

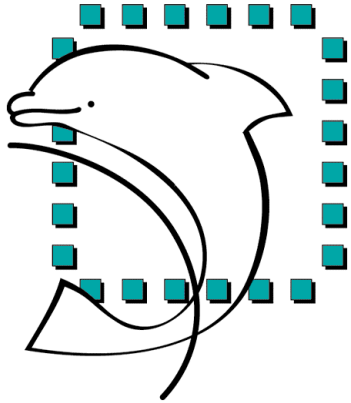
# ASIM 2002 – VHDL-AMS Simulation

Simulation eines Kfz-Energienetzes mit elektrischen und elektromechanischen VHDL-AMS Komponentenmodellen

**Dirk Dammers**

**Dolphin Integration GmbH**



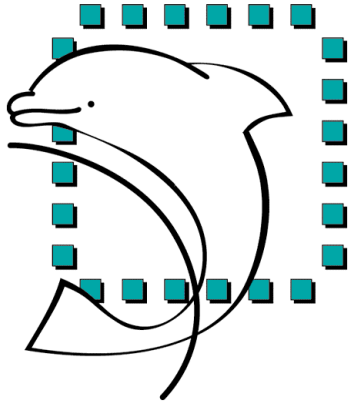


# ASIM 2002 – VHDL-AMS Simulation

## Was ist gegeben?

- **Netzliste eines KFZ-Energienetzes**
- **Modelle diverser elektrischer und elektromechanischer Komponenten**
- **Verschiedene Testumgebungen**
- **Modelle sind in VHDL-AMS implementiert**



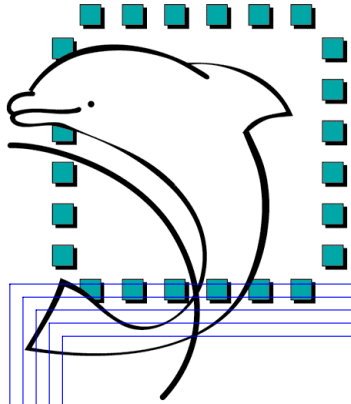


# ASIM 2002 – VHDL-AMS Simulation

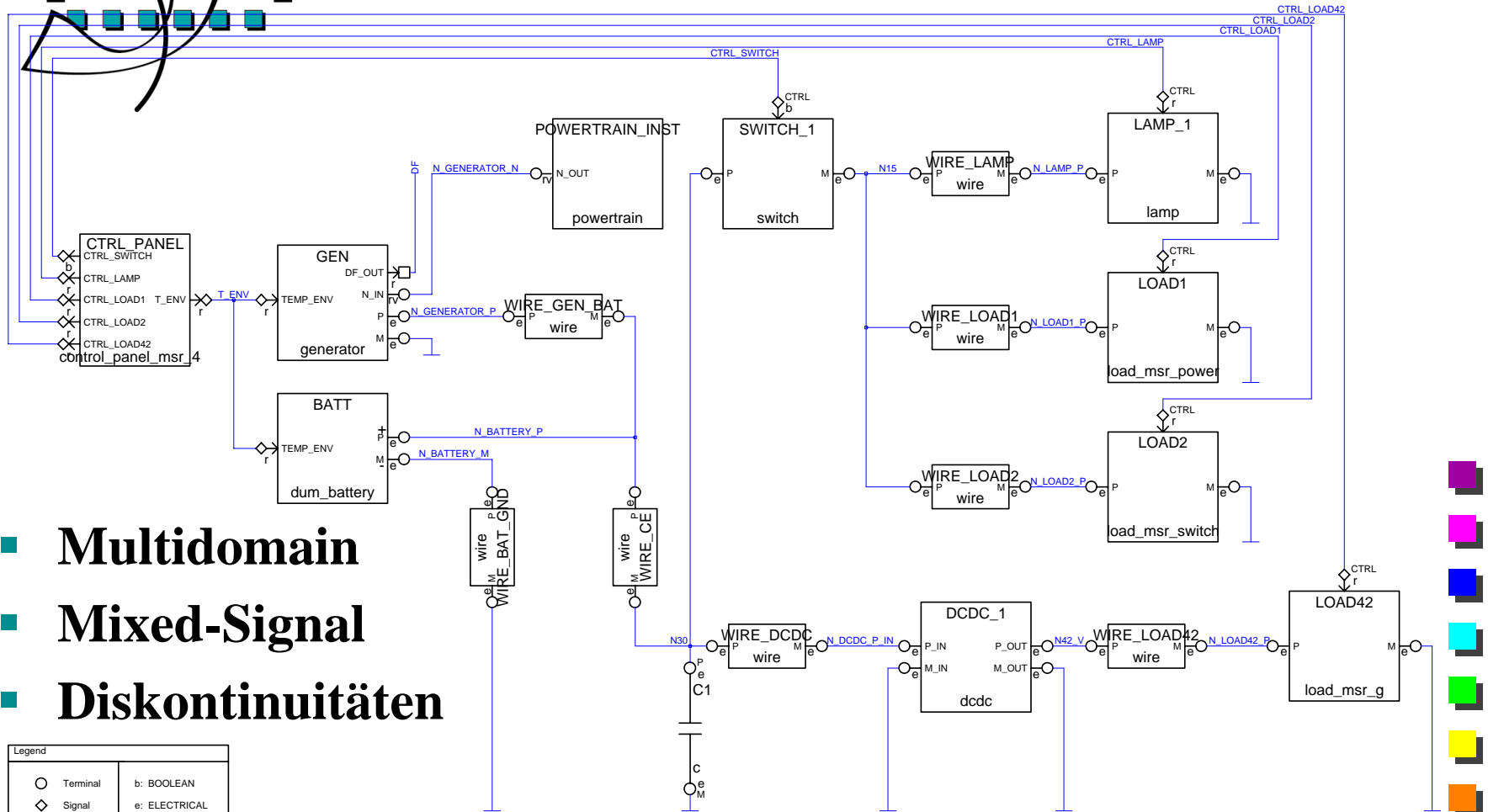
## Was bietet VHDL-AMS?

