

Gut geregelt

Neue und bewährte Einsatzorte für Regeln in Softwarearchitekturen

Caroline Buck

Beim Einsatz von Business Rules Management (BRM)-Systemen werden vermehrt neben den namensgebenden Geschäftsregeln auch Regeln für Workflow-, Präsentations- und Integrationslogik abgebildet. So wird das Verhalten von Anwendungen ganz oder in großen Teilen durch Regeln bestimmt. Dieser Artikel beschreibt, an welchen Stellen in Architekturkonzepten ein regelbasierter Ansatz sehr gut und wo herkömmliche Programmierung weiterhin besser passt.

Zu den Grundüberlegungen für die Konzeption von Systemarchitekturen und für die Auswahl der passenden Vorgehensweise gehören eine ganze Reihe zentraler Fragen: Welche Entwurfsmuster und welche Frameworks werden verwendet? Wie und wo soll die Anwendung dem Benutzer präsentiert werden? Welche bestehenden Datenbankstrukturen können genutzt oder welche müssen neu aufgebaut werden? Wie sollen die Daten in der Anwendung persistiert werden? Welche Altsysteme dienen als Datenquelle, welche als Zielsystem? Und so weiter.

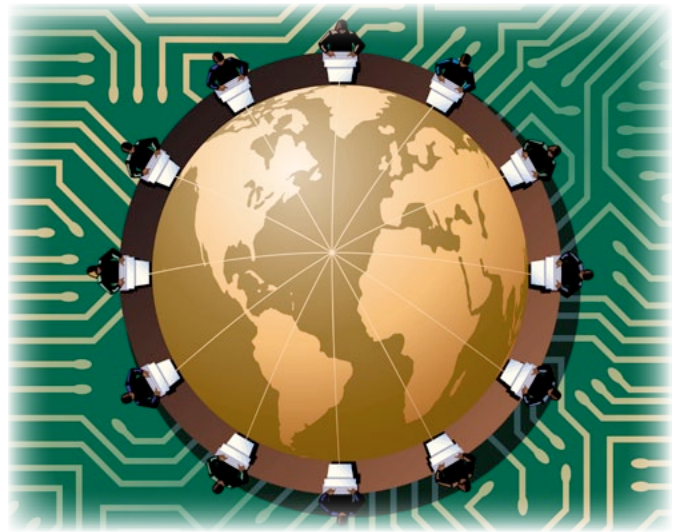
Die Herausforderung ist immer, ein unter Kosten-Nutzen-Aspekten gesehen zukunftssicheres Architekturkonzept zu entwerfen. Dafür gibt es selbstverständlich kein Patentrezept. Anhand gesammelter Erfahrungen aus Projekten in unterschiedlichen Branchen lässt sich aber illustrieren, an welchen Stellen in der Softwarearchitektur und im Entwicklungsprozess für Geschäftsanwendungen Regelsysteme operative Vorteile liefern.

Moderne BRM-Systeme umfassen Regelartefakte für die Definition von Fachlogik, Abläufen und Ereignissen, für Service-Aufrufe zur Anbindung der Systemumwelt sowie für die Steuerung und die Gestaltung von Benutzeroberflächen. In Kombination mit dynamischen Frameworks für Unternehmensanwendungen können theoretisch sämtliche Anwendungsteile in Form von Regeln definiert oder modelliert werden, ohne eine einzige Zeile Programmcode zu schreiben. Wie steht es damit in der Praxis?

Am Anfang steht Struktur

Im Zentrum einer Geschäftsanwendung befindet sich der fachliche Ablauf, der schlussendlich unterstützt werden soll. Das Wissen über diesen Prozess liegt originär in den Fachbereichen und stellt deren Kernkompetenz dar. Die Festlegung einer ersten, groben Struktur, in der die zentralen Prozessschritte definiert sind, bildet den Ausgangspunkt für alle weiteren Schritte und muss notwendigerweise in Abstimmung mit den Fachexperten erfolgen.

