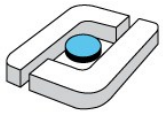
From Business Modeling to Verified Applications

Christian Ammann, Stephan Kleuker und Elke Pulvermüller



Hochschule Osnabrück
University of Applied Sciences

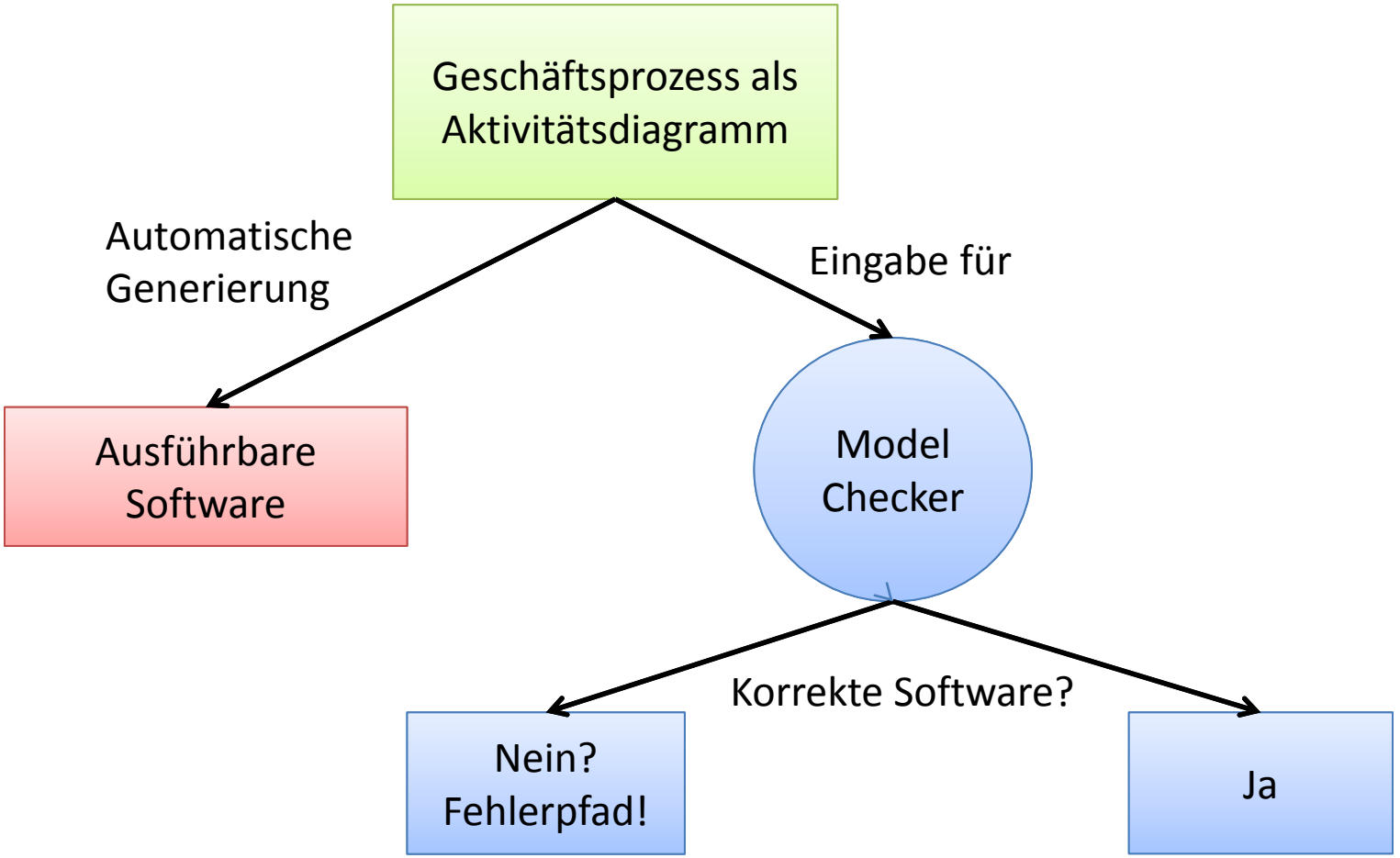
Teil des Forschungsprojekts
„KoverJa“ und gefördert vom:

