

Translation DEV&DESS Model into Linear Hybrid Automata

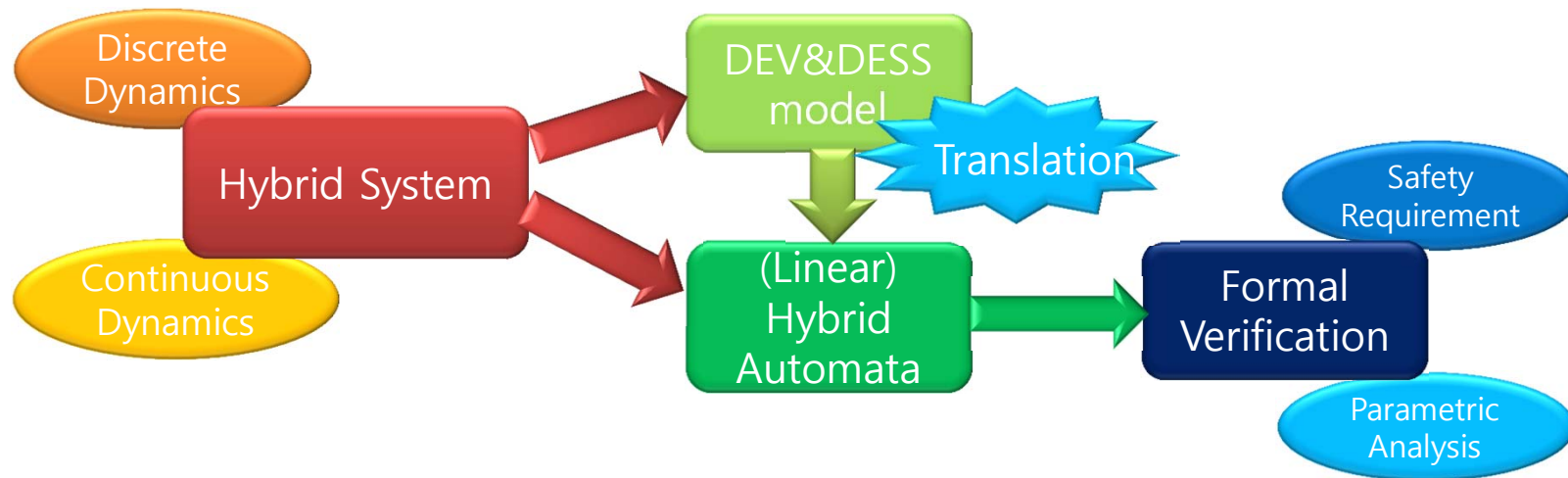
고려대학교
고신뢰소프트웨어 연구실
최 한

연구 배경

- Hybrid system
 - 이산 행위와 연속 행위가 서로 긴밀하게 반응하는 시스템
 - 의료, 항공, 국방 등 다양한 분야에서 사용
- Hybrid system 모델링 기법
 - 정형 검증 목적 기법 - high-level 모델링, 비결정적(non-deterministic) 행위
 - ✓ Timed automata, (Linear) Hybrid automata ...
 - 시뮬레이션 목적 기법 - low-level 모델링, 결정적(deterministic) 행위
 - ✓ Discrete Event & Differential Equation Specified System (DEV&DESS) formalism, Hybrid state transition, Simulink ...
- 시뮬레이션 목적 기법만을 사용하여 시스템을 분석할 경우, 안전 요구사항과 같은 중요 속성을 보장할 수 없다는 단점이 존재
- 두 가지 기법을 모두 사용할 경우, 시간·비용적 문제와 두 모델 사이의 검증 문제가 발생

연구 목표

- 시뮬레이션 목적 기법인 DEV&DESS 모델을 (Linear) Hybrid Automata로 변환, 기존의 도구를 활용하여 모델의 중요 속성 검증
 - 시스템 안전성 확보
 - 시간, 비용 절감
 - 기 개발된 정형 검증 도구 활용 가능
 - 두 모델 사이의 검증 작업 불필요



