

Von der Idee zum Anforderungsmodell ohne Medienbruch

Dustin Wüest, Norbert Seyff, Martin Glinz
Universität Zürich, Requirements Engineering Research Group,
Binzmühlestrasse 14, 8050 Zürich, Schweiz
Email: wueest@ifi.uzh.ch, seyff@ifi.uzh.ch, glinz@ifi.uzh.ch

Motivation und Problem

In frühen Projektphasen sind Kreativität sowie das Skizzieren und Austauschen von Ideen oft wichtige Bestandteile der Anforderungsermittlung [2]. Modellierungswerkzeuge wie z.B. UML-Tools werden in diesen Phasen oft nicht verwendet, weil sie – bedingt durch die Formalität der jeweils unterstützten Notation – nicht flexibel genug sind, um verschiedene Sachverhalte schnell und simpel darstellen zu können. Des Weiteren werden diese Notationen oft nicht von allen Interesseneignern verstanden. Oftmals werden freie Skizzen auf Papier oder Whiteboards zum Dokumentieren und initialen Modellieren von Anforderungen verwendet [3]. Dies erleichtert allen Projektbeteiligten den Zugang und unterstützt auch die Kreativität. Solche Diagramme können auf beliebigen Abstraktionsebenen und unter Verwendung verschiedener Notationen gezeichnet werden.

Für die Spezifikation und das Management von Anforderungen müssen diese Diagramme dann aber in einer maschinell verarbeitbaren Form vorliegen. Das heißt, dass zu einem späteren Zeitpunkt die auf Papier oder Whiteboard skizzierten Modelle von Grund auf neu erstellt werden müssen. Dazu benutzt der Anforderungsanalytiker meist ein (halb-)formales Modellierungstool.

Zwischen den initial skizzierten Diagrammen und den mit Hilfe von Werkzeugen erstellten Modellen entsteht somit ein Medienbruch. Ebenso ist das Erstellen der neuen Modelle zeitintensiv und fehleranfällig. Der Kontext, in dem die Zeichnungen entstanden sind, ist inzwischen möglicherweise nicht mehr ersichtlich, wodurch diese verschieden interpretiert und falsch übersetzt werden können.

Flexibles Modellieren im RE

Unsere aktuelle Forschung befasst sich mit einem Ansatz, welcher eine softwareunterstützte Alternative zu Papier und Whiteboards bietet [1]. Unter Verwendung dieses Ansatzes können diagrammähnliche Anforderungsskizzen halbautomatisch in halbformale Modelle transformiert werden, indem sie schrittweise verfeinert und formalisiert werden. Das erlaubt freies Zeichnen, und bei Bedarf kann den gezeichneten Symbolen jederzeit eine Bedeutung durch das Definieren von Syntax und/oder Semantik zugeordnet werden. So werden aus Skizzen Modelle; ein Medienbruch entfällt. Auf diese Weise können außerdem auch domänenspezifische Modellierungssprachen erstellt werden. Im Gegensatz zu anderen Lösungen (z.B. [4]) muss hier die Sprache

also nicht bereits definiert sein, bevor man mit der eigentlichen Modellierung beginnen kann – was ein sehr flexibles Modellieren ermöglicht.

Unter der Annahme, dass die in frühen Projektphasen verwendeten Notationen nicht zu komplex ausfallen, erlaubt dieses Vorgehen das Erstellen von Modellierungssprachen auch durch Anforderungsanalytiker, welche keine Experten in Bezug auf Metamodellierung sind.

Der Erfolg unseres Ansatzes hängt neben methodischen Herausforderungen auch von der Gebrauchstauglichkeit eines möglichen Werkzeugs ab. Deshalb haben wir einen Prototypen, *FlexiSketch*, entwickelt. Flexibles Modellieren bedeutet für uns, dem Benutzer nicht nur freie Wahl bezüglich Notation zu lassen, sondern auch, wo und wann er/sie Anforderungsdiagramme zeichnen will. Daher haben wir *FlexiSketch* für auf Android 3.0+ basierende Tablet-Computer entwickelt (*FlexiSketch* ist auf Google Play verfügbar). Der Prototyp erlaubt dem Benutzer das Ausführen von drei wesentlichen Aktivitäten [1]: das *Skizzieren* beziehungsweise Modellieren von Anforderungen, das Klassifizieren und Wiederverwenden von Symbolen als erster Schritt der *Metamodellierung*, und das automatische Erkennen sowie Ersetzen von ähnlichen Symbolen mittels *Sketch Recognition*. Nachfolgend beschreiben wir *FlexiSketch* kurz.

