

KOMPAKT  
INFO

# **LOGIK**beherrschung mit Entscheidungstabellen

## 1 Einleitung

Themen wie *Business Rules Management*, *Die Kluft zwischen Fachbereich und IT*, *Die IT leidet unter unklaren Anforderungen*, *Business-IT Alignment*, etc., sind nach wie vor seit Jahren aktuell.

In diesem Kontext fristet eine extrem leistungsstarke Methode zu Unrecht ein Schattendasein: Die Entscheidungstabelle.

Dieses Dokument beleuchtet den Nutzen von Entscheidungstabellen und erläutert, warum gerade diese Methode - richtig angewandt - für eine **durchgängige und sichere Logikbeherrschung in allen Projektphasen** konkurrenzlos leistungsstark ist.

## 2 Geschäftsregeln (engl. Business Rules)

Eine Geschäftsregel legt eindeutig fest, welche Folge von Bedingungen zu welcher Folge von Aktionen führt:

<b>Wenn</b>	Saldo vorhanden
<b>und</b>	Mahnfrist erreicht
<b>und</b>	kein VIP-Kunde
<b>und</b>	Prämienart Bagatellbestandsprämie
<b>und</b>	Vorgangsart Erstprämie
<b>dann</b>	§38-Mahnung
<b>und</b>	Vormerkung Vertragsstorno

Was oft nicht beachtet wird: Eine Geschäftsregel ist immer zwingend logisch verwandt mit zahlreichen anderen Geschäftsregeln, wobei sich die Verwandtschaft direkt aus den verwendeten Bedingungen ergibt.

Hier ein typisches Beispiel für eine **verwandte Regel**:

<b>Wenn</b>	Saldo vorhanden
<b>und</b>	Mahnfrist erreicht
<b>und</b>	kein VIP-Kunde
<b>und</b>	Prämienart Bagatellbestandsprämie
<b>und</b>	Vorgangsart Folgeprämie
<b>dann</b>	§39-Mahnung
<b>und</b>	Vertragskündigung

Damit Entscheidungsprozesse korrekt beschrieben sind, müssen Geschäftsregeln in Gruppen erfasst werden, wobei sichergestellt sein muss, dass dabei alle Kombinationsmöglichkeiten der verwendeten Bedingungen berücksichtigt sind und dies vollständig, redundanz- und widerspruchsfrei.

### 3 Entscheidungstabellen

Die Entscheidungstabelle ist eine sehr kompakte und weltweit genormte (z.B. DIN 66241) Darstellungsform für Gruppen von Geschäftsregeln.

Die beiden verbalen Regel-Beispiele aus dem vorherigen Kapitel entsprechen hier den Entscheidungstabellen-Regeln R04 und R05:

		R01	R02	R03	R04	R05	R06	R07	R08
B01	Saldo?	<0	=0	>0	>0	>0	>0	>0	>0
B02	Mahnfrist erreicht?	-	-	J	J	J	J	J	N
B03	VIP-Kunde?	-	-	J	N	N	N	N	-
B04	Bagatellbestandsprämie?	-	-	-	J	J	N	N	-
B05	Vorgangsart?	-	-	-	EP	FP	EP	FP	-
A01	Keine Aktion	-	X	-	-	-	-	-	X
A02	Protokolleintrag	X	-	X	-	-	-	-	-
A03	Mahnung	-	-	-	§38	§39	§38	§39	-
A04	Vormerkung Vertragsstorno	-	-	-	X	-	-	-	-
A05	Vertragskündigung	-	-	-	-	X	-	-	-

Wichtig bei einer Werkzeugunterstützung ist, dass die hier abgebildete genormte Darstellung unterstützt wird:

■ Im linken oberen Quadrant werden alle Bedingungen aufgelistet (B01 bis B05), im linken unteren Quadrant alle Aktionen (A01 bis A05) und jede Spalte auf der rechten Seite repräsentiert genau eine Geschäftsregel (R01 bis R08).

Diese Darstellungsform ist Voraussetzung für zahlreiche wichtige Funktionen, die erst einen leistungsstarken und sicheren Umgang mit den Regeln ermöglichen:

■ Regeln verdichten, Regeln expandieren, Regeln umsortieren, Bedingungen und Aktionen verschieben, fehlende Regeln automatisch ergänzen, Redundanzen und Widersprüche erkennen, u.v.m.

### 4 Perfekte Logik

Die Verwendung von Entscheidungstabellen hat mehrere Vorteile:

■ Schon bei der Beschreibung von Entscheidungsprozessen kann die logische Qualität der Regelgruppe geprüft werden: Vollständigkeit, Redundanzfreiheit und Widerspruchsfreiheit

■ Die Komplexität der Beschreibungen wird messbar (Anzahl Bedingungen, Anzahl Aktionen, Anzahl Regeln), wodurch auch die Folgeaufwände, z.B. für Realisierung, Test und Abnahme, deutlich sicherer geplant werden können

## **5 Programm-Generierung aus Entscheidungstabellen**

Wenn garantiert werden soll, dass das Systemverhalten zu 100% mit der Beschreibung übereinstimmt, dann gibt es dafür nur einen Weg: Der Programmcode muss direkt aus der Entscheidungstabelle generiert werden.

