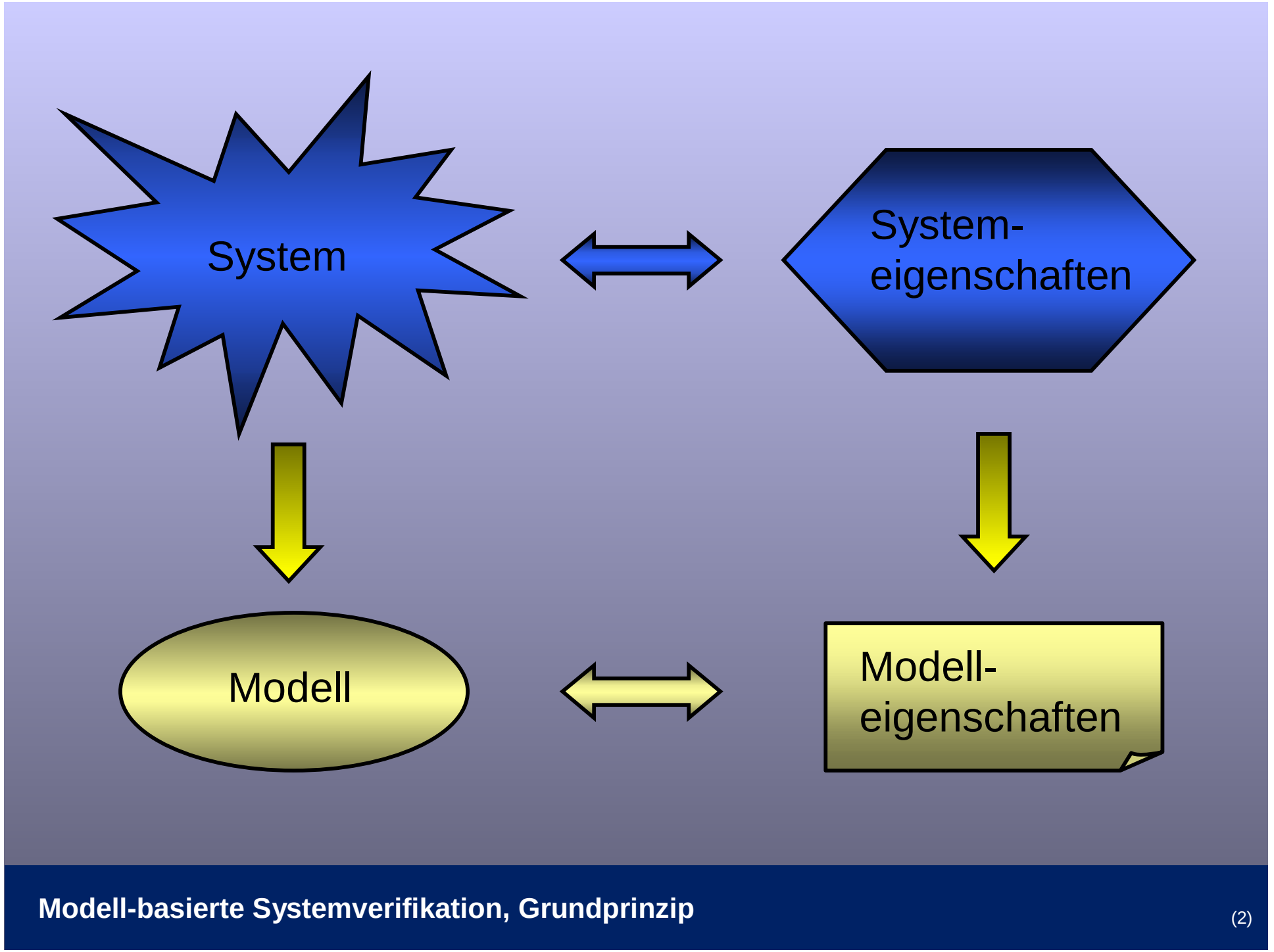
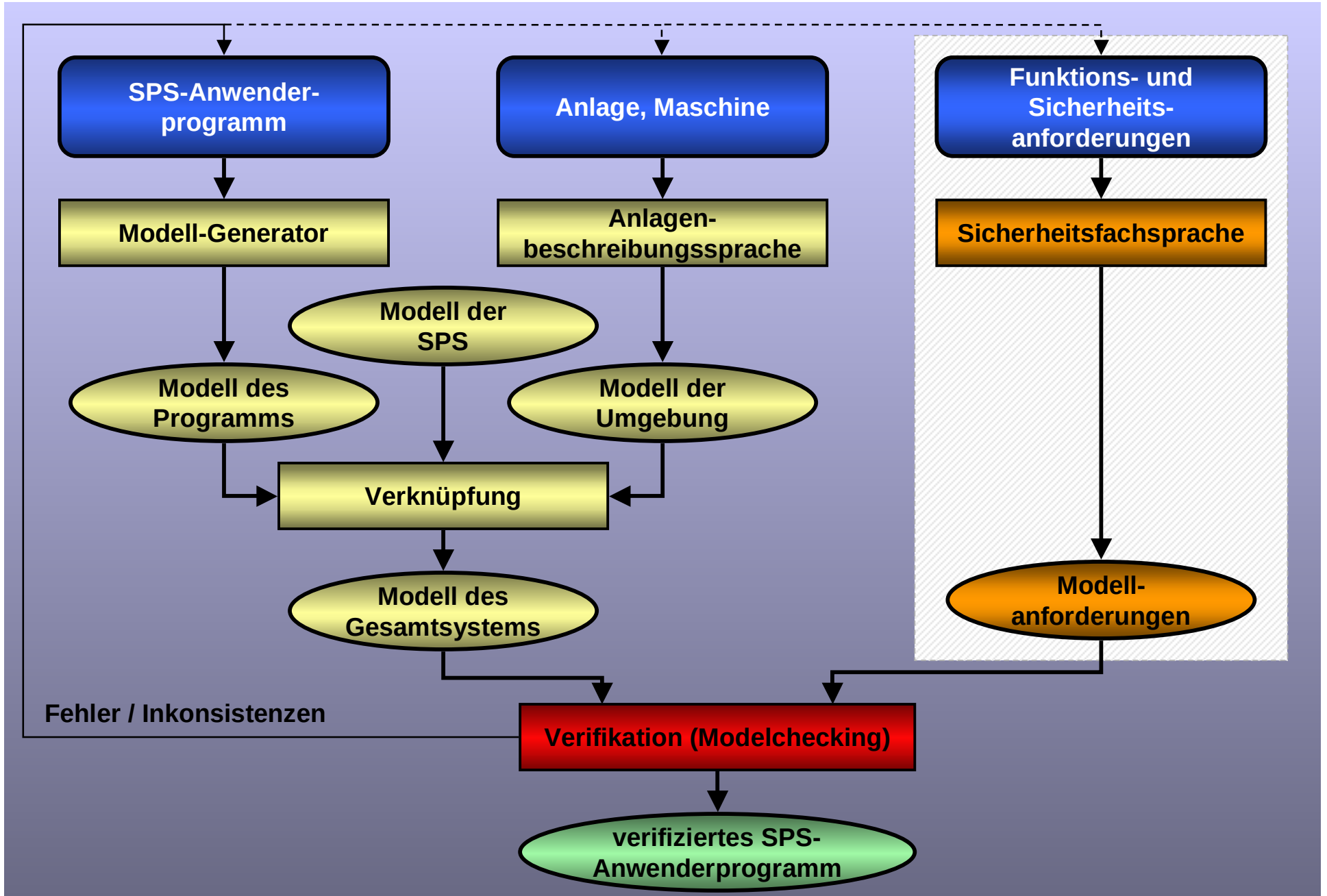
Modelchecking in der Automatisierungstechnik