- **VHDL-AMS strikte Obermenge von IEEE 1076-1993**
  - Voller VHDL 1076-1993 Syntax und Semantik Support
- **Analoge und Mixed-Signal Simulationen**
- **Hierarchische Beschreibungsmöglichkeiten**
- **Modellierung auf unterschiedlichen Abstraktionsebenen**
- **Behandlung von ein- und multidomain Systemen**
- **Management von Diskontinuitäten während der Simulation im analogen Schaltungsteil**



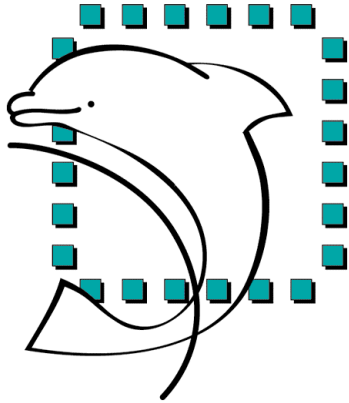


## ASIM 2002 – VHDL-AMS Simulation Was wurde verwendet?

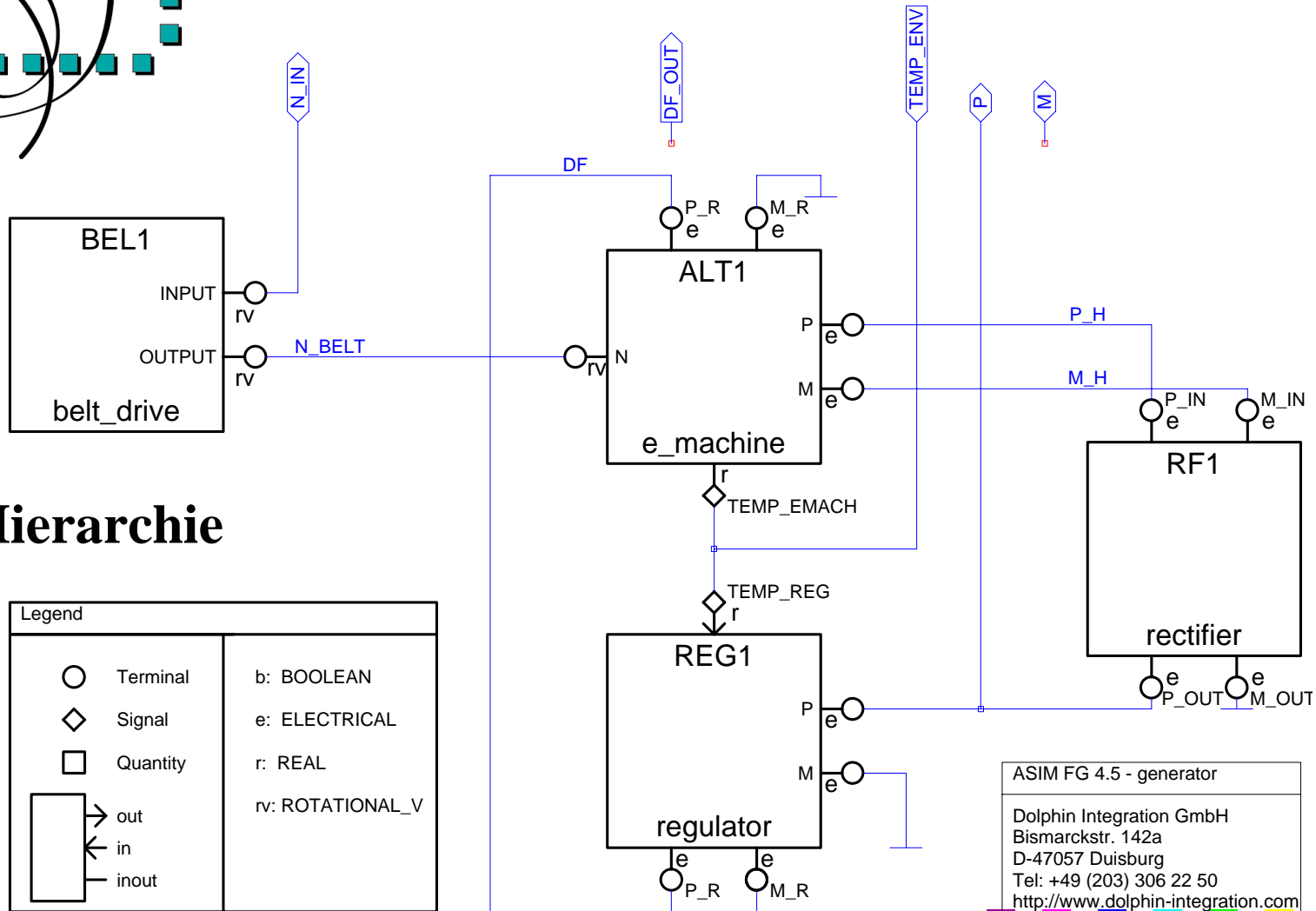


- Multidomain
- Mixed-Signal
- Diskontinuitäten

Legend	
○	b: BOOLEAN
◇	e: ELECTRICAL
□	r: REAL
⊞	rv: ROTATIONAL_V
→	out
←	in
↔	inout



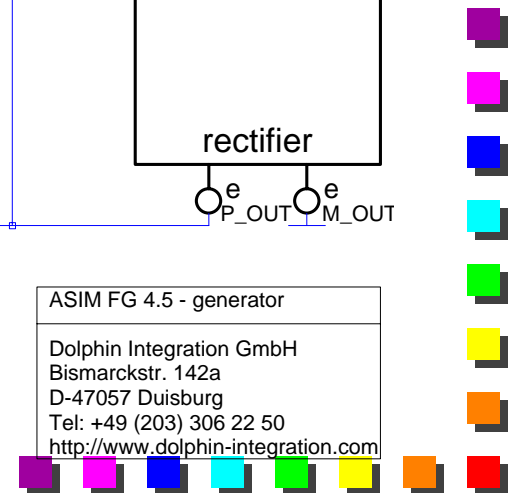
## ASIM 2002 – VHDL-AMS Simulation Was wurde verwendet?

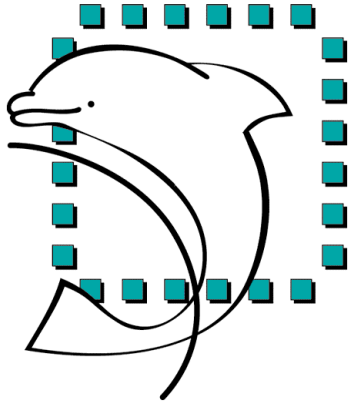


### ■ Hierarchie

Legend	
○ Terminal	b: BOOLEAN
