



Aktuelles Interview mit Bill Friend

Was sind aus Ihrer Sicht die Stärken von Oracle Forms?

Bill Friend: Ich würde sagen, die Automatisierung der Transaktionen, also das dynamische SQL und der korrekte Ablauf von Triggern bei der Änderung von Formular-Daten durch den Benutzer; eine Benutzerführung gesteuert durch Transaktionen und Trigger. Das Herzstück ist jedoch die Produktivität, die viele Funktionen umfasst. Das schien etwas zu sein, das unsere Konkurrenz nicht erreichen konnte. Es war ein Ansatz von 4GL, und wo 4GL aufhörte, ließ sich die Logik prozedural erweitern.

Forms wird dieses Jahr vierzig Jahre alt und erfreut sich wachsender Beliebtheit. Konnten Sie den Erfolg von Forms vorhersehen?

Bill Friend: Es ist für mich eine große Ehre, dass Oracle Forms, von einigen kleinen Modifikationen abgesehen, immer noch im Stil der anfänglichen Version in Betrieb ist. Ich habe nicht gedacht, dass es so lange existieren würde, und dachte eher, Oracle würde es selber einmal komplett ablösen. Ich wollte auch einmal eine neue Architektur einführen, um einige Restriktionen zu beseitigen. Ich denke, es gibt nur sehr wenige Beispiele eines Systems oder Produkts, das diese Laufzeit in einer Industrie aufweist, die sich so stark geändert hat. Zum ersten Mal hatte ich im Jahr 1984 das Gefühl, etwas Bedeutendes entwickelt zu haben, als ich die vielen Job-Angebote für Forms- und Reports-Entwickler gesehen habe.

Jemand sagte einmal, dass die Datenbank und Forms die einzigen Produkte gewesen wären, die Oracle selber entwickelt hat. Stimmt das?

Bill Friend: In der Zeit, als wir mit Oracle starteten, war Computer Associates so wie heute Oracle – sie kauften alles auf, um zu wachsen. Heute scheint Oracle eine ähnliche Position eingenommen zu haben. Jedoch sind Tools wie Oracle Designer, Oracle Applications (die originale Version) und viele Data-Warehouse-Tools so wie die, die ich von 1998 bis 2000 entwickelt habe, bei Oracle selbst entstanden.

Wie hat Larry Ellison den Entwicklungsprozess von Forms geprägt?

Bill Friend: Larry ist der smarteste Typ, den ich je kennengelernt habe, und ein Visionär – nicht nur in Technologie-Fragen, sondern auch darin, wie er eine Firma groß machen kann. Er hat auch einen feinen, sich selbst infrage stellenden Humor, der ihn nahbar macht. Ich mag Larry sehr, aber es kann auch für manche sehr frustrierend sein, mit ihm zusammenzuarbeiten. Genauso mochte ich Bob Minor, er hatte großen Einfluss auf mich. Es gab bei Oracle in den Anfangstagen bei allen einen starken Willen, intelligent, direkt und, wenn nötig, auch sehr scharf zu sein, wenn es um kritische Fragen zu Software-Produkten ging. Jeder war in seinem Bereich sehr gegenwärtig und stand ständig mit anderen in Konkurrenz, um bei dem, was er tat, die besten Ergebnisse zu erzielen. Das ging sogar bis hin zu Hobbies wie Tennis, Poker oder anderen Freizeitbeschäftigungen. Larry stand immer mit allen im Wettbewerb und wollte ständig der Beste sein, auch die besten Ergebnisse bei unserer Arbeit erzielen. Das hatte Einfluss auf unsere entwickelten Produkte.

Was bedeutet ein Wechsel von Forms (4GL) zu einer 3GL-Sprache?

Bill Friend: In meiner Zeit gab es eine Bewegung, die 4GL-Forms-Lösungen mit Java ablösen wollte. Ich bin mir nicht sicher, ob dieser Weg der richtige war. Bei dem Wechsel zu einer 3GL-Sprache geht viel von dem Funktionsumfang einer 4GL-Applikation verloren, etwa die Fähigkeit, automatisch, zuverlässig und für den Entwickler unsichtbar SQL-Transaktionen abzuwickeln. Das muss in 3GL-Sprachen komplett selbst übernommen werden. Der Vorteil einer nicht prozeduralen 4GL-Sprache besteht darin, dass sie viele komplexe Vorgänge sehr einfach beschreiben kann. Daher lieben Entwickler 4GL-Werkzeuge, nehmen allerdings dabei in Kauf, dass Vorgaben, die mit den vorgegebenen Funktionen in 4GL nicht umsetzbar sind, auch nicht ohne Weiteres möglich sind. Dann wird das 4GL-Werkzeug oft infrage gestellt. Eine prozedurale Sprache bietet dagegen unbegrenz-

te Flexibilität, was Entwickler lieben, aber dann ohne das ganze Spektrum der 4GL-Funktionen. Forms kann die Vorteile beider Welten vereinen – ein automatisches Transaktions-Management sowie die Flexibilität und Einbindung von prozeduralen Triggern und Funktionen, um den Ablauf programmtechnisch zu beeinflussen.

Wie viele Entwickler hatte Forms in den ersten Versionen?

Bill Friend: Einen, und am Ende meiner Entwicklungszeit vielleicht noch mit Sohaib Abbasi einen zweiten dazu, der mich dann ablöste. Ich meine, er ist später auch ein Senior Vice President der Tool-Devision geworden. Vielleicht hätte ich in den ersten Jahren noch mehr Mitarbeiter für die Entwicklung von Forms anfordern sollen – aber es machte mir einfach so viel Spaß, alles selbst zu machen.

Warum macht Oracle so wenig Werbung für Forms?

Bill Friend: Ich würde sagen, es frustriert Einige bei Oracle, dass Forms noch lebt (und anscheinend gedeiht). Die Menschen möchten neue Produkte vorantreiben, von denen sie glauben, dass sie etwas so Altes wie Forms ersetzen könnten. Wie ich in meiner Biografie schon gesagt habe, ist Oracle ein sehr wettbewerbsliebender Ort. Wenn aber die neuen Produkte nicht das Wesentliche dessen ausmachen, was Forms so leistungsfähig und hochproduktiv macht, sowohl für Programmierer als auch für Anwender, wird Forms wahrscheinlich fortfahren und gedeihen. Ich bedaure es manchmal, dass ich nicht selbst eine neue Forms-Version erstellt habe, weil ich glaube, dass ich aus meinen Erfahrungen mit Applications und Data Warehousing einige sehr gute Ideen habe, die, auf den Kernfunktionen von Forms aufbauend, eine Menge der Einschränkungen, die Forms heute hat, beseitigen könnte.

Eine letzte Frage: Wären Sie bereit, beim nächsten Forms Day per Videokonferenz live dabei zu sein, um die deutsche DOAG-Forms-Gruppe kennenzulernen?

Bill Friend: Sehr gerne.