Wie formal müssen formale Methoden sein?

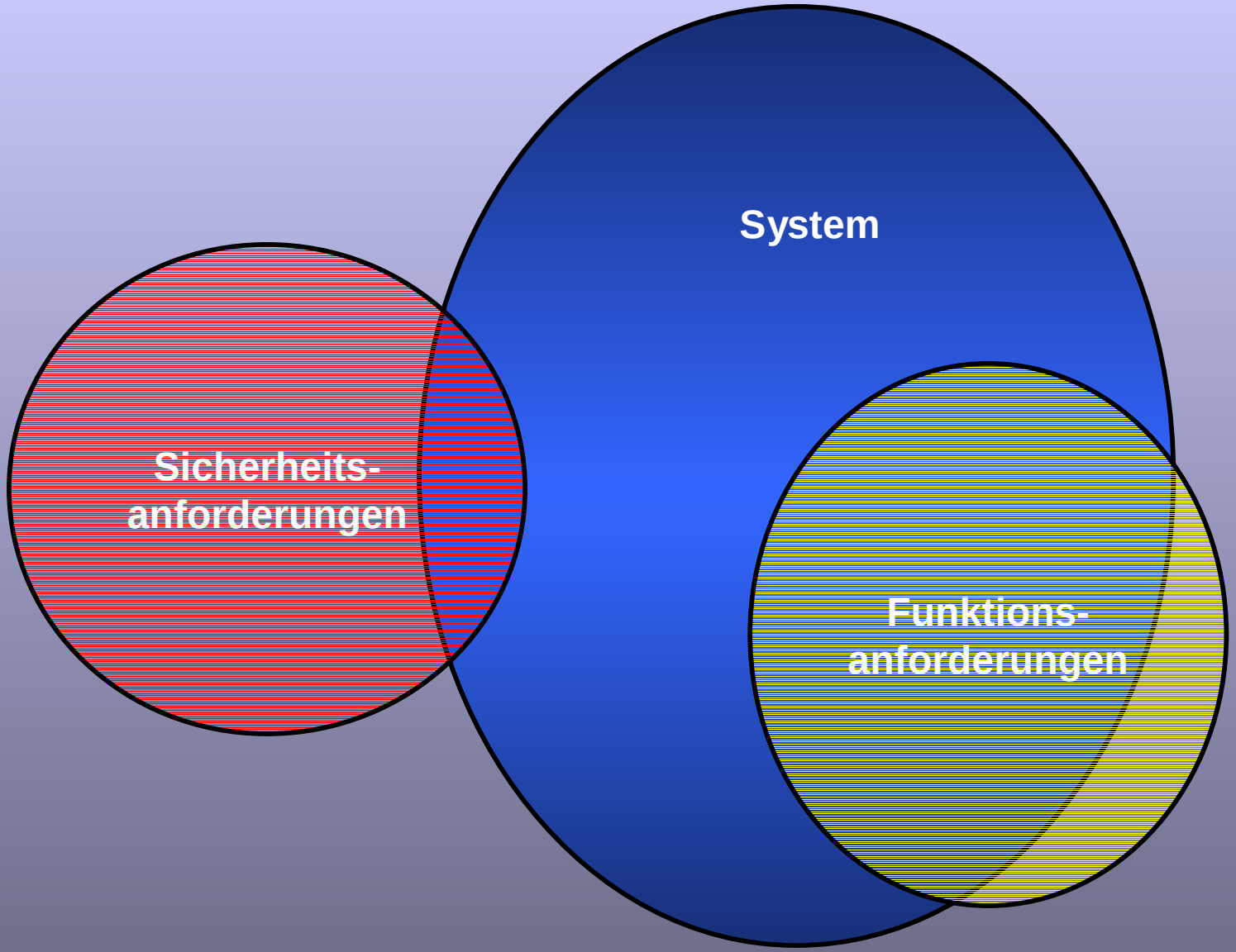
Monika Heiner

BTU Cottbus, Institut für Informatik
Datenstrukturen & Softwarezuverlässigkeit
http://www-dssz.informatik.tu-cottbus.de


System
System-
eigenschaften
Modell
Modell-
