa construction est plus complexe que les deux modèles précédents et nécessite des conaissances en programmation Java et JavaScript. L'utilisateur saisit trois valeurs numériques - date, poids et taille - l'IMC est calculé automatiquement. Malgré cette utilisation simple, le fonctionnement du formulaire complexe car il cumule les valeurs historiques pour les afficher en forme de courbes de croissance dans 3 tableaux différents. Combiné avec l'affichage des courbes de croissance comparatives, cela crée un outil analytique pour evaluer la croissance du patient actuel par rapport à une croissance “normale”. Nous donnons avec l'illustration ci-dessous un aperçu de résultat possible.



Date de mesure	Âge	Taille	Poids	IMC
01.06.10 00:00	20	169.0	60.0	21.10
01.06.09 00:00	19	169	59	20.66
01.06.08 00:00	18	169.0	58.0	20.31
01.06.07 00:00	17	168	57	20.2
01.06.06 00:00	16	167.0	55.0	19.72
01.06.05 00:00	15	166.0	54.0	19.60
01.06.04 00:00	14	163.0	51.0	19.20
01.06.03 00:00	13	160.0	46.0	17.97
01.06.02 00:00	12	154.0	41.0	17.29
01.06.01 00:00	11	148.0	36.0	16.44
01.06.00 00:00	10	142.0	31.0	15.37
01.06.99 00:00	9	135.0	29.0	15.91
01.06.98 00:00	8	129.0	26.0	15.62
01.06.97 00:00	7	122.0	23.0	15.45
01.06.96 00:00	6	117.0	21.0	15.34
01.06.95 00:00	5	112.0	19.0	15.15
01.06.94 00:00	4	104.0	16.0	14.79
01.06.93 00:00	3	96.0	14.0	15.19
01.06.92 00:00	2	90.0	13.0	16.50

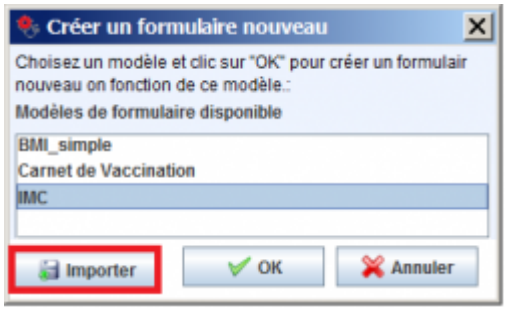
Comme dans le carnet de vaccination, ce formulaire utilise également les valeurs précédents pour créer un historique en forme de tableau, mais aussi en diagramme temporelle (chart). L'age du patient se situe sur l'achse X, la valeur du IMC sur l'Achse Y.

Le formulaire est aussi un bon exemple pour faire la démonstration de l'import d'une série de jeux de données. Si des données historiques sur la croissance d'un patient existent, celles-ci peuvent être préparées et importées dans le formulaire. Pour cela on commence avec la création d'un fichier au format CSV contenant les points de mesures à importer.

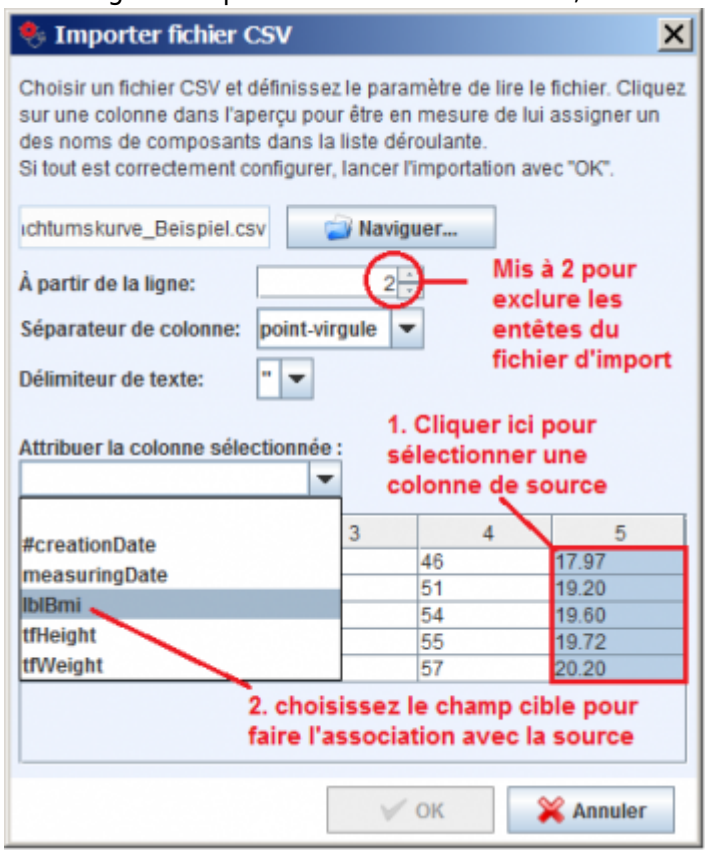
Exemple d'un fichier d'import avec plusieurs points de mesures :

```
date_mesurement; 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”
```

Pour importer ces données on choisit l'option **Importer** au moment de la création d'une instance du formulaire IMC.



Le dialogue d'importation ci-contre s'ouvre, et l'on saisit dans l'ordre les informations suivantes :



- **Naviguer** : Chemin du fichier CSV avec les mesures à importer.
- **À partir de la ligne** : Si le fichier d'import contient, comme dans cet exemple, une ligne d'entête, celle-ci doit être ignoré lors de l'immport, en mettant la ligne de début à 2.
- **Séparateur de colonne** : Le charactère séparant deux colonnes successives dans le fichier d'import. Valeurs possibles: point-virgule, virgule, tabulateur et espace.
- **Délimiteur de texte** : Le caractère entourant un élément de données (simple ou double quotes).
- **Attribuer la colonne sélectionnée** : En dernier il faut associer les colonnes dans le fichier d'import aux éléments de données du formulaire. En ouvrant le fichier d'import son contenu a déjà été lu et affiché en forme de tableau. On sélectionne une colonne de ce tableau en cliquant dans son contenu, ce qui permet de l'associer a un des éléments de données du formulaire présent dans la liste de choix. Touts les éléments de données du formulaire doivent être associés à une colonne du ficher d'import.