Spätere Regeländerungen müssen immer zuerst in der Entscheidungstabelle durchgeführt werden, die Synchronisation des Programmcodes an fachliche Änderungen erfolgt wieder durch Generierung.

Die Vorgaben, d.h. die Entscheidungstabellen, bleiben die führenden Dokumente und werden bei Einhaltung dieser Vorgehensweise niemals veralten.

## **6 Plattformunabhängigkeit**

Grundsätzlich kann aus Entscheidungstabellen Programmcode für beliebige Plattformen generiert werden, bei Bedarf auch für mehrere Plattformen gleichzeitig.

Aus einer Entscheidungstabelle kann beispielsweise gleichzeitig Programmcode für Java, C# und ABAP generiert werden.

## **7 Unterstützung der Kommunikation**

Erleichterung der Kommunikation durch die einheitliche und kompakte Darstellungsform, gerade auch bei komplexen Sachverhalten.

Durch die klaren und eindeutigen Bezüge (wie in einem Koordinatensystem: Entscheidungstabelle, Regel, Bedingung, Aktion) sind mühelos hohe Detaillierungsgrade erreichbar.

## **8 Arbeitserleichterung bei der Beschreibung**

Die Beschreibung von Entscheidungsprozessen mit Entscheidungstabellen wird von Anwendern in der Regel als „weniger anstrengend“ und „weniger ermüdend“ empfunden als die Beschreibung z.B. in Textform.

Dieser Effekt wird noch verstärkt, wenn die verwendeten Werkzeuge Funktionen bereitstellen, wie z.B. eine automatische Ergänzung von noch fehlenden Regeln oder eine Simulation des Regelverhaltens für konkrete Fallbeispiele.

## **9 Arbeitserleichterung bei der technischen Umsetzung**

Durch den Einsatz von Generatoren kann der Programmieraufwand auf das notwendige Minimum reduziert werden, im Extremfall muss nach Regeländerungen sogar einfach nur neu generiert werden.

## 10 Bestehende Anwendungen, Altlastsanierung, Migration

Die detaillierte Realisierung von komplexer fachlicher Logik erfolgt auch heute leider immer noch in vielen Fällen in der jeweiligen Zielsprache in Form von ausführbarem Programmcode.

Mit Hilfe von Entscheidungstabellen kann die Komplexität ausgewählter Programmabschnitte

- auf ein abstraktes und leicht verständliches Niveau portiert werden (Nachdokumentation mit einer genormten Methode „Entflechtung“) und

- anschließend kann die technische Umsetzung mittels Generierung ohne Qualitätsverlust in dieselbe oder in eine andere Zielplattform erfolgen.

Daher sind Entscheidungstabellen auch für Reengineering-Projekte sehr gut geeignet.

## 11 Zukunftssicherheit

Die Methode „Entscheidungstabelle“ ist ausgereift, langzeitstabil und weltweit genormt, in Deutschland z.B. in der DIN 66241.

Entscheidungstabellen sind nicht an eine konkrete technische Plattform gebunden, so kann aus Entscheidungstabellen zu einem späteren Zeitpunkt problemlos zusätzlich Programmcode für weitere oder neue Plattformen generiert werden.

## 12 Anforderungen an ein Werkzeug für Entscheidungstabellen

Der am meisten erfolgskritische Punkt ist die Akzeptanz des Anwenders, deswegen muss sich das Werkzeug möglichst gut in die jeweilige gewohnte Arbeitsumgebung einfügen:

- Der rein fachliche Anwender soll das Werkzeug z.B. wie eine MS-Office-Anwendung (MS-Word, MS-Excel) empfinden und bedienen können

- Leistungsstarke Hilfsfunktionen müssen den Anwender begeistern („so etwas hatten wir bisher noch nicht“, „wir spüren, dass unsere Beschreibungen jetzt besser und vollständiger sind“)

- Für den Software-Entwickler muss das Werkzeug gut mit der jeweiligen Entwicklungsumgebung kommunizieren können

- Der Software-Entwickler muss die Code-Generierung als hilfreich empfinden und den generierten Code als „hochwertig“ ansehen können

- Die Funktionsweisen und Mechanismen des Werkzeuges müssen einfach sein und in kurzer Zeit verstanden werden können

### ***Wichtige Funktionen für einen leistungsstarken und sicheren Umgang mit Entscheidungstabellen dürfen nicht fehlen:***