eigenschaften



Modell-basierte Systemverifikation, Vorgehen



Funktions- und Sicherheitsanforderungen

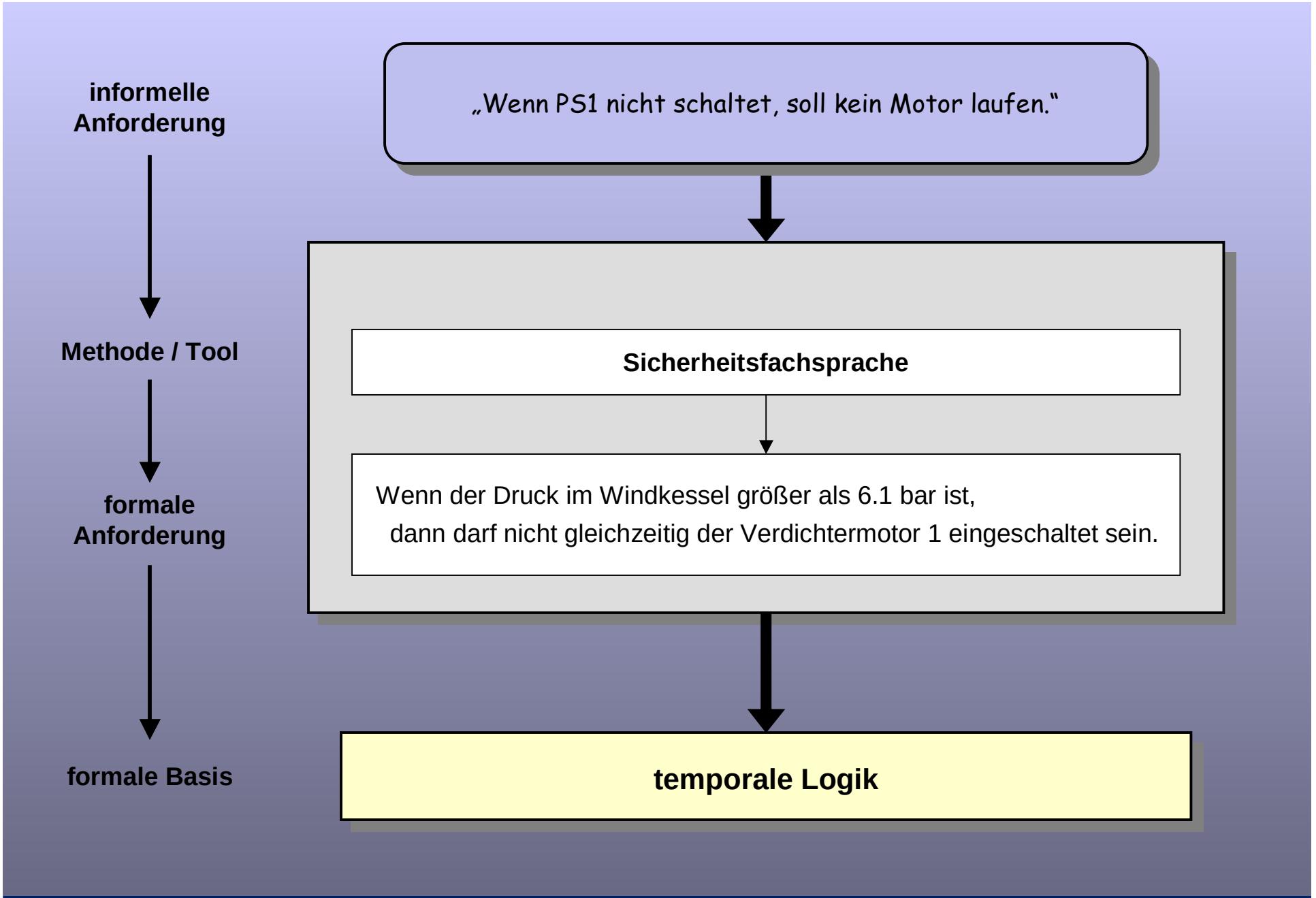
$$AG\ (\ (\ rdy\_in \wedge e\_113\ )\ \rightarrow\ A\ [\ \neg\ rdy\_plc\ U\ (\ rdy\_plc \wedge a\_110 \wedge a\_210\ )\ ]\ )$$