Cliquez sur **OK** pour démarrer l'importation. A la fin il faut encore manuellement ajouter un dernier point de mesure pour déclencher le calcul des diagrammes et des tableaux. Le résultat final ressemble à cela :

Éditeur de modèle de formulaire

L'éditeur de modèle de formulaire (onglet **Éditeur**) est activé quand on ouvre un modèle depuis la vue de liste (onglet **Modèle**). Dans la barre de titre trois boutons deviennet actifs, pour l'appel de l'éditeur de script ( ), pour la prévisualisation du formulaire ( ), et pour la sauvegarde du modèle ( ).

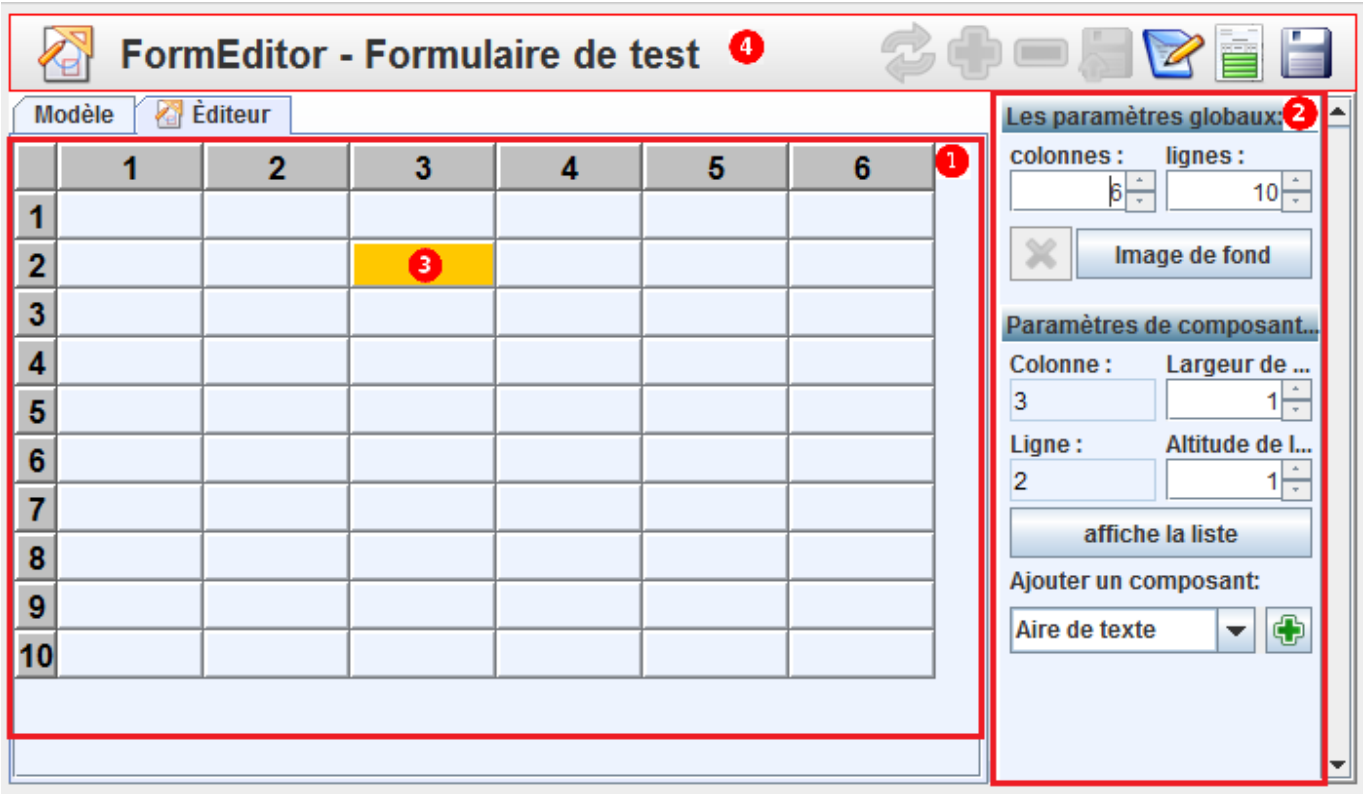


Fig. 3: Éditeur de modèle de formulaire

La partie principale gauche comporte une grille dont chaque cellule peut être sélectionnée par un

click gauche de la souris (voir figure 3 ❶). La grille sert à positionner les composants du formulaire. La cellule sélectionnée est colorée orange (siehe figure 3 ❸). Selon le type de cellule sélectionnée - une en-tête de ligne ou de colonne, un champ vide, un composant - les paramètres de composant changent (voir figure 3 ❷ et figure 4). Ces types de cellules ont des usages différents (voir figure 4).

