

Modellbasierte Generierung von Software für mobile Datenerfassungen

Mathias Kühn, Peter Forbrig

Institut für Informatik, Universität Rostock

Zusammenfassung

Die Verwendung modellbasierter Sprachen zum Zwecke der kontextabhängigen Generierung von Software für verschiedene Plattformen ist ein geeigneter Ansatz im Rahmen der Erfassung von Daten mit mobilen Geräten. Die Sprachen UsiXML und SCXML stellen eine solide Basis für die Ableitung von Instanzen für Struktur und Verhalten einer Software für interaktive Formulare dar. Das in dem Beitrag vorgestellte Konzept verwendet eine serviceorientierte Architektur zur Förderung der Plattformunabhängigkeit. Werkzeugunterstützung ist im Rahmen mobiler Datenerfassung sowohl für erfahrene als auch unerfahrene Anwender möglich.

1 Einleitung

Für empirische Untersuchungen ist es notwendig Daten von Probanden zu sammeln, die für die zugrundeliegende Forschung von Interesse sind. Dazu können Befragungen im klassischen Sinne sowohl nach als auch während einer Testdurchführung gemacht werden. In diesem Kontext können Fragebögen verwendet werden, die als Formular aufbereitet sind und dem Befragten oder dem Befragenden zur Verfügung stehen. Die Verbreitung mobiler Geräte, wie Smartphones, Tablets und Laptops, kann für die Durchführung von Interesse sein. Interaktive Formulare auf mobilen Geräten können genutzt werden, um Probanden zu befragen oder diese durch den eigentlichen Testdurchlauf zu leiten. Des Weiteren können mit den Geräten selbst relevante Daten gesammelt und für die Auswertung herangezogen werden. Sowohl Befrager als auch Befragte können für das Erstellen und Ausfüllen durch Werkzeuge unterstützt werden. Die Generierung und Nutzung von Formularen für genannte Zwecke wird im Folgenden genauer betrachtet.

2 Hintergrund

Einige modellbasierte Sprachen (Paterno et al. 2009, Limbourg et al. 2004) erlauben eine Spezifikation nach dem Cameleon Reference Framework (Calvary et al. 2003). Dieses Rahmenwerk berücksichtigt explizit einen Kontext¹ für die Transformation von User Interfaces [UIs]. Die Transformationen können Konkretisierungen beginnend von abstrakten Spezifikationen [AUI→FUI], Abstraktionen beginnend von konkreten Spezifikationen [FUI→AUI] oder Translationen [CUI→CUI'] auf gemeinsamen Ebenen sein. Um eine kontextabhängige Generierung von UIs auf einem interaktiven Gerät zu unterstützen, bieten sich abstrakte Spezifikationen an. Diese erlauben zwar eine Ableitung der Struktur einer Oberfläche, aber nicht eine Reihenfolge ihrer Präsentation zur Laufzeit (Schlungbaum & Elwert 1996). In diesem Bereich hat sich u.a. die Verwendung von Statecharts (Harel 1987) zur Beschreibung eines Dialogs (Kistner & Nuernberger 2014) bewährt. Zustände und Transitionen legen fest wann und unter welchen Bedingungen bestimmte Teile einer Benutzungsschnittstelle für den Nutzer zugänglich sind.

Die Benutzung von Webservices² erlaubt die Verwendung von Software auf verschiedenen Plattformen in einem Netzwerk und führt zur Instanziierung gemeinsamer Domänenmodelle innerhalb von Client-Server-Architekturen. Für eine Nutzung zur Laufzeit bietet sich das Konzept der De-/Serialisierung³ von Domäneobjekten an, die an eine Nutzerschnittstelle gebunden sind. Somit lassen sich Instanzen zwischen Clients und Servern übertragen und Dienste nutzen, die den Applikationskern einer Software bilden. Die Herausforderung für eine möglichst flexible Nutzung beliebiger Dienste auf Seite der Clients ist die Transformation [Castings] der Werte zwischen Typen die von Oberfläche geliefert und an das Domäneobjekt übergeben bzw. der Werte die vom Domäneobjekt geliefert und auf der Oberfläche präsentiert werden. Ebenso müssen nutzerbasierte Ereignisse [UI Events] der Benutzungsschnittstelle betrachtet werden, welche für Interaktionen innerhalb der Anwendung relevant sind.

