



GEBEN SIE IHRER GESCHÄFTSLOGIK EIN FRAMEWORK

Durch den Einsatz eines Business-Rule-Managementsystems (BRMS) können Verwaltungen Richtlinien, Ausführungs- und Verwaltungsvorschriften effizient in Anwendungen integrieren und spätere Änderungen schnell umsetzen.

| von DENNIS HUNING

Die Geschäftslogik ist nach den Daten der wichtigste Teil einer Verwaltungsanwendung. Sie besteht zumeist aus einer Vielzahl ganz unterschiedlicher Regeln. Das Spektrum ist breit und erstreckt sich beispielsweise von Regeln zum Validieren der Eingabe über solche, die den Zugriff auf Daten steuern und so sicherstellen, dass nur autorisierte Sachbearbeiter einen Datensatz einsehen können, bis hin zu solchen, die auf Basis der Eingabe die Gebühr für die Bearbeitung berechnen.

Eines haben diese Regeln heutzutage häufig gemeinsam: Sie sind in der gesamten Anwendung verteilt, im Programmcode verborgen und daher bei Anpassungen schwer zu finden.

DREISCHICHTENMODELL

Wird ein Softwaresystem vereinfacht betrachtet, teilt es sich in drei Schichten auf:

- Präsentationsschicht
- Logikschicht
- Datenhaltungsschicht

Die **Präsentationsschicht** bereitet die Daten auf und präsentiert sie dem Sachbearbeiter. In der **Logikschicht** wird der größte Teil der Geschäftslogik umgesetzt. In dieser Schicht werden Abläufe gesteuert und Berechnungen durchgeführt. Und in der **Datenhaltungsschicht** erfolgt das Laden und Speichern von Daten. Für die Präsentationsschicht und die Datenhaltungsschicht werden häufig Frameworks eingesetzt, die Standardfunktionen anbieten und komplexe technische Abläufe abstrahieren. In der Logikschicht hingegen wird mit Hinweis auf die Individualität der jeweiligen Verwaltungsabläufe und -regeln auf den Einsatz von Frameworks verzichtet. Dieses Vorgehen führt immer wieder dazu, dass in dieser Schicht sehr komplexe Entscheidungen in „If-else“-Konstrukten abgebildet werden

(siehe Beispiel 1). Im Listing aus Beispiel 1 wird ein Teil der Verwaltungsvorschrift zur Berechnung der Kfz-Steuer umgesetzt. Dieser Teil berechnet für Personenkraftwagen, die älter als 30 Jahre sind oder vor dem 4. November 2008 zugelassen wurden, die Kfz-Steuer.

```
if(kfz instanceof PKW){
    if(aelterAls30Jahre(kfz.getErstzulassung())){
        kfz.setSteuer(191.73)
    }else if (datumBis2008Nov04(kfz.getErstzulassung())){
        if(kfz.getEmissionsgruppe().equals(Emissionsgruppe.
            Euro4)){
            if(kfz.getMotorart().equals(Motorart.Diesel)){
                kfz.setSteuer(kfz.getHubraum()/100*15.44);
            }else if(kfz.getMotorart().equals(Motorart.Diesel){
                kfz.setSteuer(kfz.getHubraum()/100*6.75);
            }
        }else if(kfz.getEmissionsgruppe().equals
            (Emissionsgruppe.Euro3){
        ...
    }
}
```

Beispiel 1: „If-else“-Konstrukt in Pseudocode

Neben diesen eher unübersichtlichen „If-else“-Konstrukten ist die Geschäftslogik nicht an zentraler Stelle in der Logikschicht zu finden. Ganz im Gegenteil: Eingabevalidierungen finden meistens direkt in der Präsentationsschicht statt, der Datenzugriff wird in der Datenhaltungsschicht gesteuert.

In vielen Systemen ist die Logik in ähnlicher Form in verschiedenen Schichten mehrfach vorhanden. Änderungen an der Geschäftslogik führen nicht selten dazu, dass Anpassungen in allen Schichten der Anwendung durchgeführt werden müssen. Auch Zusammenhänge und Abhängigkeiten zwischen den Regeln verschiedener Geschäftsobjekte werden bei solchen Änderungen häufig übersehen. Doch eigentlich will man in einer modernen IT so agil sein, dass Änderungen mit möglichst wenig Aufwand, idealerweise zentral an einer Stelle, umgesetzt werden können.