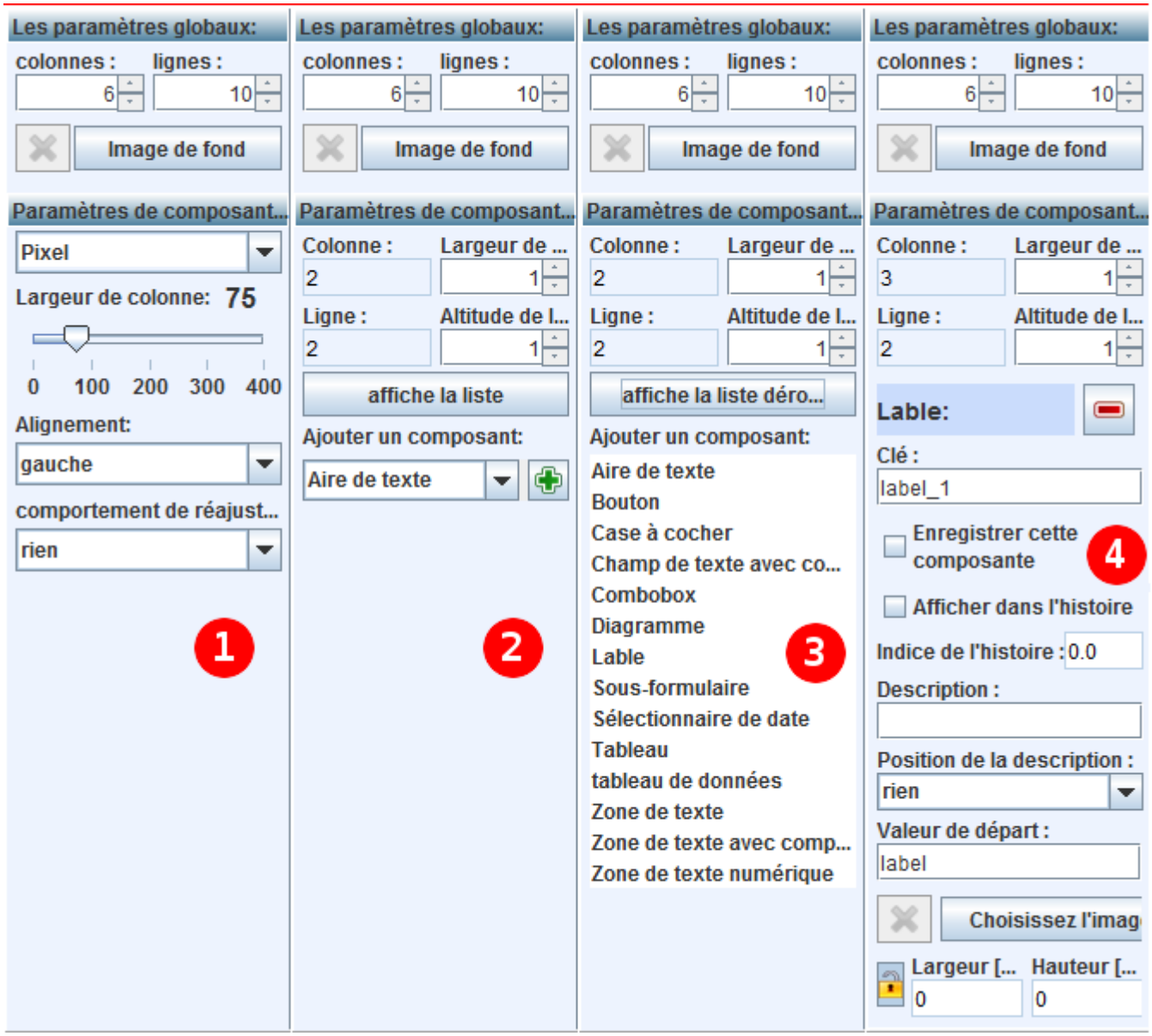


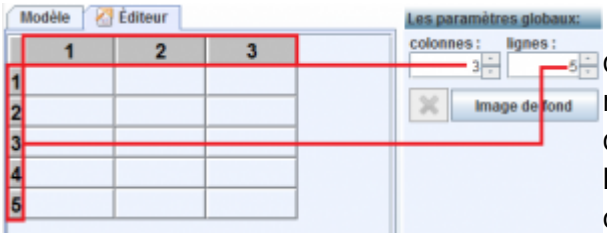
Fig. 4: Les paramètres de composants

Les **en-tête de ligne / colonne** (❶) servent à configurer la hauteur des lignes et la largeur des colonnes, ainsi que le comportement d'allignement et de réajustement des cellules concernées (allignement = gauche, centre, à droit, assumer; réajustement = augmenter ou rien). Elles se trouvent sur le bord à gauche (lignes) et en haut (colonnes) et sont numérotées 1, 2, 3, ... (voir figure 3 ❶). Les en-têtes n'aparaissent pas sur le formulaire final.

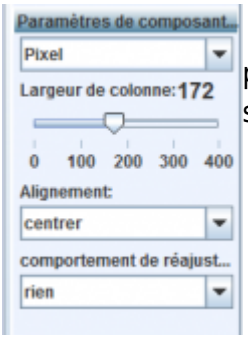
Les **champs vides** (❷ & ❸) n'aparaîtront pas en tant que tels sur le formulaire final, mais ils occupent un certain espace. Pour remplacer un champ vide par un composant, on se sert du menu déroulant ou de la liste **Ajouter un composant**.

Les **champs de composants** (❹) représentent des éléments du formulaire tel que les libellés, des boutons, des champs de saisie de texte etc., bref les éléments visibles du le formulaire final.

Ajuster un modèle de formulaire



La structure de base d'un formulaire est une grille. Elle détermine la dimension matricielle du formulaire, par le nombre de lignes et colonnes. Ce réglage se fait pas les deux compteurs correspondants dans la zone **Paramètres globalaux**. La modification de ces compteurs agmente ou diminue le nombre de colonnes vers la droite resp. de lignes vers le bas. On ne peut plus enlever une colonne ou une ligne si un de ses champs contient déjà un composant, au quel cas il faut l'enlever dabord.