2.1 Eigener Ansatz

Sämtliche Formulare werden vorab spezifiziert und als AUI-Modell in UsiXML abgelegt. Zusätzlich wird der Dialog in Zuständen, als Eigenschaften der Elemente des AUI-Modells, und Transitionen, als Ereignisse bestimmter Elemente des AUI-Modells, in SCXML abgelegt. Bedingte Transitionen hängen von Zuständen des Domänenmodells ab. Das Domänenmodell der Anwendung wird in XML Schema Definition [XSD] abgelegt. Alle Spezifikationen müssen auf einen Server geladen werden und sind dort über einen Webservice abrufbar. Zur Laufzeit werden die Spezifikationen von Clients geladen, um kontextabhängige Instanzen plattformspezifisch zu generieren. Im Anschluss daran wird ein initiales Domäneobjekt an den Client gesendet, sowie in den Startzustand des Dialogmodells gewechselt.

¹ Context of Use: <http://www.w3.org/2005/Incubator/model-based-ui/XGR-mbui>

² Java API for XML Web Services: <http://jax-ws.java.net>

³ Java Architecture for XML Binding: <http://jaxb.java.net>

Bei Benutzung werden nutzerbasierte Ereignisse der UI sowie manipulierte Domäneobjekte an den Webservice gesendet und dort anwendungsspezifisch interpretiert. Anschließend wird das interpretierte Domäneobjekt zusammen mit dem Ereignis, welches die Transition des Dialogs ist, zurück an den Client gesendet. Während der Anwendung wird dieser Zyklus durchlaufen bis der Dialog in den Endzustand wechselt oder der Client länger nicht im Netzwerk erreichbar ist.

3 Mobile Datenerfassung

Interaktive Formulare für das Sammeln von Daten werden beispielsweise in der Ausbildung (Kühn et al. 2013) oder für die Durchführung von Befragungen (Dillman et al. 2009) genutzt. Werden häufig unterschiedliche Formulare für verschiedene Zwecke eingesetzt, so wäre es wünschenswert Änderungen zum einen ohne Aufwand durchführen und diese zum anderen unmittelbar auf Seite der Befragten anwenden zu können.

