

# SOA verspielt - rekursive BPEL Prozesse

Guido Neander  
MT AG  
Ratingen

## Schlüsselworte

SOA, BPEL, rekursive Programmierung, Development, Deployment

## Einleitung

Bei komplexen Problemstellungen (z. B. Aufgaben mit Graphen) wird die iterative Programmierung schnell unübersichtlich.

Hier bietet die rekursive Programmierung eine Methode, diese Aufgaben elegant zu lösen. Das Grundprinzip ist, eine komplexe Aufgabe in viele kleine Aufgaben derselben Klasse zu zerlegen.

Lässt sich diese Lösungsstrategie auch mit BPEL realisieren? Treffen hier auch die klassischen Vorteile und Nachteile gegenüber einer iterativen Programmierung zu?

In diesem Vortrag wird eine Lösung vorgestellt, welche über einen rekursiven BPEL-Prozess ein Sudoku löst.

## Sudoku

Sudoku ist ein Logikrätsel. Ziel ist es, ein 9×9-Gitter mit den Ziffern 1 bis 9 so zu füllen, dass jede Ziffer in jeder Spalte, in jeder Zeile und in jedem Block (3×3-Unterquadrat) genau einmal vorkommt. Ausgangspunkt ist ein Gitter, in dem bereits mehrere Ziffern vorgegeben sind.

	3							
			1	9	5			
	9	8					6	
8				6				
4					3			1
				2				
	6					2	8	
			4	1	9			
							7	

Abb. 1: Ein Sudoku-Rätsel

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Abb. 2: ...und dessen Lösung

## Backtracking-Algorithmus

Der Backtracking-Algorithmus testet, ausgehend vom Startpunkt, alle Möglichkeiten aus, ähnlich der Vorgehensweise des Rätsellösers. Sobald erkennbar ist, dass eine Möglichkeit nicht funktioniert, wird diese verworfen und eine andere Richtung verfolgt. Hierdurch wird gewährleistet, dass eine vorhandene Lösung immer gefunden wird, aber nicht alle Varianten berechnet werden müssen. Ist keine Lösung möglich, wird auch dies festgestellt.

Bei komplexen Aufgaben kann die Suche sehr lange dauern. Der Backtracking-Algorithmus lässt sich leicht rekursiv implementieren.

## Rekursive Programmierung

Bei der rekursiven Programmierung ruft ein Programm sich selber wieder auf. Wichtig ist die Abbruchbedingung des eigenen Aufrufs, weil sich sonst das Programm endlos aufrufen würde.

Vorteile rekursiver Programmierung liegen je nach Algorithmus, bei einfacher Implementierung und besserer Lesbarkeit.