$$AG\ (\ (rdy\_in \wedge e\_12)\ \rightarrow\ A\ (\ \neg\ rdy\_plc\ U\ (rdy\_plc \wedge a\_234)\ )\ )$$

$$AG\ \neg\ (\ rdy\_plc \wedge \neg\ e\_123 \wedge a\_456\ )$$

$$AG\ \neg\ (\ rdy\_plc \wedge \neg\ e\_456 \wedge a\_789\ )$$

$$AG\ (\ (rdy\_in \wedge e\_0815)\ \rightarrow\ AF\ (rdy\_plc \wedge a\_7890)\ )$$


informelle Anforderung
„Wenn PS1 nicht schaltet, soll kein Motor laufen."
Methode / Tool
Sicherheitsfachsprache
formale Anforderung
Wenn der Druck im Windkessel größer als 6.1 bar ist,
dann darf nicht gleichzeitig der Verdichtermotor 1 eingeschaltet sein.
formale Basis
temporale Logik

1. **Allgemeine Anforderungen**
   - durchgängiger Einsatz in allen Phasen des Entwicklungsprozesses
   - **Akzeptanz** und Beherrschbarkeit durch alle Beteiligten
   - Entwicklung verschiedenster technischer Systeme
   - größere **Eindeutigkeit** durch einengende Notation
   - Aufdeckung von Inkonsistenzen, Zwang zu einfachem Entwurf
   - **nachweisbare** und auffindbare Anforderungen
2. **Formale Anforderungen**
   - formale Basis (Semiotik, Syntax, Semantik)
   - Modularität, Erweiterbarkeit
3. **Inhaltliche Anforderungen**
   - Beschreibung des Sollverhaltens -> Funktionsanforderungen
   - Beschreibung des unerwünschten Verhaltens -> Sicherheitsanforderungen
4. **Spezifische Anforderungen der Steuerungstechnik**
   - Berücksichtigung der Arbeitsweise speicherprogrammierbarer Steuerungen

1. Nutzung eines abgegrenzten Teils der **natürlichen Sprache**

2. Verwendung von **Regeln** zur Beschreibung der Anforderungen
   (Implikation: „wenn ..., dann ...“,
   Äquivalenz: „nur wenn ..., dann ...“)

3. Kategorisierung von Anforderungen

   → **Modalparameter** - logische Struktur der Anforderung
   - Forderungen - geforderter Programmablauf
   - Verbote - auszuschließende Reaktionen
   - Möglichkeiten - Freigaben

   → **Zeitparameter** - in welchem Zeitbereich soll diese Anforderung gelten
   - kurzfristig - „gleichzeitig“, „sofort“
     (unter Berücksichtigung des SPS-Zyklus)
   - langfristig - „solange“, „bevor“, „bis“
     (innerhalb eines bestimmten Zeitspanne)

Zeitparameter / Modalparameter	**Zustand** "gleichzeitig"	**Direkt** „unmittelbar"	**Selbst-begrenzt** "solange"	**Fremd-begrenzt** "bis irgendwann"	**Un-begrenzt** "irgendwann"
Einfache Forderung "Wenn B, dann muss F."	✓	✓	✓	✓	✓
Erweiterte Forderung "Nur wenn B, dann muss F."	✓	✓	✓	✓	✓
Einfaches Verbot "Wenn B, dann darf nicht F."	✓		✓	✓	✓
Erweiterte Möglichkeit "Nur wenn B, dann darf F."	✓		✓	✓	✓

B – Bedingung
F – Folgerung, Aktion

Einfache Forderung – Zustand („Wenn B, dann muss gleichzeitig F sein.“)

$$\mathbf{AG\ (\ (rdy\_plc \wedge B) \rightarrow F)}$$

Einfaches Verbot– Zustand („Wenn B, dann darf nicht gleichzeitig F sein.“)

$$\mathbf{AG\ \neg\ (rdy\_plc \wedge B \wedge F)}$$

Einfache Forderung – Direkt („Wenn B, dann muss unmittelbar F werden.“)

$$\mathbf{AG\ (\ (rdy\_in \wedge B) \rightarrow A\ (\ \neg\ rdy\_plc\ U\ (rdy\_plc \wedge F)\ )\ )}$$

