Temporal Change in Product Documentation for Manufacturing

Carsten Sinz and Wolfgang Küchlin
Symbolic Computation Group, WSI for Computer Science,
University of Tübingen, Germany

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

Outline of Talk

1st Part: Product documentation for manufacturing at DaimlerChrysler (Mercedes lines):

- Basic documentation schema
- Documentation of temporal aspects

2nd Part: Quality assurance of documentation

- Verification of consistency conditions
- Detection of errors caused by temporal change
- Computation of induced change on parts level

1st Part

Product Documentation for Manufacturing at DaimlerChrysler (Mercedes lines)

Basic Documentation Schema

- Batch configuration for engineering/production
  - Configuration task: order completion, order checking, parts list generation
  - Non-interactive, high throughput system
  - Rule-based, operations controlled by Boolean logic formulae:

    S(x): supplementing rule for code x

    C(x): constructability rule for code x

    R(p): part selection rule for part p
  - Atoms: Equipment codes (sales options) and control codes
  - Two levels
    1. function-oriented (equipment codes, product overview)
    2. parts-oriented (modularized hierarchical parts list)

Basic Documentation Schema: Example

F202: model class 202 (C class)

Order completion:

$S(584) = 906 \vee 955 \vee 625 \wedge (M104 \vee M112)$

$S(570) = 494 \vee 498 \vee 625 \wedge \neg 959 \vee 955$

$S(GM) = \neg GA \wedge (M112 \wedge L \vee M111 \vee M104)$

⋮ ⋮

584: electrical window lift

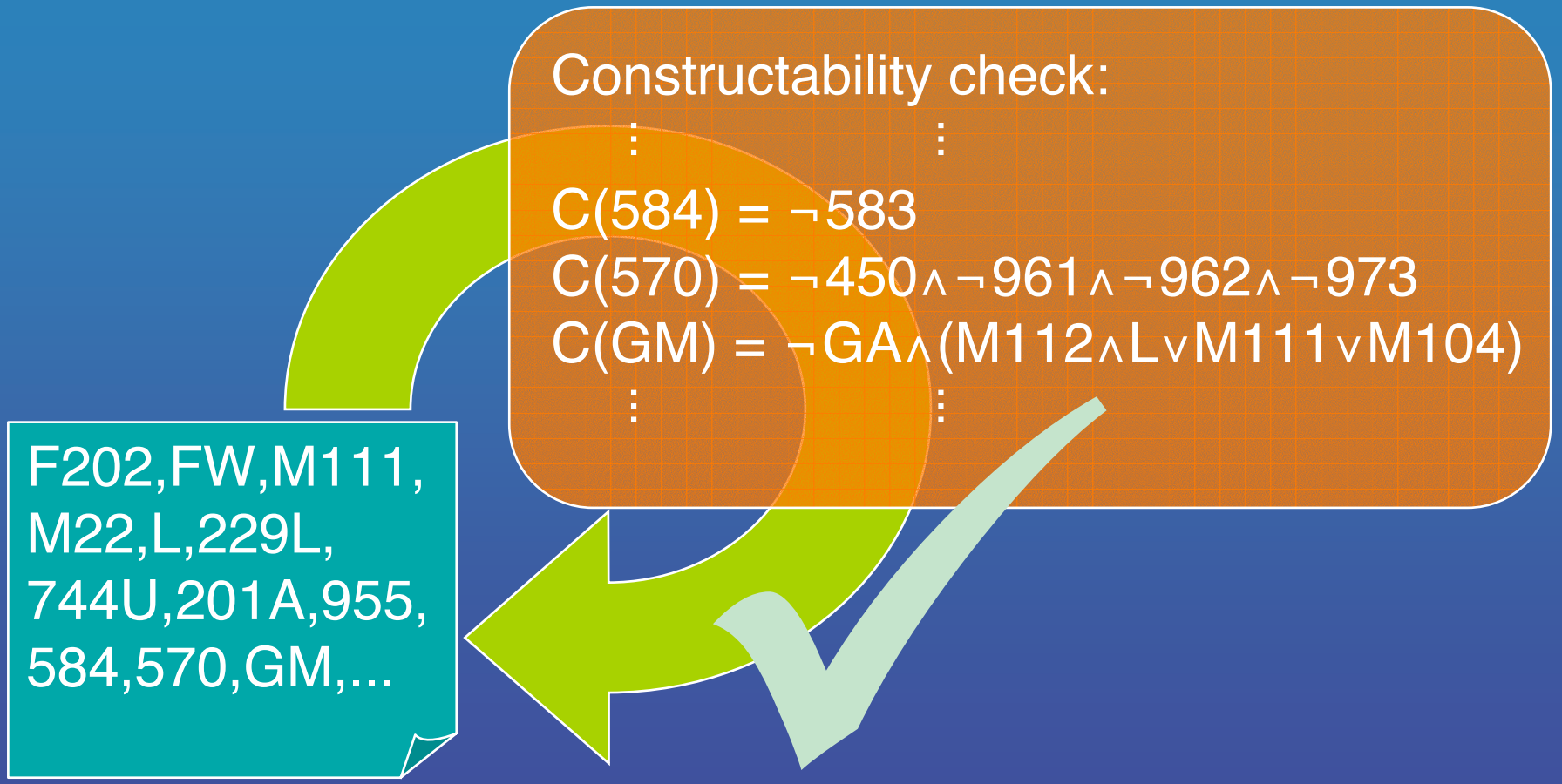
570: shiftable armrest in front

