SAT-Based Consistency Checking of Automotive Electronic Product Data

Carsten Sinz, Andreas Kaiser and Wolfgang Küchlin
Symbolic Computation Group
University of Tübingen
and Steinbeis Technology Transfer Center OIT

http://www-sr.informatik.uni-tuebingen.de

Introduction

Scenario:

- Electronic configuration of Mercedes car and truck lines
- Rule-based EPDM system already present
- Boolean logic employed to express constraints and to control processing of orders

Problem:

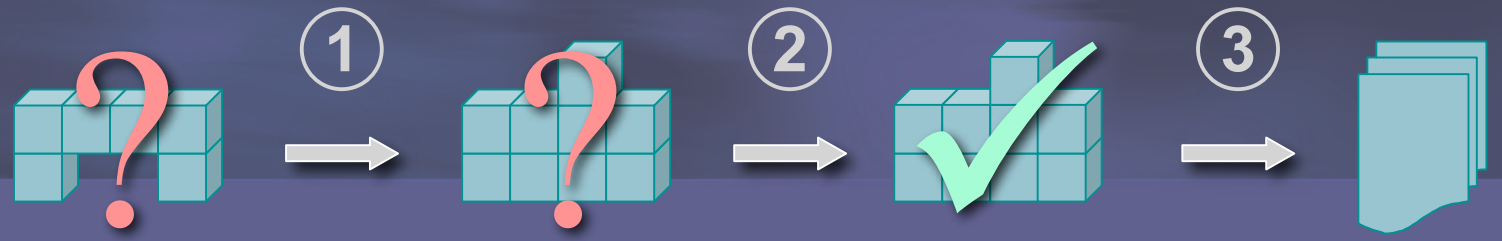
- Complexity of product and documentation induces errors

Goals:

- Computer-based assistance in finding potential errors
- Increasing documentation quality

DaimlerChrysler´s EPDM System DIALOG

- Customer´s order consists of a set of Boolean variables (*codes*) describing the model class and additional features
- Order processing performed in three steps:
  1. Order completion
  2. Constructibility check
  3. Parts list generation
- All steps controlled by evaluating logical rules



Customer´s Order | Extended Customer´s Order | Checked and Extended Customer´s Order | Order´s Parts List

DC´s EPDM System DIALOG (contd.)

Step 1: Order completion

- Interpretation of supplementing rules $Cond^S \rightarrow x$:

  Code *x* is added to a customer´s order *O*,

  if *x*´s supplementing rule ($Cond^S$) evaluates to true under *O*

- Notes:
  1. Steering restriction has to be considered
  2. Group controls order in which additional codes are tested
  3. Order of rule application can be relevant
- Example:

Code	Steering	Group	Supplementing Rule
GM	-	CAA-1030	M111/M605/M611
GM	L	CAA-1030	M112
GA	R	CAA-1030	M113

DC´s EPDM System DIALOG (contd.)

Step 2: Constructibility check

- Interpretation of constructibility rules $x \rightarrow Cond^C$:

  Code *x* is constructible (valid) in a customer´s order *O*,
  if *x´*s constructibility rule ($Cond^C$) evaluates to true under *O*

- Notes:
  1. Additional constructibility rules independent of model class
  2. Rules hierarchically organized in positions and variants
  3. For a valid order all codes have to be constructible

- Example:

Code	Steering	Pos.	Var.	Constructibility Rule
M111	-	60	1	M18/M20//M23+494+-M001
M111	L	60	2	M20+M001
423	-	100	1	M111/M605/M611

DC´s EPDM System DIALOG (contd.)

Step 3: Parts list generation

- Interpretation of code rules $Cond^P \rightarrow p$:

  Part *p* is contained in a customer´s order *O*,
  if *p´s* code rule ($Cond^P$) evaluates to true under *O*

- Notes:
  1. Parts list grouped by modules, positions, variants depending on functional and geometrical aspects
  2. Variants of each position are mutually exclusive

- Example:

Part	St.	Module	Pos.	Var.	Code Rule
124893	-	040522	200	1	221+292+(500/611)
242488	L	040522	200	2	(800/801)+-704
486919	R	012400	100	10	M18/M20+M111

BIS: Consistency of Product Documentation

- Critical points with DIALOG system:
  - Complexity of rules may cause errors
    - unnecessary rejected orders
    - loss of production
    - wrong scheduling of parts
  - Maintaining the rule base is a demanding job
- Our solution: BIS (*Baubarkeits-Informations-System*)
  - Add-on tool to check global consistency criteria of the product data-base as a whole
  - Increase the quality of product documentation
  - Deliver assistance in maintaining the product data-base

Global Consistency Criteria

- A priori criteria: (not requiring special product knowledge)
  - Necessary Codes
  - Inadmissable Codes
  - Consistency of the order completion process
    - order dependency
    - stability (no valid orders are invalidated)
  - Superfluous parts
  - Ambiguities in the parts list
- User-specified criteria

SAT Encoding of Consistency Assertions

- Outline of Encoding
  1. Generate Boolean formula *B* describing all supplemented and checked orders