◇ Signal	e: ELECTRICAL
□ Quantity	r: REAL
	rv: ROTATIONAL_V

ASIM FG 4.5 - generator  
 Dolphin Integration GmbH  
 Bismarckstr. 142a  
 D-47057 Duisburg  
 Tel: +49 (203) 306 22 50  
<http://www.dolphin-integration.com>



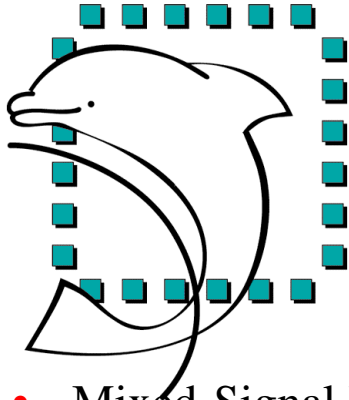


# ASIM 2002 – VHDL-AMS Simulation

## Was sind die Erweiterungen?

- **Erweiterte strukturelle Semantik**
  - Konservative Semantik für die Modellierung physikalischer Systeme (z.B. Kirchhoffs Gesetz für elektrische Stromkreise)
  - Nichtkonservative Semantik für die abstrakte Modellierung (Signalflußbeschreibungen)
  - Mixed-Signal Interface
  
- **Mixed-Signal Semantik**
  - Vereinheitlichtes Modell der Zeit für eine konsistente Synchronisation von gemischtem ereignisgesteuerten/kontinuierlichem Verhalten
  - Mixed-Signal Initialisierung und Simulationszyklus
  - Mixed-Signal Beschreibung des Verhaltens
  
