

OOP: Anwender suchen Alternativen zur Unified Modeling Language

## Modellierungsstandard verliert an Gewicht

**München (fg) – Domänenspezifische Techniken für die Beschreibung von Software-systemen gewinnen an Akzeptanz. Der Grund dafür liegt in der Komplexität des etablierten Modellierungsstandards UML (Unified Modeling Language).**

„Wir haben von den Kunden gelernt, dass UML zu umfangreich ist und trotzdem manchmal nicht alle Anforderungen an ein System so abbilden kann, wie der Anwender es möchte“, erläutert Bastiaan Schönhage, Produktmanager beim Entwicklungsspezialisten Compuware, auf der OOP-Konferenz. Daher unterstützt der Hersteller neben

UML-basierten Techniken nun auch die Definition domänenspezifischer Modellierungssprachen (DSL), mit denen sich die Anforderungen eines bestimmten Fachbereichs erfassen lassen. Mit dem Hamburger Hersteller Gentleware offeriert inzwischen ein weiterer Anbieter aus dem UML-Lager DSL-Lösungen.

Verständnis für diesen Trend äußert Professor Heinrich Mayr, Rektor der Alpen-Adria-Universität in Klagenfurt: „Warum soll es nicht ein domänenspezifisches Modell geben, wenn der Benutzer in einer bestimmten Domäne tätig ist?“ Mayr plädiert dafür, beim Design von IT-Systemen nur eine kleine Anzahl grafischer

Notationselemente zu verwenden und nennt als Beispiel die Petrinetze, ein Konzept, das in den sechziger Jahren entworfen wurde. Dieses einfache Modell lasse sich sehr vielseitig verwenden. „Ich bin ein Gegner von Ansätzen, die versuchen, immer mehr in eine Modellierungssprache zu packen, bis es aufwändiger ist, die Beschreibungssprache zu lernen als das System zu implementieren“, so Mayr.

Die Definition einer domänenspezifischen Sprache gilt aber als außerordentlich schwierig. Compuware-Manager Schönhage rät, eine DSL im Dialog zwischen IT-Experten und Anwendern zu definieren und ständig

zu verfeinern: „Auf iterativem Weg erhält man so eine Lösung, mit der man arbeiten kann.“

Für Embedded-Software, die in Autos oder Handys eingesetzt wird, sei es dagegen leichter, DSLs zu definieren, meint Juha-Pekka Tolvanen, Chef des finnischen DSL-Pioniers Metacase.

„Wenn es für ein Notationselement eine physikalische Entsprechung gibt, lässt sich die Sprache einfacher bauen“, so Tolvanen. „Embedded-Systeme besitzen auch den Luxus, dass der Anwendungsbereich klar abgegrenzt ist. Bei typischen Unternehmensanwendungen gibt es dagegen Millionen möglicher Richtungen.“

Software as a Service benötigt spezielle Designkriterien – Datenbankabfragen sind zu optimieren

## Vier Sekunden entscheiden

**München (fg) – Anwendungen, die via Internet als Service angeboten werden sollen, müssen entsprechend programmiert werden, warnt Brian Behlendorf, Chief Technology Officer von Collabnet.**

„Im Jahr 2000 nahmen die meisten Application Service Provider einfach existierende Standardanwendungen und offerierten sie als Service per Web. Die Applikationen waren aber nicht darauf ausgelegt“, begründete Behlendorf am Rand

der Entwicklertagung OOP die damals geringe Akzeptanz. „Außerdem gingen die Anbieter davon aus, dass die Benutzer über Netzwerke mit hoher Bandbreite verfügten. Das war aber oft nicht der Fall.“

Beim Design von Software-as-a-Service-Applikationen (SaaS) spielt die Aufmerksamkeits-spanne der Benutzer eine wesentliche Rolle, so Behlendorf: „Wenn die Antwortzeit länger als vier Sekunden beträgt, wird der Anwender abge-



**Behlendorf: Der User soll nicht lange warten.** Foto: Collabnet

lenkt.“ Architekten von SaaS-Applikationen müssten daher wie Webdesigner denken und die via Web übertragene Datenmenge möglichst gering halten. Zudem rät Behlendorf, komplexe Datenbankabfragen, die für die Darstellung auf einer Seite nötig sind, auf ein Minimum zu beschränken. Sind solche Abfragen unumgänglich, empfiehlt er, dies dem Nutzer klar mitzuteilen – etwa durch die Meldung, dass die Anfrage gerade bearbeitet wird.