

# Codegenerator van SceneBuilder-FXML naar Haskell

Kai Broux

Master IW elektronica-ict

Luigi Guerriero

Master IW elektronica-ict

## Situering

FunTTop te Diepenbeek heeft als doel het gebruik van functioneel programmeren in het werkveld te verhogen om zo de voordelen op het gebied van efficiëntie en het aantal regels code t.o.v. imperatieve programmeertalen te kunnen valoriseren. Tegelijkertijd stelt FunTTop vast dat het programmeren van een grafische user interface (GUI) nog steeds omslachtig is in Haskell, de academisch ontwikkelde functionele taal. Om die reden beoogt deze masterproef een codegenerator te ontwikkelen die de output van SceneBuilder (FXML) kan omzetten naar Haskell-code.

## Methode

### 1. Demo-applicatie

Enkele GUI-bibliotheken worden aan de hand van een literatuurstudie geselecteerd waarbij voor iedere bibliotheek een voorbeeld-GUI-applicatie wordt gemaakt. Figuur 1 (a) toont de demo-applicatie en figuur 1 (b) toont de boomstructuur van de widgets.

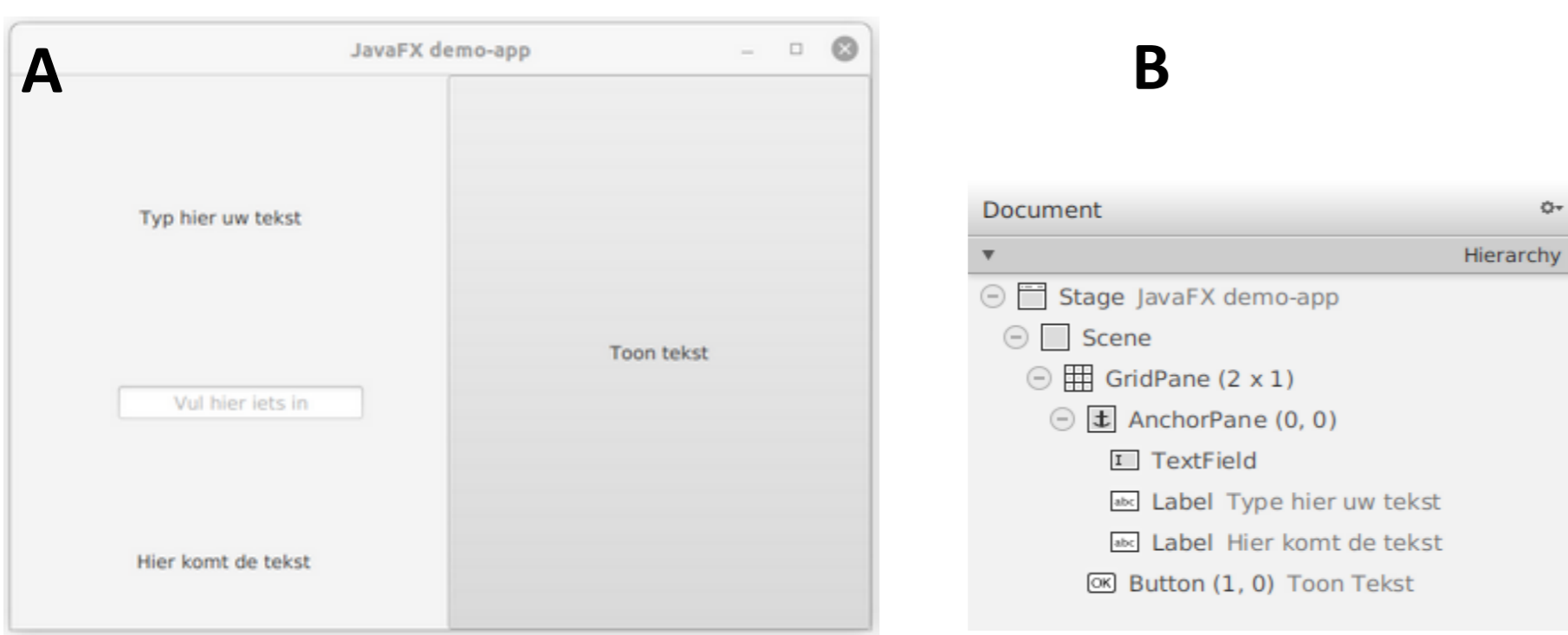


Fig. 1: Demo GUI-applicatie in JavaFX (a) en de hiërarchie van de widgets in SceneBuilder (b)