연구 진행 상황

- 모델 변환 전제 조건 및 모델 변환 규칙 수립

연구 진행 상황

- 모델 변환 전제 조건 및 모델 변환 규칙 수립

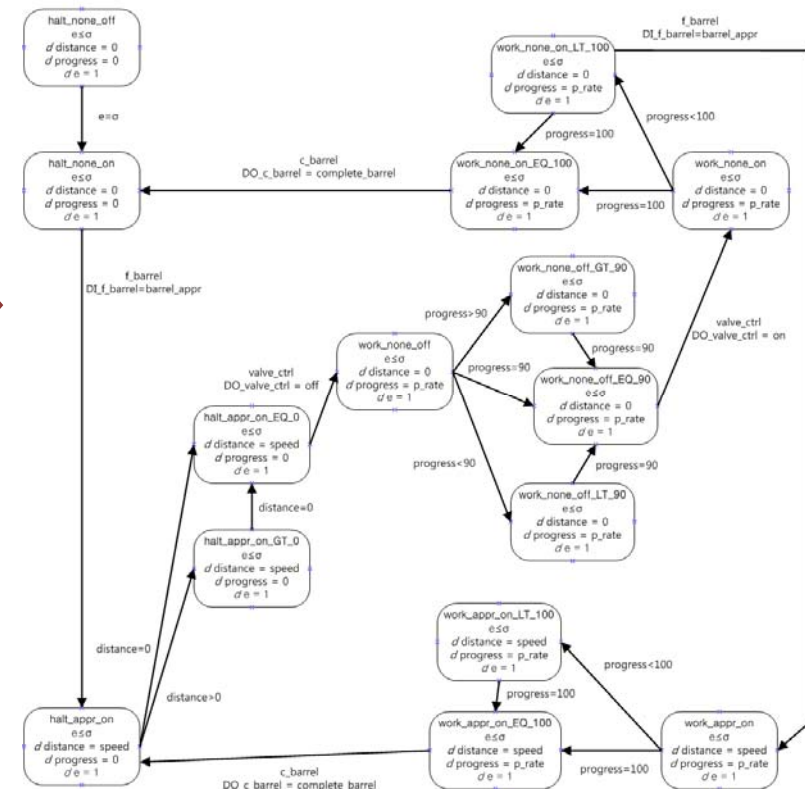
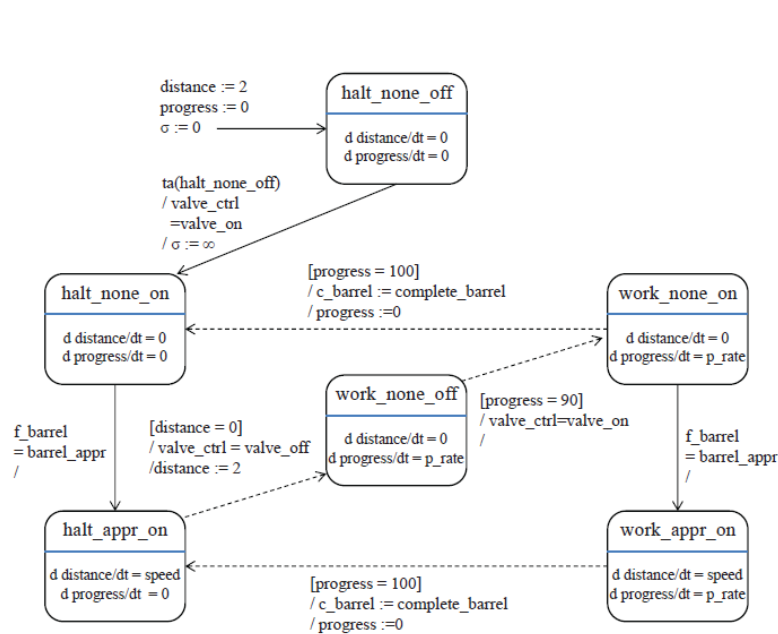
1. $X^{\text{discr}}, X^{\text{cont}}, Y^{\text{discr}}, Y^{\text{cont}}, S^{\text{discr}}, S^{\text{cont}}$ 에 대응되는 variable set X 생성 및 e, σ 추가
2. Discrete input automata(DIA) 생성
3. DEV&DESS model의 phase에 1:1로 대응되는 state(control mode) set V 생성 및 flow condition function 생성
 1. e, σ 에 대한 invariant condition 추가
 2. e 에 대한 flow condition ($de = 1$) 추가
4. δ_{int} (state event) 및 λ^{discr} 변환
5. δ_{ext} 변환
6. δ_{int} (time event) 및 λ^{discr} 변환
7. initial condition 변환