     $B = Z \wedge C$

     $Z = (Cond^S_1 \Rightarrow x_1) \wedge \ldots \wedge (Cond^S_n \Rightarrow x_n)$

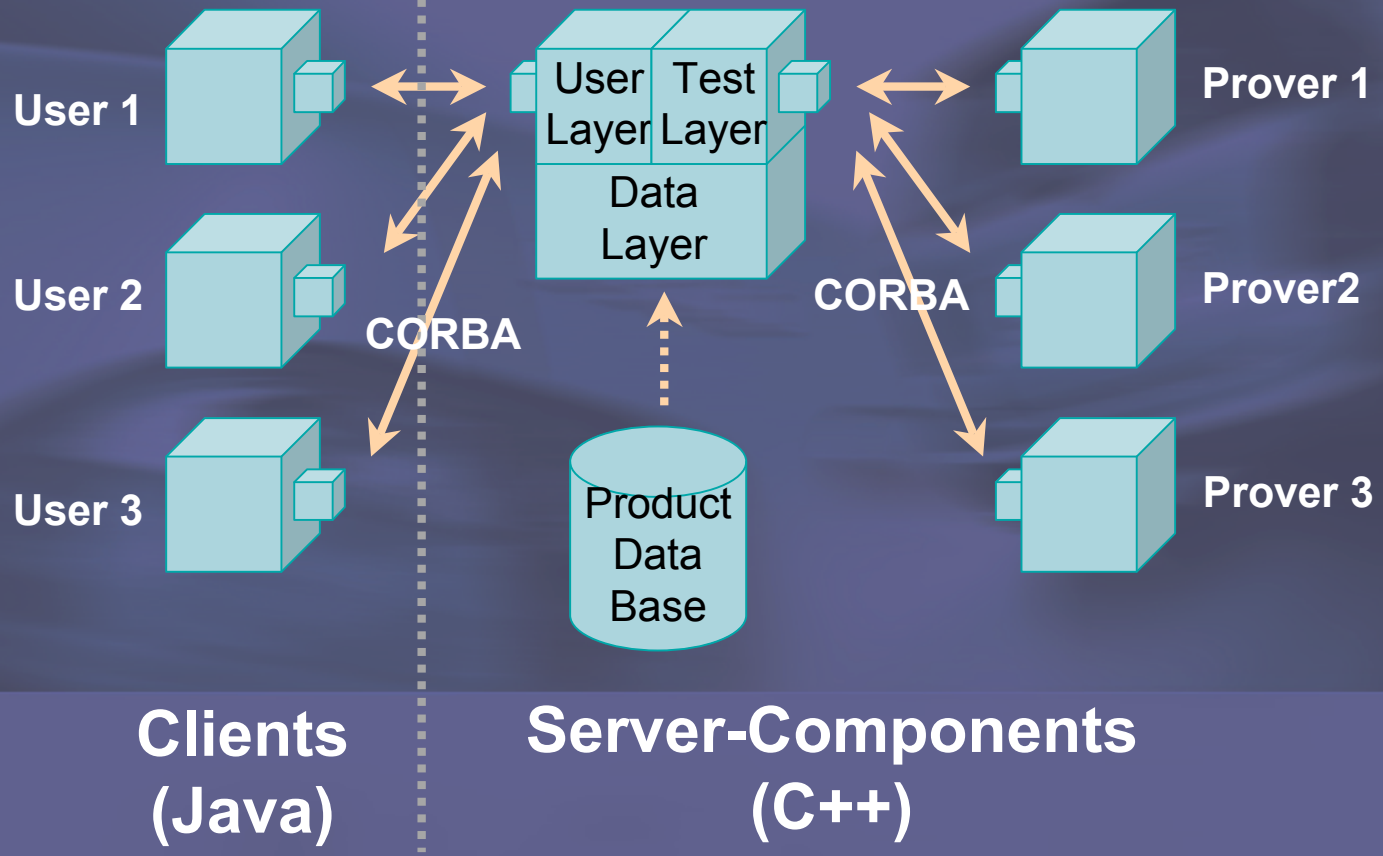
     $C = (x_1 \Rightarrow Cond^C_1) \wedge \ldots \wedge (x_m \Rightarrow Cond^C_m)$
  2. Specify consistency criteria as side condition *S*
  3. Check satisfiability of formula $T = B \wedge S$
- Example:

  Part *p* with code rule $Cond^P$ = 221+292+(500/611) is superfluous if formula T = B ∧ 221 ∧ 292 ∧ (500 ∨ 611) is unsatisfiable.