## technische Umsetzung des Backtracking-Algorithmus für ein Sudoku

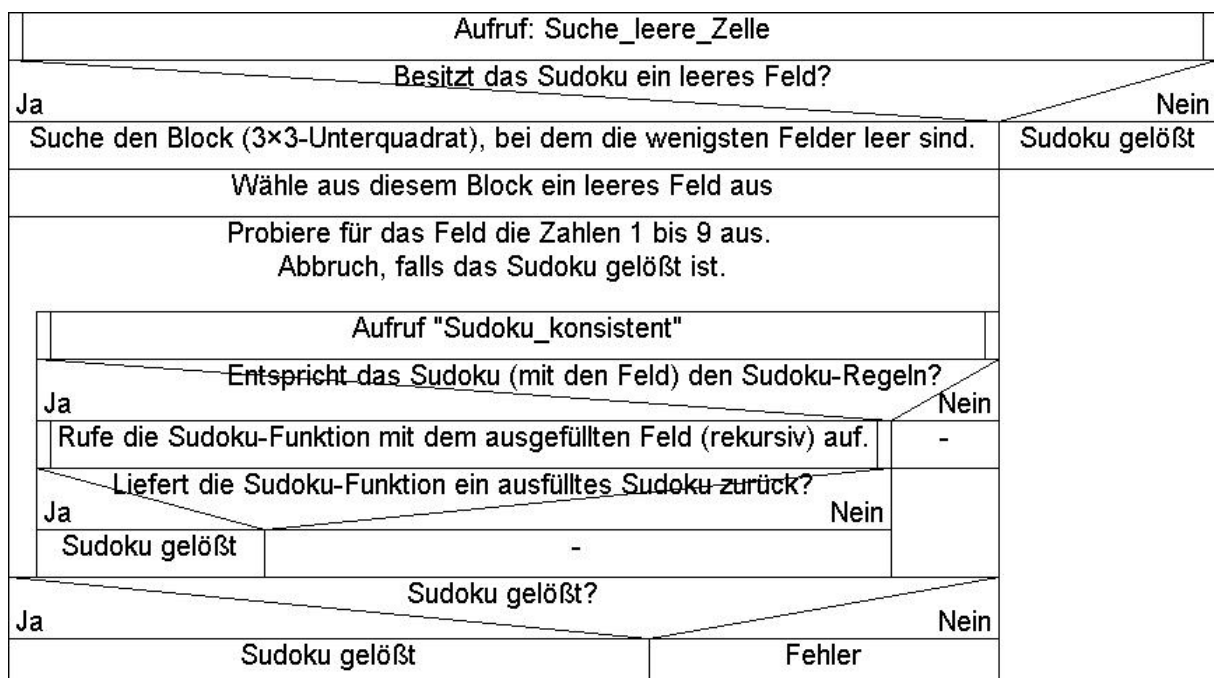


Abb. 3: Sudoku-Struktogramm

Der Backtracking-Algorithmus wurde mit 3 BPEL-Modulen realisiert.

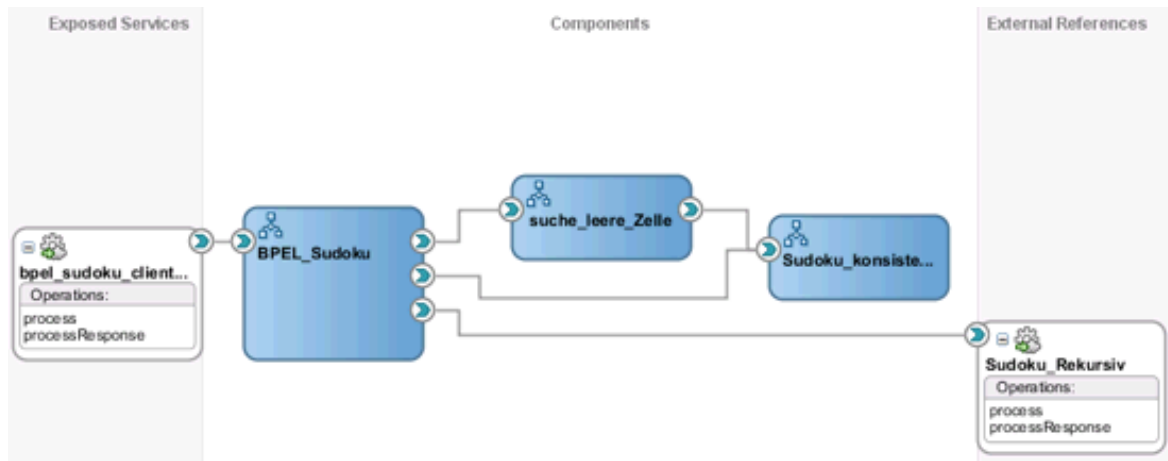
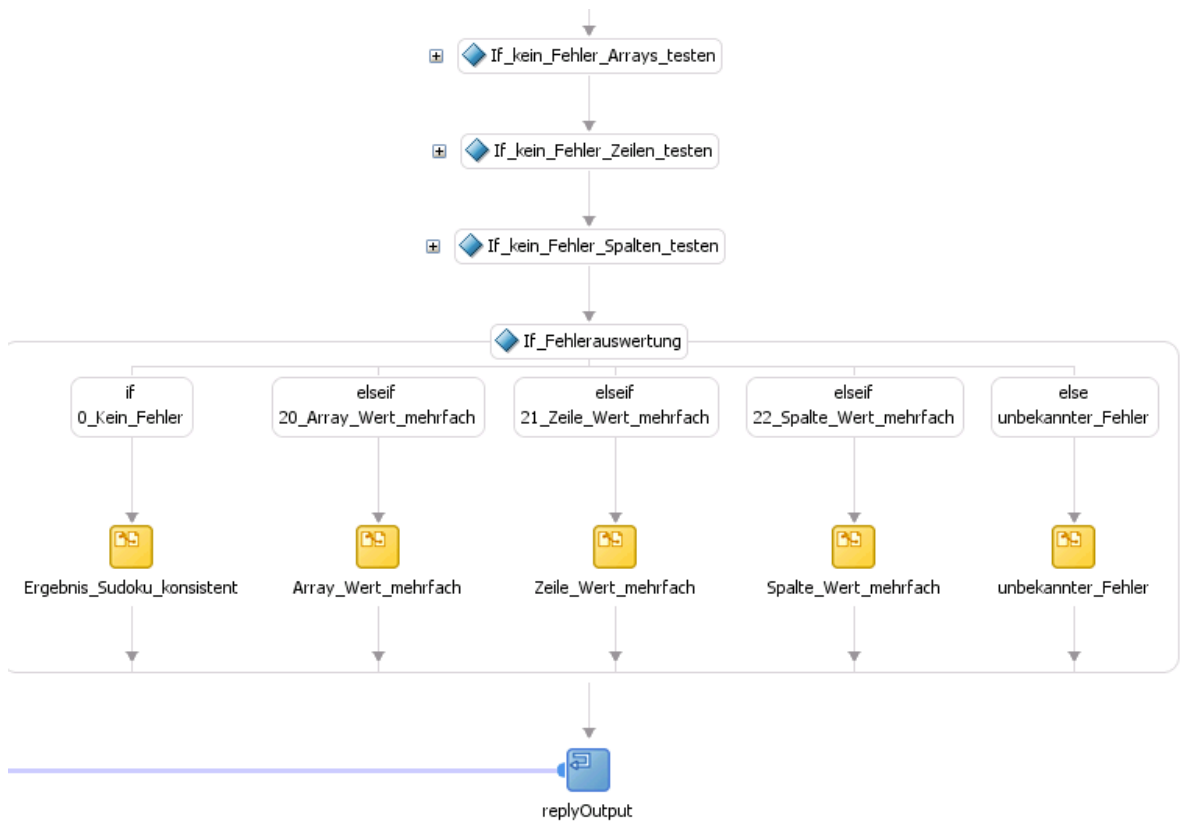