GM: mechanical gearing

F202,FW,M111,
M23,L,229L,
744U,201A,955

744U,201A,955,
584,570,GM,...

Basic Documentation Schema: Example



Basic Documentation Schema: Example

Parts list generation:

⋮ ⋮

$R(P49581) = M111 \wedge \neg 718$

$R(P49582) = M112 \wedge 423$

$R(P49583) = 201A \wedge 584 \wedge \neg 715$

⋮ ⋮

F202,FW,M111,
M22,L,229L,
744U,201A,955,
584,570,GM,...

⋮

49581

49583

⋮

Documentation of Temporal Aspects

- All rules R equipped with activity time interval:

  $I(R) = (t_\alpha(R), t_\omega(R))$

- Additional control codes (CC_α,CC_ω) may override activity time interval:

  CC_α starts rule activity (even before t_α)

  CC_ω stops rule activity (even before t_ω)

  [Control codes simplify order-dependent temporal rule selection (e.g. to model overlapping phases).]

Temporal Rules

- Used in production environment:
  - to control part availability and exchange
  - to model equipment code start-up and run out
  - to accompany assembly line reconfiguration

- Central question:

How to determine

changing part requirements

induced by

documentation changes?

2nd Part

Quality Assurance of the Product Documentation in the Presence of Temporal Change

Quality Assurance of Documentation

- Static consistency assertions [Küchlin&Sinz,2000]
  - E.g. no contradictory constructability/supplementing rules, no unused part list entries
  - Boolean formulae, satisfiable if consistency assertion is fulfilled
  - Assertions declare properties holding for all possible constructible orders
- Computation of induced change on parts level
  - Comparison of parts requirement for two scenarios using method for static consistency assertions