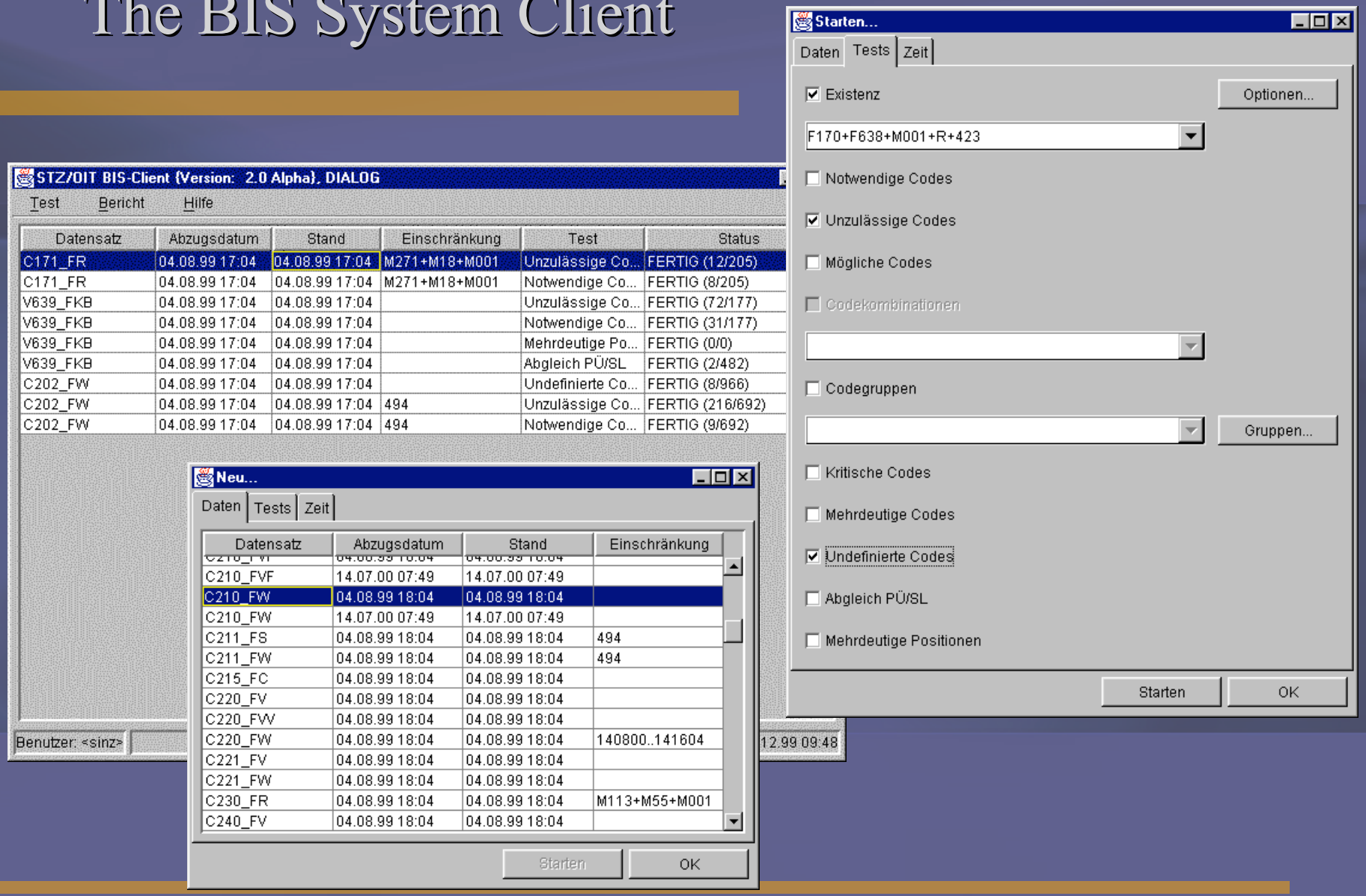
First Experience with BIS

- Formula *B* usually contains 200-2000 variables and 10000-100000 symbols (depending on model class)
- Davis-Putnam style satisfiability checkers solve most of the generated SAT-instances in under a second
- Push-button technology (no user interaction during proofs required)
- Inconsistencies in the DIALOG system data-base could be found

Architecture of the BIS System



The BIS System Client



Extensions Based on Experience

- Additional Functionality
  - Restricting the set of valid orders
  - Valid additional equipment options
  - Combinations of codes
  - Groups of mutually exclusive codes
- Satisfiability checking without prior CNF-conversion
- Extended Propositional Language

Extended Propositional Language

Special operators for symmetrically releated codes

- General form: $Rk : X_1,...,X_n$

  $R \in \{ =, \neq, \leq, <, \geq, > \}$, k a positive number,

  $X_1,...,X_n$ arbitrary formulae of the extended language

- Example

  ≤ 1: A, B, C is equivalent to $\neg(A \wedge B) \wedge \neg(A \wedge C) \wedge \neg(B \wedge C)$

- Advantages
  - More compact notation for symmetrically related codes
  - Pattern occurs frequently in product configuration (e.g. one country code in each order)
  - Specialized algorithms

Summary & Prospects

- Summary
  - BIS complements existing EPDM system
  - Increase in product documentation quality
  - Global consistency assertions are
    1. converted to Boolean logic satisfiability problems
    2. solved by Davis-Putnam style prover
- Prospects
  - Adaptation to other EPDM systems
  - On-line product configuration (E-commerce) requires high-quality EPDM systems with low error rates