

Die Zukunft der Entwicklung von Geschäftsapplikationen

Stephan Kraft, Andreas Essigkrug

IBM Rational Software Hollerithstr. 1 81829 München
stephan.kraft@at.ibm.com, Andreas.Essigkrug@de.ibm.com

Abstract: Mit Techniken aus Model Driven Design (MDD) und Architected Rapid Application Development (ARAD) können mit dem neuen IBM Rational Rapid Developer moderne mehrschichtige Geschäftsapplikationen in Rekordzeit entwickelt werden. Insbesondere die strikte Trennung von fachlichem Modell und voll generierter technischer Plattform erlaubt den fachlich erfahrenen Entwicklern ausgereifte Web Applikationen für verschiedene technische Plattformen wie J2EE oder Microsoft DNA/COM ohne tiefere Kenntnis von diesen Plattformen zu entwickeln. Ein späterer Plattformwechsel ist durch den voll generativen Ansatz ebenfalls mit minimalem Aufwand verbunden.

Mit Techniken aus Model Driven Design (MDD) und Architected Rapid Application Development (ARAD) können mit dem neuen IBM Rational Rapid Developer moderne mehrschichtige Geschäftsapplikationen in Rekordzeit entwickelt werden. Insbesondere die strikte Trennung von fachlichem Modell und voll generierter technischer Plattform erlaubt den fachlich erfahrenen Entwicklern ausgereifte Web Applikationen für verschiedene technische Plattformen wie J2EE oder Microsoft DNA/COM ohne tiefere Kenntnis von diesen Plattformen zu entwickeln. Ein späterer Plattformwechsel ist durch den voll generativen Ansatz ebenfalls mit minimalem Aufwand verbunden.

Eine Aufgabe der Informationstechnologie ist die computerunterstützte Optimierung von Geschäftsprozessen. Dabei treffen zwei Welten aufeinander. Einerseits kennen Rechner nur Nullen und Einsen als Basis ihrer gesamten Welt. Menschen und Geschäfte "funktionieren" auf der anderen Seite mit einer Vielzahl von abstrakten Begriffen und Interaktionen (z.B. "Kunde Maier", "Bankkonto 4711").

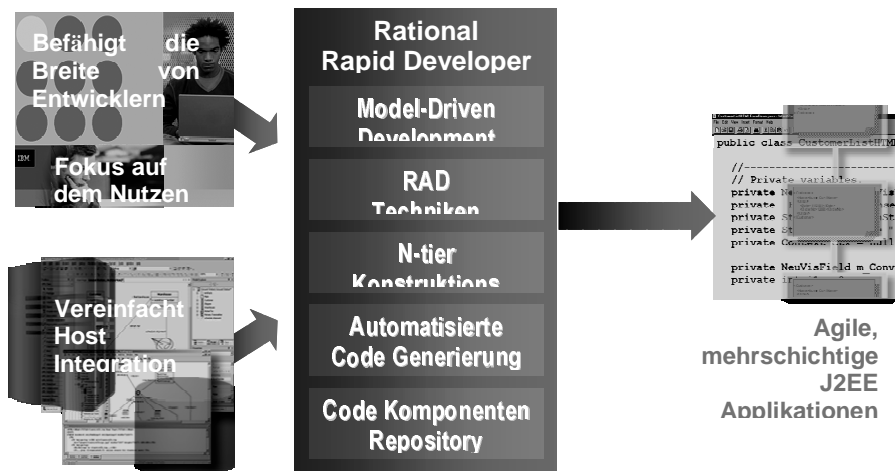
Die ersten Computer-Systeme wurden noch durch sehr computernahe Methoden und Programmiersprachen erstellt (z.B. Assembler), was zwar wenig Transformationen notwendig macht, allerdings für die Implementation ein sehr spezielles Wissen erfordert und in Bezug auf Wartbarkeit und teamübergreifender Zusammenarbeit große Hürden darstellt. Eine Lösung dieser Herausforderungen bieten Programmiersprachen mit höherem (geschäftlich- und realitätsnäherem) Abstraktionslevel und entsprechenden Transformationswerkzeugen (Compilern), die den spezifizierten Output in maschinenverständlichen und ausführbaren Code umwandeln. Zusätzlich dazu wurden sogenannte Rapid Application Development -Umgebungen (RAD) am Markt angeboten, mit denen man häufig benötigte Programmieraufgaben parametrisiert und werkzeuggestützt wesentlich schneller erledigen konnte.