- **Unterstützung des Frequenzbereichs**
  - Kleinsignal- und Rauschfrequenz Modellierung und Simulation





# ASIM 2002 – VHDL-AMS Simulation

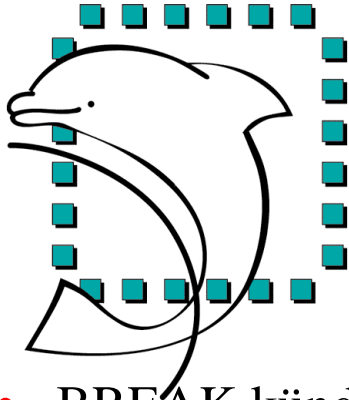
## Was wurde verwendet?

- Mixed-Signal Interface
- Konservative Semantik
- Terminal als Interface ist hier dem Nature „electrical“ zugehörig
- Nichtkonservative Semantik, hier Signal „ctrl“ mit Mode „in“
- Quantity repräsentiert den Potentialwert zwischen, bzw. den Flußwert an Terminals; es ist eine Unbekannte im DAE Gleichungssystem
- Mixed-Signal Beschreibung des Verhaltens
- In jedem Entity muß  $\Sigma$  through Quantities +  $\Sigma$  freie Quantities +  $\Sigma$  Interface Quantities mit Mode Out =  $\Sigma$  Gleichungen
- Behandlung von Diskontinuitäten mit der Break-Anweisung

```
ENTITY switch IS
PORT (
    TERMINAL p,m: electrical;
    SIGNAL ctrl: IN boolean);
END entity;
```

```
ARCHITECTURE msr OF switch IS
    QUANTITY v ACROSS i THROUGH p TO m;
BEGIN
    IF ctrl USE
        v == i * r_on;
    ELSE
        i == v * g_off;
    END use;
BREAK ON ctrl;
END architecture;
```





# ASIM 2002 – VHDL-AMS Simulation

## Wie wird BREAK verwendet?

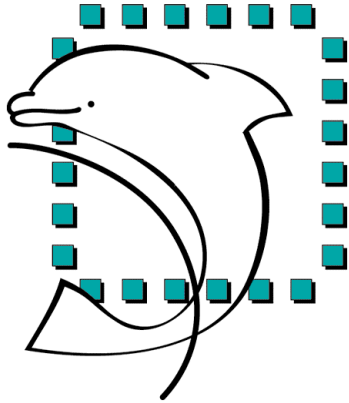
- BREAK kündigt dem Simulator Diskontinuitäten an
- Analog Solver reinitialisiert am Anfang des nächsten kontinuierlichen Intervalls
- Verwendung: BREAK ON „Ereignis“ oder BREAK WHEN „Bedingung“
- **Ein VHDL-AMS Modell, das eine Diskontinuität bei einem Quantity zum Zeitpunkt T verursacht und zu diesem Zeitpunkt T keine BREAK Anweisung ausführt, ist fehlerhaft!**
  - Ausgenommen davon sind Diskontinuitäten verursacht von S'Ramp, S'Slew und Q'Slew

```
ENTITY regulator IS
  GENERIC (vsoll : real := 14.0);
  PORT (TERMINAL p, m: electrical;);
END ENTITY regulator;
```

```
ARCHITECTURE msr OF regulator IS
  QUANTITY vin ACROSS p TO m;
  QUANTITY dv: REAL;
BEGIN
  dv == vin-vsoll_temp;
  IF dv'ABOVE(0.0) USE
    unfilterd == 0.0;
  ELSE
    IF NOT dv'ABOVE(-0.1) USE
      unfilterd == 1.0;
    ELSE
      unfilterd == dv*(1.0/(-0.1));
    END USE;
  END USE;
  BREAK ON dv'ABOVE(0.0), dv'ABOVE(-0.1);
END ARCHITECTURE msr;
```