Einfache Forderung – Unbegrenzt („Wenn B, dann muss irgendwann F werden.“)

$$\mathbf{AG\ (\ (rdy\_in \wedge B) \rightarrow AF\ (rdy\_plc \wedge F)\ )}$$

B – Bedingung
F – Folgerung, Aktion
rdy_plc, rdy_in - Beobachtungsvariablen

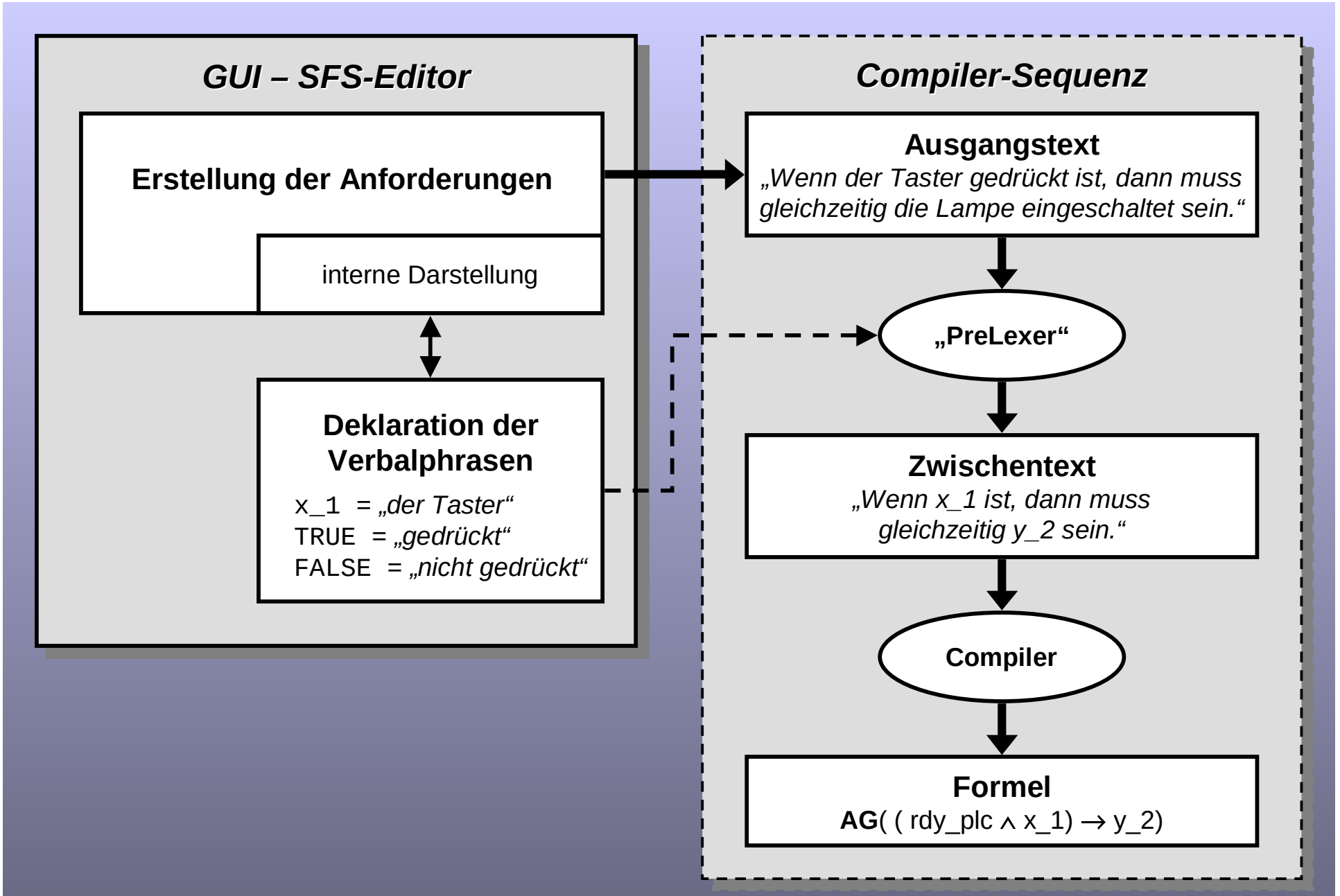
- durch die Auswahl der Parameter werden **allgemeine** Satzstrukturen vorgegeben

  -> insgesamt 18 Satzstrukturen

- Angebot inhaltlich **redundanter** Formulierungsmöglichkeiten je Satzstruktur

- jeder Satzstruktur liegt eine eindeutige temporal-logische Formel zugrunde

- die allgemeinen Satzstrukturen sind mit
  konkreten Aussagen (Verbalphrasen) auszufüllen



Struktur des SFS-Prototypen

informelle Anforderung

„Wenn PS1 nicht schaltet, soll kein Motor laufen."

Formalisierung mit der SFS

Einfaches Verbot - Zustand

Wenn ..... ist, dann darf nicht gleichzeitig ..... sein.

