

Dissertationen

Simon M. Becker: Integratoren zur Konsistenzsicherung von Dokumenten in Entwicklungsprozessen

Promotion: RWTH Aachen

Erstgutachter: Prof. Dr.-Ing. Manfred Nagl, RWTH Aachen

Zweitgutachter: Prof. Dr. Gregor Engels, Universität Paderborn

Datum der Prüfung: 26. Juni 2007

Veröffentlichung: Shaker-Verlag, 2007

Kurzfassung:

In Entwicklungs- und Entwurfsprozessen entsteht eine Vielzahl von Dokumenten. Diese werden meist arbeitsteilig, unabhängig voneinander und mit heterogenen Werkzeugen bearbeitet. Da die Dokumente dasselbe zu entwerfende Produkt unter unterschiedlichen Gesichtspunkten beschreiben, bestehen zahlreiche feingranulare Abhängigkeiten zwischen ihnen. Um die Qualität des Produkts zu sichern, ist es unabdingbar, sie zueinander bezüglich dieser Abhängigkeiten konsistent zu halten. In realen Entwicklungsprozessen geschieht dies oft zeitaufwändig und fehleranfällig von Hand. Gängige Werkzeugunterstützung erlaubt lediglich die automatische Generierung abhängiger Dokumente, wobei die inkrementelle Natur von Entwicklungsprozessen und die Notwendigkeit des interaktiven Vorgehens unberücksichtigt bleiben.

Diese Arbeit stellt einen darüber hinausgehenden Ansatz und die entsprechenden Werkzeuge zur Konsistenzsicherung vor, so genannte Integratoren. Basis des Ansatzes sind Graphen, Graphersetzungen und Tripelgraphgrammatiken. Als Anwendungsszenario werden verfahrenstechnische Entwicklungsprozesse im Anlagenbau betrachtet.

Integratoren erlauben die bidirektionale Transformation zwischen Dokumenten sowie deren Konsistenzprüfung untereinander. Die feingranularen Beziehungen der Dokumente werden persistent in einem so genannten Integrationsdokument abgelegt. Integratoren arbeiten inkrementell, sodass bei der Transformation jeweils nur die Änderungen zwischen Dokumenten übertragen werden, ohne unbeabsichtigt von der Integra-

tion nicht betroffene Teile der Dokumente zu überschreiben. Die Integration wird durch Integrationsregeln gesteuert. Da die Zusammenhänge der Dokumente nicht eindeutig sind, wird Benutzerinteraktion bei der Integration unterstützt. Integratoren leiten neue Regeln zur Laufzeit aus Benutzerinteraktionen ab. Da es in den unterschiedlichsten Anwendungsdomänen jeweils eine Vielzahl möglicher Anwendungsstellen für Integratoren gibt, müssen diese mit geringem Aufwand realisiert und an bestehende Werkzeuge angepasst werden können.

Integrationsregeln werden mittels eines größtenteils deklarativen, mehrschichtigen Modellierungsformalismus definiert. Er ist als Erweiterung des Metamodells der Unified Modeling Language (UML) realisiert. Innerhalb des Modells werden eine Typ- und eine Instanzebene unterschieden. Auf der Typebene werden zunächst in Dokumentenmodellen die Typen der in den zu integrierenden Dokumenten auftretenden Entitäten und deren mögliche Beziehungen als Klassendiagramme erfasst. Darauf aufbauend können als erste Klärung der möglichen feingranularen dokumentübergreifenden Beziehungen so genannte Linktypen zwischen den Entitätstypen verschiedener Dokumentenmodelle definiert werden. Konkreter werden die Beziehungen auf der Instanzebene des Modells formuliert. In Link-Templates werden mittels Instanzen der Entitäts- und Linktypen Graphmuster als UML-Objektdiagramme spezifiziert, die korrespondierende Situationen aus Quell- und Zieldokument sowie ihre Beziehungen beschreiben. Link-Templates sind deklarativ, sie beschreiben statische Situationen und nicht deren Erzeugung. Aus ihnen werden mittels des Tripelgraphgrammatik-Ansatzes drei Arten operationaler, ausführbarer Regeln abgeleitet: Vorwärtsregeln finden ein Muster im Quelldokument und erzeugen ein korrespondierendes im Zieldokument sowie die Beziehung im Integrationsdokument, Rückwärtsregeln arbeiten umgekehrt. Korrespondenzanalyseregeln erzeugen lediglich die Beziehung zwischen vorhandenen Mustern. Der mehrschichtige Modellierungsformalismus erlaubt die stufen-

