

# XML-Dokumente bündeln Daten und Prozesslogik

**Herkömmliche Entwicklungskonzepte zwingen Softwareprojekte in ein Korsett, das Umbauten erschwert. Organische Programmiermethoden versuchen dagegen, das Wachstum der Natur zu kopieren.**

Die Komplexität der realen Welt resultiert aus der Interaktion vieler einfacher Teile. Man spricht hier auch von Emergenz. Bisher gibt es im Bereich des Software-Engineering erst ein Beispiel für Emergenz, nämlich das Web: Aus einfachsten Anfängen hat sich im Zusammenspiel mit dem Menschen spontan eine Entität mit der bisher größten Softwarekomplexität entwickelt.

Die Idee ist einfach: Man setze komplexe Softwaresysteme aus simplen, aber autonomen Teilen zusammen – jedes einfacher als ein Programmiersprachenartefakt, aber zugleich dem Vorbild aus der realen Welt viel ähnlicher als ein Objekt einer objektorientierten Sprache. Die Interaktionen zwischen diesen Dingen sollen so vielfältig sein, dass sich ein Softwaresystem entwickeln kann. Veränderliche Implementierungsdetails werden virtualisiert und in einer Virtual Machine gekapselt.

So genannte Ercatons liefern die Umsetzung des Ansatzes. Das sind einfache XML-Dokumente,

die sowohl Daten als auch Verhaltensmuster enthalten.

Objektorientierte Prinzipien wie Vererbung und Kapselung werden direkt erfüllt. Zusätzliche Verweise auf Binärdaten – etwa Java-Code – sind möglich. Um Duplizierung zu vermeiden, kann ein Ercaton auf ein anderes als seinen Prototypen verweisen und damit von ihm – auf Instanzebene – erben.

Während die Daten eines Ercatons im XML-Inhalt liegen, ergibt sich sein Verhaltensmuster aus den ebenfalls auf XML-Ebene deklarierten Actions (Verhaltensweisen), Target-Pipelines (den Sichten und Bezügen) und Triggern (bei Ereignissen). Dabei verweist eine Action auf andere Ercatons, also auf Java- oder XSLT-Code (Extensible Stylesheet Language Transformations).

Da wiederverwendbare Algorithmen durch Ercatons gekapselt werden, ist die Prozesslogik so meist vollständig auf XML-Ebene beschrieben. Prozessrelevante Daten eines Ercatons werden in einer SQL-Datenbank indiziert.

In einem Softwareprojekt mit Ercatons erzeugen diverse Fachautoren XML-Dokumente fachlichen Inhalts für die Geschäftsobjekte und -prozesse. Fachanwender überarbeiten diese

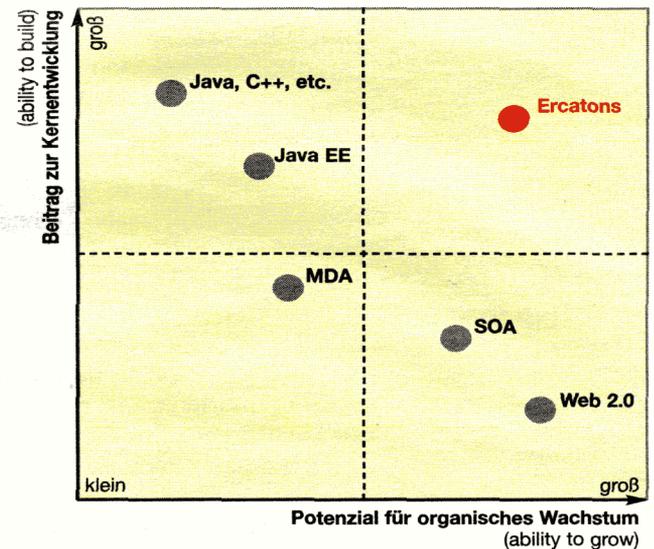
später. Programmierer schreiben fehlende Algorithmen in Java oder anderen Sprachen – von Ercatons gekapselt und allgemein verfügbar gemacht. Der Unterschied zwischen Programmierer, Fachexperte und

Anwender verwischt. Der Prozess ist ein stetes Wachsen und benötigt nur eine grobgranulare Vorabplanung, üblicherweise in Form prototypischer Ercatons.

*Falk Langhammer, Geschäftsführer Living Pages Research/fg*

## Korsett und kreatives Chaos

Vergleich unterschiedlicher Programmiertechniken unter dem Aspekt der organischen Programmierung



Mit Ercatons programmierte Softwaresysteme können quasi organisch wachsen. **Traditionelle Ansätze, aber auch die Model Driven Architecture (MDA), wirken hier einengend.** Serviceorientierte Architekturen (SOA) und Web 2.0 besitzen zwar Potenzial für organisches Wachstum, verfügen allerdings nicht immer über einen klaren Fokus. *fg*