Bundesministerium
für Bildung
und Forschung

Gliederung

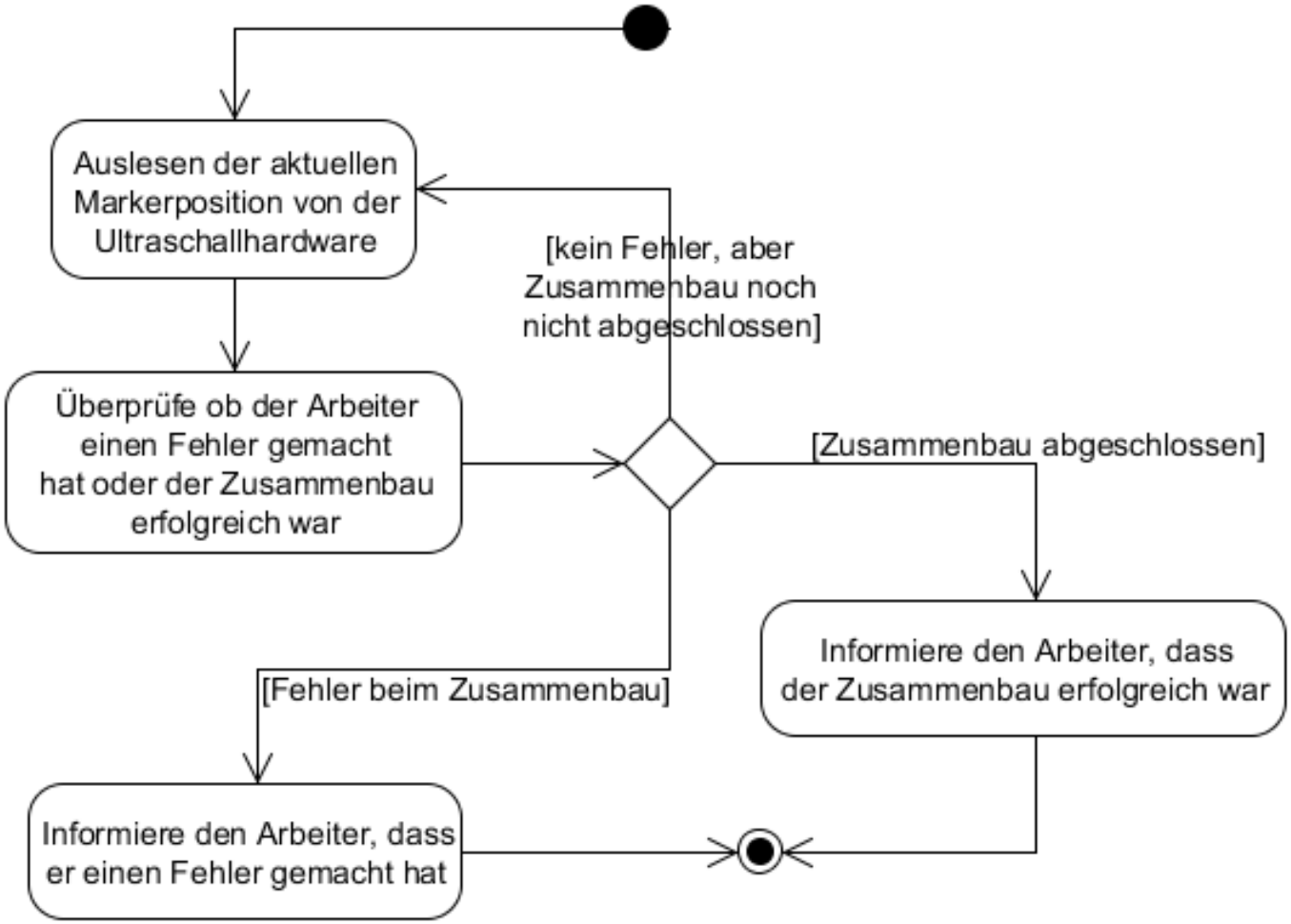
1. Einleitung
2. Fallstudie: AssyControl
3. Verifikation von Aktivitätsdiagrammen
4. Optimierung des Verifikationsprozesses