weise Konkretisierung von Integrationswissen. Dadurch wird auch die Anbindung bereits existierender Datenmodelle erleichtert. Die Schichten werden untereinander auf Konsistenz geprüft.

Die Ausführung von Integrationsregeln erfolgt durch Graphtransformationen. Im Gegensatz zum ursprünglichen Tripelgraphgrammatik-Ansatz wird jede Regel einem Integrationsalgorithmus folgend durch mehrere Graphtransformationen ausgeführt. Dies dient der Unterstützung mehrdeutiger Regelsätze. Es werden zunächst mögliche Regelanwendungen und deren Konflikte untereinander bestimmt, diese durch Benutzerinteraktion aufgelöst und erst dann Integrationsregeln angewendet. Um diese Art der Ausführung zu realisieren, werden Regelsätze in Graphtransformationssysteme überführt, die aus regelspezifischen und allgemeingültigen Anteilen bestehen. Mit dem resultierenden Graphtransformationssystem können Quell- und Zieldokumente dargestellt und Läufe des Integrators durchgeführt werden. Dabei werden verzahnt Vorwärts-, Rückwärts- und Korrespondenzanalyserregeln ausgeführt. Neben der Ausführung operationaler Integrationsregeln erlaubt der Algorithmus auch die Prüfung und gegebenenfalls die Reparatur bereits bestehender Beziehungen im Integrationsdokument. Der Integrationsalgorithmus wird anhand der Darstellung als Graphtransformationssystem in der Arbeit detailliert beschrieben, wobei auch mögliche Optimierungen und Erweiterungen diskutiert werden.

Die Konzepte dieser Arbeit sind in Form verschiedener Prototypen realisiert. Ein auf einem kommerziell verfügbaren UML-Werkzeug aufbauender Regeleditor dient der Definition und Konsistenzprüfung von Integrationsregeln. Außerdem wurden zwei unterschiedliche Ansätze zur Realisierung konkreter Integratoren geschaffen. Eine Evaluierungsrealisierung leitet unmittelbar aus den aus Regelsätzen gewonnenen Graphtransformationssystemen Prototypen ab. Diese sind nicht an reale Werkzeuge angebunden, da sie lediglich zum Testen von Integrationsregelsätzen und des Integrationsalgorithmus dienen. Für den A-posteriori-Anschluss existierender Entwicklungswerkzeuge und somit den Einsatz in realen Entwicklungsprozessen wurde ein Integratorrahmen-

werk geschaffen, das aus einer Architektur und vordefinierten Komponenten für Integratoren besteht. Zentraler Bestandteil ist ein Interpreter für Integrationsregeln, der Regelsätze analog zu den daraus ableitbaren Graphtransformationssystemen ausführt.

Neben dem Integrator für das Anwendungsszenario werden in der Arbeit weitere Integratoren vorgestellt. Diese decken zumindest einen Teil der Bandbreite möglicher Integrationsprobleme ab. Besonders interessant sind dabei solche, die ein komplexes Geflecht von Dokumenten integrieren, wobei auch die Kopplung zwischen der technischen und der administrativen Ebene von Entwicklungsprozessen untersucht wird.

Die Ergebnisse dieser Arbeit werden in Zukunft in Kooperation mit dem bisherigen Industriepartner im Rahmen eines Transferprojekts in die Praxis übertragen.