

Kurzfassung

Konzeption und Realisierung einer Referenzarchitektur für die einfachere Entwicklung von MDS-Generatorsystemen mit dem Schwerpunkt auf Unternehmensanwendungen

Zheng Zhang

An eine modellgetriebene Entwicklung von Unternehmensanwendungen werden folgende Anforderungen gestellt: Flexible Adaptierbarkeit, hohe Modellqualität, Unterstützung für Modularität, Änderbarkeit und Updatefähigkeit der generierten Anwendungen sowie benutzerfreundliche Entwicklungsumgebung. Anhand dieser Anforderungen wurden folgende Lösungsansätze untersucht: AndroMDA, oAW, JANUS, UWE und openMDX. Keines der untersuchten Ansätze erfüllt die Anforderungen vollständig. Die Anpassung eines vorhandenen Generatorsystems an eine bestimmte Zieldomäne ist sehr aufwendig und kostspielig. Außerdem werden die Modularität und die Anwendungsaktualisierung nur beschränkt oder gar nicht berücksichtigt. Die Bedienbarkeit und Erlernbarkeit sind bei den meisten Lösungsansätzen unzureichend. Diese Defizite haben dazu geführt, dass sich die modellgetriebene Entwicklung – obwohl bereits seit 20 Jahren bekannt – in der Praxis noch nicht breit durchgesetzt hat.

In dieser Arbeit wird ein neuer Ansatz vorgestellt, der aus einem konzeptionellen Entwurf, einer Referenzarchitektur und einem Vorgehensmodell besteht. Es wird eine Meta-Architektur als Modellierungsgrundlage für beliebige Anwendungsdomänen vorgestellt. Domänenspezifische Modellierungsansichten erleichtern die Lesbarkeit von Modellen. Beispielsweise kann eine domänenspezifische Sprache (DSL) mit Hilfe der Meta-Architektur für die GUI-Beschreibung definiert und eine dazugehörige Modellierungsumgebung entwickelt werden, in der die Benutzungsoberflächen (DSL-Modellen) nach dem WYSIWYG-Konzept erstellt werden können. Die fachliche Korrektheit der Modelle wird durch ein kontextsensitives Validierungsverfahren, welches während der Modellierungszeit kontinuierlich durchgeführt wird, gewährleistet. Eine Anwendung kann in der Modellierungsphase in unterschiedlichen Paketen (Teilmodellen) beschrieben und zur Laufzeit auf den daraus generierten Anwendungsmodulen aufgebaut werden (Modularitätskonzept). Eine Softwareaktualisierung der generierten Anwendungen wird auf der Modellebene ermöglicht. Eine Referenzarchitektur für die Implementierung von Generatorsystemen sowie ein infrastrukturelles Rahmenwerk erleichtert und „verschlangt“ den Bau von Generatoren. Es wird zwischen Basisfunktionen (domänenübergreifend) und domänenspezifischen Funktionen unterschieden. Die Basisfunktionen, wie z. B. die Funktion zum Einlesen von Modellen aus standardisierten Austauschformaten, die Versionsverwaltung von Modellen usw., werden zentral in einem Kernsystem zur Verfügung gestellt. Auf der Grundlage des zentralen Systems können Generatoren für unterschiedliche Zieldomänen entwickelt und in das Kernsystem integriert werden. Die Referenzarchitektur spezifiziert vier weitere domänenspezifische Funktionen, die ein Generator bereitstellen muss, nämlich die Validierung, die die Korrektheit einer Problembeschreibung sicherstellt, die Code-Generierung, die Anwendungen aus den Modellen generiert, die Konflikterkennung für die modellbasierte Softwareaktualisierung und eine Modellierungsumgebung, in der DSL-Modelle erstellt werden können. Das zentrale System stellt eine Reihe von vorgefertigten Baukästen zur Verfügung, die an eine Zieldomäne angepasst werden können. Außerdem wird ein systematisches Vorgehensmodell beschrieben.

Der vorgestellte Ansatz wurde technisch umgesetzt und befindet sich seit mehreren Jahren im praktischen Einsatz. Anhand einer Fallstudie werden die praktische Relevanz sowie die Tauglichkeit des vorgestellten Lösungsansatzes evaluiert. Auf die Erweiterung des Funktionsumfangs des zentralen Systems und die Unterstützung für BPMN-Modelle bei der Anwendungsmodellierung, -generierung sowie -aktualisierung wird im Ausblick hingewiesen.