# ASIM 2002 – VHDL-AMS Simulation

## Wie werden Diskontinuitäten behandelt?

- Ein VHDL-AMS Modell, das eine Diskontinuität bei einem Quantity zum Zeitpunkt T verursacht und zu diesem Zeitpunkt T keine BREAK Anweisung ausführt, ist fehlerhaft!
  - **Ausgenommen davon sind Diskontinuitäten verursacht von S'Ramp, S'Slew und Q'Slew**
- Das Ergebnis diskontinuierlicher Funktionen kann Signalen zugewiesen werden
- Diese Signale können mit Hilfe den Attributen SLEW oder RAMP in der Bestimmungsgleichung eines Quantities verwendet werden

```

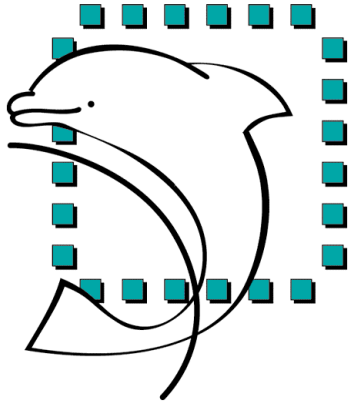
ENTITY e_machine IS
  PORT (SIGNAL temp_emach: IN real);
END ENTITY e_machine;

ARCHITECTURE msr OF e_machine IS
  QUANTITY vout ACROSS iout THROUGH m TO p;
  ...
  SIGNAL s_temp_beh : REAL := 0.0;
  SIGNAL s_imax : REAL := 0.0;
  FUNCTION temp_beh (te: REAL) RETURN REAL IS
  ...
  FUNCTION imax (rs: REAL) RETURN REAL IS
  ...
  BEGIN
    p_temp_beh: PROCESS (temp_emach) IS
      BEGIN
        s_temp_beh <= temp_beh(temp_emach);
      END PROCESS p_temp_beh;

    p_imax: PROCESS IS
      BEGIN
        s_imax <= imax(radps_to_rpm*n_inp);
        WAIT FOR 0.1sec;           -- Attention!
      END PROCESS p_imax;

    iout == reg_in*inenn*s_imax'SLEW*s_temp_beh'SLEW;
    ...
  END ARCHITECTURE msr;
  
```

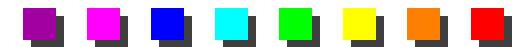


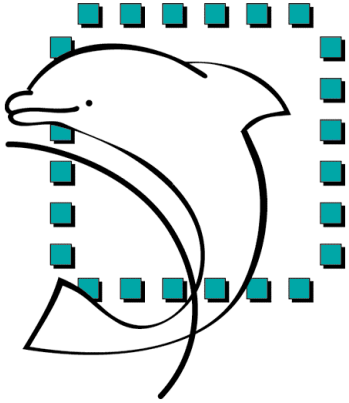


# ASIM 2002 – VHDL-AMS Simulation

## Was sind Anfangsbedingungen?

- **Eine Anfangsbedingung spezifiziert den Wert eines Quantities am Beginn eines kontinuierlichen Intervalls**
  - Am Anfang einer zeitlichen Simulation
  - Nach Diskontinuitäten
- **Anfangsbedingungen können mit der Break Anweisung spezifiziert werden**
- **Anfangsbedingungen ersetzen implizierte Gleichungen. Eine Anfangsbedingung für Q ersetzt die Gleichungen:**
  - $Q'_{\text{Dot}} == 0$  während der Arbeitspunktanalyse
  - $Q == Q(T-)$  bei der Reinitialisierung nach einer Unstetigkeitsstelle
- **Falls eine Anfangsbedingung für ein Quantity Q spezifiziert werden muß, dessen Ableitung Q'Dot nicht explizit angegeben ist, muß die Gleichung aufgeführt werden, welche ersetzt werden soll**





# ASIM 2002 – VHDL-AMS Simulation

## Wie werden Anfangsbedingungen verwendet?

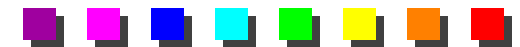
```
ENTITY c IS
  GENERIC (c_nom: REAL := 1.0; v_init: REAL := REAL'LOW);
  PORT (TERMINAL p,m : ELECTRICAL);
END ENTITY c;
```

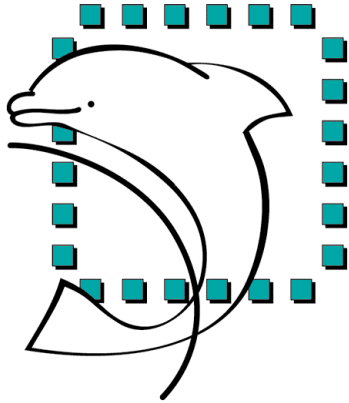
**v==v\_init**  
**ersetzt v'DOT == 0**

```
ARCHITECTURE one OF c IS
  QUANTITY v ACROSS i THROUGH p TO m;
  BEGIN
    BREAK v => v_init WHEN v_init /= REAL'LOW;
    i == c_nom * v'DOT;
  END ARCHITECTURE one;
```

**v==v\_init**  
**ersetzt q'DOT == 0**

```
ARCHITECTURE two OF c IS
  QUANTITY v ACROSS i THROUGH p TO m;
  QUANTITY q: charge;
  BEGIN
    BREAK FOR q USE v => v_init WHEN v_init /= REAL'LOW;
    q == c_nom * v;
    i == q'DOT;
  END ARCHITECTURE two;
```





