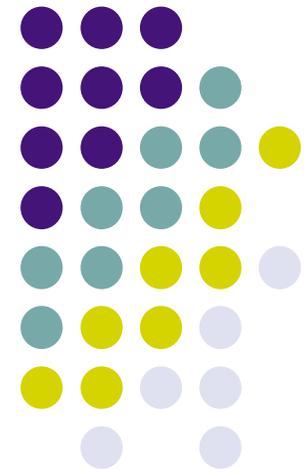


Produktkonfiguration und SAT-Solving

Carsten Sinz

Institut für formale Modelle und Verifikation

JKU Linz



Was ist Produktkonfiguration?



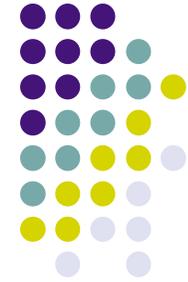
- Komplexe, variantenreiche Produkte
 - Z.B. PKW/LKW, Telekommunikations-Anlagen, Flugzeuge, Computer, medizinische Geräte
 - Produkt-Modell-Linien (z.B. C-Klasse, Harmony)
- Fragestellungen:
 - **Modellierung:** Wie können die Varianten zur automatisierten Verarbeitung dargestellt werden?
 - **Schlussfolgern:** Wie können Produktdaten automatisch geprüft bzw. Folgerungen daraus abgeleitet werden?

Produkt-Modellierung: Fragestellungen



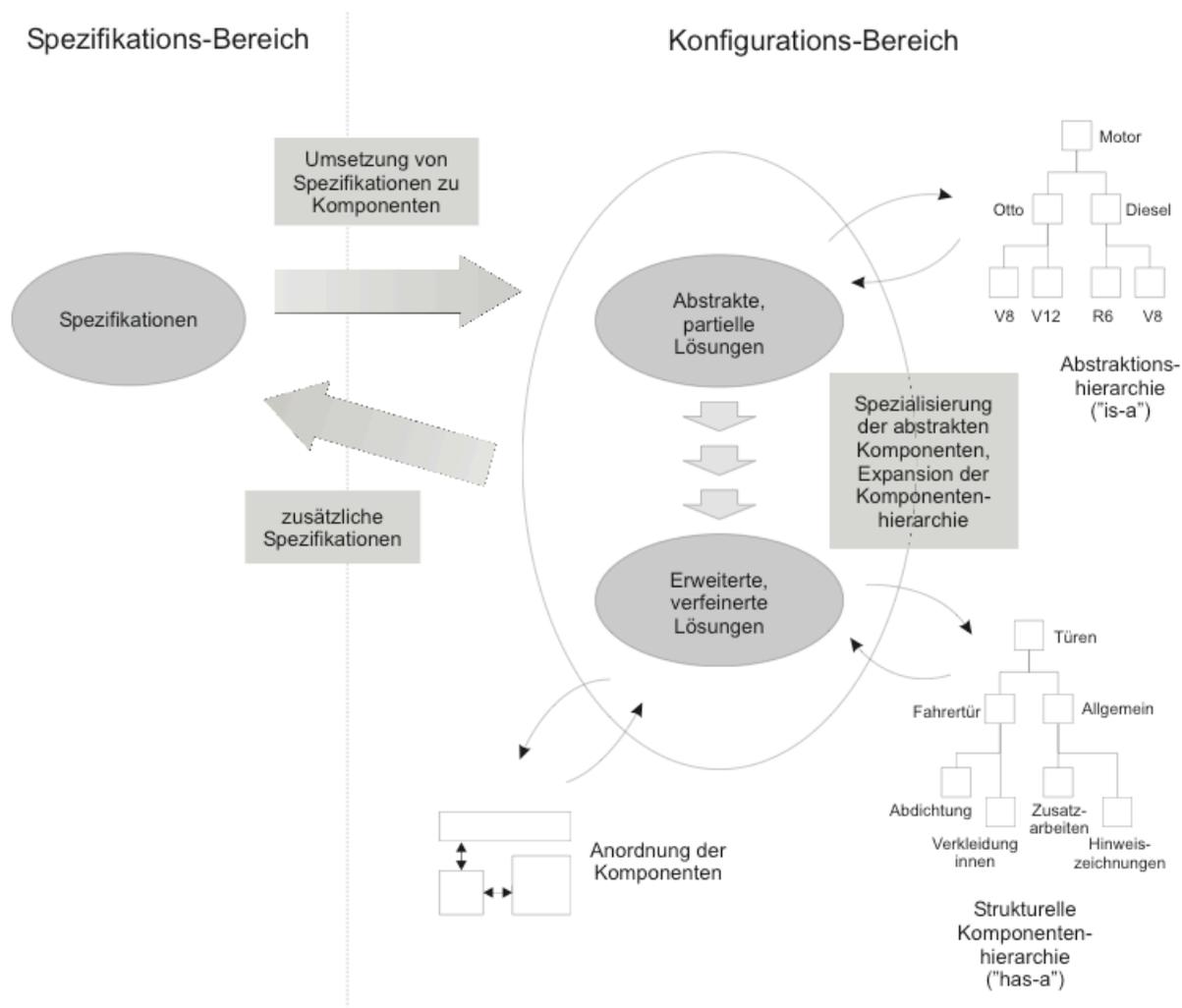
- Welche Zusammenhänge müssen/sollen abgebildet werden?
 - Kombinierbarkeit der Optionen
 - Option A geht nicht mit Option B
 - Wenn A vorhanden, muss auch B oder C vorhanden sein.
 - Komponenten-Hierarchie
 - Z.B.: PKW enthält Motor, dieser enthält Zylinder, ...
 - Funktionalität (z.B. mehrere Radiomodelle erfüllen gleiche Fkt.)
 - Ressourcen (z.B. Stromversorgung)
- Welchen Formalismus verwenden?
 - Beschreibungs- o. Featurelogiken (Rollen/Konzepte/Features; vgl. OWL / Semantic Web)
 - CSP (Constraint-Programming; vgl. ILOG-Configurator)
 - Aussagenlogik / SAT-Solving (Mercedes-Produktdatenbank)
 - Grafische Formalismen (z.B. AND-OR-Trees)