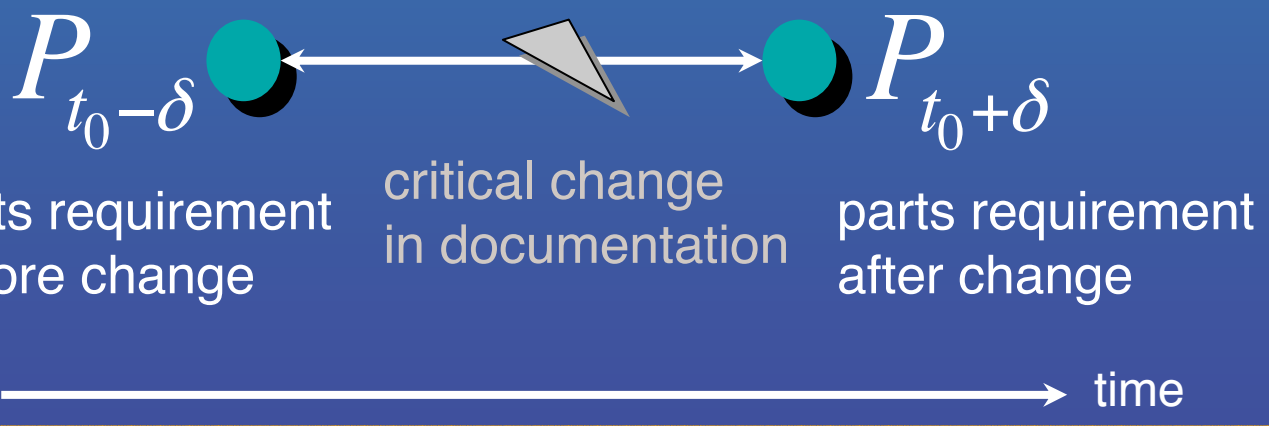
Induced Change on Parts Level: Example

code/part	rule type	formula	t_α	t_ω
P49583	R	$201A \wedge 584 \wedge \neg 715$	01.04.2001	31.08.2001
584	C	$\neg 583$	01.01.2001	10.08.2001
584	C	$\bot$	11.08.2001	30.09.2001

- Code 584 is valid until 10.08.2001
- Entry P49583 in parts list is unused after 10.08.2001
- Causal chain of dependencies may be arbitrary complex
- Unnecessary parts list entries may also have non-temporal reasons