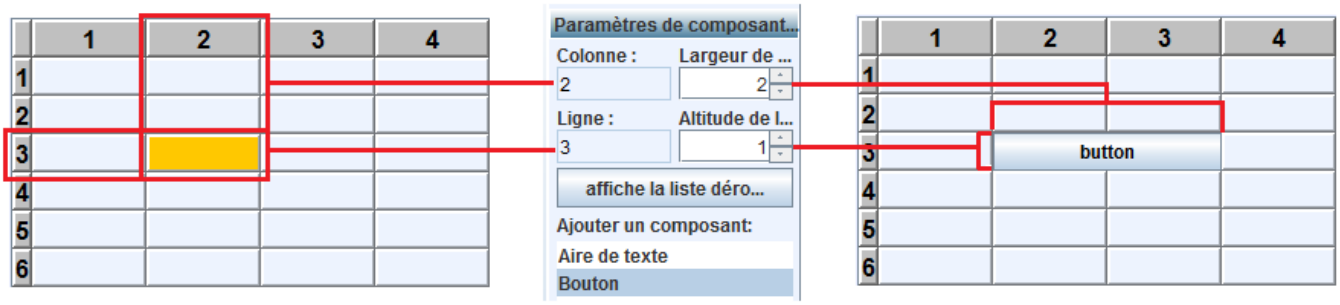


Si un des champs d'en-tête (colonne ou ligne) est sélectionné (voir figure 3) on peut régler plusieurs de ses propriétés dans la zone Paramètres de composant, à savoir:

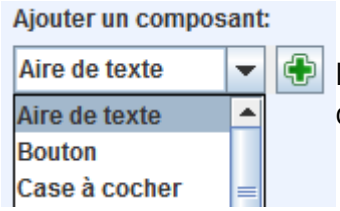
- La **largeur de la colonne** resp. **hauteur de la ligne** dans une unité à choisir (pixel, points, inch, ...).
- L'**alignement** des composants à l'intérieur d'un champ touché par la colonne / ligne (gauche, droite, centrer, assumer)
- Le **comportement de réajustement** qui détermine si les champs touchés peuvet augmenter en taille en cas de besoin.

Placer des composants

Pour placer un composant dans un modèle de formulaire, sélectionnez dabord son emplacement dans un champ vide de la grille, qui devient alors orange. Dans la zone de paramétrage apparaissent alors les options pour créer le nouveau composant (voir ci-dessous).



Colonne et **Ligne** (non-modifiables) rappellent les coordonnées où se situera le composant par rapport à la grille. Par les compteurs **Largeur de la colonne** et **Hauteur de la ligne** on peut choisir combien de colonnes (2) resp. lignes (1) le composant va occuper, par rapport à son emplacement original (2 ,3). Un composant n'est donc pas limité à un *seul* champ, mais peut s'étaler sur *plusieurs*. Il est possible de modifier la taille d'un composant après son insertion.

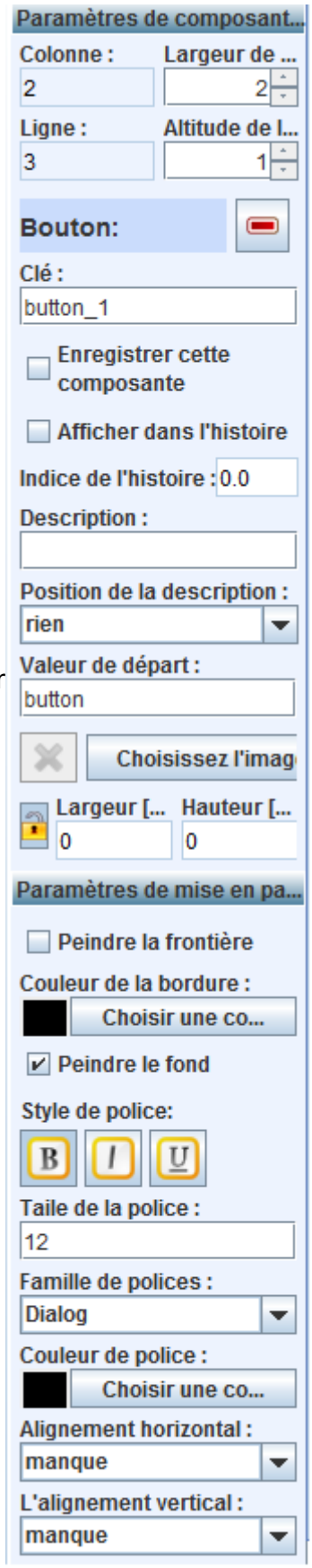


A gauche du bouton pour insérer un nouveau composant se situe une liste déroulante avec tous les choix possibles. La liste des composants disponibles se trouve hier.

Après l'ajout du composant celui-ci est directement sélectionné. En conséquence, la zone de paramétrage à droite change pour afficher les réglages possibles du composant choisi (voir p.ex. figure 4).

Ajuster les composants

Pour adjuster un composant celui-ci doit être sélectionné. Cela se fait en cliquant sur le composant, comme si c'était une celule vide. Un composant sélectionné est normalement affiché en couleur orange. Les réglages se font alors dans la barre de paramétrage droite (figure 4)



Dans la partie haute se trouvent l'emplacement du composant dans la grille (colonne et ligne). A coté se trouvent les nombres de colonnes ou de lignes sur lesquelles s'étend le composant. Ceux-ci peuvent être modifiés pour l'aggrandir ou le rapetisser. La taille minimale est d'une cellule, et le composant ne peut pas chevaucher avec un autre composant ou dépasser la grille.

A l'emplacement du menu de choix de la cellule vide s'affiche le type du composant (Libellé, Bouton, Texte, ...), et just à coté le bouton pour enlever le composant.

