

Fallstudien über den Einsatz des Enterprise Architect

Anwendungsfall 4: Enterprise Architect für die Cloud-basierte Modernisierung von Legacy Applikationen

Hintergrund. Moderne Cloud Computing Technologien, welche in den letzten Jahren entwickelt worden sind, bescheren sich einem stetigem Zuwachs an Nachfrage aufgrund inhärenter Vorteile, wie beispielsweise theoretisch-unbegrenzter Skalierbarkeit und der Bereitstellung von Software als Service. Mittels Cloud Computing werden beispielsweise hohe initiale Kosten, welche bei der Bereitstellung von eigenen Servern entstehen, vermieden und durch Operationskosten auf bestehender hoch-optimierter Infrastruktur ersetzt, welche zusätzlich durch dynamischen Zukauf und Freigabe von Ressourcen skaliert werden kann. Cloud Ressourcen, welche von Cloud Providern, wie beispielsweise Amazon, Google oder Microsoft angeboten werden, reichen von maschinenorientierten Ressourcen, wie einzelne Rechenknoten, bis hin zu vollständigen Laufzeitumgebungen auf welchen Artefakte von modernen Programmiersprachen, wie Java und .NET, direkt eingesetzt werden können. Da sich jedoch traditionelle Software von Cloud-basierter Software in einigen Aspekten, wie dynamischer Skalierbarkeit, stark unterscheidet, werden involvierte Akteure durch die Migration von traditioneller Software, als auch die Entwicklung neuer Cloud-basierter Software, vor entscheidenden Herausforderungen gestellt.

Ansatz. Im FP7 EU-Forschungsprojekt „ARTIST“ wurden Herausforderungen der Migration von traditioneller Software auf Cloud-basierte Software untersucht und Lösungen entwickelt diese zu überbrücken. In dem ARTIST-Ansatz kommen automatisierte modellgetriebene Techniken zum Einsatz, welche bestehende Software einem Prozess unterzieht der Abhängigkeiten von traditionellen Technologien überbrückt. Im Anschluss werden Design-Entscheidungen in einer Art und Weise dargestellt, welche es ermöglicht Software unabhängig von Cloud-Providern auszudrücken, sodass der „Lock-in-Effekt“ – hier: die Bindung an einen spezifischen Cloud-Provider – vermieden werden kann.

Umsetzung. Um dies zu ermöglichen wurde in ARTIST die Cloud-Modellierungssprache „Cloud Application Modeling Language (CAML)“ entwickelt, welche es einerseits erlaubt Design-Entscheidungen auf plattformunabhängiger Ebene darzustellen und andererseits Cloud-Provider spezifische-Implementationen durch den Einsatz von modellgetriebenen Technologien automatisiert zu generieren.

Vorteile. Da CAML mittels UML-Profilen sowie UML-basierten Cloud-Bibliotheken realisiert ist, kann es in unterschiedlichsten UML-Werkzeugen, wie beispielsweise SparxSystems Enterprise Architect (EA), eingesetzt werden. Da CAML-Konzepte bereits auf den Cloud-Applikationsmodellierungsstandard TOSCA (welcher während der Entwicklung von CAML entstanden ist) abgebildet worden sind, ist es möglich bestehende CAML Modelle automatisiert in TOSCA-konforme Artefakte zu transformieren welche im Anschluss mittels TOSCA-konforme Werkzeuge weiterbearbeitet sowie auf Cloud-Plattformen ausgerollt und verwaltet werden können. EA ermöglicht die Generierung von TOSCA-konformen Abbildungen aus CAML Modellen.

Weitere Informationen: <http://www.artist-project.eu>

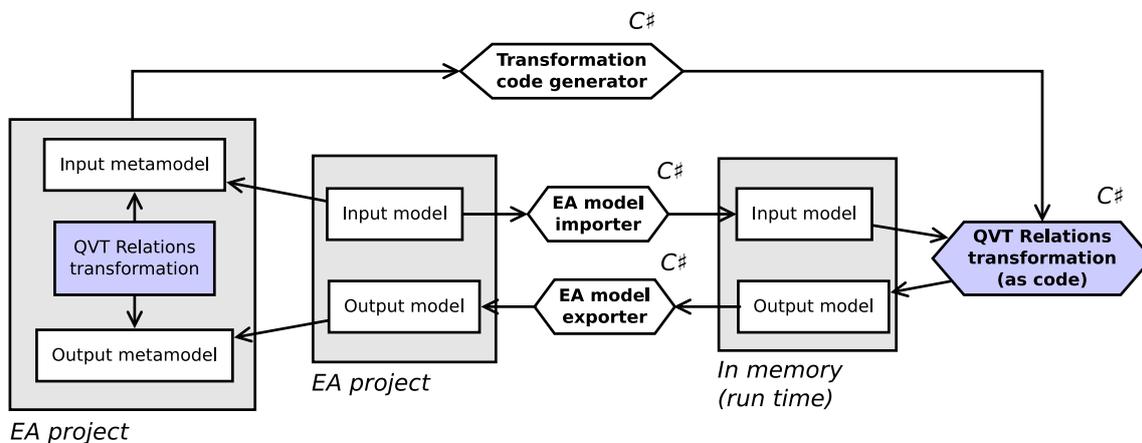
Anwendungsfall 5: Enterprise Architect in der Forschung über Modelltransformationen