# ASIM 2002 – VHDL-AMS Simulation

## Was wurde verwendet?

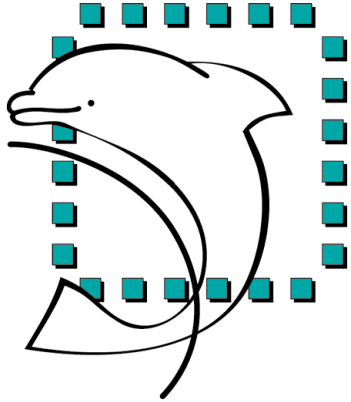
### ■ VHDL-AMS spezifische Schlüsselworte

ACROSS	BREAK	INTEGER	LIMIT
NATURE	NOISE	NOW	PROCEDURAL
QUANTITY	REAL	REFERENCE	SPECTRUM
SUBNATURE	TERMINAL	THROUGH	TOLERANCE

### ■ VHDL-AMS spezifische Attribute

'ABOVE	'ACROSS	'CONTRIBUTION	'DELAYED
'DOT	'INTEG	'LTF	'RAMP
'REFERENCE	'SLEW	'THROUGH	
'TOLERANCE	'ZOH	'ZTF	





# ASIM 2002 – VHDL-AMS Simulation

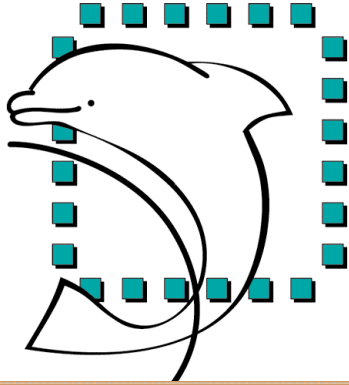
## Welche Änderungen waren notwendig?

- **SMASH unterstützt momentan nur eine Architektur pro ENTITY**

**=> folgende Architekturen wurden Umbenannt:**

- control\_panel(msr\_1) => control\_panel\_msr\_1(msr\_1)
- control\_panel(msr\_2) => control\_panel\_msr\_2(msr\_2)
- control\_panel(msr\_3) => control\_panel\_msr\_3(msr\_3)
- control\_panel(msr\_4) => control\_panel\_msr\_4(msr\_4)
- load(msr\_g) => load\_msr\_g(msr\_g)
- load(msr\_power) => load\_msr\_power(msr\_power)
- load(msr\_switch) => load\_msr\_switch(msr\_switch)