Le champ **Clé** défini un nom unique pour le composant par rapport au formulaire. C'est le nom de référence de ce composant tel que utilisé dans l'éditeur de script. Il doit donc être conforme au format d'un nom d'objet Java (composé de lettres, chiffres, et de soulignés (‚_'), sans commencer par un chiffre). Un caractère ne respectant pas ces règles ne sera pas accepté. Un nom double est signalé par un fond rouge du champ de saisie, un nom correct par un fond vert.

La case à cocher **Enregistrer ce composant** détermine si la valeur du composant sera enregistrée dans la base de données,c.à.d. si elle fait partie du jeux de données à ajouter dans le dossier patient. Elle est activée par défaut pour des composants de saisie tel que champ ou zone de text, case à cocher etc. D'autres types de composant comme les tableaux ou les graphes ne permettent pas de

l'activer car leur contenu n'est jamais saisie dans un formulaire. Ils reçoivent les données à afficher par une requête de base de données, à spécifier séparément.

La case à cocher **Afficher dans l'historique** indique si la valeur du composant devrait être apparaître dans la vue historique du patient, au cas où aucun modèle d'impression (en format XSL) n'a été défiré pour la vue historique. L'**Indice dans l'histoirique** sert à déterminer l'ordre de l'affichage de la valeur dans l'historique. Les valeurs apparaîtront dans la forme suivante, en ordre alphanumérique croissante de l'indice :

Nom du formulaire
Description 1: *Valeur 1*
Description 2: *Valeur 2*
Description 3: *Valeur 3*
...
Description n: *Valeur n*

Un texte de “Description” peut être ajouté. Il sera affiché dans la zone d'emplacement du composant, position à choisir dans “Position de la description” (rien, en bas, à droite, en haut, gauche). La description sera également utilisée comme libellé dans l'historique du patient, pourvue que le composant est sensé d'y figurer (“Afficher dans l'historique”). On choisissant “rien”, la description apparaîtra uniquement dans l'historique, mais pas dans le formulaire.

La valeur par défaut d'un composant peut être définie de plusieurs façons, en fonction du type de composant. Elle détermine l'aparance et le contenue du formulaire au moment de sa création.

- Pour un libellé c'est le texte du libellé
- Pour une case à cocher c'est l'état de la case plus un texte
- pour une liste de sélection multiple (combo box) on définit par choix une valeur à stocker et un libellé à afficher (= traduction), après quoi on peut sélectionner la valeur par défaut.

Pour deux types de composant (bouton et libellé) il est possible d'attacher une image et de la dimensionner (bouton “Choisissez l'immage”, champs “Hauter” et “Largeur”).

Pour tout composant affichant un texte il y a les propriétés de mise en page texte habituelle à régler, tel que police, style et taille de caractère, ainsi que couleur, allignement etc. Nous n'allons pas détailler toutes les possibilitées de configuration pour tout type de composant, qui sont, somme tout assez standard et qu'on trouve dans d'autres applications avec formulaires tel que MS-Access. A l'exception du diagramme, qui est relativement complexe et dont la configuration nécessite plus d'explications.

Fix Me! Cette page n'est pas encore traduite en français.
Merci de nous aider à faire la traduction
Les captures des écrans par contre ont déjà été tirées en français.

(Supprimez ce paragraphe une fois la traduction terminée)

Das Diagramm

Avec “Type de diagramme” on choisit parmi deux types “Diagramme du temps” et “Diagramme X Y”.
Dans “Record d'entrée” on peut ajouter / enlever des séries de données à afficher et , qui normalement correspondent à d'autres composants du formulaire.

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.

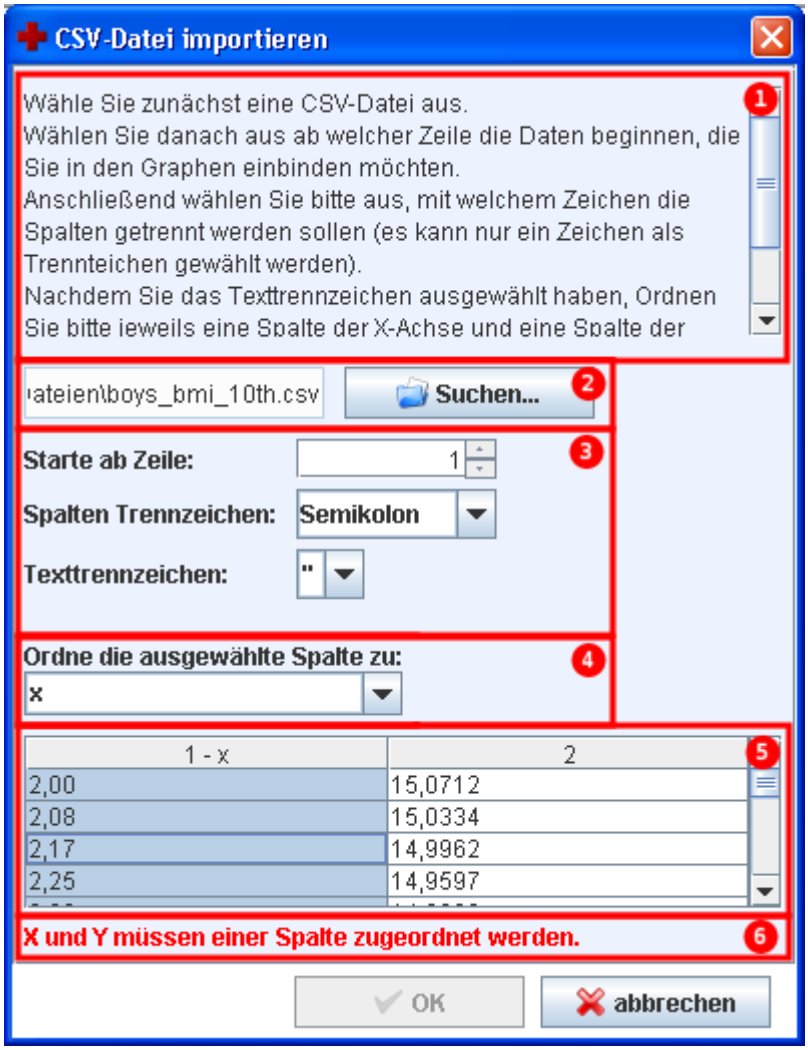


Fig. 5: CSV-Import Dialog

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 figure 5).