Skizzieren/Modellieren. Das GUI ist minimalistisch gehalten und beschränkt sich auf das Wesentliche. Wenn unser Prototyp gestartet wird, zeigt er eine leere, weiße Fläche an, die zu freiem Zeichnen animieren soll. Zusätzliche Funktionalitäten verstecken sich in Kontextmenüs und ausklappbaren Registern (Abbildung 1). Jedes Mal, wenn der Benutzer beim Zeichnen eine Pause einlegt, verwandelt *FlexiSketch* die zuletzt gezeichneten Striche in ein separates Symbol (oder in eine gerade Verbindungslinie, falls ein Strich zwischen zwei Symbolen gemalt wurde). Symbole sind eigenständige Objekte, die selektiert und manipuliert (z.B. verschoben, skaliert) werden können.

Metamodellierung. Symbole können über ihr Kontextmenü klassifiziert werden. Alle klassifizierten Symbole werden in ein Register auf der Seite des Bildschirms kopiert. Diese Symbole bilden dann sozusagen das Grundvokabular der Modellierungssprache. Per „Drag & Drop“ können Kopien dieser Symbole auf die Zeichenfläche gezogen werden.

Sketch Recognition. Die vom Benutzer klassifizierten Symbole stellen die Grundlage für die automatische Erkennung weiterer Symbole dar. Zeichnet der Benut-

zer ein ähnliches Symbol wie es schon im Register vorhanden ist, so wird dies von *FlexiSketch* erkannt und der Benutzer wird gefragt, ob die zwei Symbole dieselbe Bedeutung haben bzw. ob sie identisch sind. Der Benutzer kann die Frage beantworten, muss aber nicht. Falls er es tut, lernt der Erkennungsalgorithmus dazu und wird so über die Zeit treffsicherer.

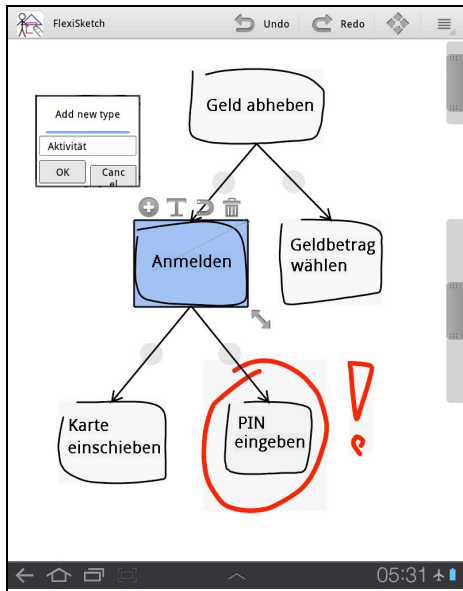


Abbildung 1: *FlexiSketch* unterstützt freies Zeichnen und das Formalisieren von skizzierten Diagrammen.

Resultate und Ausblick

In ersten Studien haben wir 9 Anforderungsanalytiker aus der Industrie und 8 Informatikstudenten um Feedback zu *FlexiSketch* gebeten [1]. Fast alle Teilnehmer gaben an, dass sie gerne ein Werkzeug wie *FlexiSketch* einsetzen würden. Kritische Rückmeldungen betrafen in erster Linie die begrenzten Fähigkeiten des aktuellen Prototyps, während unser genereller Ansatz positiv und mit viel Interesse aufgenommen wurde.

Um die Nützlichkeit unseres Ansatzes zu evaluieren, baten wir die Studienteilnehmer als Erstes, Use Case Diagramme zu zeichnen und dabei alle verwendeten Symbole zu klassifizieren. Diese Aufgabe wurde von allen Teilnehmern gemeistert. Sie schätzten die Einfachheit und Flexibilität des Prototyps. Die Mehrheit meinte, dass sie lieber auf Formalisierungsmöglichkeiten als auf das freie Zeichnen verzichten würden, wenn sie eine Wahl treffen müssten.