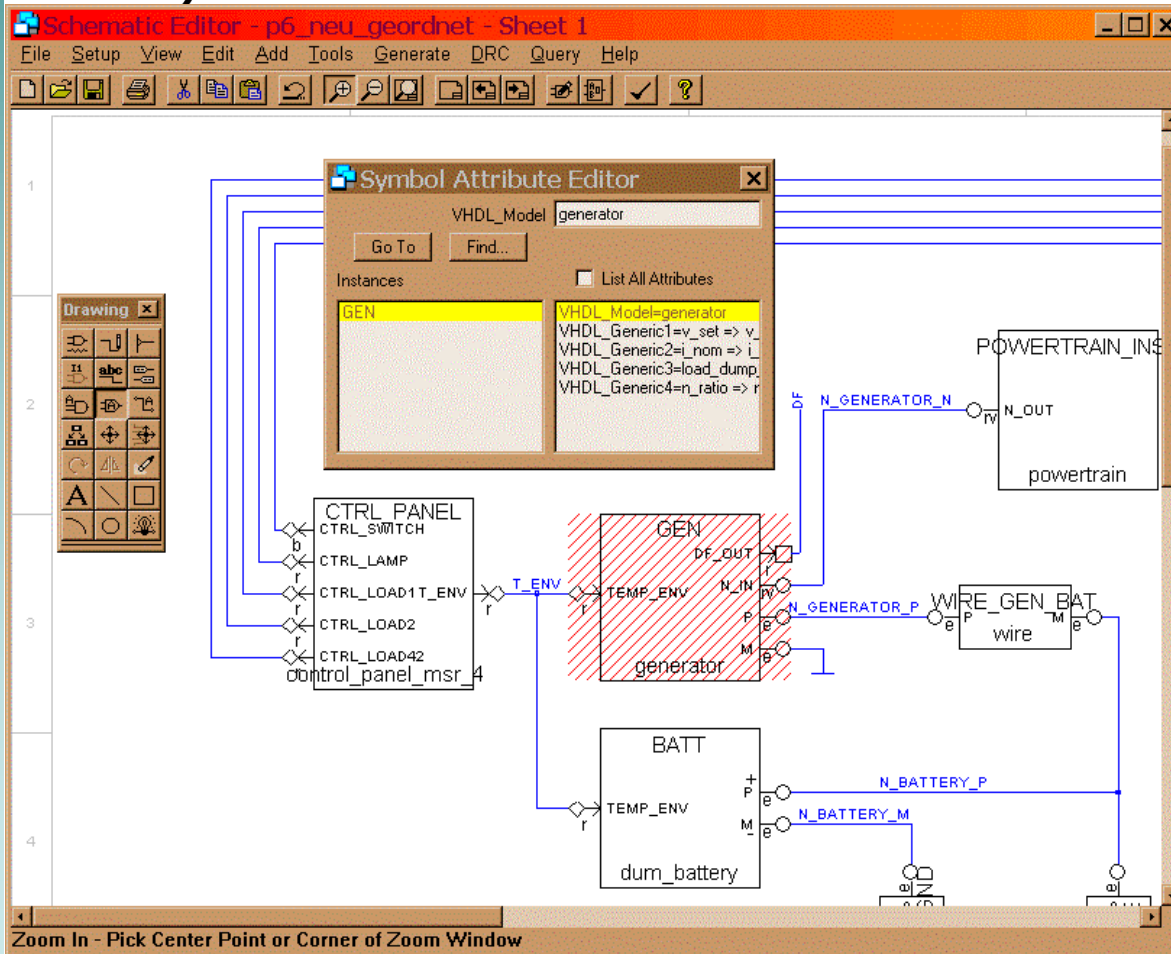


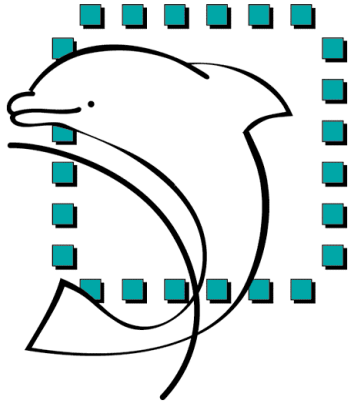


# ASIM 2002 – VHDL-AMS Simulation Welche Erweiterungen sind möglich?

## Integration in einen Schaltplanneditor

- Domänen-  
übergreifende  
visuelle  
Modellierung
- Einfachere Kontrolle  
der Anschlüsse
- Übersichtliche  
Parametrierung
- Automatische  
Generation der  
Netzliste





# ASIM 2002 – VHDL-AMS Simulation

## Wie werden die Modelle eingebunden?

- **SMASH verwendet zwei Dateien: die .NSX und die .PAT Datei**

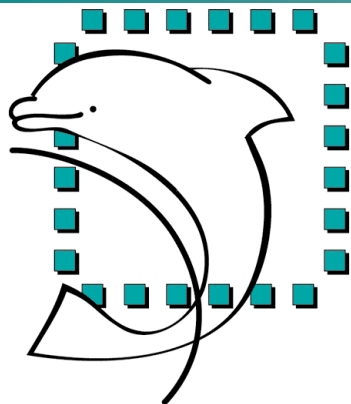
```
>>> VHDL
`include "powernet_6.vhd"
```

**powernet\_6.nsx**

```
.VHDL SET KIND=AMS
.VHDL compile library=ieee source=../packages/math_real.vhd
.VHDL compile library=work source=../packages/standard_packages.vhd
.VHDL compile library=work source=../models/belt_drive.vhd
.VHDL compile library=work source=../models/e_machine.vhd
...
.VHDL elaborate entity=power_net unit=msr
```

**powernet\_6.pat**





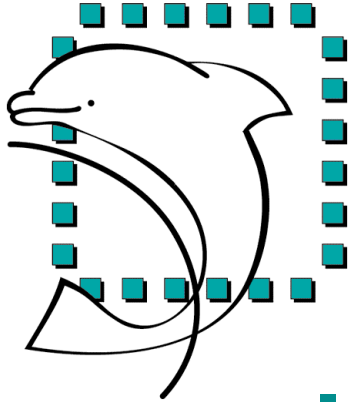
# ASIM 2002 – VHDL-AMS Simulation

## Wie wird das Modell ausgeführt?

- **Ein VHDL-AMS Modell besteht aus**
  - digitalem Anteil => Prozesse + digitaler Simulationskern
  - analogen Anteil => Gleichungssatz + Analog Solver
- **Zwei Phasen**
  - Initialisierung: Berechnung des Arbeitspunktes
  - Simulation: Zeit-, Kleinsignal- bzw. Rauschverhalten
- **Reduzierung auf VHDL 1076 Initialisierung und Simulation, falls das Modell keine Quantities enthält**
- **Nur der Analog Solver wird ausgeführt, falls im Modell keine Signale vorhanden sind**





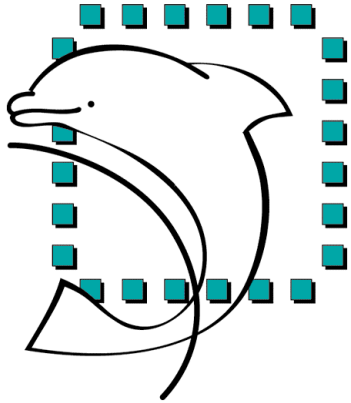


# ASIM 2002 – VHDL-AMS Simulation

## Wie lange dauerte die Kompilation?

- **Kompilationszeit: ca. 51 Sekunden**
  - CPU: Pentium III 933MHz
  - RAM: 256MB
  - OS: Windows 2000
  - Simulator: SMASH 4.3.4ß5
  