Produkt-Modellierung: Weitere Fragestellungen



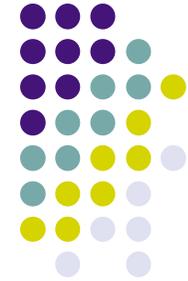
- Welcher Einsatzzweck?
 - Verkaufs-Konfiguration (interaktiv, Erklärungen)
 - Batch-Konfiguration (vollautomatisch, z.B. in Produktion)
- Welcher Detaillierungsgrad erwünscht?
 - Ursachen von Inkompatibilitäten mit formalisieren? (*Ressourcen*; z.B. Stromverbrauch)
- Welche Prüf-/Inferenzbedingungen?
 - Erfüllt gegebene Produktinstanz alle Randbedingungen?
 - Vervollständigung von Teilspezifikationen
 - z.B. egal ob AMD oder Intel CPU, aber 8 GB RAM
 - Kundenpräferenzen evtl. mitberücksichtigen
 - Konfigurationsdaten in sich stimmig?

Konfigurations-Modell [Stefik 1995]



Fallstudie 1:

Mercedes-Benz PKW



- 550 Anhängervorrichtung mit abnehmbarem Kugelhals
- 500 Außenspiegel links und rechts elektrisch heranklappbar
- 673 Batterie mit größerer Kapazität
- 231 Garagentoröffner im Innenspiegel integriert
- 551 Einbruch-Diebstahl-Warnanlage (EDW) mit Abschleppschutz
- 581 Komfort-Klimatisierungsautomatik THERMOTRONIC
- 280 Lenkrad in Lederausführung (zweifarbige) mit Chromspange
- 921 Motor mit Pflanzenölmethylester-Betrieb (Bio-Diesel)
- 353 Audio 30 APS (Navigationssystem mit integr. Radio und CD-Laufwerk)
- 671 Leichtmetallräder 4-fach, 7-Speichen-Design
- 772 Styling AMG

AMG-Styling (772) kann nicht mit Anhängervorrichtung 550 kombiniert werden.

Klimatisierungsautomatik (581) benötigt größere Batterie (673), außer bei den Benzinmotorvarianten mit 2,6 und 3,2 Litern Hubraum

Fallstudie 1:

Mercedes-Benz PKW



- Batch-Konfigurations-System zur Auftragsverarbeitung (Host-basiert, Cobol / 390-Assembler)
- Kundenauftrag als Liste von Bestellcodes
 - z.B. L, M612, M43, 200A, 158L, 423, 772, 550
- (Aussagenlogische) Regeln zur
 - automatischen Auftragsvervollständigung
 - Z.B. Wenn M612 und nicht GM, dann füge GA hinzu
 - automatischen Auftragsprüfung
 - Z.B. Wenn 772, dann darf 550 nicht im Auftrag sein
 - automatischen Generierung der Teileliste
 - Z.B. Wenn 550 und 423 und nicht 600, dann füge Teil A123456789 zum Auftrag hinzu

Fallstudie 1: Mercedes-Benz PKW



- Beispiel Prüfregeln:



- Größe des Konfigurationssystems (C-Klasse):
 - 692 Bestellcodes
 - 952 Prüf-, 197 Vervollständigungs- und 18508 Teileselektions-Regeln