formale Anforderung

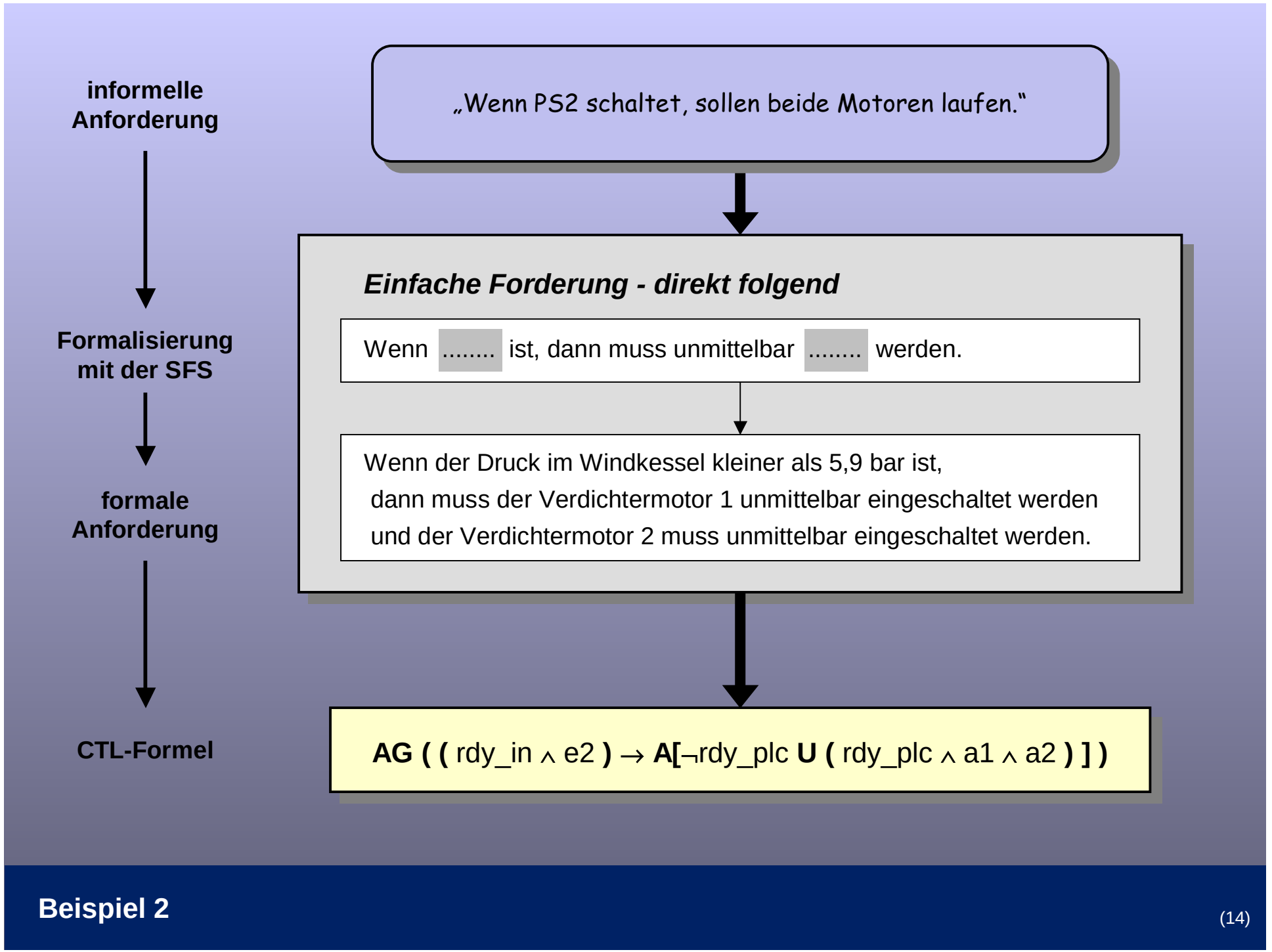
Wenn der Druck im Windkessel größer als 6.1 bar ist,
dann darf nicht gleichzeitig der Verdichtermotor 1 eingeschaltet sein.