Die Beschreibung (siehe figure 5 ❶) erklärt was hier getan werden muss. Mit der Suchen-Schaltfläche kann die gewünschte CSV-Datei ausgewählt werden (siehe figure 5 ❷). 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 figure 5 ❸).

Sind diese Einstellungen vorgenommen wird unten ❺ eine Tabelle angezeigt, die die CSV-Datei widerspiegelt. Klicken Sie auf eine Spalte der Tabelle und wählen Sie in der Auswahlbox darüber (siehe figure 5 ❹) 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 figure 5 ❻). Solange ein Fehler besteht, bleibt die OK-Schaltfläche deaktiviert. Sobald eine Datei geladen und alle Fehler behoben wurden, kann sie angeklickt werden.

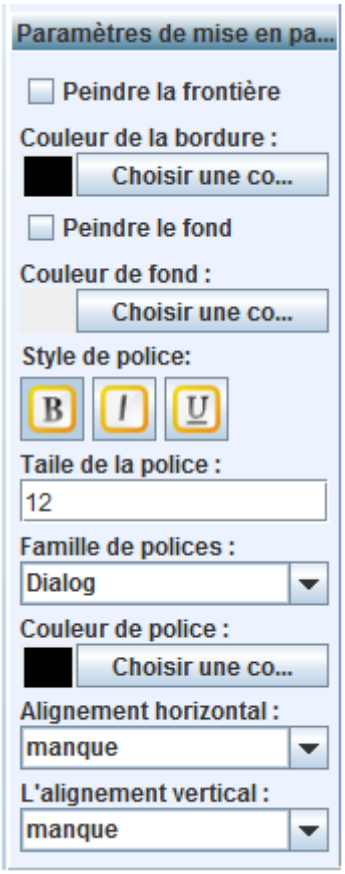


Fig. 6: Eigenschaften

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 figure 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

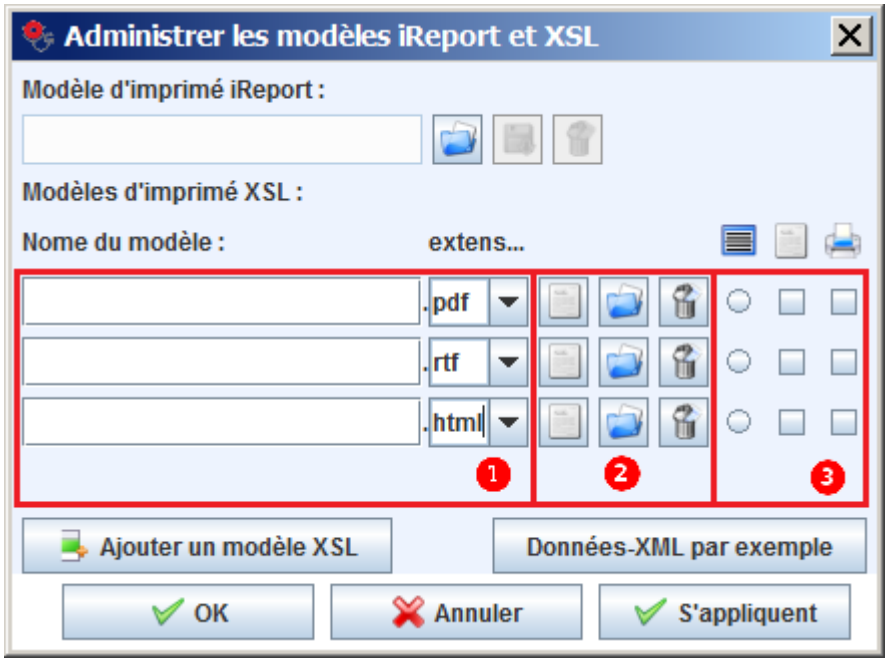


Fig. 7: Druckvorlagen Verwaltung

Durch Klicken auf die Schaltfläche “Vorlagen verwalten” öffnet sich die Formular-Druckvorlagen-Verwaltung (siehe figure 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 (E**X**tensable **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 Programmiertechniken verwendet werden.
Man unterscheidet zwischen XSLT (E**X**tensable **S**tylesheet **L**anguage **T**ransformation) und XSL-FO (E**X**tensable **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 Dateiendung der aus XML mittels XSL erstellten Datei. Anhand der Dateiendung 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 2 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.

Editeur de script

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-Komponen 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 einzeiliges 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

Fig. 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 figure 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.