QVT Relations Code Generator für Enterprise Architect

Grundlage. Die deklarative Modelltransformationssprache QVT Relations ist ein Teil des QVT Standards der OMG. Eine QVT Relations Modelltransformation besteht aus einem Set von deklarativen Relationen zwischen Metamodellen. Einerseits kann die Transformation zur Überprüfung, ob zwei existierende Modelle die definierten Relationen einhalten, verwendet werden. Andererseits kann sie dazu genutzt werden ein Eingabemodell in ein Ausgabemodell so zu transformieren, dass alle Relationen erfüllt sind. Durch die Nutzung einer deklarative Sprache für Modelltransformationen liegt der Fokus auf dem erwarteten Resultat und nicht notwendigerweise auf der Implementierung. Dies führt zu sehr präzisen Transformationen.

Es gibt jedoch noch immer sehr wenige QVT Relations Implementierungen und die meisten Umsetzungen in der Enterprise Architect Modellierungsumgebung sind nicht effizient. Allgemein erreicht ein deklarativer im Gegensatz zu einem imperativen Lösungsansatz oft nur schwer eine gute Performance, da hierfür mehr Logik in der Transformationsausführungseingine erforderlich ist.

Lösungsansatz. Um dieses Problem zu lösen stützt sich unser Ansatz auf die Codegenerierung: C# Code wird automatisch aus dem QVT Relations Modell, welches im Enterprise Architect definiert wurde, erzeugt. Zur Laufzeit verwendet das generierte C# Programm aus Enterprise Architect importierte Modelle und produziert dazu Ausgabemodelle, d.h. die Resultate der Transformation.



Bei der Implementierung werden zwei Sprachen genutzt: C# für die hauptsächliche Logik und T4 für die Codegenerierungstemplates. Darüber hinaus wird das .NET Modeling Modeling Framework (NMF) verwendet um das offizielle QVT Relations Metamodell in C# benutzen zu können.

Nutzen. Der Hauptvorteil dieses Ansatzes ist, dass die Transformationen in EA graphisch spezifiziert und in jeder .NET basierten Umgebung verwendet werden können, sobald der Code generiert wurde. Außerdem ist der generierte Code signifikant schneller als die Verwendung von QVT Interpreter, da die Logik und die Bedingungen hauptsächlich statisch aufgelöst und in einen einfacheren imperativen Code synthetisiert werden.

Mehr Informationen: http://www.sysml4industry.org/?page_id=635



Die Business Informatics Group (BIG) ist eine Forschungsgruppe des Instituts für Softwaretechnik und Interaktive Systeme an der Technischen Universität

Wien. Wie aus dem Namen ersichtlich, liegt der Forschungsschwerpunkt dieser Gruppe im Bereich der Wirtschaftsinformatik, welche Theorie und Methodik von Computerwissenschaften und Wirtschaftswissenschaften vereint. Mehr Informationen finden Sie unter www.big.tuwien.ac.at



Sparx Systems Pty. wurde 1996 in Australien gegründet und ist Hersteller von Enterprise Architect, einer weltweit erfolgreichen UML Modellierungsplattform mit über 360.000 Anwendern.

Enterprise Architect in der aktuellen Version 13 wird für seine Leistungsfähigkeit zu einem hervorragenden Preis-Leistungs-Verhältnis geschätzt. Enterprise Architect unterstützt unter anderem UML 2.5, BPMN 2.0, SOMF 2.1 SysML 1.4, DDS und TOGAF. Die umfassende Integration in andere Entwicklungsumgebungen wie Visual Studio, TFS sowie Eclipse und die Anbindung an Versionierungssysteme machen es zum idealen Bindeglied in jeder Prozesslandschaft. Um den zahlreichen Kunden in ihrer Sprache und Zeitzone bestes Service rund um Enterprise Architect bieten zu können, wurde 2004 die SparxSystems Software GmbH Central Europe gegründet, die die gesamte deutschsprachige Region beim Lizenzerwerb sowie durch Training und Consulting unterstützt. Mehr Informationen finden Sie unter www.sparxsystems.de