

Programmierung versus Industrialisierung – der Ausgang bleibt offen

Die industrialisierte Fertigung von Software ist ein ewiges Ziel unserer Branche. Es gibt heute ein paar interessante Ansätze für die produktive Bereitstellung von Software. Die Frage ist lediglich, ob sie nicht von den neuen Mobile-Technologien und der zunehmenden Flexibilität der 3-GL-Programmierungsumgebungen wieder eingeholt werden.



«Eine Abgrenzung der Einsatzgebiete von Programmiersprachen wird sich erst entwickeln.»

Daniel Liebhart ist Dozent für Informatik an der Hochschule für Technik in Zürich und Solution Manager der Trivadis AG.

Zwei der wichtigsten modernen Werkzeuge, die eine signifikante Produktivitätssteigerung versprechen, sind die generative Programmierung durch Domain Specific Languages und die grafische Programmierung mittels Mashups, Process und Rule Engines. Beide Ansätze versprechen einen Sprung in Sachen Industrialisierung der Softwareentwicklung.

Generative und grafische Programmierung

Domain Specific Modelling (DSM) ist der Überbegriff im Bereich Software Engineering, der für die Generierung von Code aus einem abstrakten Modell steht. Die Kernkomponente von DSM ist die Domain Specific Language (DSL). DSL werden auch als 4-GL-Sprache bezeichnet. Eine DSL ist eine Sprache, die für eine definierte Problemdomäne entwickelt wird. Sie enthält im konkreten Fall Elemente, die typischerweise den Grundbausteinen des abzubildenden Fachbereiches entsprechen. Es wird die Begriffswelt der Domäne übernommen, um Anwendungen basierend auf diesen Grundbausteinen modellieren zu können. Den Elementen einer DSL werden grafische Repräsentationen zugewiesen, und anschliessend werden die zu erstellenden Anwendungen anhand dieser Repräsentationen modelliert, was auch als Workbench DSL bezeichnet wird. Die standardisierte Ausprägung von DSM ist MDA (Modell Driven Architecture), die auf Basis von UML Profilen Code generiert.

Die grafische Programmierung basiert auf einer sehr einfachen Idee: Eine Software

wird mittels eines Modellierungstools grafisch gestaltet. Das Tool generiert aus der grafischen Darstellung ausführbaren Code, der entweder durch eine spezielle Engine oder jedoch durch vorherige konventionelle Kompilierung direkt auf dem Rechner ausgeführt werden kann. Es gibt eine Vielzahl von solchen grafischen Programmierungsumgebungen. Heute sind zwei Arten dieser Programmierungsumgebungen verbreitet: Instrumente zur Bereitstellung von sogenannten Business Mashups und Werkzeuge für ausführbare Prozesse und Regeln.

Mashups

Die Programmierungsumgebungen für Business Mashups eignen sich für einfache Anwendungen im Bereich Office-Processing. So zum Beispiel einfache Workflows wie die Erfassung einer Reklamation mit anschliessender Nachbearbeitung bis hin zur Erzeugung eines Reports und möglicher automatisierter Antworten an den Kunden und Ablage im zentralen System zur Archivierung. Business Meshups gehen davon aus, dass die Funktionalität bestehender Systeme zur Bereitstellung solcher Abläufe aufgerufen werden kann. Der Ablauf selbst wird jedoch nicht im bestehenden System realisiert, sondern als schlanke separate Anwendung umgesetzt.

Process & Rule Development

Eine völlig andere Art von grafischer Programmierungsumgebung sind die Werkzeuge für ausführbare Prozesse und Regeln. Sie arbei-

ten mit formalen und meist standardisierten Sprachen als Grundlage. Beispiele solcher Sprachen sind BPEL, BPMN oder XPDLL für ausführbare Prozesse und BRML, SWRL oder RuleML für Regeln. Die Modellierung erfolgt mittels grafischer Elemente, die den Sprachkonstrukten exakt entsprechen. Sie werden meist im Kontext einer auf SOA oder Cloud Computing basierenden Grundarchitektur eingesetzt und decken den dynamischen Teil der Businesslogik, also die Steuerung der Abläufe einer Anwendung ab. Jeder der grossen Hersteller hat in seinem Portfolio die entsprechenden grafischen Programmierungsumgebungen. Sie eignen sich für eine grosse Anzahl von Einsatzgebieten. Ob nun die Abbildung von Validierungsregeln, die Steuerung eines Datenaustauschs, die Umsetzung der Meldungsverteilung in einem Enterprise Service Bus oder jedoch die Gesamtsteuerung der Anwendung – ausführbare Prozesse und Regeln sind immer im Spiel.

Mobile Programming und die Klassiker

Neben den beiden am meisten in der Entwicklung verwendeten Programmiersprachen Java und C# und deren Entwicklungsumgebungen werden zunehmend im Rahmen der Bereitstellung von mobilen Anwendungen Programmiersprachen wie Objective C und sogenannte CPT(Cross Plattform Tools)-Frameworks wie beispielsweise PhoneGap eingesetzt. Mit Windows 8 wird zudem eine weitere Programmiersprache – HTML 5 – ihren Einzug in die Entwicklung von Individualsoftware auf Unternehmensebene feiern. Zur Verbesserung der Produktivität haben verschiedene Hersteller umfangreiche Toolkits bereitgestellt, die den ganzen Entwicklungsprozess unterstützen und damit auch mit den innovativen Instrumenten der generativen und grafischen Entwicklung konkurrieren. Eine Abgrenzung der Einsatzgebiete wird sich erst entwickeln. <