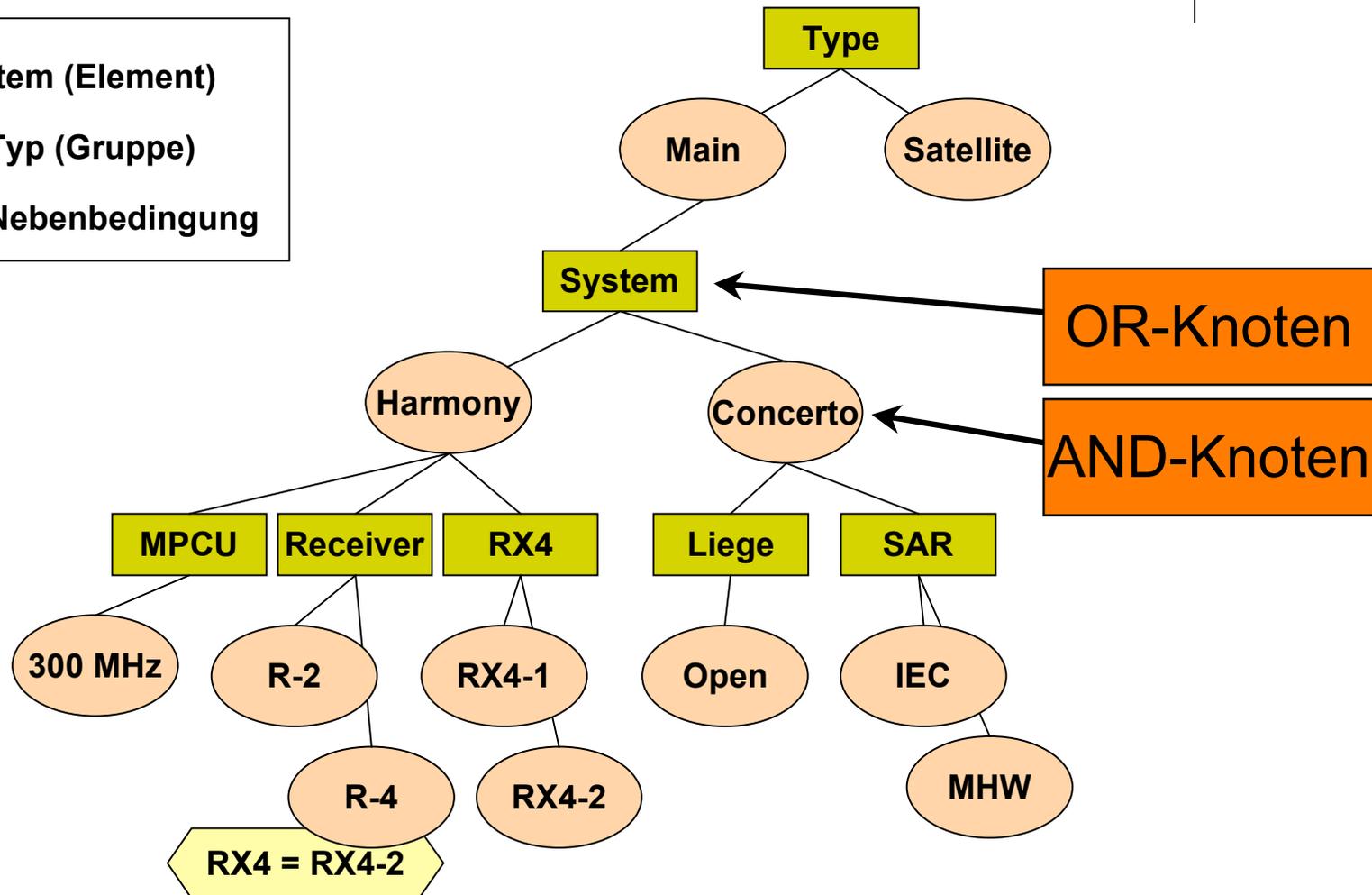
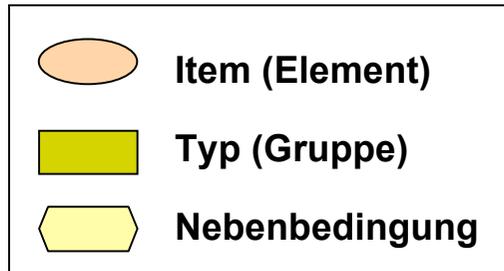
Fallstudie 1:

Mercedes-Benz PKW



- Verwendung des Konfigurationssystems
 - Prüfung einzelner Aufträge
 - Umsetzung Spezifikation → Teileliste
- Eingliederung Modellierungs-Formalismus:
 - Unterscheidung Bestellcodes / Teile
 - Constraints im Spezifikations-Bereich, auf Bestellcode-Ebene (nicht auf Teile-Ebene)
 - Keine Ursachen von Inkompatibilitäten erfasst (keine Ressourcen-Spezifikation)

