CPS 기술 & LVCG 기술 융합 공동 워크샵 2013
르네상스 서울호텔
2013.10.23

고신뢰 CPS 검증을 위한 정형 검증 기술

유 준 범

Dependable Software Laboratory
(http://dslab.konkuk.ac.kr)
KONKUK University

발표 내용

- CPS 모델링 기법
  - DEV&DESS
  - Hybrid Automata
  - ECML(ETRI CPS Modeling Language)
- CPS 정형검증 기법
- ECML을 위한 정형검증 기법 연구
- 향후 연구 방향

CPS 모델링 기법

CPS(Cyber Physical System)

- 이산적인(Discrete) 행위와 연속적인(Continuous) 행위가 결합되어 서로 긴밀히 상호 반응하는 시스템
  → **Hybrid System**

- 대표적인 Hybrid System 모델링 기법
  - DEV&DESS (Discrete EVent & Differential Equation Specified System)
    - 목적: 시뮬레이션
    - Low-level modeling (기본 semantics만 정의됨)
    - 시뮬레이션 도구: Simulink 등
  - (Linear) Hybrid Automata
    - 목적: 정형검증/분석 (+ 시뮬레이션)
    - High-level modeling
    - 정형검증/분석 도구: HyTech, SpaceEx, PHAver 등
  - ECML (ETRI CPS Modeling Language)
    - ETRI 개발
    - DEV&DESS를 기반으로 다양한 모델링 편의 사항 및 시뮬레이션 구현 (EcoSIM, EcoPOD)
    - 정형검증/분석 도구: HyTech, SpaceEx 사용 가능

DEV&DESS

Discrete Dynamics
시스템의 operation mode (즉, state)를 표현

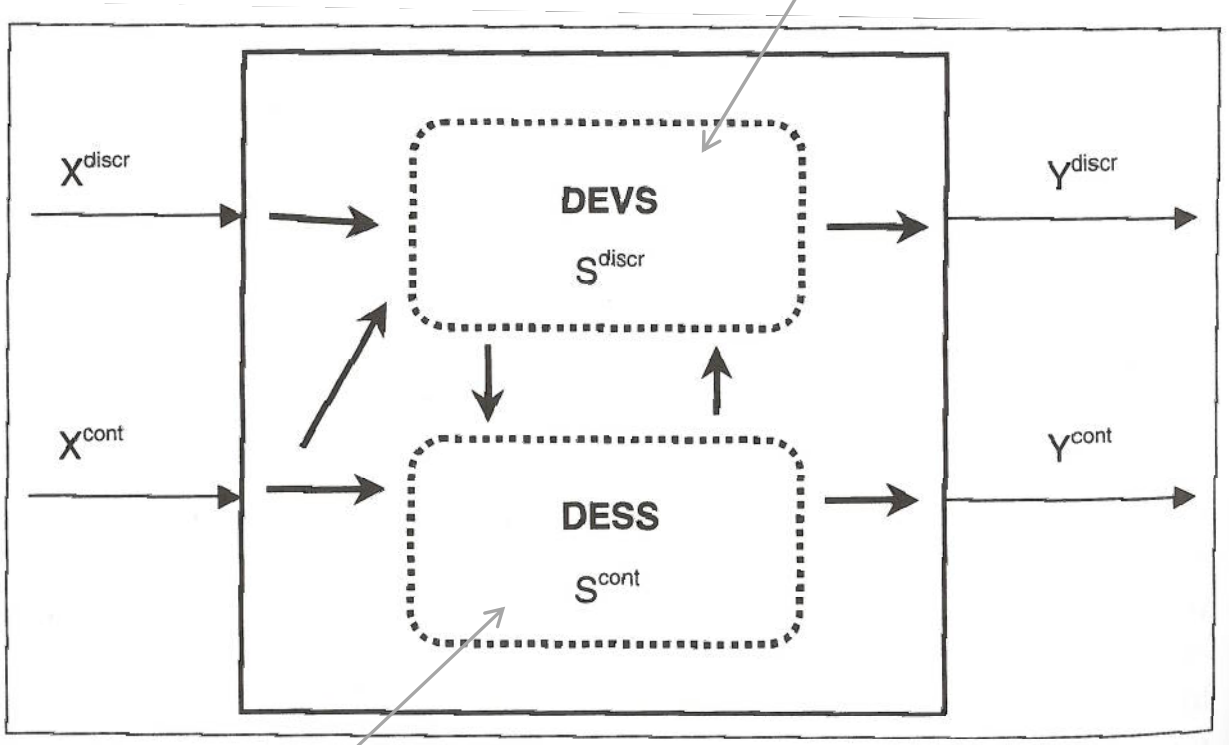


Figure 2 DEVS and DESS combined model.