WAS IST EIN BRMS?

Die Lösung für dieses Problem ist ein Framework, das Möglichkeiten bietet, die Geschäftslogik zentral und intelligent zu verwalten. Ein sogenanntes BRMS – ein Business Rules Management System – ist ein solches Framework. Es besteht im

Wesentlichen aus drei Komponenten:

- dem Regel-Repository zum Auslagern und Versionieren der Geschäftslogik,
- Werkzeugen zum Erzeugen und Bearbeiten der Geschäftslogik,
- der Business Rules Engine (BRE) zum Auswerten der Geschäftslogik.

Wird ein BRMS eingesetzt, wird zunächst die Geschäftslogik deklarativ in Form von „Wenn-dann“-Regeln beschrieben. Wenn also das Listing aus Beispiel 1 für ein BRMS umgeschrieben wird, sehen die Regeln in etwa so aus, wie im Listing in Beispiel 2 dargestellt:

```
wenn
    pkw:PKW( erstzulassung < „2008-Nov-04“
        und kfz.motorart == Motorart.Diesel
        und kfz.emissionsgruppe = Emissionsgruppe.Euro4)
dann
    pkw.steuer = kfz.hubraum / 100 * 15.44
wenn
    pkw:PKW( erstzulassung < „2008-Nov-04“
        und kfz.motorart == Motorart.Ottomotor
        und kfz.emissionsgruppe = Emissionsgruppe.Euro4)
dann
    pkw.steuer = kfz.hubraum / 100 * 15.44
```

Beispiel 2: Code aus Listing 1 neu geschrieben als deklarative Regeln

Diese Regeln werden direkt im BRMS spezifiziert. Da sich damit sämtliche Geschäftsregeln an einer zentralen Stelle befinden, lassen sie sich auch leichter lokalisieren, analysieren und pflegen. Und wenn sich nun die Anforderungen an die Geschäftslogik ändern, muss nicht mehr die ganze Anwendung untersucht und geändert werden. Mit einem BRMS können Änderungen zielgenau an einer Stelle durchgeführt werden. An denjenigen Stellen im Programmcode, an denen in Systemen ohne BRMS Teile der Geschäftslogik implementiert waren, finden mit BRMS nur noch Aufrufe der BRE statt.

```
ruleEngine.berechneKfzSteuer(kfz)
```

Die korrekte Kfz-Steuer wird von nun an innerhalb des BRMS durch das Auswerten der Regeln berechnet.

ANBIETER VON BRMS

Am Markt gibt es viele Anbieter von BRMS. Sie alle bieten eine eigene Variation an, wie Regeln erfasst und verwaltet werden.

So lassen sich bei einigen Anbietern die Regeln mithilfe von geführten Editoren (Wizards) erstellen. Andere wiederum bieten die Möglichkeit, Regeln in Excel-Tabellen als sogenannte Entscheidungstabelle zu definieren.

Aber unabhängig von der Art und Weise, wie sie die Regeln definieren, realisieren alle Anbieter das gleiche Grundschema: Eine Regel besteht aus einer oder mehreren Prämissen (WENN-Teil). Wenn diese Prämissen erfüllt sind, folgt die Schlüsselfolgerung (DANN-Teil). In der Schlüsselfolgerung werden eine oder mehrere Aktionen ausgeführt.

DAS BRMS DROOLS

Im Java-Bereich gehört Drools – ein Open-Source-Projekt der JBoss-Community – zu den bekanntesten und am meisten eingesetzten Produkten. Drools bietet mehrere Möglichkeiten zur Regelerstellung an, so zum Beispiel die Möglichkeit, Regeln in der Drools Rules Language (DRL) zu definieren (siehe Beispiel 3).

```
rule "REGEL 1: Oldtimer?"
  when
    pkw : PKW(alter >= 30 )
  then
    pkw.setSteuer(191.73);
  end
```