### 2. Bibliotheekkeuze

Tabel 1: Overzicht voor- en nadelen van de onderzochte bibliotheken

	GTK2Hs	gi-gtk	FLTKHS	Qtah
Voordelen	+ Callbackfuncties zijn eenvoudig te integreren + Zelfgemaakte widgets mogelijk	+ Dezelfde voordelen als GTK2Hs + Grotere subset bindings dan GTK2Hs	+ Korte compileertijden bij successieve builds	+ Grote subset aan widgets beschikbaar
Nadelen	- Niet de volledige GTK+ subset/bindings beschikbaar	- Exacte types bij bepaalde widgets en functies expliciet meegeven	- Meeste widget zelf maken door laag-niveau - Callback-functies moeizaam implementeerbaar	- Geen directe aanhanger mogelijk voor bepaalde JavaFX-widgts - Niet evident om widgets met coördinaten te plaatsen - Veel imports

Tabel 1 toont de voor- en nadelen van de vier meest belovende bibliotheken. FLTKHS en Qtah zijn niet geschikt door de nadelen. GTK2Hs en gi-gtk zijn vergelijkbaar, maar gi-gtk bezit een grotere set bindings. Er werd dus voor gi-gtk gekozen voor het ontwikkelen van de codegenerator.

## Resultaten



Figuur 3 (a) toont dezelfde demo GUI-applicatie van figuur 1 (a) in Haskell, die gemaakt is door de codegenerator. De applicaties in Haskell en JavaFX zien er quasi hetzelfde uit. Bepaalde widgets komen echter niet exact overeen zoals de Button.

Figuur 3 (b) toont een andere applicatie waarin meerdere widgets voorkomen. De codegenerator kan een groot aantal widgets met precisie omzetten.

Tabel 2 toont alle elementen die de codegenerator succesvol kan omzetten

Tabel 2: Subset widgets die de codegenerator kan omzetten

Containers	Controls (1)	Controls (2)
Hbox en VBox	Button	RadioButton
AnchorPane	Label	LinkButton
Layout	Entry	
GridPane	Checkbox	
Notebook	ComboBoxText	

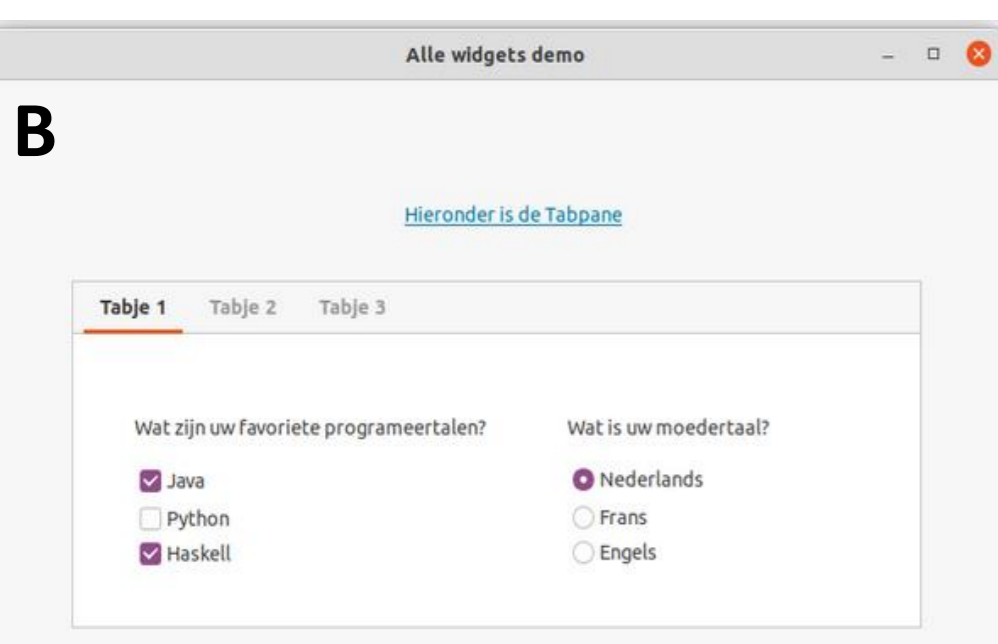
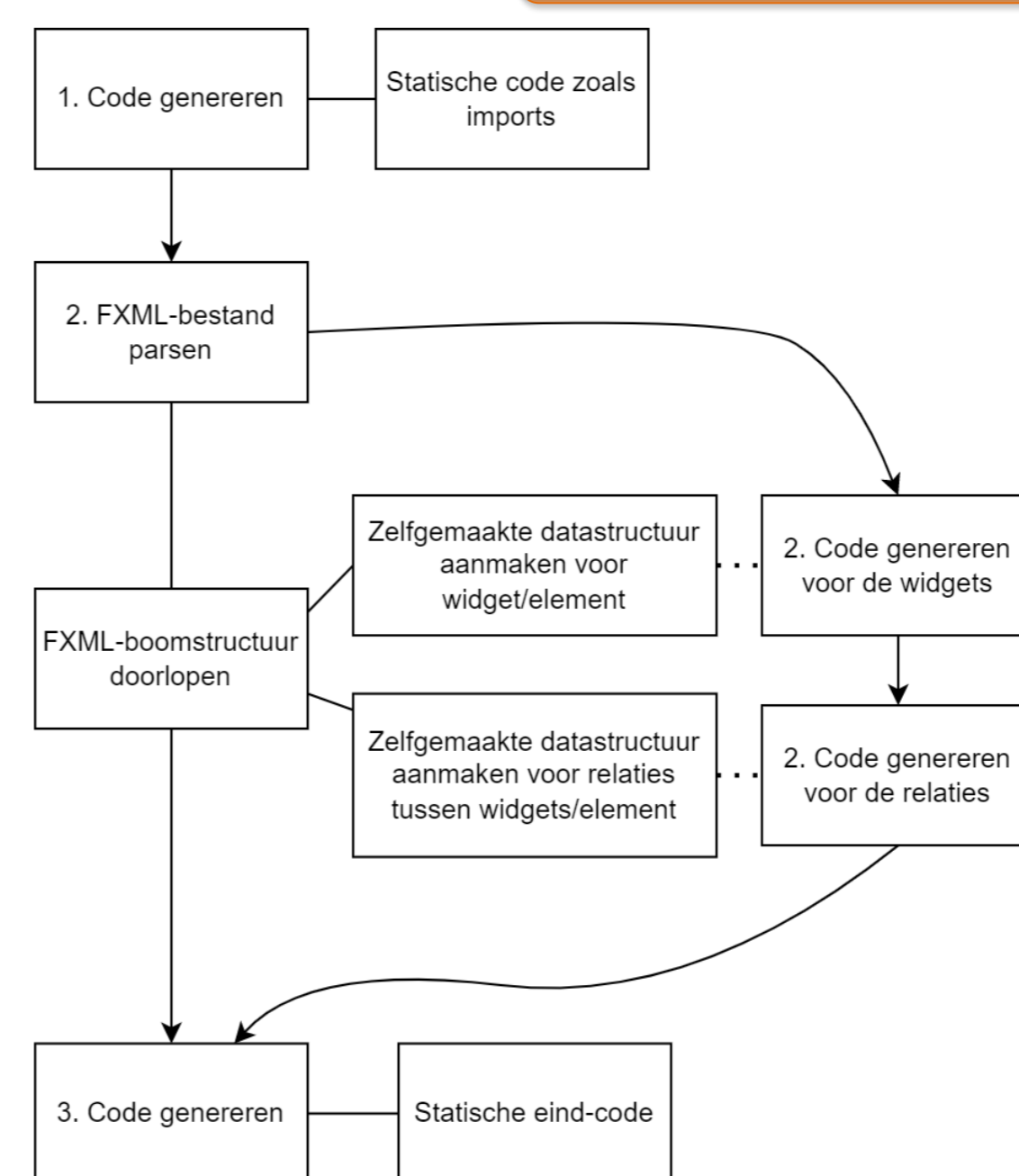