Fallstudie 2: Siemens MR-Geräte



Inferenz



- Tools Beschreibungs- / Featurelogik:
 - **FaCT**: Hierarchie, funktionale Rollen, Anzahl-Einschränkungen
 - **RACER**: OWL-Reasoner, eingebaute *Domains* (*integer, strings*)
 - Noch nicht ausgereift
- Tools CSP: z.B. ILOG Solver / Configurator
- SAT-Solver: Aussagenlogik
 - Etabliert in der Hardware-Verifikation
 - Sehr performant (Formeln bis mehrere 100 MB Größe)
 - Eingebaute Domains relativ aufwändig zu realisieren

abnehmende Bereichs-Spezifität

SAT-Solver in der Konfiguration



- Voraussetzung: Transformation des Konfigurationsmodells nach Aussagenlogik
 - Struktur (z.B. Hierarchie) muss nach Aussagenlogik übersetzt werden
 - Einfach, sofern nur endliche Bereiche benötigt
 - Modellierung von z.B. *integers* schwieriger
- SAT-Solver können auch Beweise / Gegenbeispiele liefern
 - Z.B. zur Klärung von Fehlerursachen

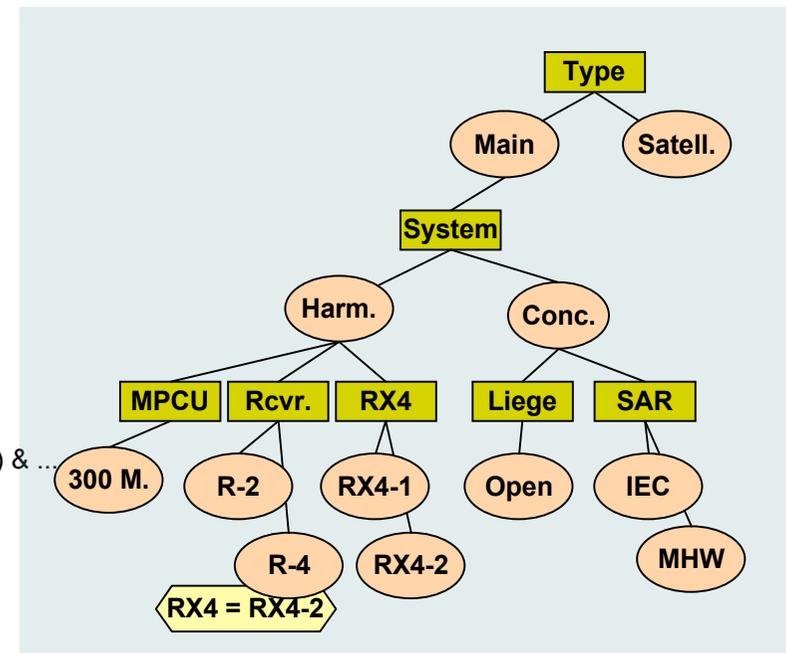
Transformation nach SAT: Beispiel Siemens MR-Geräte



```

1 [ValConfT(Type), SubItemsValid.Satellite]
2
3 (INI_ConsoleType_Sat =>
4   (~INT_System & ~INT_Comp_MPCU & ~INT_Comp_RXNumOf &
5    ~INT_Comp_ReceiverNumOf & ~INT_Comp_COT & ~INT_Comp_SAR)
6 ) &
7
8 [ValConfT(Type), SubItemsValid.Main]
9
10 (INI_ConsoleType_Main =>
11   (INI_System024 [Harmony] =>
12     INI_Comp_MPCU300 & ~INI_Comp_MPCU133 & ~INI_Comp_MPCU266 &
13     =1:(INI_Comp_RXNumOf1, INI_Comp_RXNumOf2) &
14     (INI_Comp_ReceiverNumOf4 => INI_Comp_RXNumOf2) &
15     =1:(INI_Comp_ReceiverNumOf2, INI_Comp_ReceiverNumOf4) &
16     ~INT_Comp_COT &
17     ~INT_Comp_SAR
18   ) &
19   (INI_System007 [Concerto] =>
20     ...
21     ...
22     ...
23     ...
24     ...
25   ) &
26   =1:(INI_System024, INI_System007)
27 ) &
28 =1:(INI_ConsoleType_Sat, INI_ConsoleType_Main) &
29
30 [Type Definitions]
31
32 (INI_ConsoleType_Sat | INI_ConsoleType_Main => INT_ConsoleType) & ...

```





Zusammenfassung

- Anforderungen an Software- und Produktkonfiguration leicht unterschiedlich
- Produktkonfigurations-Verfahren in Industrie bereits etabliert
 - **Systeme:** SAP, Oracle, ILOG, Tacton, ...
 - **Anwender:** DaimlerChrysler, Siemens, BMW, ABB, HP, ...
- Ideen / Verfahren übertragbar auf Software-Konfiguration?