- Automatische Prüfung auf Vollständigkeit, Redundanz- und Widerspruchsfreiheit
- Automatische Ergänzung von noch fehlenden Regeln
- Automatisches Verdichten von Regeln, z.B. durch Eliminierung irrelevanter Bedingungen
- Umsortieren der Regeln nach bestimmten Gesichtspunkten
- Ein- und Ausblenden von Regeln, um sich gezielt auf Teilbereiche konzentrieren zu können
- Simulation des Verhaltens von Entscheidungstabellen für konkrete Fallbeispiele
- Verknüpfen oder Verschachteln von Entscheidungstabellen, z.B. um eine Entscheidungstabelle in einer anderen Entscheidungstabelle als Bedingung oder Aktion zu verwenden
- Systemweite Analysen (Retrieval) von bereits „irgendwo“ verwendeten Bedingungen oder Aktionen
- Ermittlung und Darstellung von Mengengerüsten für Projekte oder Teilprojekte: Anzahl Entscheidungstabellen, Bedingungen, Aktionen und Regeln

### ***Sonstige Aspekte:***

- Das Werkzeug soll plattformunabhängig sein, d.h. es soll mit und für beliebige Plattformen verwendbar sein
- Aus einer Entscheidungstabelle muss gleichzeitig Programmcode für verschiedene Programmiersprachen erzeugt werden können
- Das Werkzeug soll ein reines Entwicklungswerkzeug sein, in den jeweiligen Produktionsumgebungen sollen keinerlei Laufzeitkomponenten des Herstellers notwendig sein (z.B. keine Rule-Engines, keine Rule-Server, keine „Black-Box“)
- Maximale Transparenz: Der generierte Programmcode soll die Möglichkeit bieten, das Systemverhalten (d.h. die Regelausführungen) zur Laufzeit beliebig detailliert analysieren und auswerten zu können

### 13 Weiterführende Themen

Dieses Dokument erläutert nur ganz grob, im Sinne einer ersten Übersicht, die wichtigsten Gesichtspunkte für die Verwendung von Entscheidungstabellen bei der Beschreibung fachlicher Regelwerke.

Weiterführende Themen wären beispielsweise:

- Abbildung von Schleifen-Verarbeitungen, ggf. auch mehrfach geschachtelt
- Abbildung von Automaten
- Verschachtelung von Entscheidungstabellen, d.h. Entscheidungstabellen als Bedingungen oder Aktionen in anderen Entscheidungstabellen
- Sichere Ableitung aller notwendigen Testfälle direkt aus den Entscheidungstabellen
- Generierung von Programmcode, z.B. für: Java, C#, ABAP, PL/SQL, Natural, Cobol, VB, VBA, C++, C, Delphi, PHP5,...
- Automatische Protokollierung der Regel-Ausführungen
- Statistische Auswertung der Regel-Ausführungen, auch über längere Zeiträume
- Automatische Ermittlung der Test-Deckungsgrade

### 14 Das Entwicklungswerkzeug LF-ET

Der Entscheidungstabellen-Editor LF-ET ([www.lohrfink.de/lf-et](http://www.lohrfink.de/lf-et)) erfüllt alle in diesem Dokument beschriebenen Anforderungen und ermöglicht die leichte und trotzdem leistungsstarke und durchgängige Verwendung von Entscheidungstabellen in beliebigen IT-Projekten.

Der Umgang mit Entscheidungstabellen ist schnell erlernbar und LF-ET ermöglicht, mit seiner Excel-ähnlichen Oberfläche und zahlreichen leistungsstarken Funktionen, die sichere Beherrschung auch sehr komplexer Regelwerke.

Durch den optionalen Einsatz von Generatoren kann maschinell garantiert werden, dass die erstellten Softwaremodule exakt mit den Vorgaben übereinstimmen. Auch der tatsächlich anfallende Programmieraufwand kann stark reduziert werden.

Insbesondere bei späterer Wartung und Pflege sind erfahrungsgemäß erhebliche Aufwands-Einsparungen zu erwarten, bei gleichzeitig deutlich reduziertem Fehler-Risiko.

Bei Bedarf kann LF-ET aus einer Entscheidungstabelle auch Programmcode für mehrere Programmiersprachen erzeugen.

LF-ET ist ein reines Entwicklungswerkzeug, zur Ausführung der generierten Programme ist **keinerlei zusätzliche Software erforderlich**.

## 15 Kontakt

Als IT-Beratungsunternehmen und -Dienstleister unterstützen wir seit vielen Jahren zahlreiche Behörden und Unternehmen aus den unterschiedlichsten Branchen.

Unabhängig vom technischen Umfeld ist dabei die Entscheidungstabelle ein wesentliches Hilfsmittel, um leichter und präziser Prozessabläufe beschreiben, implementieren und testen zu können.

Gerne stehen wir für weitere Informationen zur Verfügung.

**LOHRFINK**  
software engineering GmbH & Co. KG

Marie-Curie-Str. 6  
D-70736 Fellbach

Telefon 0711/3424 897-0  
Telefax 0711/3424 897-15  
info@lohrfink.de  
www.lohrfink.de  
www.lohrfink.de/lf-et