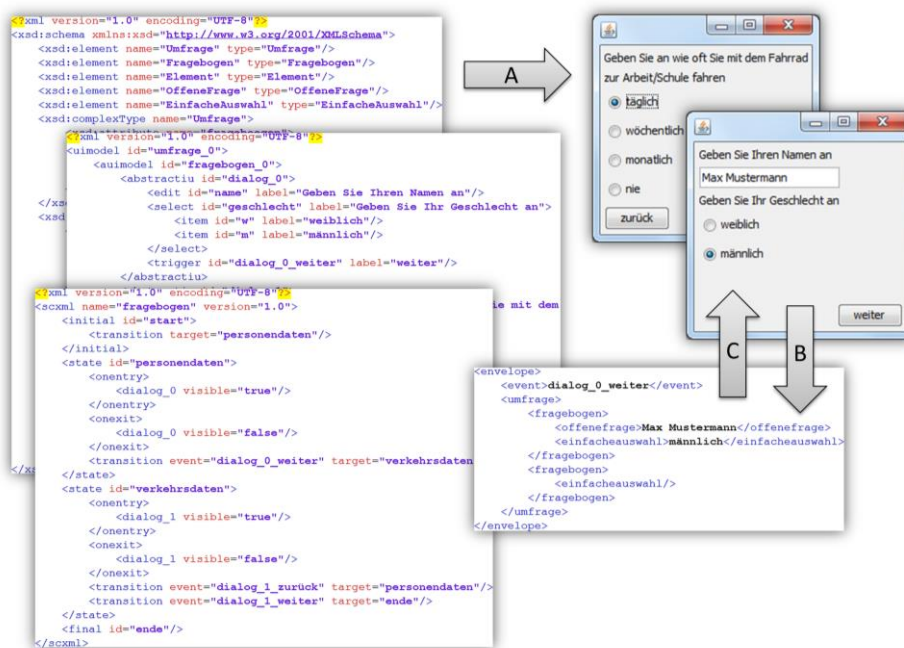


Abbildung 1: Schematische Darstellung am Beispiel einer Verkehrsbefragung

Abbildung 1 zeigt schematisch eine Anwendung des Konzepts am Beispiel einer Befragung zur Benutzung von Verkehrsmitteln. Das Domänenmodell betrachtet Umfragen als Komposition von Fragebögen. Ein Fragebogen ist eine Komposition von offenen Fragen oder Einfachauswahl. Im Beispiel wird nach dem Namen und dem Geschlecht des Befragten gefragt. Dazu wurden vorab in Schritt A alle Spezifikationen vom Client geladen und

kontextabhängig [Nutzer mit Desktop PC] generiert. Die UI [siehe Abbildung 1] ermöglicht dem Befragten die Eingabe seiner Daten. In Schritt B wird nach Klick auf den *weiter*-Button das serialisierte Domäneobjekt zusammen mit dem Ereignis *dialog_0weiter* an den Webservice geschickt. Dieser antwortet in Schritt C unverändert mit den gleichen Daten, da die eingegeben Daten valide sind und zum nächsten Dialog gewechselt werden kann. Der Zyklus [Schritt B und C] läuft weiter bis die Umfrage abgeschlossen oder abgebrochen ist.

4 Ausblick

In diesem Beitrag wurde ein Konzept für die modellbasierte Generierung von Software zum Zwecke der Datenerfassung vorgestellt. Für die Anwendung wurden die Sprachen UsiXML zusammen mit SCXML, sowie die Verwendung von Webservices als funktionaler Kern vorgestellt. Weiterführend wäre eine Anwendung des Konzepts für andere Domänen, neben der interaktiver Formulare, interessant. Die Generierung von Benutzungsschnittstellen basierend auf abstrakten Oberflächenbeschreibungen erlaubt keine exakte Definition von konkret zu interpretierenden Ereignissen der UI, sondern betrachtet ausschließlich solche, die für das zugrundeliegende Framework einen Standard repräsentieren [siehe *ActionListener* in Java]. Eine Erweiterung der Werkzeugunterstützung wäre, dass konkrete Beschreibungen der UI und damit auch detailliertere Dialoge spezifiziert werden können, diese aber die Kontextunabhängigkeit einschränken. In Zukunft wird dazu untersucht Modellinstanzen erst clientseitig zu abstrahieren, um sie anschließend wieder kontextabhängig zu konkretisieren.

Literaturverzeichnis

- Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L. & Vanderdonckt, J. (2003). *A unifying reference framework for multi-target user interfaces*. Interacting with computers, 15, 289-308.
- Dillman, D. A., Smyth, J. D. & Christian, L.M. (2009). *Internet, Mail and Mixed-Mode Surveys: The Tailored Design Method*, 3rd ed. John Wiley & Sons.
- Harel, D. (1987). *A Visual Formalism for Complex Systems*. Science of Computer Programming 8, 3, 231-274.
- Kistner, G. & Nuernberger, C. (2014). *Developing User Interfaces using SCXML Statecharts*. EICS 14 Workshop. 5-11.
- Kühn, M., Forbrig, P. & Dittmar, A. (2013). *Werkzeugunterstützung für die mobile Dokumentation in praktischen Phasen der akademischen Ausbildung*. DeLFI 2013 Workshop. 65-66.
- Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., Florins, M. & Trevisan, D. (2004). *UsiXML: A User Interface Description Language for Context-Sensitive User Interfaces*. AVI 2004 Workshop. 55-62.
- Paternò, F., Santoro, C. & Spano, L. D. (2009). *MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments*. Transactions on Computer-Human Interaction, 16, 4.
- Schlunbaum, E. & Elwert, T. (1996). *Dialogue graphs: a formal and visual specification technique for dialogue modelling*. BCS-FACS. 13-13.