Abb. 4: Sudoku-Composit

### Erstes BPEL-Modul: Sudoku\_konsistent

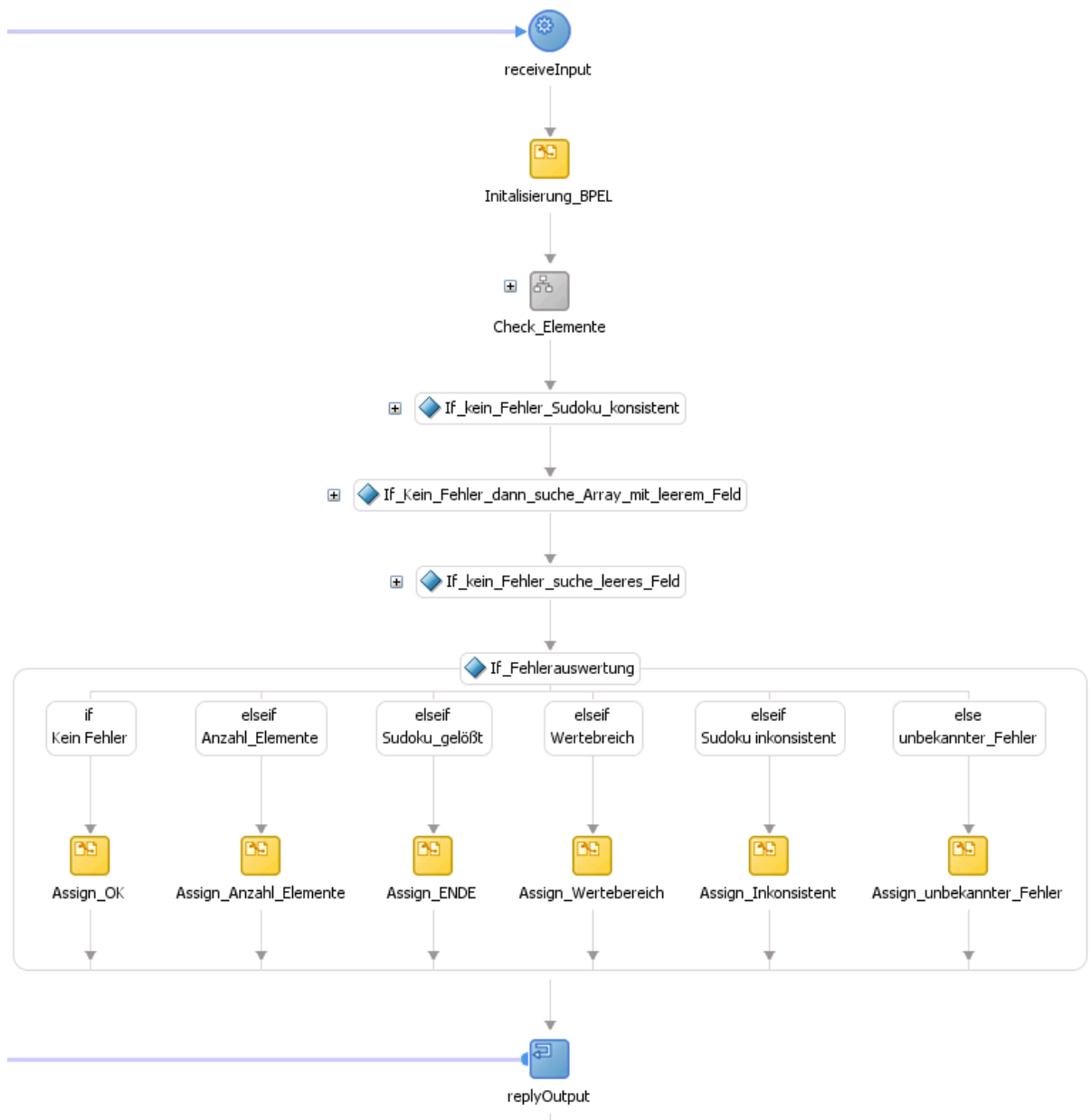
Das Modul prüft die Regeln, dass jede Ziffer in jeder Spalte, in jeder Zeile und in jedem Block (3×3-Unterquadrat) genau einmal vorkommt.



--- Abb. 5: Modul: Sudoku\_konsistent