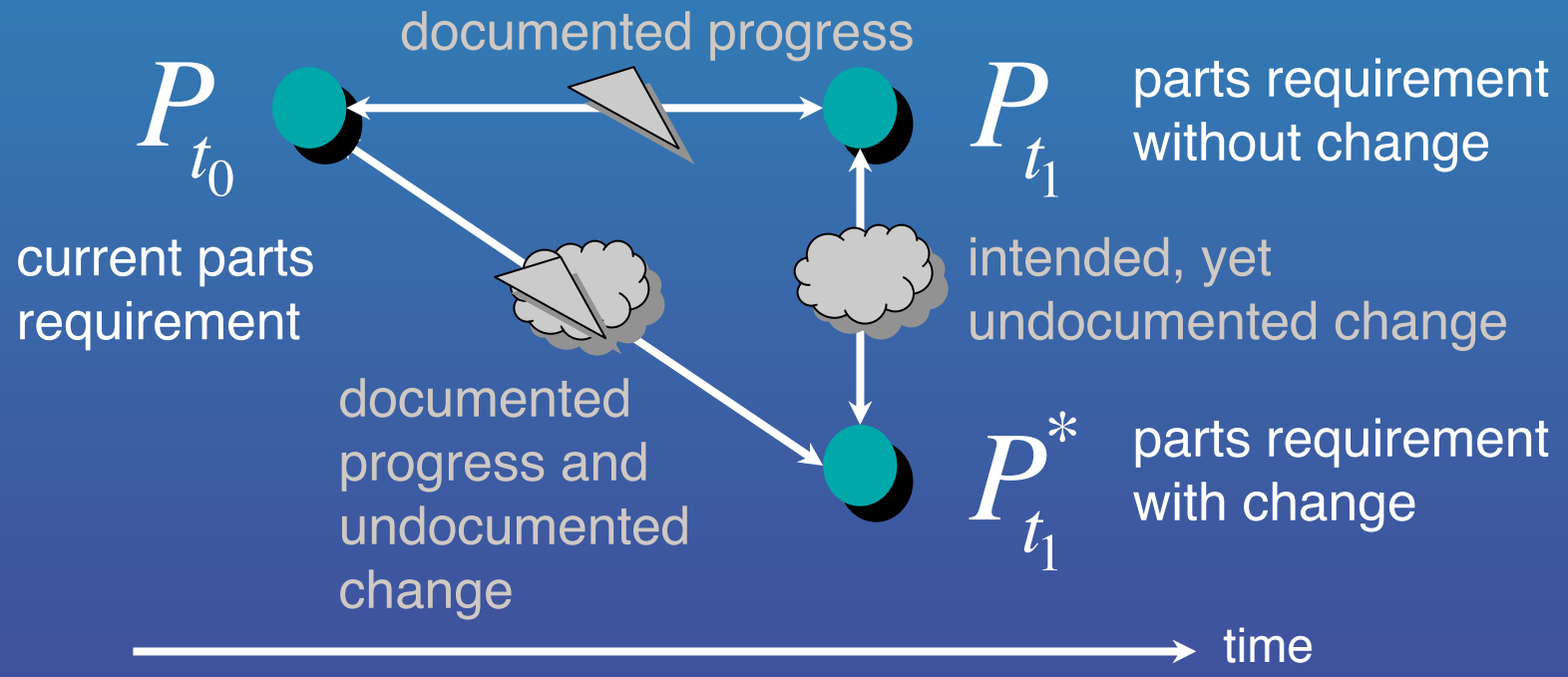
$\pm\delta$–Method

- Critical change in documentation at time t_0, e.g. model year change
- Comparison of parts requirements shortly before and after t_0
- Computation of parts requirement at $t_0 \pm \delta$ using techniques from static consistency analysis (SAT)



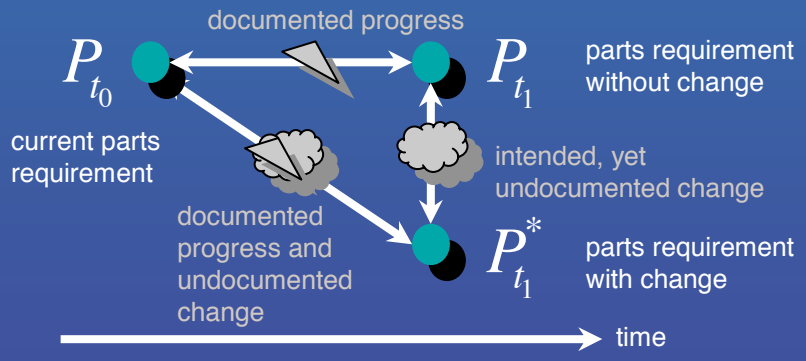
Extension: 3-Point Method

- Comparison of parts requirement at fixed time t_1 in the future with and without change



Extension: 3-Point Method (continued)

- Intended change is specified only for consistency checking purpose; possible changes:
  - Equipment codes becoming valid or invalid
  - Arbitrary code combinations becoming invalid, e.g. all limousines of model class 202 with 2300cc direct-injection engine (FW∧F202∧M111∧M23)
- Allows handling of production relocation
- Consideration of current parts requirement gives additional information
- Details in the proceedings



Implementation

- Prototypical implementation in C++ as part of our BIS system
- Complete comparison of parts requirements computed in approx. 15-30 min.
- Typical size of input data:
  - 500-1000 Boolean variables (equipment and control codes)
  - 400-750 rules in product overview
  - 4000-10000 rules in parts list
- Promising experiments in parallelization of checks

Summary & Conclusions

- Changes in production/manufacturing are very frequent, thus also in the documentation
- Determination of varying parts requirements is of prime importance
- Conversion to Boolean logic satisfiability problems straightforward
- Application of advanced SAT-checking technology adequate
- Two computation methods for different applications:
  - $\pm\delta$–method for minor changes at fixed time
  - 3-point method for substantial, vaguely specified changes

Thank you for your attention!

Carsten Sinz and Wolfgang Küchlin
Symb. Comp. Group, WSI for Computer Science,
University of Tübingen, Germany

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