Continuous Dynamics
시스템 내부간 및 시스템과 외부의 상호작용을 표현

Rate of Change
미분방정식(Differential equation)으로 표현

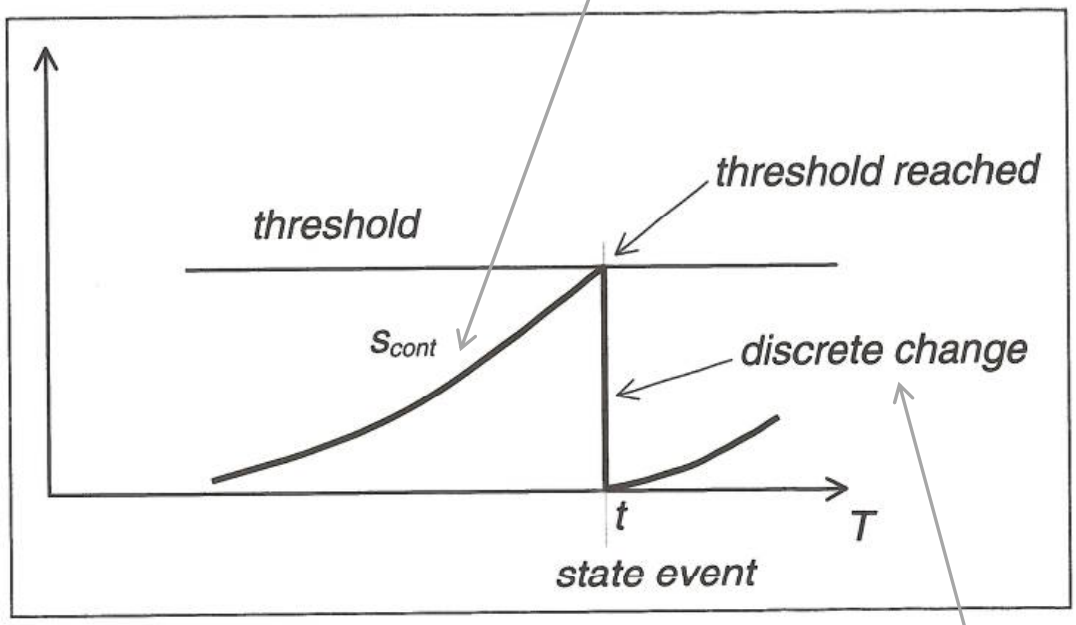


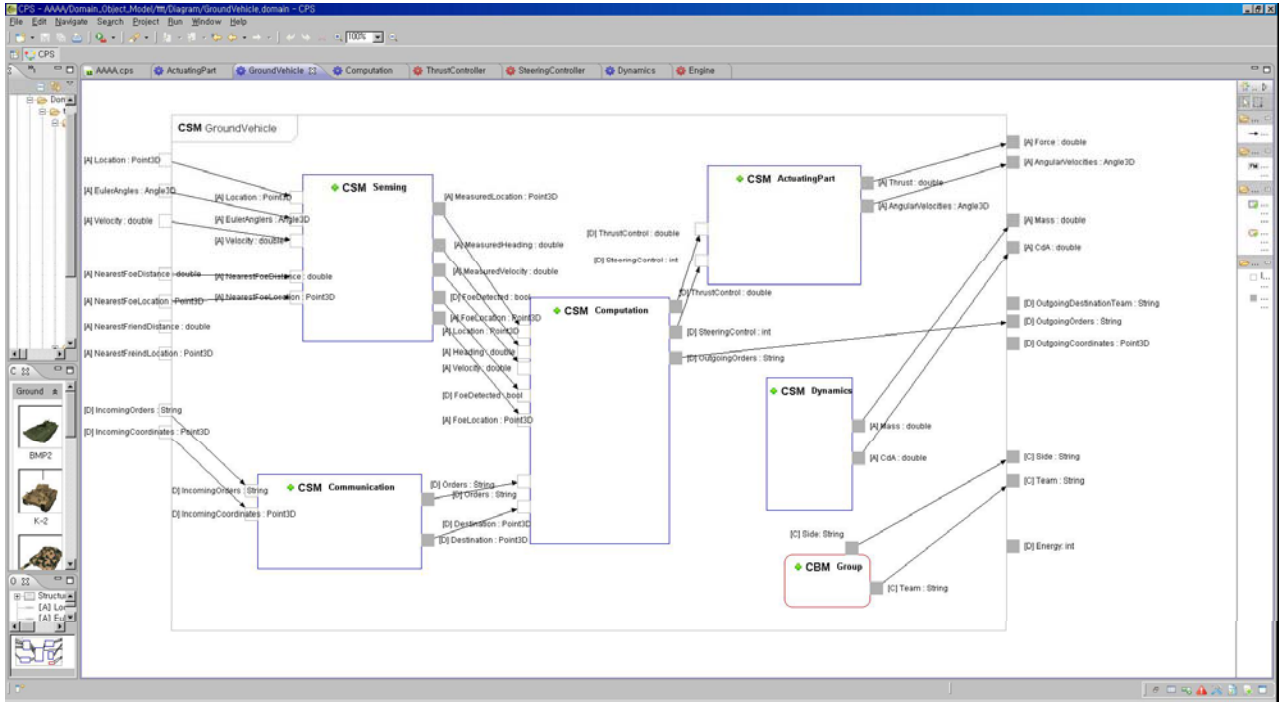
Figure 3 State event.