Beispiel 3: Regeln in Drools Rules Language

Im Gegensatz zum Listing aus Beispiel 1 ist eine solche Regel schon sehr gut lesbar und verständlich. Um die Verständlichkeit noch weiter zu verbessern, können Textbausteine definiert wer-

den. Mithilfe dieser Bausteine kann eine Organisation auch eine eigene domänenspezifische Sprache (DSL) entwickeln, die die spezifischen regelnden Sachverhalte, zum Beispiel einer Verwaltungsbehörde, abbildet. Ein Beispiel für solche Sachverhalte ist in Beispiel 4 gezeigt. Diese Textbausteine ersetzen die DRL durch eine für den Fachbereich verständliche Sprache. Die Regel in Listing 5 wurde um die DSL erweitert. Was die Regel berechnet, kann jeder Anwender direkt ablesen.

```
[condition][]PKW ist älter als {alter} Jahre=pkw : PKW
({alter} >= 30 )
[consequence][]Lege Kfz-Steuer auf {betrag} fest=pkw.
setSteuer({betrag});
```

Beispiel 4: Textbausteine zum Erstellen einer eigenen domänenspezifischen Sprache

```
rule "REGEL 1 mit DSL: Oldtimer?"
  when
    PKW ist älter als 30 Jahre
  then
    Lege Kfz-Steuer auf 191,73 fest
  end
```

Beispiel 5: Regel mit DSL-Elementen

Bei BRMS handelt es sich um Frameworks, die potenziell sehr generisch sehr komplexe Sachverhalte abstrahieren und – vor dem Entwickler verborgen – in Programmcode umsetzen. Daher liegt die Vermutung nahe, dass der Einsatz eines BRMS das Laufzeitverhalten von Anwendungen negativ beeinflussen könnte. Die Praxis zeigt jedoch, dass das Gegenteil der Fall ist. Anders als die Entwickler im Code hat das BRMS – ähnlich wie bei

| | Rule Table | Berechnung Kfz-Steuer PKW | | | |
|---|-----------------|---------------------------|-----------------|-------------------|------------------------|
| Erläuterung | Regelname | Motorart | Emissionsgruppe | Erstzulassung bis | Steuer pro 100ccm in € |
| Berechnet die Steuer für einen PKW mit einer Erstzulassung bis zum 04.11.2008, mit der Motorart Diesel und Emissionsgruppe Euro4 | Diesel_Euro4 | Diesel | Euro4 | 04-Nov-2008 | 15,44 |
| Berechnet die Steuer für einen PKW mit einer Erstzulassung bis zum 04.11.2008, mit der Motorart Ottomotor und Emissionsgruppe Euro4 | Ottomotor_Euro4 | Ottomotor | Euro4 | 04-Nov-2008 | 6,75 |
| Berechnet die Steuer für einen PKW mit einer Erstzulassung bis zum 04.11.2008, mit der Motorart Diesel und Emissionsgruppe Euro2 | Diesel_Euro2 | Diesel | Euro2 | 04-Nov-2008 | 16,05 |
| Berechnet die Steuer für einen PKW mit einer Erstzulassung bis zum 04.11.2008, mit der Motorart Ottomotor und Emissionsgruppe Euro2 | Ottomotor_Euro2 | Ottomotor | Euro2 | 04-Nov-2008 | 7,36 |

Beispiel 6: Entscheidungstabelle

modernen Datenbanksystemen – die Möglichkeit, sämtliche Zusammenhänge in der Geschäftslogik zu strukturieren und diese zu optimieren. Der Algorithmus, der hierfür eingesetzt wird, heißt Rete-Algorithmus. Drools verwendet diesen Algorithmus in einer sehr gelungenen und effizienten Weise und kann so direkt mit dem fachlichen Datenmodell einer Fachanwendung arbeiten. In den Beispielen wird dies durch die Verwendung des fachlichen Objektes PKW verdeutlicht.

ENTSCHEIDUNGSTABELLE

Will man eine große Menge gleichartiger Regeln möglichst übersichtlich darstellen, wird die Darstellung der Regeln in einer Entscheidungstabelle empfohlen. Drools bietet die Möglichkeit, Entscheidungstabellen in Excel-Dateien zu entwerfen (siehe Beispiel 6).