연구 진행 상황

- 모델 변환 전제 조건 및 모델 변환 규칙 수립
- 예제 모델을 통한 변환 유효성 및 효과 확인 (HyTech 사용)
 - ※ HyTech
 - Hybrid Automata의 제약 클래스인 Linear Hybrid Automata를 입력으로 모델의 reachable state를 탐색
 - 간단한 키워드를 통해 안전 요구사항 정형 검증

연구 진행 상황

- 모델 변환 전제 조건 및 모델 변환 규칙 수립
- 예제 모델을 통한 변환 유효성 및 효과 확인 (HyTech 사용)



연구 진행 상황

- 모델 변환 전제 조건 및 모델 변환 규칙 수립
- 예제 모델을 통한 변환 유효성 및 효과 확인 (HyTech 사용)
- HyTech 도구 입력 자동 변환 도구 개발

연구 진행 상황

- 모델 변환 전제 조건 및 모델 변환 규칙 수립
- 예제 모델을 통한 변환 유효성 및 효과 확인
- HyTech 도구 입력 자동 변환 도구 개발

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2012 rel. 2 sp1 (x64) (http://www.altova.com) by Han Choi (DSLAB) -->
<!-- based on v.1.21 -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xds="AtomicDEVDESS" targetNamespace="AtomicDEVDESS" elem
  X_dlskr
  -->
  <xsd:element name="AtomicDEVDESS">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Xdiscr" type="xsd:VarRangeType"/>
        <xsd:element name="Xcont" type="xsd:VarRangeType"/>
        <xsd:element name="Xdiucr" type="xsd:VarRangeType"/>
        <xsd:element name="Ycont" type="xsd:VarRangeType"/>
        <xsd:element name="phases" type="xsd:PhasesType"/>
        <xsd:element name="stateVars" type="xsd:StateVarsType"/>
        <xsd:element name="deltaExts" type="xsd:DeltaExtsType"/>
        <xsd:element name="deltaInts" type="xsd:DeltaIntsType"/>
        <xsd:element name="Cints" type="xsd:CintsType"/>
        <xsd:element name="lambdaDiscrs" type="xsd:lambdaDiscrsType"/>
        <xsd:element name="lambdaConts" type="xsd:lambdaContsType"/>
        <xsd:element name="initValues" type="xsd:InitialPredicatesType"/>
      </xsd:sequence>
      <xsd:attribute name="modelName" type="xsd:NMTOKEN" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <!-- name type -->
  <xsd:simpleType name="NameType">
    <xsd:annotation>
      <xsd:documentation>
        'IN_ALL_PHASES' is a keyword using in lambda expression.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:NMTOKEN"/>
  </xsd:simpleType>
  <!-- input, output, state variable의 범위 -->
  <xsd:attributeGroup name="ValueType">
    <xsd:attribute name="type" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="number"/>
          <xsd:enumeration value="word"/>
          <xsd:enumeration value="varName"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:attributeGroup>
  <xsd:complexType name="UnaryType">
    <xsd:sequence>
      <xsd:element name="value">
        <xsd:annotation>
```



```
C:\Users\WDSLAB\workspace\WDEVDESStoLHAWBF.hy - Notepad++
파일(F) 편집(E) 찾기(S) 보기(V) 인코딩(N) 언어(L) 설정(T) 매크로 실행 플러그인 창 ?
sf_packing.hy sf_bfs.hy BF.hy
1 define(INFINITY, 1000)
2 define(INFINITY_REACHED, 1000)
3 define(ON, 0)
4 define(OFF, 1)
5 define(FULL_FILLED_BARREL, 0)
6 define(OPEN, 0)
7 define(CLOSING, 1)
8 define(CLOSED, 2)
9
10
11 var
12 BF_switch
13 , DO_BF_barrel
14 , BF_valve
15 , BF_sigma
16 , DI_BF_on_off
17 , DI_BF_test
18 : discrete;
19 BF_e
20 , BF_contents
21 : analog;
22
23 automaton BF
24 syncLabs: BF_test, BF_barrel, BF_on_off;
25 initially BF valve=CLOSED & BF switch=OFF & BF contents=0 & closed off;
```

연구 진행 상황

- 모델 변환 전제 조건 및 모델 변환 규칙 수립
- 예제 모델을 통한 변환 유효성 및 효과 확인
- HyTech 도구 입력 자동 변환 도구 개발
- 현재 변환 규칙 정형화 및 논문 작성 중

감사합니다