Dies führte zu einer wesentlichen Beschleunigung der Entwicklung und erleichterte zudem den Zugang zu komplexen Technologien auch für weniger erfahrene Programmierer. Problematisch an herkömmlichen RAD-Umgebungen war allerdings der Umstand, daß sich die Umwandlung zu tatsächlich ausführbarem Code als Black-Box darstellte und man damit auf Gedeih und Verderb auf den Anbieter der Entwicklungsumgebung angewiesen war.

Architected Rapid Application Development (ARAD) bietet als evolutionären Schritt in der Software-Entwicklung eine Technologie, bei der die Umsetzung der Geschäftsprozesse in einen maschinenverständlichen Code durch eine Reihe von transparenten und praxiserprobten Transformationsschritten automatisch durchgeführt wird. Das Resultat ist eine skalierbare, weil schichtenbasierte, 100% J2EE-kompatible Applikation.

Ein weiterer Ansatz, der in den letzten Monaten für Furore gesorgt hat, ist die "Model Driven Architecture"-Technologie (MDA), der das Geschäftsmodell komplett von der technischen Umsetzung trennt und damit höhere Investitionssicherheit und Änderungsfreundlichkeit ermöglicht. Ein Wechsel des Applikationsservers oder der darunter liegenden Datenbank, der naturgemäß keinen Einfluß auf das Geschäftsmodell hat, soll damit weitgehend automatisch durch Neugenierung der Applikation bewerkstelligt werden.

Diese Ansätze - ARAD und MDA - verbindet der Rational Rapid Developer zu einer richtungsweisenden Softwareentwicklungsumgebung. Das wichtigste Modell des Rational Rapid Developers ist ein UML Klassenmodell, das eine Abstraktion der natürlichen Begriffe des Geschäftsvorgangs und deren Beziehungen darstellt. Auf dieses Modell beziehen sich die meisten anderen Modelle. Zu den Begriffen werden Eigenschaften (Attribute) und Verarbeitungsschritte oder Geschäftsregeln in Java-Notation definiert. Das Klassenmodell kann von UML Werkzeugen wie Rational Rose importiert oder aus einer Datenbankstruktur bezogen und beliebig erweitert werden.



Die Präsentation im Web wird über ein Modell der Seiten und Ihrer Links und über ein WYSIWYG-Seitenlayout definiert. Der Rational Rapid Developer erlaubt es dabei, komplexe Webseiten mit Datenbankinhalten zu definieren, ohne eine Zeile Code oder SQL zu schreiben.

Die konkret verwendeten Technologien, wie Applikationsserver, Datenbank, Messageserver, usw. werden von Technologie-Spezialisten unabhängig vom Rest der Applikation eingestellt. Der komfortable Partition Architect erlaubt darüber hinaus ein visuelles Deployment der verschiedenen Applikationselemente (Webseiten, Komponenten, Messageschnittstellen, ...) auf die eingesetzten technischen Plattformen, vor allem die verschiedenen Applikationsserver.

Das Werkzeug generiert Software-Code auf Basis des J2EE-Standards. Dabei wird eine wohl definierte Architektur aufgebaut. Diese garantiert wiederum optimale Performance und Wartbarkeit. Der Architekt kann dabei das generierte Ergebnis beeinflussen.

In einer Studie wurde von einem unabhängigen Unternehmen (The Middleware Company) die Erstellung der J2EE-Referenzapplikation PetStore mit Rational Rapid Developer im Vergleich zu einer herkömmlichen Java-Programmierungsumgebung getestet. Das Ergebnis war eindeutig: Rapid Developer bewirkte eine Reduktion der Produktionszeit um 90 Prozent.