Zur Zusammenarbeit eignen sich grafische Werkzeuge hervorragend, denn selbst Programmierlaien können damit schnell ein Verständnis für die abgebildete Logik entwickeln. Bietet das verwendete Werkzeug einen Entwurfsmodus, können die grundlegenden Schritte schnell in einer Ideenskizze festgehalten (Neudeutsch geskribbelt) werden, ohne zuvor eine umfangreiche domänenspezifische Sprache definieren zu müssen.



Basierend auf einer ersten Struktur (s. Abb. 1) kann eingeplant werden, für welche Schritte die Anbindung an andere Systeme oder eine Benutzerinteraktion notwendig wird und an welchen Stellen die Regeln für die einzelnen Prozessschritte weiter verfeinert werden müssen. Dieses Vorgehen hat sich als Projektstart bewährt und wird in Anwendungsentwicklungsprojekten als erster Schritt nach den Schulungsmaßnahmen durchgeführt.

Alles bleibt anders

Offen für Änderungen zu sein, ist keine esoterische Philosophie, sondern eine notwendige Strategie für die Entwicklung agiler Applikationen, die sich einfach an ständig neue Anforderungen anpassen lassen. Sind viele Anforderungen an eine Applikation nicht von vornherein bekannt und muss die implementierte Logik laufend an veränderte Marktbedingungen angepasst werden, schreit das förmlich nach dem Einsatz eines Regelsystems.

Unterstützt das Regelsystem eine testgetriebene, evolutionäre Vorgehensweise, werden die Tests im Idealfall nach jeder Änderung automatisiert ausgeführt. Möglichst kleine Regeleinheiten werden parallel zu den korrespondierenden Unit-Tests entwickelt (s. Abb. 2) und – wo nötig – wird die Integrationslogik via Service-Calls, die eventuell nur zur

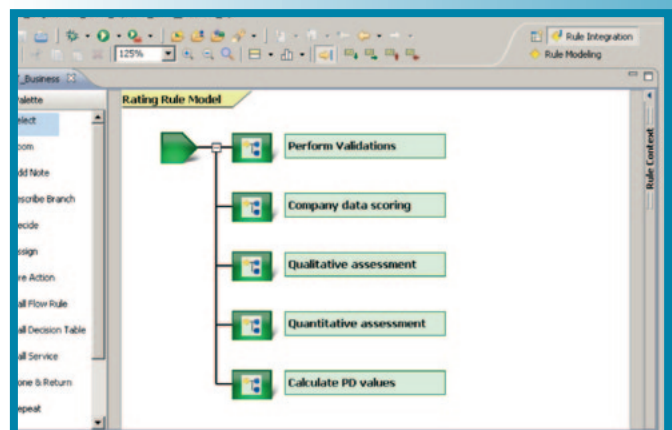


Abb. 1: Beispiel für eine grafisch modellierte Strukturregel

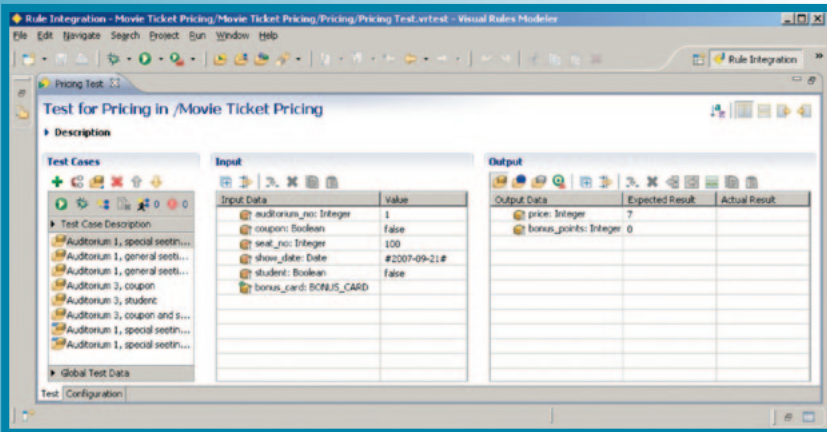


Abb. 2: Beispiel für einen Regeltest

Laufzeit zur Verfügung stehen, in Mock-Objekten gekapselt. So kann eine hohe Testabdeckung erreicht werden und die Ergebnisse, die von der Regellogik geliefert werden sollen, können gleichzeitig spezifiziert und dokumentiert werden.