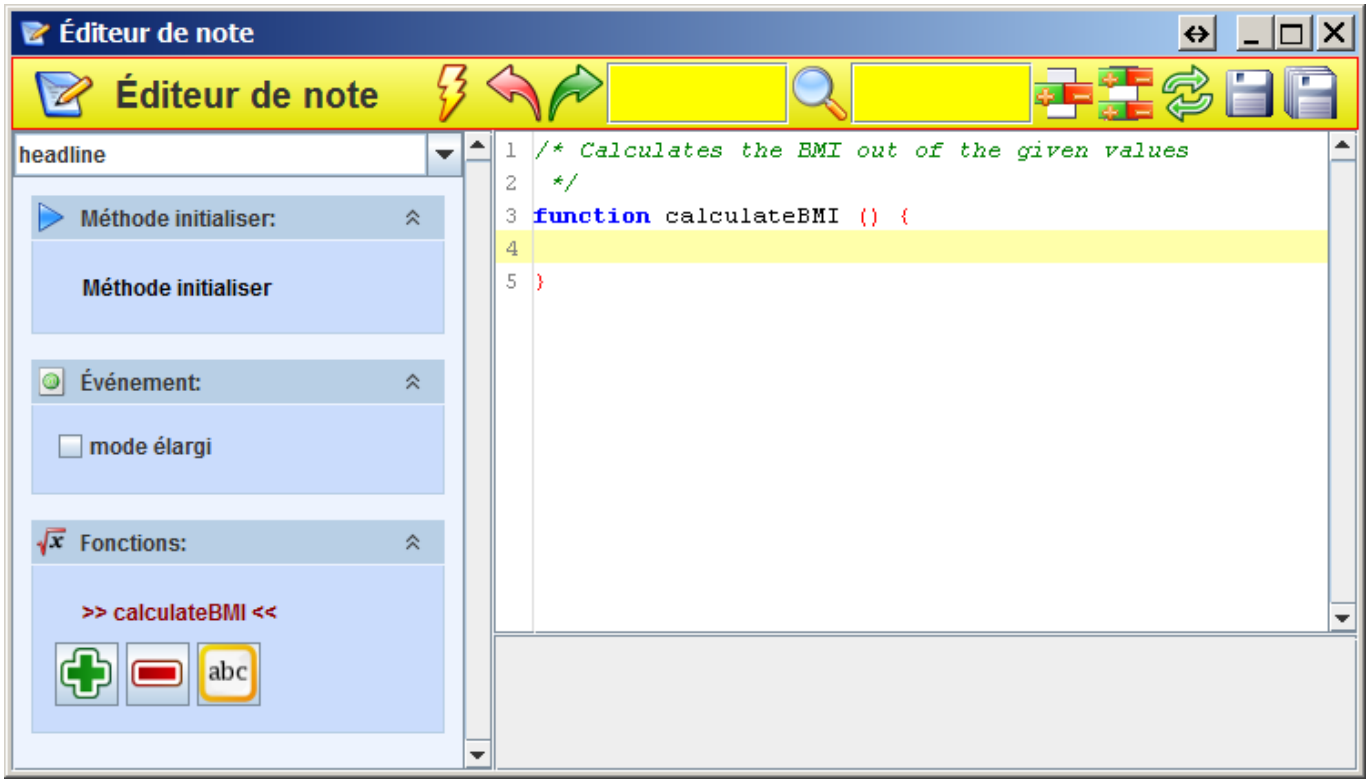


Fig. 9: Skript-Editor

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 figure 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:

BMI = Gewicht [kg] / 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 setzten 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 devide 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, das 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 *claculateBmi* 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 (figure 10) dargestellt. Hier kann es bearbeitet, gespeichert, gelöscht und gedruckt werden.