Ebenso wurden die Studienteilnehmer gefragt, welche Arten von Diagrammen sie zeichnen, und wir erhielten darauf viele verschiedene Nennungen. Die meisten genannten Diagrammtypen sind aus Knoten und Kanten aufgebaut. Des Weiteren wurde sehr oft erwähnt, dass sich die Skizzen an bekannte Diagrammarten (z.B. aus der UML) anlehnen, aber meist die Notationen bewusst vereinfacht werden und/oder von den Standards abweichen. Manche Teilnehmer zeichneten die erwähnten Diagramme dann auch mit Hilfe von *FlexiSketch*.

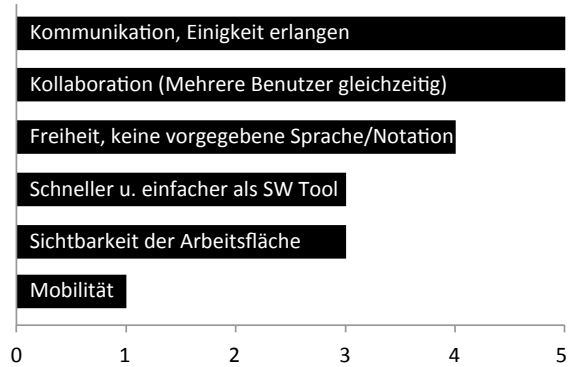


Abbildung 2: Gründe, aus denen Anforderungsanalytiker Papier und Bleistift sowie Whiteboards benutzen.

Wir fragten die Studienteilnehmer ebenfalls, wieso sie Papier und Bleistift sowie Whiteboards verwenden (Abbildung 2). Zwei der am häufigsten genannten Gründe, die Sichtbarkeit der Arbeitsfläche und die Kollaboration mehrerer Benutzer, können durch einen Tablet-Computer nur beschränkt unterstützt werden. Würde man stattdessen eine vergleichbare Applikation für ein elektronisches Whiteboard verwenden, müsste man aber auf die Mobilität und die größere Verbreitung von mobilen Geräten verzichten.

Ein Ziel für nächste Forschungsschritte ist, *FlexiSketch* ebenfalls für elektronische Whiteboards verfügbar zu machen, wobei ein Augenmerk auf die Zusammenarbeit mehrerer Benutzer gerichtet werden muss. Zweitens befassen wir uns damit, wie wir eine benutzerfreundliche Metamodellierung verwirklichen können, welche über das Klassifizieren von Symbolen hinaus geht und das Definieren von Assoziationen und Einschränkungen erlaubt. Drittens soll unser Ansatz in der Praxis getestet werden, um dessen Gebrauchstauglichkeit und Nützlichkeit genauer evaluieren zu können.

Referenzen

[1] Wüest D, Seyff N, Glinz M: „FlexiSketch: a mobile sketching tool for software modeling“, wird veröffentlicht in *Mobile Computing, Applications, and Services, Lecture Notes of the Institute for Computer Science, Social Informatics and Telecommunications Engineering, Springer Berlin / Heidelberg, 2012.*

[2] Cherubini M, Venolia G, DeLine R, Ko A. J. „Let’s go to the whiteboard: how and why software developers use drawings“, in *Proc. SIGCHI Conf. on Human Factors in Computing Systems, 2007, pp. 557–566.*

[3] Mangano N, Baker A, Dempsey M, Navarro E, van der Hoek A, „Software design sketching with Calico“, in *Proc. IEEE/ACM Int. Conf. on ASE, 2010, pp. 23–32.*

[4] Kelly S, Lyytinen K, Rossi M, „Metaedit+: A fully configurable multi-user and multi-tool CASE and CAME environment“, in *Advanced Information Systems Engineering, LNCS, P. Constantopoulos, J. Mylopoulos, and Y. Vassiliou, Eds. Springer Berlin / Heidelberg, 1996, vol. 1080, pp. 1–21.*