## Zweites BPEL-Modul: suche leere Zelle

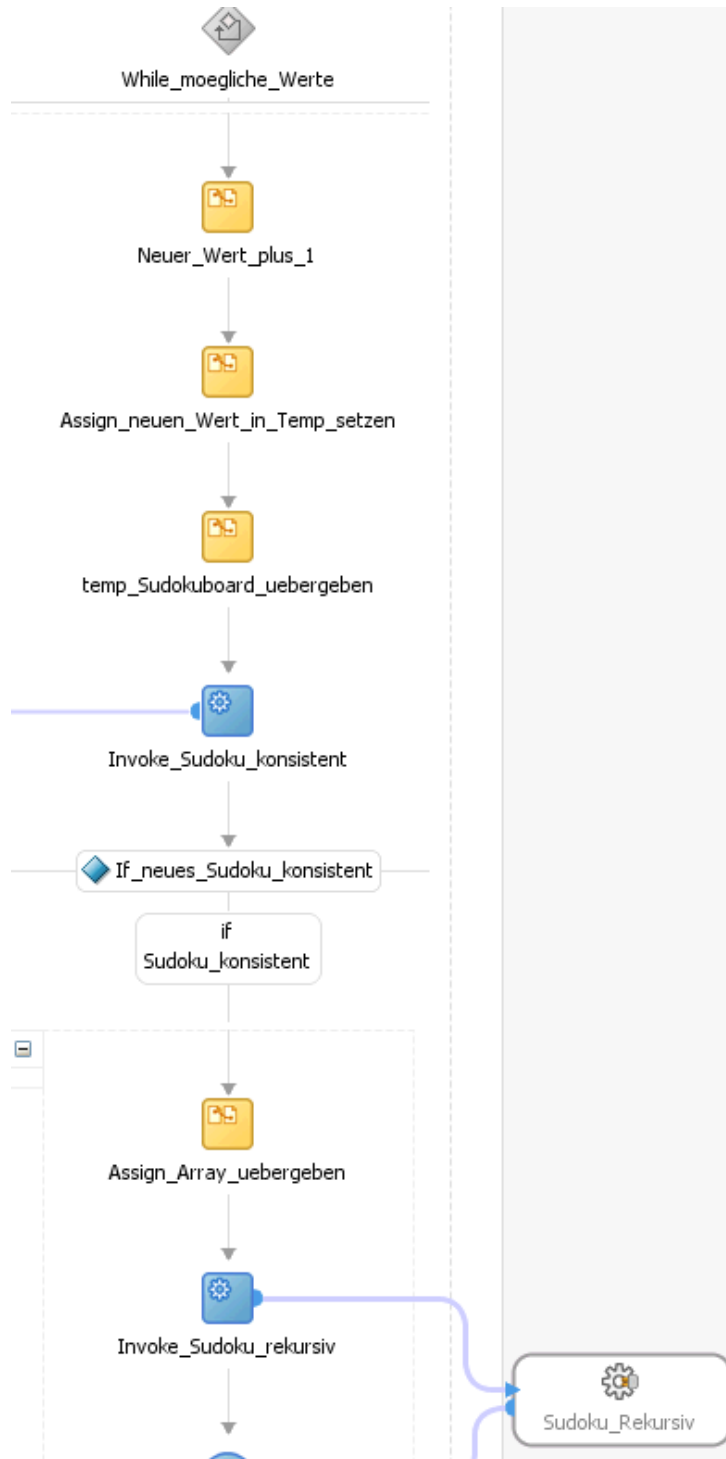
Das Modul prüft über das Modul Sudoku\_konsistent, ob das Sudoku gültig ist. Falls ja, wird der Block (3×3-Unterquadrat) gesucht, bei dem die wenigsten Felder leer sind. Danach wird das erste leere Feld in diesem Block ermittelt.



