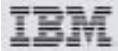


Inkonsistenzen in der Modellierung



Software Group



Claudio Grolimund: „Inkonsistenzen in Modellen sollten schnell erkannt und behoben werden. Im Rahmen der Rational Software Developer Platform 2009 stellte Prof. Dr. Egyed ein Verfahren vor, mit dem Modelle nach einer Veränderung schnell, genau und automatisch auf Inkonsistenzen geprüft werden können. Nach seiner Habilitation arbeitete Herr Egyed sieben Jahre lang in der Software-Entwicklung im Rahmen von industrienahen Forschungsprojekten für die Techknowledge Corporation in den USA. Später wechselte er zum University College in London. Heute ist er Leiter des Instituts für System Engineering & Automation an der Johannes Kepler Universität in Linz und beschäftigt sich momentan hauptsächlich mit modellbasierter Softwareentwicklung. In einem kurzen Gespräch stellte uns Professor Egyed seinen Ansatz vor. Er erklärte, warum Inkonsistenzen in der modellbasierten Softwareentwicklung auch nach jahrelanger Erfahrung immer noch ein grosses Problem für Entwickler darstellen.“

Alexander Egyed: „Das Besondere an Inkonsistenzen ist, dass sie im Grunde genommen kein wirkliches Problem sind - sie haben vielmehr Vorteile und Nachteile. Inkonsistenzen treten in einer Software ganz einfach deswegen auf, weil versucht wird, verschiedene Gesichtspunkte unterschiedlich zu modellieren. Als Analogie dazu bietet sich die zeichnerische Darstellung eines Hauses. Ein Haus kann man aus verschiedenen Blickwinkeln darstellen, etwa in einer Frontansicht und einer Seitenansicht. Man kann zusätzlich andere Sichtweisen wählen, wie etwa eine Darstellung der Kabel- und Leitungswege. Auch dort herrschen bestimmte Konsistenzregeln. So sollte das Höhenmass des Hauses in der Seitenansicht identisch mit dem der Frontansicht sein. Gibt es hier eine Abweichung, passen die Darstellungen nicht zusammen. Wir sehen also, auch im normalen Ingenieurwesen können für verschiedene Sichtweisen Konsistenzregeln entwickelt werden. In der Softwareentwicklung liegt ein ähnlicher Sachverhalt vor. Wir haben verschiedene

Sichtweisen und können sie unterschiedlich beschreiben, und genau deswegen müssen wir uns auch explizit mit dem Problem der Konsistenz beschäftigen. Man sollte dazu anmerken, dass Unterschiede bei der Modellierung nicht nur Nachteile bringen - sie sind durchaus beabsichtigt. Wir wollen in einem Modell schliesslich nicht das gesamte Produkt auf einmal, sondern nur diejenigen Teile beschreiben, die gerade für ein Argument oder eine Analyse massgeblich sind. Wenn verschiedene Akteure jeweils einen Teil eines Systems beschreiben, muss dafür gesorgt werden, dass die Einzelteile am Ende wieder zusammenpassen.“

Claudio Grolimund: „Im Verlauf des Gesprächs erläuterte Herr Egyed, wie Inkonsistenzen nach einer Änderung schnell entdeckt, überprüft und behoben werden können.“