Fig. 3: Demo GUI-applicatie in Haskell met gi-gtk (a) en een GUI-applicatie waarin meerdere widgets zijn getest (b)

### 3. Codegenerator



Figuur 2 toont een schematische weergave van het stappenplan van de codegenerator. Om te beginnen genereert de codegenerator statische code waaronder imports en het begin van de main-functie. Daarna volgt de dump-stap waarbij het FXML bestand doorlopen wordt. Eerst wordt voor elke widget in JavaFX een gelijkaardig widget gemaakt in Haskell met de gi-gtk-bibliotheek. Daarna worden de relaties tussen de widgets onderling aangemaakt. Tot slot bestaat de laatste stap uit het afsluiten van de main-functie.

Fig. 2: Schematische weergave van het stappenplan voor de codegenerator

## Conclusie

De keuze om gi-gtk als bibliotheek te gebruiken brengt een aanzienlijk aantal voordelen met zich mee. Ten eerste kan de bibliotheek op een eenvoudige manier callbackfuncties en zelfgemaakte widgets integreren. Bovendien heeft gi-gtk het voordeel dat het een grote subset aan bindings bezit. Hierdoor is de bibliotheek beter geschikt voor toekomstbestendigheid en daarom is het dus de geschikte bibliotheek. Daarnaast is de codegenerator robuust door de dump-functie. Deze functie herkent de eigenschappen van de widgets en de relaties tussen de widgets correct en zorgt ervoor dat de codegenerator eenvoudig uit te breiden is met meerdere widgets of eventueel andere GUI-bibliotheken. Toch is de codegenerator niet perfect. Er is op dit moment een beperkt aantal widgets beschikbaar die de codegenerator kan omzetten. Echter kan door de robuustheid van de codegenerator, het aantal widgets uitgebreid worden.

Promotoren / Copromotoren / Begeleiders

Prof. dr. Kris Aerts

De heer Wouter Groeneveld