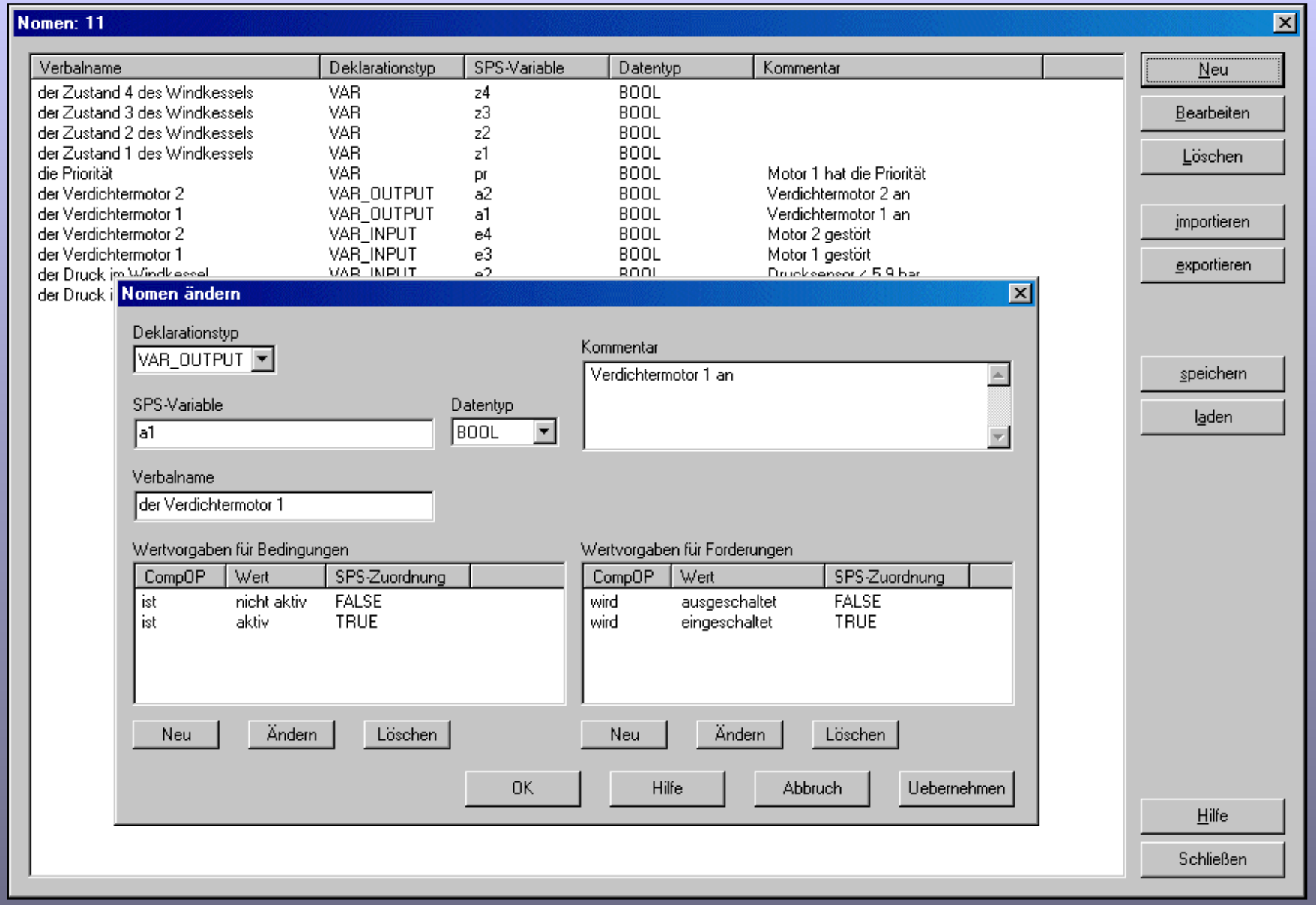
Wenn der Druck im Windkessel größer als 6.1 bar ist,
dann darf nicht gleichzeitig der Verdichtermotor 2 eingeschaltet sein.