In diesem Beispiel wird jede einzelne Zeile in eine Regel übersetzt. Bei der Erstellung können sämtliche Funktionen von Excel verwendet werden. So wurden hinter den Spalten Motorart und Emissionsgruppe Wertebereiche definiert. Durch diese wird schon während der Erstellung die Eingabe von falschen Werten verhindert. Im Beispiel fällt direkt auf, dass verschiedene Farben eingesetzt wurden. Jede Farbe hat eine andere Bedeutung. So werden

- fachliche Beschreibungen hellblau,
- technische Details dunkelgrau,
- Aktionen rot,
- Tests auf Gleichheit blau
- und arithmetische Vergleiche hellgrau

dargestellt. Die Farbcodierung der einzelnen Spalten unterstützt den Benutzer dabei, die Regeln zu verstehen, auch wenn er die technischen Details nicht kennt.

Die unterschiedlichen Farben zeigen, wie eine Bedingung (WENN-Teil) ausgewertet wird und welche Zellen einer Spalte den Aktionsteil (DANN-Teil) enthalten.

Im Beispiel wird noch eine weitere Excel-Funktion verwendet: das Gruppieren und Ausblenden von Zeilen und Spalten. Die technischen Details, die von Drools interpretiert werden, sind ausgeblendet. Drools benötigt diese, um zu wissen, was eine Spalte aussagt und wie diese zu verwenden ist. Ein fachlicher Benutzer erkennt diese Informationen direkt über die Farben und die Beschreibungen. Da die technischen Details für die Fachexperten überflüssig sind, können sie ausgeblendet werden.

Diese Gestaltung der Entscheidungstabellen in Excel-Dateien bietet ein intuitives Mittel zur Kommunikation mit dem Fachbereich. Spezifikationsfehler können zusammen mit dem Fachbereich in der Geschäftslogik schnell gefunden werden. Versierten Fachanalysten ist es mit diesem Werkzeug möglich, einen Teil der Geschäftslogik direkt in diesen Tabellen zu beschreiben. Der Fachbereich ist im Umgang mit solchen Tabellen bei Weitem geübter als mit einem Quellcode wie aus Beispiel 1.

FAZIT

Die msg hat in vielen Projekten bei namenhaften Unternehmen und in der öffentlichen Verwaltung gezeigt, dass der Einsatz eines BRMS entscheidende Vorteile bietet. Bei der Landeshauptstadt München wurde Drools im Projekt „eAkte“ Waffe zur Validierung der Eingaben genutzt. Mithilfe von Entscheidungstabellen konnten Fehler in der Validierung schnell gefunden werden. Allein in einem der vielfältigen, nach dem Register-Factory-Standard des Bundesverwaltungsamtes durchgeführten Projekte wurden etwa 700 Regeln in Entscheidungstabellen definiert. Regeln werden dort zum Validieren von Nachrichten, zum Prüfen der Berechtigungen und zur Einschränkung der Sichtbarkeit von Informationen benutzt. Erst durch den Einsatz eines BRMS wurde diese große Zahl an Regeln handhabbar. Bei Banken, Versicherungen und Automobilkonzernen werden durch den Einsatz von BRMS Kreditwürdigkeiten bestimmt und Policen berechnet. Außerdem können Kundenportale damit intelligenter und individueller gesteuert werden.

Doch ist nicht in jedem Fall der Einsatz eines BRMS empfohlen. Manchmal wird die Menge der sich ändernden Geschäftsregeln falsch, das heißt zu hoch, eingeschätzt. Wenn sich selten oder nie Änderungen ergeben, lohnt sich der höhere Erstellungsaufwand für den Einsatz eines BRMS nicht. In diesem Fall können die Regeln, wie sonst auch, verteilt über die ganze Anwendung platziert werden. Wenn jedoch mit häufigen Änderungen zu rechnen und der größte Teil der Regeln fachlich motiviert ist, sollte man den Einsatz eines BRMS prüfen. ●

ANSPRECHPARTNER – DENNIS HUNING

IT-Consultant

Public Sector

- +49 89 96101-1296
- dennis.huning@msg-systems.com

