Douglas Cunningham, Petra Hofstedt, Klaus Meer, Ingo Schmitt (Hrsg.); INFORMATIK 2015
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2015

Eine grafische, interaktive Benutzeroberfläche für den Constraint-basierten Produktkonfigurator FdConfig

Sven Löffler[1]

Produktkonfiguration gewinnt bei der ständig wachsenden Nachfrage nach Quantität und Individualität immer mehr an Bedeutung. Der *FdConfig*-Konfigurator des Lehrstuhls Programmiersprachen und Compilerbau der Brandenburgisch Technischen Universität Cottbus-Senftenberg betrachtet und löst Konfigurationsprobleme, die durch erweiterte Feature-Modelle beschrieben werden [SH11]. Erweiterte Feature-Modelle weisen eine Baumstruktur auf. Feature können dabei zu Alternativ-, Oder- bzw. Kardinalitätsgruppen zusammengefasst werden. Die Blätter des Baumes stellen Feature-Attribute dar. Jedes Blatt kann entweder ein Integer-, Enumerations- oder Bool-Attribut sein.

Der Vorteil des *FdConfig*-Konfigurators gegenüber anderen grafischen, interaktiven Konfiguratoren wie z.B. Configit oder dem CAMOS Konfigurator ist, dass *FdConfig* neben Boolschen-Attributen auch große Integer-Attribute auswerten kann.

Der *FdConfig*-Konfigurator ist Backtracking-frei, restorierbar und lässt eine beliebige Reihenfolge der Konfigurationsschritte zu, wobei der Konfigurationsraum nicht eingeschränkt wird. Bei Verwendung von komplexen Modellen ist allerdings eine Reaktionszeit von unter 250ms pro Konfigurationsschritt nicht mehr gewährleistet. Da die graphische Oberfläche nicht nebenläufig implementiert wurde, hat eine lange Berechnung zur Folge, dass der Nutzer in dieser Zeit keinerlei Steuerungsmöglichkeiten besitzt.

Im Zuge dieser Arbeit wurde der *FdConfig*-Konfigurator um eine neue interaktive graphische Oberfläche erweitert. Hauptaugenmerk lag hierbei auf der effektiven und übersichtlichen Darstellung der Integer-Attribute und einer nebenläufigen Implementierung der GUI, damit diese sich bei größeren Berechnungen des Konfigurators reaktiv verhält und nicht einfriert.

Die nebenläufige GUI Für die Darstellung des Modells wurde das Java-Toolkit *Prefuse* [Pr15] verwendet, welches es ermöglicht, Graphstrukturen graphisch darzustellen. Das Zeichnen des Graphen und die Interaktion damit wurde in einen separaten Thread ausgelagert. Dadurch wird es ermöglicht, dass konfigurationsunabhängige GUI-spezifische Einstellungen während der Berechnung eines Konfigurationsschritts weiter manipuliert werden können. Zu diesen Einstellungen, die gleichzeitig als Übersicht- und Detail-Schnittstelle dienen (overview and detail interface), gehören das Verändern der Ansicht bzw. des Ausschnittes, das Expandieren bzw. Kollabieren von Teilbäumen, das zoomen in die Grafik hinein oder aus ihr heraus, das Sichtbarmachen bzw. Verdecken

[1] Brandenburgische Technische Universität Cottbus-Senftenberg, Lehrstuhl Programmiersprachen und Compilerbau, Postfach 101344, 03013 Cottbus, Sven.Loeffler@tu-cottbus.de

von Constraint-Beziehungen und das Auswählen von Integer-Intervallen für das spätere Auswählen bzw. Eliminieren eines Integer-Attributes.

Entwurf geeigneter Visualisierungen Ein weiterer Beitrag der Arbeit war die Konzeption einer geeigneten Visualisierung für die Komponenten des Modells. Die *excludes*- und *requires*-Bedingungen zwischen Features werden durch Kanten bzw. Pfeile direkt im Graphen dargestellt. Ebenso werden Constraints durch einen Knoten, welche durch Pfeile mit den an diesem Constraint beteiligten Attributen verbunden sind, im Graphen dargestellt. Beide genannten Maßnahmen ermöglichen es dem Nutzer, leichter Zusammenhänge zwischen den Attributen abzulesen. Durch die Möglichkeit, einzelne Constraints auszublenden, können überfüllte Ansichten umgangen werden.

Interaktiven Intervallansichten Die Darstellung von Integer-Attributen wurde durch das Implementieren von interaktiven Intervallansichten kompakter gestaltet. Jedes Integer-Attribut hat eine Miniaturansicht, die dem Nutzer standardmäßig angezeigt wird (siehe Abb. 1). Durch Klick auf diese wird dem Nutzer die oberste Intervallebene angezeigt. Diese beinhaltet eine Aufteilung aller Auswahlwerte in maximal zehn Intervalle. Durch Auswahl eines der Intervalle werden dessen Unterintervalle bzw. auf der untersten Ebene dessen Werte angezeigt (siehe Abb. 1).