Präsentation ist alles

Änderungen an der zugrunde liegenden Fachlogik wirken sich in vielen Fällen direkt auf die Benutzeroberfläche aus; hier wird ein neuer Eintrag in einem Listenfeld benötigt, da eine weitere Checkbox usw. Selbst wenn der Fachbereich in der Lage ist, seine Fachlogik alleine zu ändern, braucht es für die Anpassung der Oberflächen dann doch wieder die Unterstützung der IT. In den meisten Fällen ist die dynamische Programmierung von Benutzeroberflächen ein komplexes Thema, das sich nur schwer mit generischen Ansätzen lösen lässt.

Der Einsatz regelbasierter Frameworks kann eine interessante Alternative zur herkömmlichen Oberflächenprogrammierung sein, wenn sich die Oberfläche aus einer Abfolge standardisierbarer Dialoge zusammensetzt. Dabei wird die Fach-, Präsentations- und Ablauflogik mit dem Regelsystem definiert bzw. modelliert. Im Beispiel aus Abbildung 3 übersetzt ein HTML-Renderer, der Bestandteil solcher Frameworks ist, zur Laufzeit die Regeln in Webformulare. Deren Abfolge wird dann wieder über Regeln gesteuert, unter Auswertung

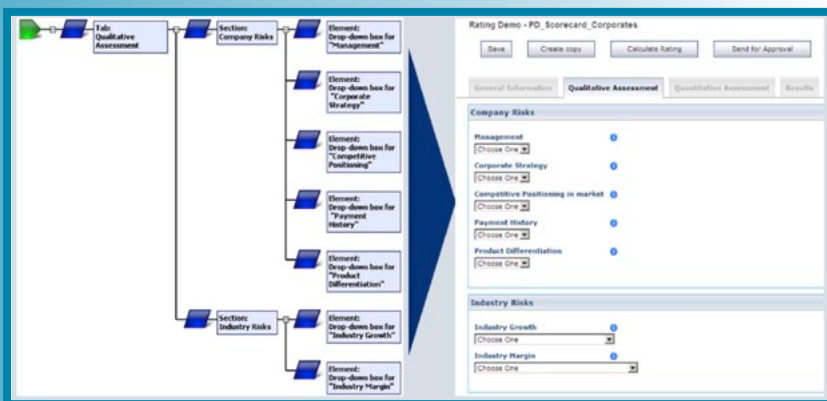


Abb. 3: Regelbasiert erzeugte Benutzeroberfläche

der tatsächlichen Formulareingaben und der definierten Ablaufregeln.

Ein Anwendungsbeispiel für eine vollständig regelgesteuerte Oberfläche ist der Wizard für einen komplexen Kontoeröffnungsprozess eines Banksystems: Im Wartungsfall zeigt sich immer wieder, dass die modellgestützte Synchronisierung der Oberflächen mit der zugrunde liegenden Fach- und Ablauflogik Anpassungen immens erleichtert.

Migration mal anders

Ein Verwendungszweck für regelbasierte Werkzeuge ist die Datenmigration, mit der sehr große Datenmengen zum Beispiel von einem Altsystem in die Datenbank einer modernen betrieblichen Standardsoftware überführt werden.

Für den Datentransfer aus dem Altsystem und die Implementierung der neuen Prozesse ist das Wissen der Schlüsselanwender aus dem Fachbereich entscheidend, denn vor allem sie können wertvolle Hinweise zum Migrationsprozess liefern und beurteilen, ob die migrierten Daten korrekt sind. Durch die direkte Auswirkung der Datenmigration auf den operativen Betrieb des Unternehmens ist es wichtig, die Ausfallzeiten so gering wie möglich zu halten. Erreicht wird dieses Ziel durch umfassende Tests und Probemigrationen in der extensiven Projektphase, die der eigentlichen Migration vorausgeht [Hal08].