- **Elaborated entity POWER\_NET unit MSR**
  - Analog elements summary:
    - 176 VHDL-AMS quantities
    - 176 VHDL-AMS expression(s)
  - Digital elements summary:
    - 17 Output pin(s)
    - 17 VHDL signal(s)
    - 16 VHDL port(s)
    - 17 VHDL driver(s)
    - 10 VHDL dispatcher process(es)
    - 6 VHDL process(es)





# ASIM 2002 – VHDL-AMS Simulation

## Was wurde simuliert?

POWERTRAIN\_INST.VEL

GEN.ALTI.IOUT

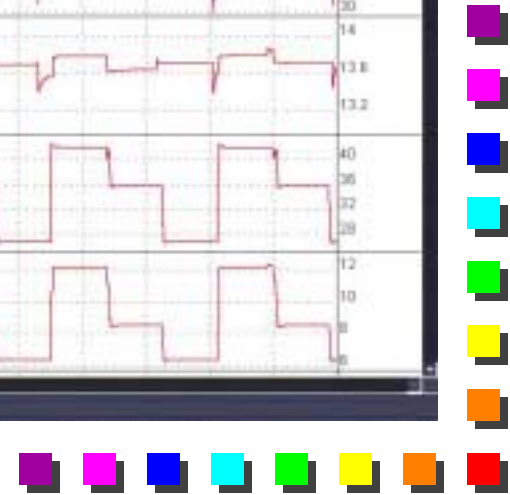
DCDC\_1.V\_IN

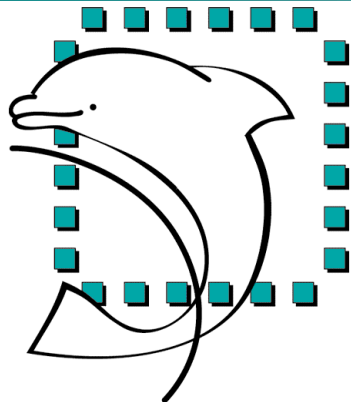
DCDC\_1.V\_OUT

DCDC\_1.V\_TANK



powernet\_6





# ASIM 2002 – VHDL-AMS Simulation

## Wie lange dauerte die Simulation?

### ■ Einstellungen

- Simulierter Zeitraum : 0 Sekunden bis 600 Sekunden
- Minimale interne Zeitschrittweite : 1e-015 sec
- Maximale interne Zeitschrittweite : 0.01 (0.1) sec
- Genauigkeit der Spannungsberechnung : 1e-006 V
- Relative Genauigkeit : 0.001
- Genauigkeit der Stromberechnung : 1e-009 A

### ■ Simulationszeit: ca. 65 Sekunden (8 Sekunden)

- Integrationsalgorithmus : BACKWARD EULER

### ■ Simulationszeit: ca. 164 Sekunden (152 Sekunden)

- Integrationsalgorithmus : BDF

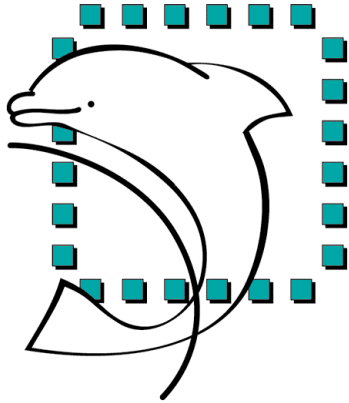
### ■ Simulationszeit: ca. 68 Sekunden (8 Sekunden)

- Integrationsalgorithmus : TRAPEZOIDAL

### ■ Simulationszeit: ca. 75 Sekunden (11 Sekunden)

- Integrationsalgorithmus : IMPROVED TRAPEZOIDAL





# ASIM 2002 – VHDL-AMS Simulation

## Zusammenfassung

- **VHDL-AMS ist gleichwertig geeignet zur Beschreibung von elektrischem und nichtelektrischem Verhalten**
  - Ermöglicht die Modellierung des kompletten elektromechanischen Systems in nur einer Modellierungssprache
  - Vereinfacht die Simulation kompletter Multi-Domain Systeme in einem Simulator
  
- **Integration in einen Schaltplaneditor**
  - Übersichtlichkeit durch visuelle Modellierung
  - Vereinfacht die Integration von mechanischen Komponenten in eine elektronische Schaltung
  - Zeitersparnis und Fehlerreduzierung durch automatische Netzlistengeneration