Alexander Egyed: „Das Entdecken von Inkonsistenzen ist eigentlich ein sehr einfaches Problem. Bereits rudimentäre Konsistenzregeln können hochkomplexe Fehler in Pop-Modellen lösen, wie ich mit dem Beispiel des Hauses veranschaulicht habe. Bei der Konsistenzprüfung geht es darum, möglichst alle wichtigen Konsistenzregeln zu beschreiben, so dass sie auch automatisch erkannt werden können. Es gibt bereits sehr viele Automatisierungstools, die nichts anders machen, als diese Konsistenzregeln im Modell auf verschiedene Art und Weise zu evaluieren und einen entsprechenden Wert zu generieren. Das Ergebnis ist entweder konsistent, also korrekt, oder inkonsistent, also falsch. Im Wesentlichen ist dies ein einfacher Vorgang. Probleme bei der Überprüfung von Konsistenzregeln entstehen unter Anderem dann, wenn man viele dieser Regeln hat und das Modell sehr gross ist. In diesem Fall kann die Überprüfung trotz Automatisierung sehr viel Zeit in Anspruch nehmen. Das automatisierte Verfahren ist immer noch besser als eine manuelle Bearbeitung, dennoch ist es noch lange nicht optimal, wenn ein Feedback über einen Fehler nicht schnell erfolgt. In heutigen Programmierumgebungen, in Eclipse zum Beispiel, tritt dieses Problem häufig auf. Man schreibt eine Zeichenkette, ein Stück Code beispielsweise, und anschliessend wird irgend etwas unterstrichen, um anzuzeigen, dass sich ein Fehler in der Zeichenkette befindet. Es werden hier nur teilweise Lösungsvorschläge angeboten, bei der Modellierung jedoch kann es möglicherweise Stunden dauern, bis man dieses Feedback bekommt. Dauert ein solcher Vorgang lange, wird er in der Regel nur selten ausgeführt.“

Wird er jedoch nur selten ausgeführt, folgt natürlich das Problem, dass man möglicherweise mit Fehlern arbeitet, die nicht erkannt wurden. Aus diesen Fehlern entstehen weitere Fehler. Bemerkt man nach einer gewissen Zeit, dass ein Fehler vorliegt, muss man sich nicht nur erinnern, was man damals beim Schreiben gemeint hat, sondern muss auch überlegen, wie man diesen Fehler mitsamt aller möglichen Folgefehler beseitigt. Bisher haben wir eigentlich nur über das Entdecken, ansatzweise bereits über das Entfernen von Inkonsistenzen gesprochen. Das Entfernen an sich stellt jedoch grundsätzlich das grösste Problem dar. Beim Entfernen von Inkonsistenzen muss man genau verstehen, wo der Ursprung einer Inkonsistenz liegt. Ein Fehler, den ich mache, kann sich auf fünf andere Sichtweisen auswirken und daher fünf verschiedene Fehlermeldungen generieren. Dies bedeutet nicht, dass ich fünf Fehler gemacht habe. Vielmehr liegt ein einzelner Fehler vor, der eine von mehreren Auswirkungen nach sich zieht. Und dies zu verstehen, also den Rückschluss von einer Auswirkung auf die Ursache zu ziehen, ist ein noch völlig ungelöstes Problem.“

Claudio Grolimund: „Im Rahmen seiner Präsentation stellte Herr Egyed eine Lösung vor, mit der die Konsistenz eines Modells auch nach Änderungen schnell, akkurat und automatisch evaluiert werden kann. Er erklärte, wie dieses Modell funktioniert und wodurch sich diese Lösung von anderen unterscheidet.“

Alexander Egyed: „Es gibt einen Ansatz, der mir sehr gut gefällt. Er besteht im Prinzip daraus, zu sagen: Wir versuchen, die Konsistenzregel zu analysieren. Jemand schreibt also auf, was diese Konsistenzregel zu sagen hat und wir prüfen einfach, was dieses Ding eigentlich macht. Anschliessend versuchen wir anhand der Ergebnisse herauszufinden, wie man am besten mit Änderungen umgeht. Und diese Methode funktioniert sowohl für die Erkennung als auch für die Lösung von Fehlern. Das Problem besteht darin, dass die automatische Analyse einer Konsistenzregel ein äusserst komplexer Vorgang ist, der jedoch im Wesentlichen nur der Beschreibung und Entdeckung einfacher Inkonsistenzen dient. Hier ist der Punkt, an dem wir ansetzen. Wir gehen genau umgekehrt vor und schlagen einen radikal neuen Weg ein. Wir beobachten die Konsistenzregel. Wir versuchen gar nicht, sie zu verstehen. Wir umgeben sie lediglich mit einem kleinen Wrap, der keine andere Aufgabe hat, als anzuzeigen, wie die Konsistenzregel auf das Modell reagiert. Es spielt hierbei keine