--- Abb. 6: Modul: suche\_leere\_zelle

### Drittes BPEL-Modul: BPEL\_Sudoku

Das Modul ermittelt über das Modul `suche_leere_Zelle` ein leeres Feld und testet die Zahlen 1 bis 9. Ist das Sudoku mit diesem Wert konsistent, wird der Sudoku-Algorithmus Rekursiv aufgerufen.



--- Abb. 7: Modul: BPEL\_Sudoku

## **Herausforderung und Lösungsansätze, die beim Deployment eines rekursiven BPEL-Prozesses auftreten können?**

### Erster Deployment Versuch

1. Deployment des Programms ohne Rekursion.
2. Implementierung der Rekursion. (Erstellung der Referenz über die Resource Palette.)
3. Beim Deployment erscheint die Fehlermeldung, dass die Referenz nicht existiert.

### Zweiter Deployment Versuch

1. Deployment des Programms ohne Rekursion.
2. Speichern der WSDL im Dateisystem.
3. Implementierung der Rekursion. (Erstellung der Referenz über die gespeicherte WSDL mit der Option, eine lokale Kopie zu erstellen.)
4. Das Deployment erfolgt ohne Fehler.

## **Buchempfehlungen**

Titel: Oracle SOA Suite 11g Handbook

Autor: Lucas Jellema

Verlag: Osborne Oracle Press Series

Titel: Oracle SOA Suite Developer's Guide

Autor: Matt Wright (Author), Antony Reynolds (Author)

Verlag: Packt Publishing

## **URL-Nachweis**

- <http://de.wikipedia.org/wiki/Sudoku>
- <http://de.wikipedia.org/wiki/Backtracking>
- [http://de.wikipedia.org/wiki/Rekursive\\_Programmierung](http://de.wikipedia.org/wiki/Rekursive_Programmierung)
- <http://blogs.bpel-people.com/2007/02/writing-recursive-bpel-process.html>

## **Fazit**

Die Vorteile rekursiver Programmierung treffen auch auf BPEL zu. Bei passenden Algorithmen lassen sich kompakte und übersichtliche rekursive Programme erstellen.

## **Kontaktadresse:**

MT AG

Guido Neander

Balcke-Dürr-Allee 9

40882 Ratingen

Telefon: +49 (0) 21 02 309 61-0

Telefax: +49 (0) 21 02 309 61-10

E-Mail: [guido.neander@mt-ag.com](mailto:guido.neander@mt-ag.com)

Internet: <http://www.mt-ag.com>