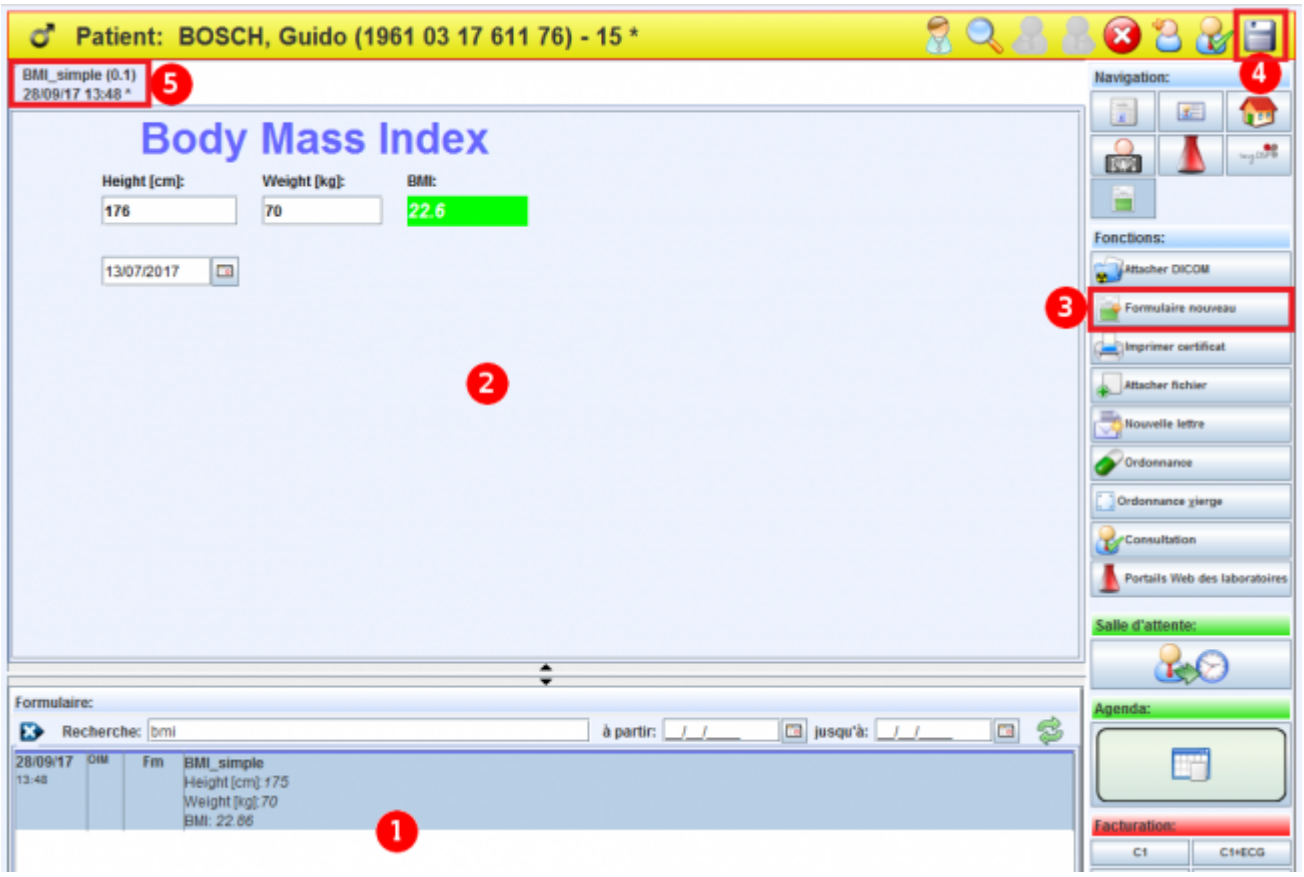


Fig. 10: Formularansicht

Im unteren Bereich der Formularansicht ist eine Historie eingeblendet (siehe figure 10 1), die nur Formulare anzeigt. Das Formular selber wird in der Mitte der Ansicht (siehe figure 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 könenn ein neues Formular erstellen, indem Sie die Schaltfläche “Neues Formular” (siehe figure 10 3) betätigen. Daraufhin öffnet sich ein Dialog wie in figure 11 zu sehen, in dem alle in dieser GECAMed-Installation vorhandenen Formular-Vorlagen angezeigt werden (siehe figure 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.

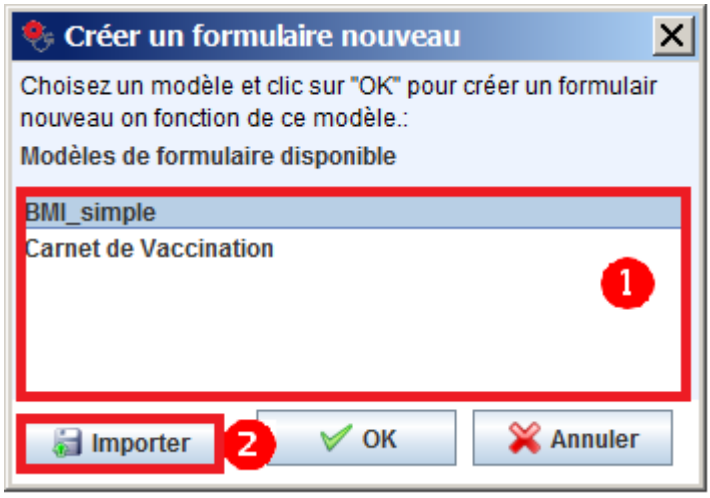


Fig. 11: neues Formular erstellen

Formulare speichern

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

Formulare öffnen und schließen

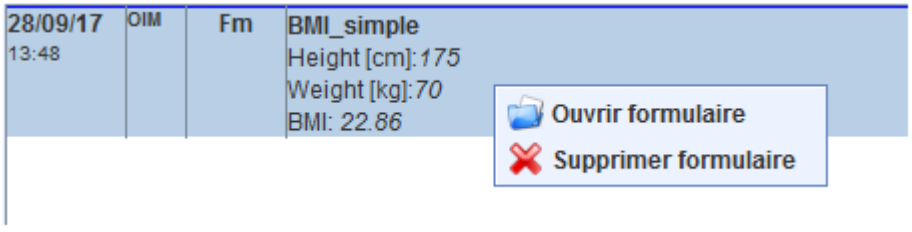


Fig. 12: Formular Rechtsklickmenü in der Historie

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 Formularzeil. Beim Rechtsklick erscheint das Kontextmenü (siehe figure 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 figure 10), und zeigt das Formular dort in einem neuen Karteireiter an.

Fig. 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 bestätig werden (siehe figure 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.

Fig. 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 figure 12) gelöscht werden. Die gleiche Funktion ist bei bereits geöffnetem Formular auch über das Kontextmenü des zugehörigen Karteireiters verfügbar (siehe figure 13).

Bevor ein Formular gelöscht wird, erscheint ein Bestätigungsdialog (siehe figure 15). Wird der Löschvorgang mit “OK” bestätigt, wird das Formular unwiderruflich gelöscht. Wird der Vorgang abgebrochen, bleibt das Formular bestehen.

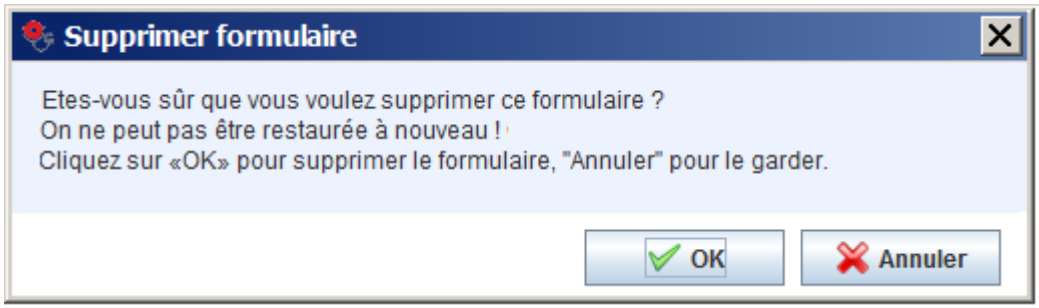


Fig. 15: Bestätigungsdialog: Formular löschen

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 figure 11). Daraufhin öffnet sich der CSV-Datei-Import-Dialog (siehe figure 5). Die Bedienung ist Äquivalent zum Importieren von statischen Graphen.

Ein ausführliches Beispiel für das Importieren vom Datensätzen befidet sich bei der Beschreibung des BMI / IMC Formulares

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 figure 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 abspeichert 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 Dateiendung, 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.