Rolle, aus welchem Grund die Konsistenzregel ein Modellelement überprüft. Wichtig ist für uns vielmehr, dass die Konsistenzregel genau dieses Modellelement einer Konsistenzprüfung unterzieht. Hat eine Konsistenzregel während der Evaluierung ein Modellelement überprüft, und ändert sich dieses Modellelement, dann muss diese Konsistenzregelung neu reguliert werden. Die Technologie, die wir hierfür entwickelt haben, basiert auf einem sogenannten Model Profiler, dessen einzige Aufgabe darin besteht, den System C Checker zu beobachten. Diese Information liefert uns im Prinzip alles, was wir brauchen, um zu entscheiden, wann und wie sich eine Änderung auf ein Modell auswirkt und wie sich dies auf die Inkonsistenzen auswirken kann. Der grosse Vorteil dieses Verfahrens ist, dass man mit Hilfe dieser Information nicht nur entdeckt, wie sich eine Inkonsistenz auswirkt, sondern auch entscheiden kann, wo man Fehler beseitigt. Greifen wir also auf das vorher genannte Beispiel zurück, in dem ich erklärte, dass zwischen Fehlerursache und Fehlererkennung unterschieden werden muss, sehen wir, dass eine Inkonsistenz nur eine Meldung ist, die anzeigt, dass irgendwo ein Fehler vorliegt. Das bedeutet, die Stellen, an denen eine Inkonsistenz korrigiert wird, ergeben sich genau aus der gleichen Information des Model Profilers, der diese Konsistenzregel vorher überprüft hat. Man kann somit in einem relativ grossen Modell mit vielen Modellelementen jene 10 oder 20 Modellelemente finden, die man beim Auftreten einer Inkonsistenz untersuchen muss.

Claudio Grolimund: „Herr Egyed erläuterte abschliessend seine Sichtweise auf die Zukunft der modellbasierten Softwareentwicklung und die Möglichkeiten, die sich im Bereich der Konsistenzsicherung von Modellen bieten.“

Alexander Egyed: „Die Zukunft der modellbasierten Softwareentwicklung liegt meiner Meinung nach in ihrer Integration in den Software-Entwicklungsprozess. Die Modellierungsphase ist aus einem sehr einfachen Grund sehr wichtig für die Softwareentwicklung: Je länger ein Fehler in der Softwareentwicklung unerkannt bleibt, desto teurer wird es, diesen Fehler zu entfernen. Die Beseitigung eines Fehlers, den man bereits in der Anfangsanalyse erkennt, also ganz zu Beginn, ist mit bestimmten Kosten verbunden. Die Entfernung des gleichen Fehlers beim Testen oder während der Integration im Rahmen der Wartung kostet etwa das 30-fache. Stellt sich erst vor Ort beim Kunden heraus, dass ein Fehler vorliegt und die Software nicht benutzbar ist, kann

dies die Behebungskosten bis ins 200-fache treiben. Die Modellierung gehört zu jener Entwicklungsphase, in der die Fehlerbehebung am billigsten ist, sofern eine Behebung nicht in der Anfangsanalyse erfolgen konnte.“

Claudio Grolimund: „Wir möchten uns an dieser Stelle bei Herrn Egyed für dieses Gespräch bedanken und weisen darauf hin, dass Sie auf unserer Homepage weitere Links zum Thema modellbasierte Softwareentwicklung finden.“



© Copyright IBM Corporation 2010. Alle Rechte vorbehalten.

IBM und das IBM Logo sind eingetragene Marken der International Business Machines Corporation in den USA und/oder anderen Ländern.

Marken anderer Unternehmen/Hersteller werden anerkannt. Vertragsbedingungen und Preise erhalten Sie bei den IBM Geschäftsstellen und den IBM Business Partnern. Die Produktinformationen geben den derzeitigen Stand wieder. Gegenstand und Umfang der Leistungen bestimmen sich ausschliesslich nach den jeweiligen Verträgen. Die vorliegende Veröffentlichung dient ausschliesslich der allgemeinen Information.