State Change
Continuous variables이 특정 조건(threshold)을 만족하면, Discrete change가 발생.

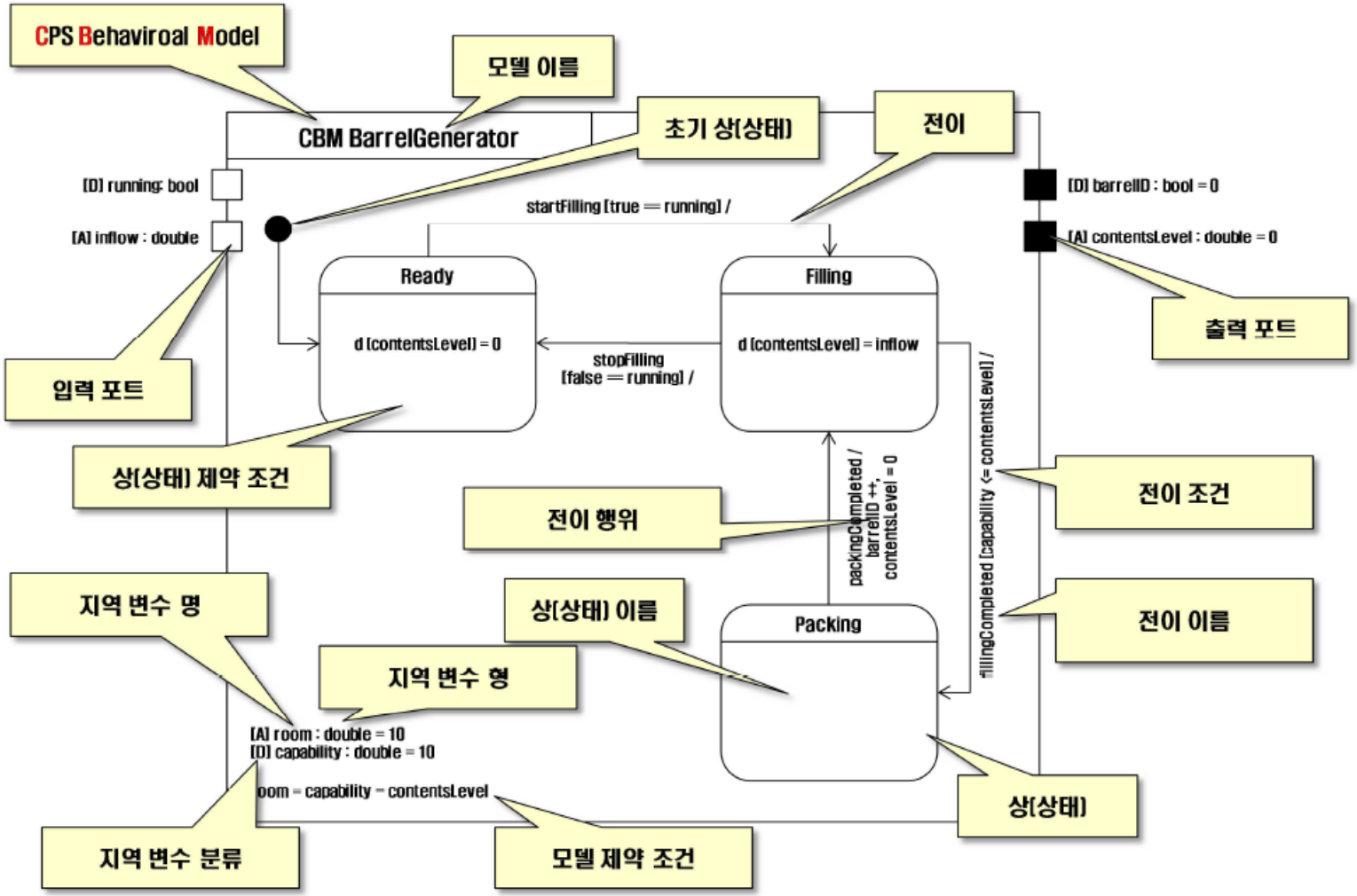
ECML(ETRI CPS Modeling Language)