Idee: Automatische Generierung und Verifikation von Software

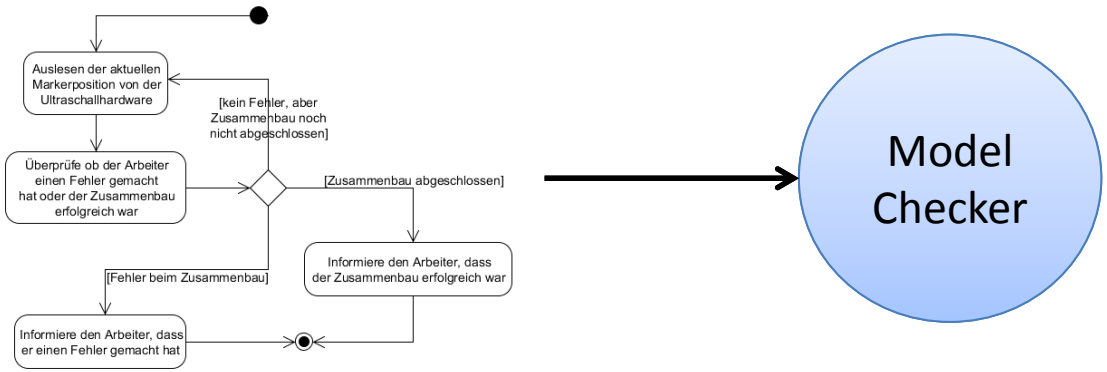


Fallstudie: Entwicklung von Software für AssyControl

AssyControl als Aktivitätsdiagramm



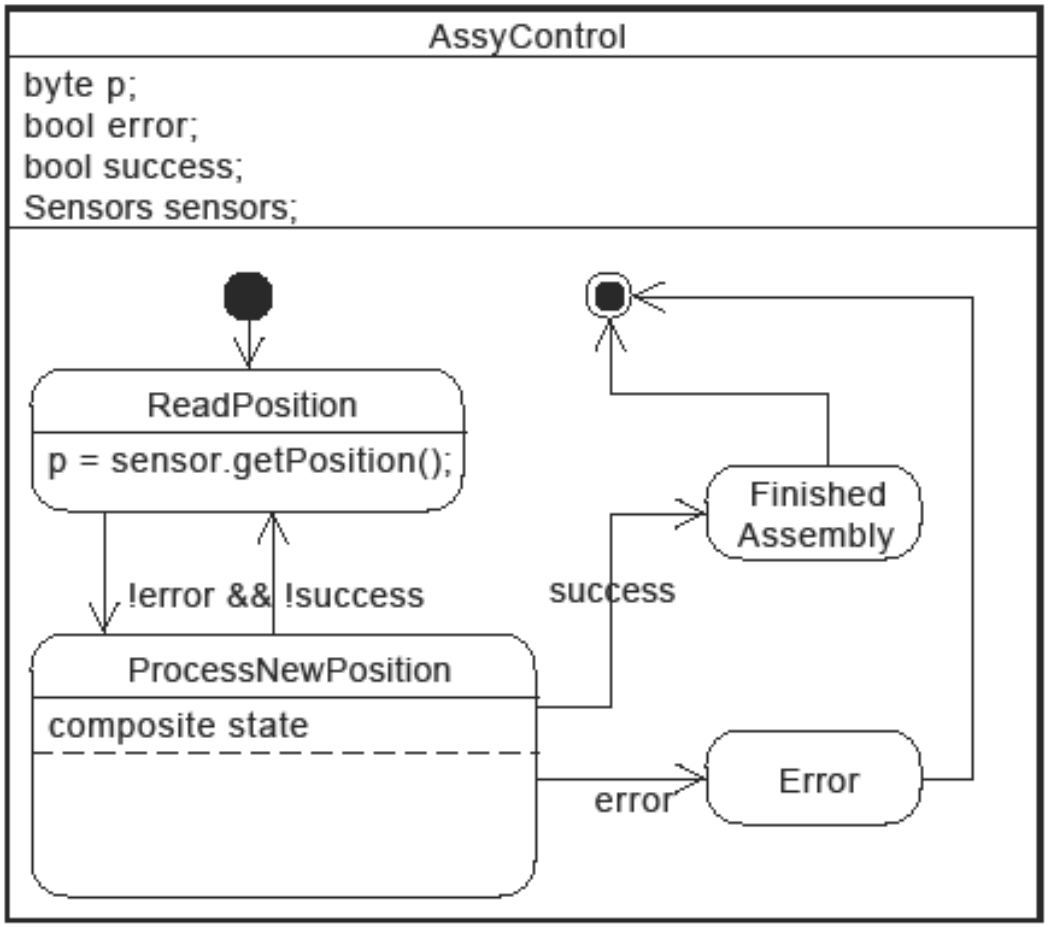
Verifikation von Aktivitätsdiagrammen



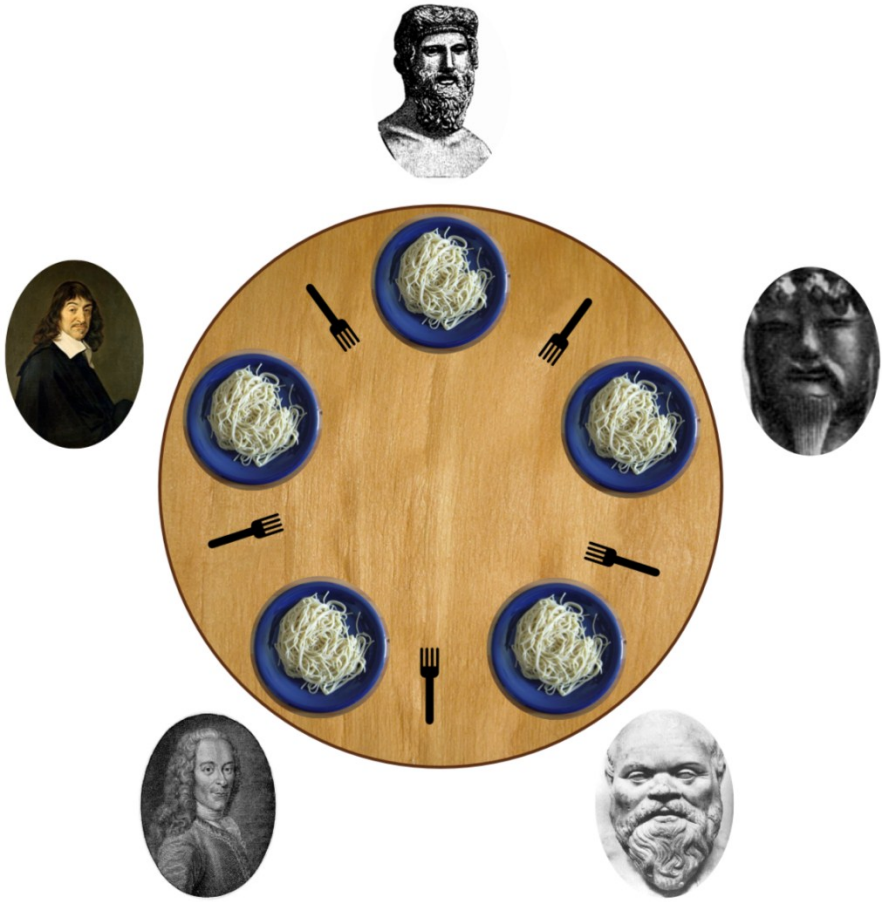
Probleme:

