
.NET arbeitet mit Daten? Nicht wie der Fuchs!

Anders Hejlsberg (der „Vater“ von C#) wurde kürzlich über die zukünftige Entwicklung der Programmierung mit C# interviewt. In dem Auszug des Interviews, den Sie unter der URL <http://www.microsoft-watch.com/article2/0,2180,1837434,00.asp> lesen können, spricht Anders über verschiedene Problembereiche im Zusammenhang mit den Daten. Dort verwendet er auch das verflixte „F“-Wort. Nein nicht **F#**, das er aber auch erwähnt, sondern **FoxPro**. Hier ein Zitat:

„Nein. Ich würde nicht sagen, dass C# Anleihen bei FoxPro nimmt. Wie ich bereits ausgeführt habe, haben wir dafür bereits in Whidbey (Visual Studio 2005) einige Grundlagen gelegt. Es handelt sich um einen großen unerforschten Bereich der tieferen Integration von Sprache und Daten. Bei FoxPro handelt es sich um eine Sprache, die für eine spezielle Anwendungsart gedacht ist.“

Wenn Sie dieses Interview lesen, werden Sie feststellen, dass Anders zum großen Teil über Features von „Orcas“ spricht, das der Codename für die nächste Version von Visual Studio nach Whidbey ist, das für November angekündigt ist. Ich erwarte daher das Erscheinen von Orcas nicht vor 2007. Wenn ich Anders richtig verstehe, sagt er, dass, wenn .NET 7 Jahre alt ist, C# eventuell Features aufweisen könnte, die „nah an den Daten“ sind, was für einige Anwendungen „furchtbar hilfreich“ sein kann. Eigentlich für jede Anwendung, die Daten verwaltet. Inzwischen erledigt VFP diese Aufgabe extrem gut und das bereits länger, als Microsoft FoxPro besitzt.

Etwas später in dem Interview führt Anders zu FoxPro und anderen dBase-Sprachen aus: „Ihnen fehlen einige Fähigkeiten, die Programmierer haben wollen.“ Vielleicht ... Marketing? Ken Levy hat bei der Werbung für VFP mit Abstand den besten Job gemacht, aber ihm sind ganz klar die Hände gebunden. Sicher haben VFP-Entwickler immer verschiedene Features auf ihrer Wunschliste stehen (sehen Sie sich dafür nur einmal das Fox-Wiki und den UT an). Aber der Großteil der „Fähigkeiten“, nach denen VFP-Entwickler immer gefragt haben, waren Marketinganstrengungen, die den technischen Fähigkeiten von VFP angemessen sind.

Hier die letzte Erwähnung von FoxPro in dem Interviewauszug:

„Ich möchte klarstellen, dass ich den Umgang mit den Daten in C# so einfach machen möchte, wie er in FoxPro ist. Ich sage aber nicht, dass dies ein besonderes Ziel ist.“

Auf der einen Seite zeigt er richtig auf, dass das Programmieren von Daten in VFP einfacher ist als in C#. Anschließend ist er darauf bedacht, jede Annahme zu verbieten, dass die Verwaltung von Daten in C# so einfach wird, wie er es von VFP kennt. Microsoft möchte aber, dass alle VFP-Entwickler so schnell wie möglich zu .NET wechseln, wohl wissend, dass alles, was wir tun, direkt mit der Datenverwaltung zusammenhängt. Bedenken Sie auch hier, dass wir nicht über VS 2005 sprechen, sondern über dessen Nachfolgeversion.

Mir gefällt alles, was Anders über die Möglichkeit eines integrierten Abfrage-Frameworks sagt, das es ermöglichen würde, Daten unterschiedlicher Datentypen abzufragen, auch in XML und Objekten, nicht nur in SQL/relationalen DBMS. Es wird vermutlich interessant werden, zu sehen, wie dies in Orcas funktioniert. Trotzdem finde ich dieses Zitat von Anders Hejlsberg am Ende des Interviewausschnitts etwas prophetisch:

„Sie müssen angeben, was Sie erledigen wollen, aber nicht exakt, wie es getan werden soll. Wir haben Generationen von Programmierern beigebracht, nicht nur darüber nachzudenken, was getan werden soll, sondern auch wie es getan werden soll, und explizit anzugeben, wie dies in ihren Programmen geschehen soll.“

In vielen Bereichen müssen Programmierer dies wieder verlernen und dafür lernen, denjenigen zu vertrauen, die das ‚Was‘ entwickeln. Die Maschine ist intelligent genug, um das ‚Was‘ so zu erledigen, wie die Programmierer es wünschen, oder auf die effizienteste Weise.“

Hmm... Das Ziel ist, die .NET-Sprachen so zu verbessern, dass der Entwickler dem Rechner nur sagen muss, was er tun soll, und es geschieht dann effizient, konsistent und korrekt. Das klingt etwas anders als das, was ich von VFP erwarte, wo ich immer verschiedene Möglichkeiten habe, eine gegebene Aufgabe zu erledigen, auch das Abfragen von Daten. Häufig habe ich dadurch die Möglichkeit, anzugeben, ‚wie‘ etwas geschehen soll, einen Code zu erstellen, der funktioniert – die erste Technik könnte langsam oder ineffizient sein, nicht wie erwartet funktionieren, oder überhaupt nicht arbeiten. Oder wie mir kürzlich ein Entwickler sagte: „Es gibt fast eine Million Wege zum gleichen Ziel. Dadurch bleiben die Dinge interessant.“

Vergeben Sie mir bitte meine klammheimliche Freude, aber warum soll ich mir die Mühe machen, dieses Konzept umzusetzen, meine Programmbefehle in Abfragen umzuwandeln, nur weil dies ein hochdotierter Manager eines Unternehmens dies fordert, dessen Betriebssystem schon Probleme hat, auf meinem Rechner all die für Windows zertifizierte Software auszuführen? Hey, ich will hier nicht auf Microsoft/C#/Anders Hejlsberg einprägen. Wirklich nicht. Ich glaube, dass seine Ziele für C#/.NET im Zeitrahmen von Orcas hochgesteckt und für Entwickler vorteilhaft sind. Wenn aber VFP-Entwickler dies lesen, werden sie ungläubig den Kopf schütteln, da sie sich über den Fehlen von datenzentrierten Features in den .NET-Sprachen wundern. Wir wundern uns, wie ein Unternehmen, das Eigentümer von VFP ist und das Hunderte Millionen Dollar (eventuell auch Milliarden?) in die Entwicklung und das Marketing von .NET gesteckt hat, immer noch sagen kann, dass die .NET-Sprachen nicht nativ mit Daten umgehen können. Und so weit ich sagen kann, behandeln Features wie das Integrated Query Framework nur die Abfrage von Daten, nicht deren Aktualisierung.

Die Gerüchte über die Aufnahme VFP-ähnlicher datenzentrierter Features in .NET sind relativ spät aufgekommen. Beispiele finden Sie im Blog von David Stevenson. Damit könnte .NET für VFP-Entwickler interessanter werden, wir müssen aber noch abwarten, in welcher Form diese Features in .NET enthalten sein werden.

Ach, und hatte ich eigentlich schon erwähnt, dass diese Themen alle die *nächste* Version von .NET betreffen? Nein, nicht Visual Studio 2005, sondern eine Version später.