- 구성
  - CSM(CPS Structural Model)
    - 시스템의 전체 구조를 표현
    - Hierarchy를 가짐
  - CBM(CPS Behavioral Model)
    - CSM의 행위를 정의

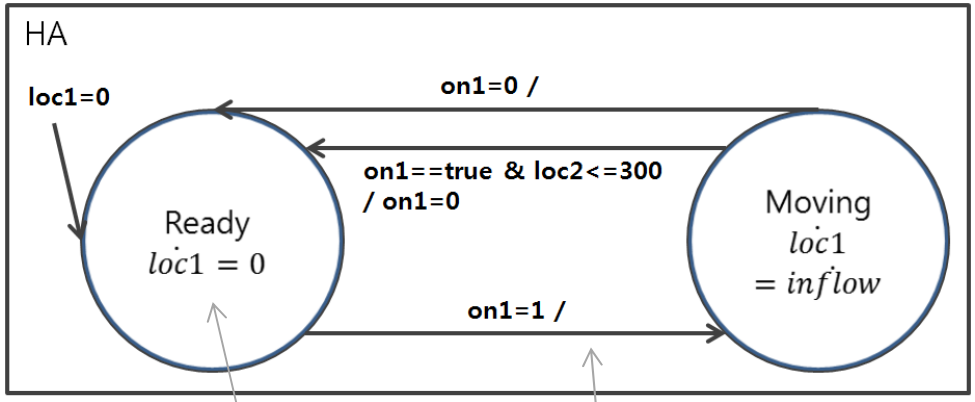
- 지원도구
  - EcoPOD
  - EcoSIM



CBM(CPS Behavioral Model)



(Linear) Hybrid Automata



Invariant
현재 상태를 유지하기 위한 조건

State Transition
비결정적인(Non-deterministic) 상태 전이

Input/Output Variables
없음 → Closed System

- $A = (\vec{x}, V, inv, dif, E, act, L, syn)$
  - $\vec{x} = (x_1, \ldots, x_n)$
    - Real-valued variables
    - v is valuation, V is set of Valuation
    - $\dot{x}$ is differential inclusion
    - x' is new value after transition
  - V - Set of Control location
  - Inv - Convex data predicate
  - dif- Rate predicate
  - E
    - Set of transitions
    - Optionally be assigned asap
  - Act - Discrete actions
  - L- Synchronization label
  - syn - Labeling function