1. Anreicherung des Modells mit Anforderungen und Implementierungsdetails
2. Überführung in eine Model Checker Eingabesprache

Problem 1: Anreicherung des Modells durch Überführung in ein Statechart



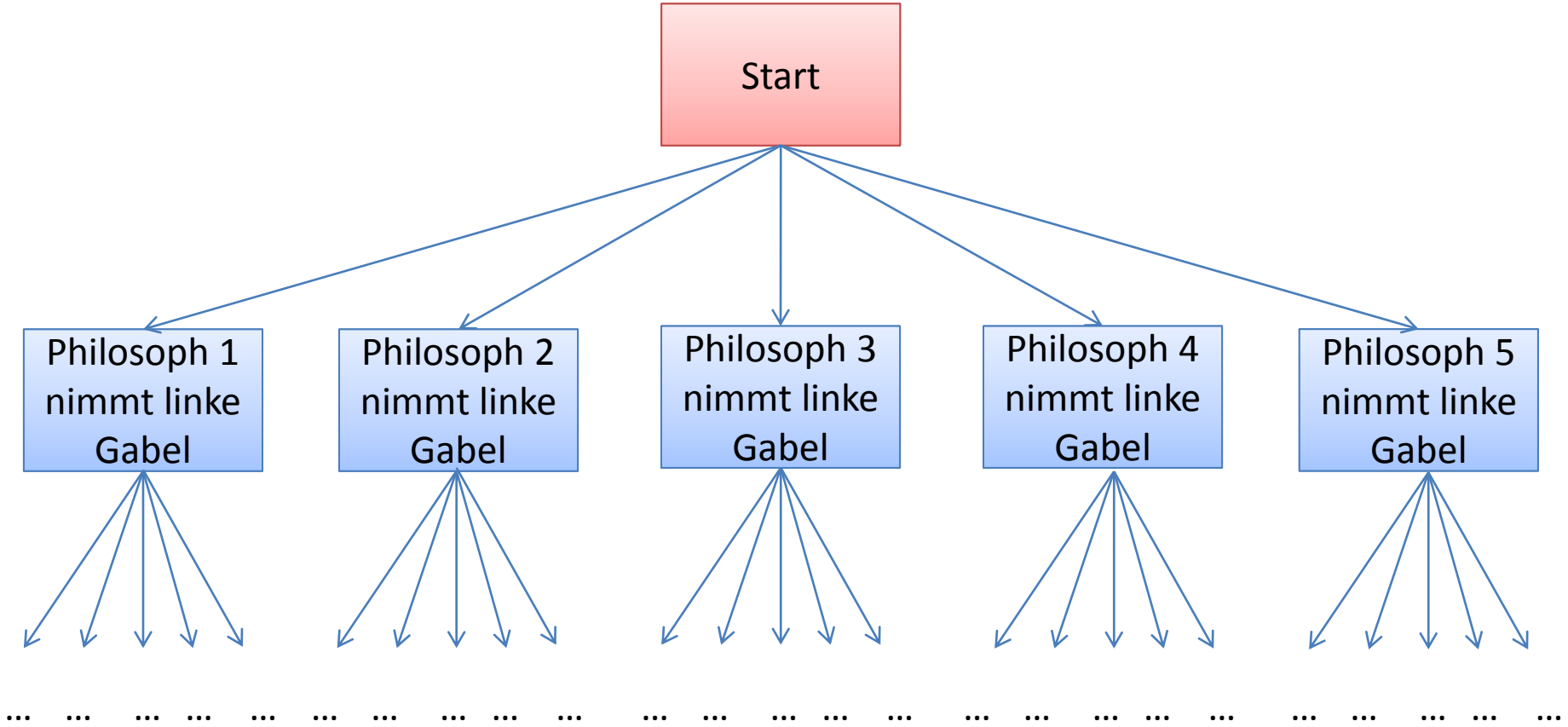
Problem 2: Überführung in die Eingabesprache Promela am Beispiel des Philosophenproblems