Regelsysteme unterstützen bei dieser Herausforderung durch die schon angesprochene vereinfachte Zusammenarbeit von IT und Fachbereich sowie durch die Testfunktionalitäten. In diesem Anwendungsszenario ist es essenziell, dass das verwendete Regelsystem Funktionen zur Datenbankanbindung und Transaktionssteuerung sowie den Zugriff auf Metadaten und Stored Procedures bietet. Zentrale Aufgaben wie das Mapping der Attribute und der Definition der Transformationsregeln können im Regelsystem von Schlüsselanwendern und IT-Experten der Quell- und Zielsysteme gemeinsam erarbeitet werden. Die Ergebnisdokumente brauchen nicht extra erzeugt zu werden, sondern entstehen mit den Regeln (s. Abb. 4).

Grenzwertig

Neben den aufgezeigten Einsatzszenarien gibt es natürlich auch Grenzen der Regeltechnologien im Umfeld von Geschäftsanwendungen. Grafisch modellierte Geschäftsregeln eignen sich zum Beispiel kaum für Optimierungs- und Vorhersageverfahren, bei denen mit unbekanntem Parametern gearbeitet wird. Anzutreffen sind solche klassischen Optimierungsprobleme beispielsweise im Operations Research. Diese Art von Problemstellung wird zumeist durch mathematische Algorithmen gelöst, die softwaregestützt modelliert oder ausprogrammiert werden und für die eine Vielzahl Java-basierter kommerzieller und Open-Source-Systeme zur Verfügung steht [Informs].

Für bereits implementierte und störungsfrei arbeitende Algorithmen, wie z. B. hocheffiziente Datenbankoperationen, ist der wirtschaft-

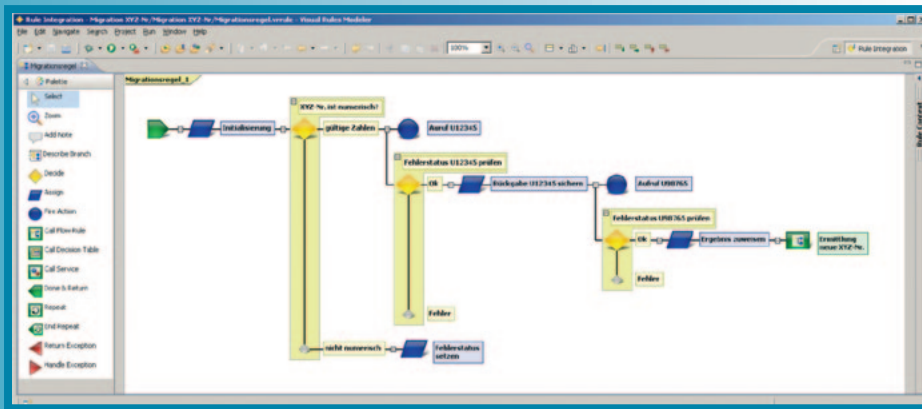


Abb. 4: Beispiel für eine Migrationsregel

[JBR] Open Source Business Rules Management System,
<http://www.jboss.com/products/platforms/brms/>
 [BuSchä09] C. Buck, M. Schärtel, Architekturkonzepte mit Business Rules, in: JavaSPEKTRUM, 5/2009
 [VR4] Kommerzielle Business Rules Enterprise Plattform,
<http://www.visual-rules.de>

liche Nutzen einer Neuimplementierung in Form von Regeln abzuwägen. Solche Algorithmen können bei Bedarf als externalisierte Logik in die Regelwelt integriert werden. Ebenso wird man es nicht schaffen, Vorteile durch die Nachbildung der Konfigurationslogik aus den XML-Dateien anderer Java-Frameworks wie Spring, JSF, Struts oder Hibernate zu generieren. Diese Form von Komplexität kann mit regelbasierten Ansätzen nicht aufgelöst werden.