CPS 정형검증 기법

Name	Year (Update)	Tool Support	Execution Environment	Functions (M/S/A/V/Tr)	Verifiability	Input Front-End	Verification Technique
CHARON	2001	Yes	JAVA	M / S	N/A	Automata	N/A
CheckMate	-	No	MATLAB	V	MATLAB	MATLAB	Approximate quotient transition systems
d/dt	**2001**	**Yes**	**Linux**	**M / S**	**N/A**	**d/dt input language**	**Forward reachability analysis**
Ellipsoidal Toolbox	2006	Yes	MATLAB	V	MATLAB	MATLAB	Forward and backward reachability analysis
GBT	2004	Yes (Commercial)	MATLAB	A	MATLAB	MATLAB	Convex hull
HSIF	2002	Yes	Windows	M / S	N/A	GME model	N/A
HSolver	2005	Yes	Linux	V	Manual	Input program	Theorem proving (Rsolver)
HyTech	**2000**	**Yes**	**Linux**	**V**	**Automatic**	**Linear hybrid automata**	**Polyhedral model checking**
HyVisual	2000 (2005)	Yes	JAVA	M / S	N/A	Ptolemy plug-in	N/A
Hybrid ToolBox	2004 (2011)	Yes	MATLAB	M / S / V	MATLAB	HYSDEL language, MATLAB	LP/QP Solver
HYSDEL	2002 (2011)	Yes	Windows, Linux, Solaris	Tr	N/A	HYSDEL language	N/A
KeYmaera	2006 (2011)	Yes	JAVA	V	Manual	Differential dynamic logic formula	Theorem Proving (KeY)
Level Set Toolbox	2004 (2011)	Yes	MATLAB	S / V	MATLAB	MATLAB	Set of Algorithms
MATISSE	2005	Yes	MATLAB	V	MATLAB	MATLAB	Bi-simulation, reachable analysis
MultiParametric Toolbox	2004 (2006)	Yes	MATLAB	M / A / V	MATLAB	MATLAB	Forward and backward reachability analysis
PHAVer	**2004 (2007)**	**Yes**	**Windows, Linux, Mac**	**V**	**Automatic**	**Linear hybrid automata**	**Forward and backward reachability analysis**
Ptolemy	**2002 (2010)**	**Yes**	**JAVA**	**M / A / V**	**Automatic MATLAB(확인 중)**	**UML (in XML), Java code, MATLAB**	**SMV**
SHIFT	1999	Yes	Linux	M / Tr	N/A	Shift language	N/A
SpaceEx	**2010 (2011)**	**Yes**	**Linux**	**V**	**Automatic**	**SX language**	**LeGuernic-Girard Algorithm**
STeP	**1994 (1998)**	**Yes**	**Linux**	**V**	**Automatic**	**STeP language**	**Deductive model checking**