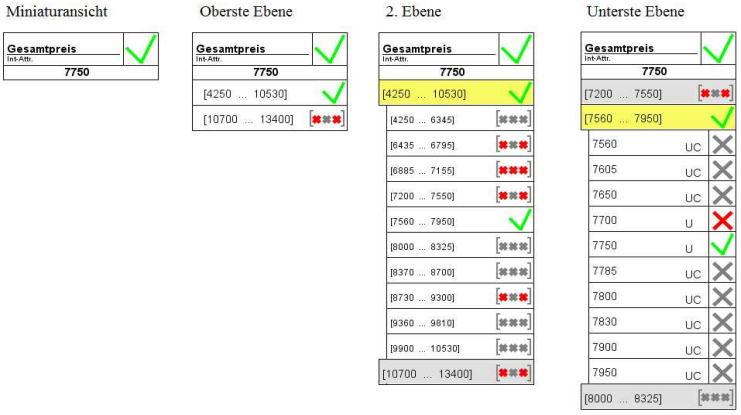

Abb. 1: Verschiedene Darstellungsformen des selben Integer-Attributes

Durch Klicken auf die grau hinterlegten Nachbarintervalle werden diese dargestellt. Ein Klick auf das gelbe Intervall lässt die Ansicht zur nächst höheren Ansicht zurückkehren. Durch ein Häkchen wird verdeutlicht, in welchem Intervall der ausgewählte Wert sich befindet, bzw. welcher der ausgewählte Wert ist. Graue Kreuze und Häkchen markierten durch den Konfigurator zur Konsistenzwahrung automatisch erzeugte Konfigurationsentscheidungen.

Sowohl mit dem Ein- und Ausblenden der verschiedenen Integer-Ansichten als auch mit dem Ein- und Ausblenden aller Knoten und Attribute des Konfigurationsbaumes wird eine Master-Detail-Beziehung unterstützt. Die verschiedenen Integer-Ansichten stellen

gleichzeitig eine sogenannte Fish-eye View dar, welche es dem Nutzer gleichzeitig ermöglicht, Ausschnitte des Konfigurationsmodells sehr detailliert und den Rest des Modells weniger detailliert darzustellen.

Die Abbildung 2 zeigt die GUI unter Verwendung eines kleinen Modells und nach Ablauf einiger Konfigurationsschritte.

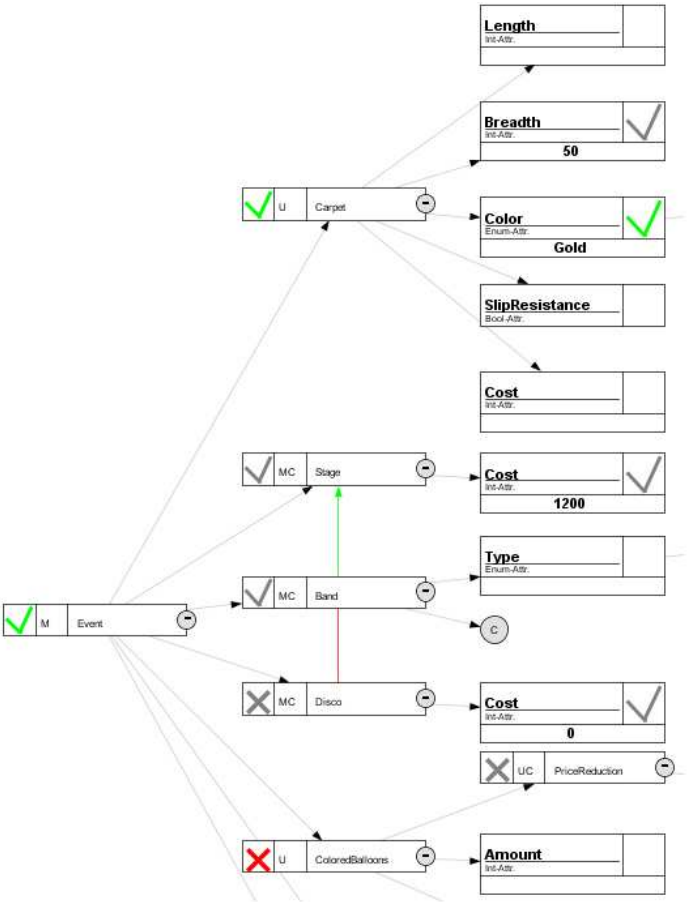

Abb. 2: Vollständige Ansicht eines Modells des *FdConfig*-Konfigurators

Die neu entstandene Visualisierung berücksichtigt bekannte HCI-Prinzipien (Human Computer Interaction) wie zum Beispiel Overview and Detail, zooming und Focus and Context. Dadurch kann die neu gestaltete Visualisierung auch Attribute mit großen Wertebereichen übersichtlich und kompakt darstellen und es dem Nutzer ermöglichen zielstrebig und ohne großen Suchaufwand bestimmte Attributwerte zu selektieren oder zu eliminieren.

[SH11] Schneeweiss, D.; Hofstedt, P. FdConfig*: A constraint-based interactive product configurator*. In: *Proceedings of 19th Conference on Applications of Declarative Programming and Knowledge Management, Wien, 2011*.

[Pr15] *prefuse – Information Visualization Toolkit,* August 2011. http://prefuse.org/.