Fazit

Regelbasierte Ansätze und Frameworks sind kein Sturm des Fachbereichs auf die Entwickler-Bastille. Im Gegenteil: Sie erlauben allen Projektbeteiligten, sich auf ihre Kernkompetenzen zu konzentrieren. Ihr Einsatz beschränkt sich nicht mehr nur auf Fachlogik: Die meisten Systemhersteller haben Möglichkeiten entwickelt, Ablauflogik zu implementieren, und arbeiten an Erweiterungen hinsichtlich Ereignisverarbeitung. Die Definition der Präsentationslogik mit Regelsystemen ist recht exotisch, hat sich aber in der Praxis da bewährt, wo die Integrationsgeschwindigkeit neuer Parameter in die Fachlogik und die Benutzeroberflächen eine große Rolle spielt und wo ähnliche Dialoge in unterschiedlichen Ausprägungen von verschiedenen Projekten implementiert werden müssen.

Dazu kommt: Wo Programmcode und Dokumentation klassischer Anwendungen mit der Zeit unvermeidlich auseinander laufen, wird die Wartung zu einer sportlichen Herausforderung. Das gilt für zehn Jahre alten Java-Code genauso wie für sagenumwobene Hostanwendungen, deren Programmierer längst ihren Ruhestand genießen. Regelbasiert implementierte Systeme sind wartungsfreundlicher, weil die abgebildete Logik sich quasi selbst dokumentiert.

Kaum zu glauben? Gekontert mit den Worten einer schwedischen Möbelkette: Entdecke die Möglichkeiten, entweder durch die Evaluierung kommerzieller Produkte wie Visual Rules [VR4] oder von Open-Source-Software wie JBoss Rules [JBR].

Literatur und Links

[Hal08] K. Haller, Datenmigration bei Standardsoftwareeinführungsprojekten, DatenbankSpektrum, 25/2008, S. 39 ff.
 [Informs] Übersicht Java-basierter OR-Ressourcen,
http://www2.informs.org/Resources/Computer_Programs/Java_Programs/

Der Einsatz von Regelsystemen empfiehlt sich:

- ▼ wenn für die Entwicklung der Applikation die *Zusammenarbeit mit dem Fachbereich* intensiviert werden soll oder muss. Warum? Die abgebildete Logik ist deutlich transparenter und einfacher zu verstehen als Programmcode. Tatsächlich sind Fachexperten mit diesen Systemen in der Lage, Teile der Anwendung selbst zu entwickeln oder zu pflegen – auch ohne Programmierkenntnisse.
- ▼ wenn für den Entwicklungs- und Lebenszyklus der Anwendung *kontinuierliche Änderungen* an der Logik erwartet werden. Warum? Regelsysteme sind gerade für dieses Szenario gebaut: Sie bieten intensive Unterstützung darin, den Änderungsaufwand zu beherrschen und neben den vielen Anpassungen die Qualität der Ergebnisse bzw. der Software sicherzustellen.
- ▼ wenn für einen Geschäftsprozess eine *komplexe Abfolge von Entscheidungen* implementiert werden soll. Warum? Regelbasierte, prozessorientierte Frameworks ermöglichen die Modellierung von Fach-, Präsentations- und Ablauflogik in einem Werkzeug und sorgen für ein reibungsloses Zusammenspiel.
- ▼ wenn *komplexe Datenmigrationen* geplant sind, selbst mit einmaligem Charakter. Warum? In Form von Regeln oder Modellen abgebildete Extraktions-, Transformations- und Ladelogik ermöglicht nachvollziehbare Migrationsprozesse, die Logikänderungen bis zur letzten Minute zulassen.



Caroline Buck ist Marketing Manager für Technology Solutions bei der Innovations Software Technology. Nach vielen Jahren Erfahrung in der Entwicklung von Enterprise-Java-Anwendungen ist sie heute verantwortlich für das Technologiemarketing rund um die Innovations-Regeltechnologie und darauf aufsetzender Plattformen.
 E-Mail: caroline.buck@innovations.de