CTL-Formel

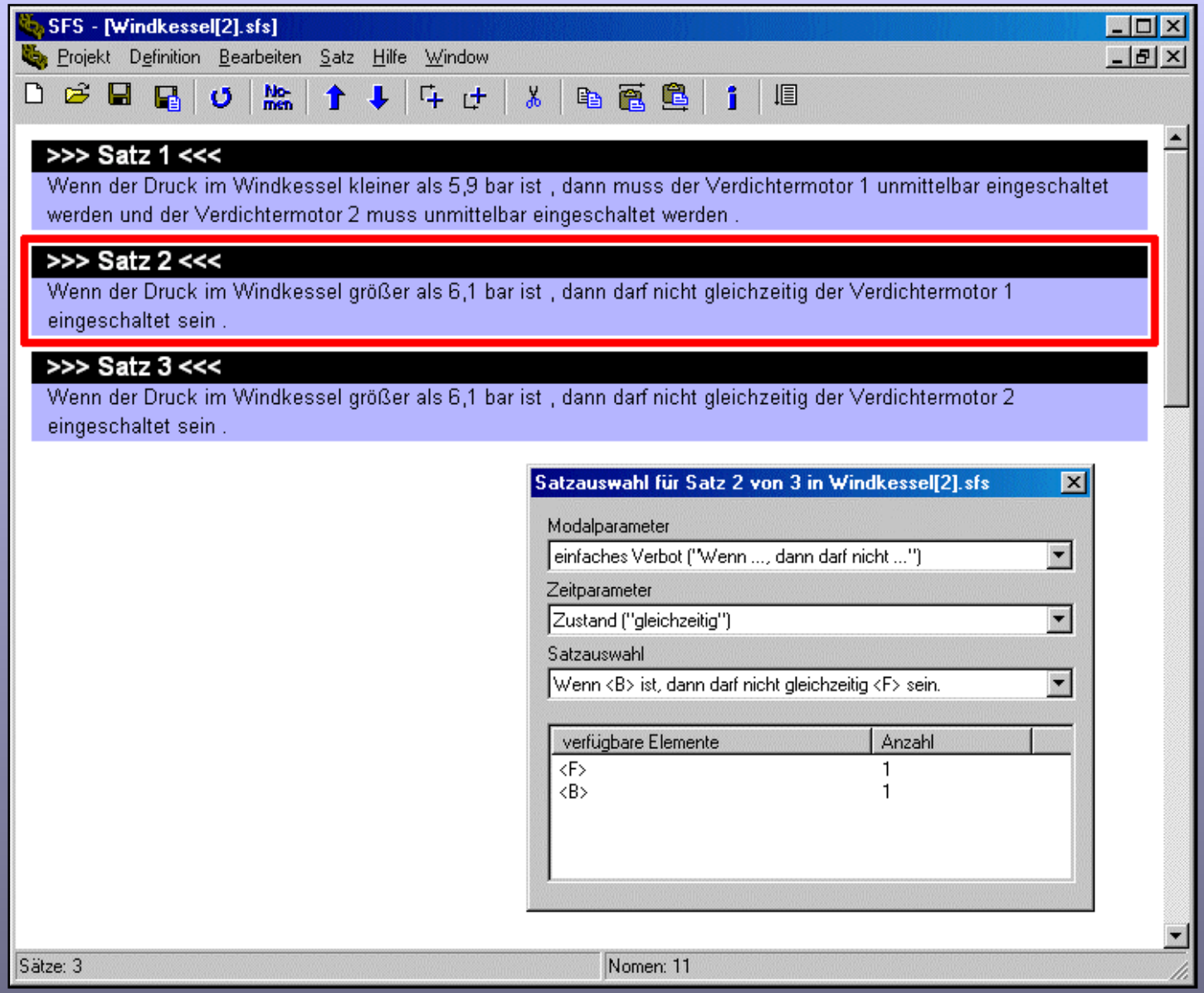
AG ¬(rdy_plc ∧ ¬e1 ∧ a1 **)**

AG ¬(rdy_plc ∧ ¬e1 ∧ a2 **)**


informelle Anforderung
„Wenn PS2 schaltet, sollen beide Motoren laufen."
Formalisierung mit der SFS
Einfache Forderung - direkt folgend
Wenn ..... ist, dann muss unmittelbar ..... werden.
formale Anforderung
Wenn der Druck im Windkessel kleiner als 5,9 bar ist,
dann muss der Verdichtermotor 1 unmittelbar eingeschaltet werden
und der Verdichtermotor 2 muss unmittelbar eingeschaltet werden.
CTL-Formel
AG ((rdy_in ∧ e2) → A[¬rdy_plc U (rdy_plc ∧ a1 ∧ a2)])



SFS-Editor, Vorbereitungsphase – Erstellen der Verbalphrasen



SFS-Editor, Durchführungsphase - Auswahl der Anforderungskategorie

/* >>>Satz 175<<< (einfache Forderung - direkt) */

Wenn "die Zustandsvariable des Ablagebands" "0" ist und "das Startsignal 'Ablageband soll Teil transportieren'" ist "gesetzt" , dann muss "das Startsignal 'Ablageband soll Teil transportieren'" unmittelbar "zurückgesetzt" werden und "die Zustandsvariable des Ablagebands" muss unmittelbar "auf 1 gesetzt" werden und "der Antrieb des Ablagebands" muss unmittelbar "gestartet" werden .

/* >>>Satz 176<<< (einfache Forderung - direkt) */

Wenn "die Zustandsvariable des Ablagebands" "1" ist und "die Lichtschranke am Ende des Ablagebands" ist "blockiert" , dann muss "der Antrieb des Ablagebands" unmittelbar "gestoppt" werden und "die Zustandsvariable des Ablagebands" muss unmittelbar "auf 2 gesetzt" werden und "die Bereitschaftsmeldung 'Ablageband ist zur Übergabe eines Teils bereit'" muss unmittelbar "gesetzt" werden und "die Bereitschaftsmeldung 'Ablageband ist zur Übernahme eines Teils bereit'" muss unmittelbar "gesetzt" werden .

/* >>>Satz 177<<< (einfache Forderung - direkt) */

Wenn "die Zustandsvariable des Ablagebands" "2" ist , dann "die Zustandsvariable des Ablagebands" muss unmittelbar "auf 0 gesetzt" werden .

```
/* >>>Satz 175<<< (einfache Forderung - direkt) */

AG ( (rdy_in & Dbelt_state=0 & Dbelt_run) ->
        A [ !rdy_plc U (rdy_plc & !Dbelt_run & Dbelt_state=1 & DBelt_start) ] )

/* >>>Satz 176<<< (einfache Forderung - direkt) */

AG ( (rdy_in & Dbelt_state=1 & !LB_at_dbelt) ->
        A [ !rdy_plc U (rdy_plc & !DBelt_start & Dbelt_state=2 &
                        Dbelt_ready_give & Dbelt_ready_take) ] )

/* >>>Satz 177<<< (einfache Forderung - direkt) */

AG ( (rdy_in & Dbelt_state=2) ->
        A [ !rdy_plc U (rdy_plc & Dbelt_state=0) ] )
```

Sicherheitsfachsprache

1. **Kommunikationsmittel für die Spezifikation reaktiver Systeme**
   - natürlichsprachlich, allgemeinverständlich
   - eindeutig
   - fachübergreifend
2. **Spezifikation in allen Entwicklungsphasen**
   - Grobentwurf
   - Feinentwurf
3. **Darstellung von Anforderungen**
   - gewünschtes Verhalten, Forderungen
   - unerwünschtes Verhalten, Verbote
4. **Formale Basis**
   - temporale Logik
   - Erweiterbarkeit des Sprachumfangs möglich

- DFG-Projekt
  „Zertifizierung speicherprogrammierbarer Steuerungen“,
  1997 - 2001

- Thomas Mertke:
  Formale Spezifikation reaktiver Systeme mit einer Sicherheitsfachsprache;
  BTU Cottbus, Diss. 10/2003

- http://www-dssz.informatik.tu-cottbus.de
  -> publications
  -> BTU Report 09/01

- monika.heiner@informatik.tu-cottbus.de