Problem 2: Überführung in die Eingabesprache Promela am Beispiel des Philosophenproblems

```
process Philosoph1{
    do(true){
        if(linke gabel){
            nimm(linke gabel);
            if(rechte gabel){
                nimm(rechte gabel);
                essen();
                zurücklegen(rechte gabel);
                zurücklegen(linke gabel);
            }
        }
    }
}
```

Funktionsweise eines Model Checkers und „State Space Explosion Problem“



Optimierungen bei der Übersetzung

Eigenschafts-Erhaltene Abstraktion

```
Enum Position{Arbeitsfläche, Box1, Box2, …}

Position getPosition(){
   float a = lese Signallaufzeit 1 von Hardware();
   float b = lese Signallaufzeit 2 von Hardware();
   float c = lese Signallaufzeit 3 von Hardware();
   Point p = trilateration(a,b,c);

   if( arbeitsfläche(p) ) return Arbeitsfläche;
   if( box1(p) ) return Box1;
   if( box2(p) ) return Box2;
   …
}
```

Optimierungen bei der Übersetzung

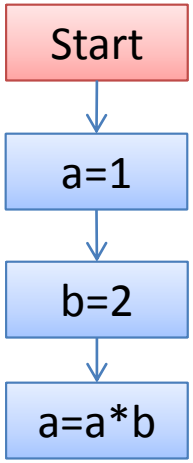
Eigenschafts-Erhaltene Abstraktion

```
Enum Position{Arbeitsfläche, Box1, Box2, …}

Position getPosition(){
   int a = random();

   if( a==0 ) return Arbeitsfläche;
   else if ( a==1 ) return Box1;
   else if ( a==2 ) return Box2;
   …
}
```

Optimierungen bei der Übersetzung

Statement Merging

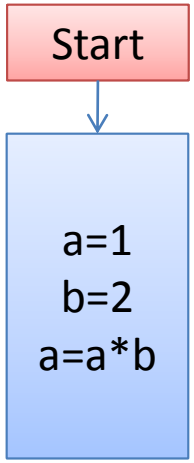
Eingabesprache	Model Checker

byte a;
byte b;
 	
a=1;
b=2;
a=a*b
	

Optimierungen bei der Übersetzung

Statement Merging

Eingabesprache	Model Checker

byte a;
byte b;
 	
d_step{
  a=1;
  b=2;
  a=a*b
}
	

Zusammenfassung

- Fallstudie AssyControl modelliert als Aktivitätsdiagramm
- Manuelle Überführung von Aktivitätsdiagramm in ein Statechart
- Automatisierte Überführung von Statecharts in die Model Checker Eingabesprache Promela
- Optimierungen in Promela zur Reduzierung der Laufzeit des Model Checkers

Ende