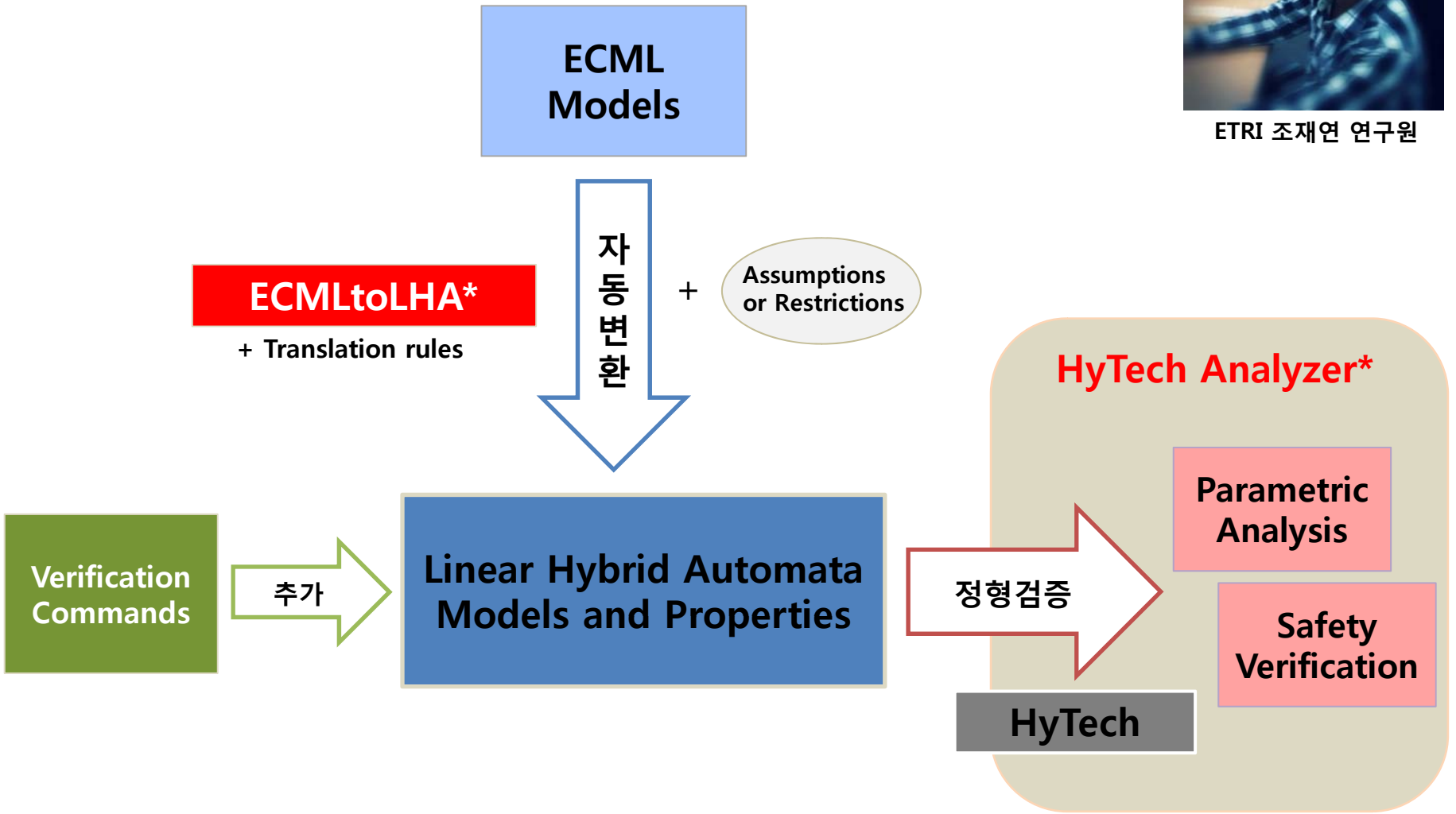
ECML을 위한 정형검증 기법 연구

건국대 : 2011.04 ~ 2013.02

ETRI : 2013.03 ~ 현재

1. HyTech을 이용한 ECML 정형검증

ETRI 조재연 연구원



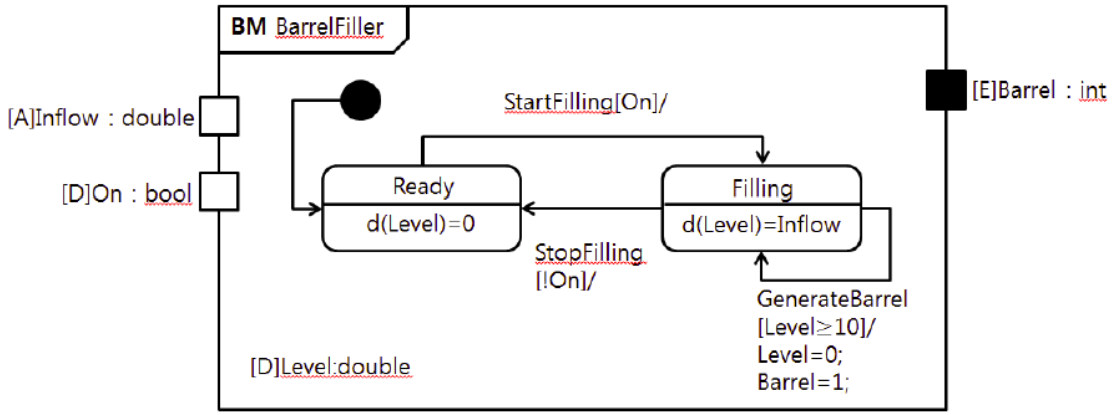


Fig. 1. A graphical representation of the *Barrelfiller* system

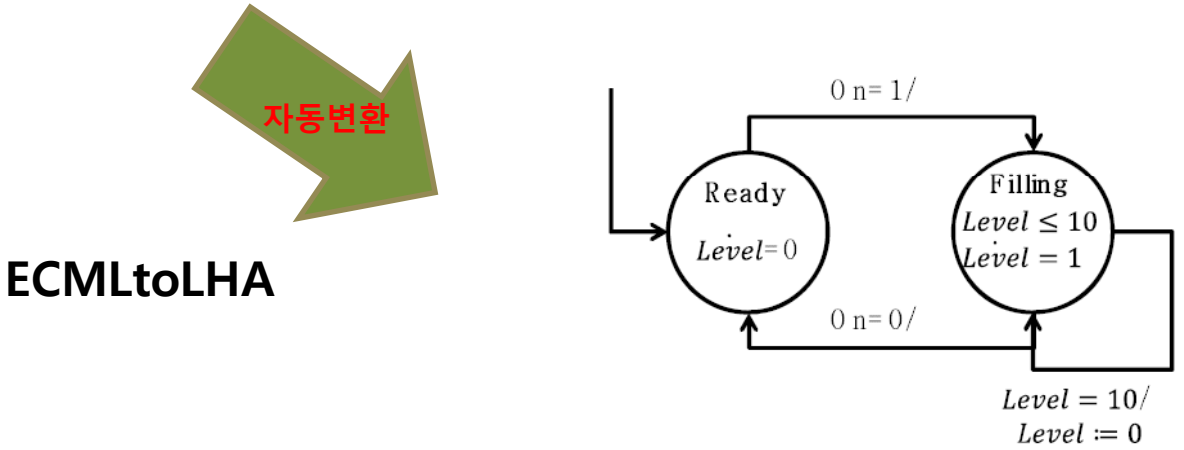


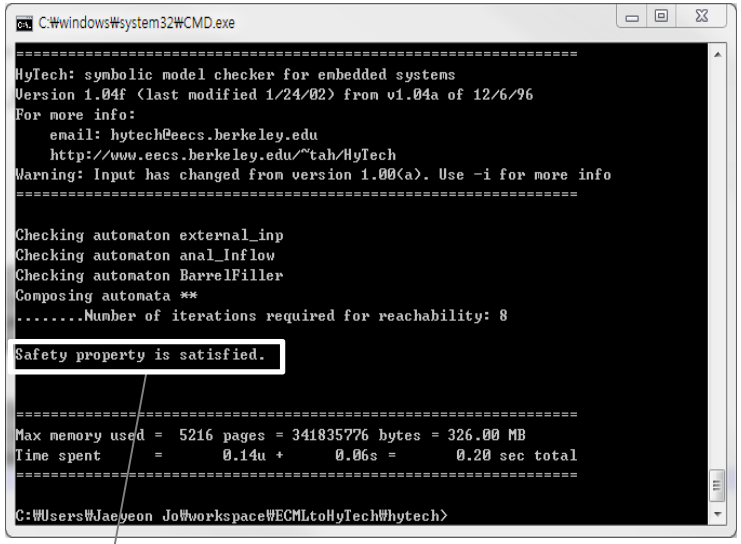
Fig. 2. A graphical representation of the *Barrelfiller* using linear hybrid automata

Safety Verification : 검증이 성공했을 경우 성공 메시지 출력

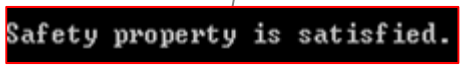
Ex) "BarrelFiller 모델의 Level 값이 10을 초과하지 않는가?"

```
init_reg:=loc[BarrelFiller]=Ready&
          loc[anal_Inflow]=idle1&
          loc[external_inp]=idle & Level=0/10 &
          Capacity=100/10;
reached := reach forward from init_reg endreach;
final_reg := loc[BarrelFiller]=Filling & Level>10;

if empty(final_reg & reached)
   then prints "Safety property is satisfied.";
   else prints "Safety property is not satisfied.";

print trace to final_reg using reached;
endif;
```

HyTech Analysis Command



검증 결과



Safety Verification : 검증이 실패했을 경우 Counter-example을 생성

Ex) "BarrelFiller 모델의 Level 값이 9를 초과할 수 있는가?"

```
init_reg:=loc[BarrelFiller]=Ready&
          loc[anal_Inflow]=idle1&
          loc[external_inp]=idle & Level=0/10 &
          Capacity=100/10;
reached := reach forward from init_reg endreach;
final_reg := loc[BarrelFiller]=Filling & Level>9;

if empty(final_reg & reached)
   then prints "Safety property is satisfied.";
   else prints "Safety property is not satisfied.";
   print trace to final_reg using reached;
endif;
```

HyTech Analysis Command

```
====== Generating trace to specified target region ========
Time: 0.000
Location: Ready.idle1.idle
    On = 0   & Level = 0   & Capacity = 10   & Inflow = 0   & t + 10 = 0
------------------------------
 VIA:
------------------------------
Time: 0.000
Location: Ready.idle1.active
    On = 1   & Level = 0   & Capacity = 10   & Inflow = 0   & t + 10 = 0
------------------------------
 VIA: ext_input_BarrelFiller
------------------------------
Time: 0.000
Location: Filling.idle1.idle
    On = 0   & Level = 0   & Capacity = 10   & Inflow = 0   & t + 10 = 0
--------------
VIA 1e+01.000 time units
--------------
Time: 1e+01.000
Location: Filling.idle1.idle
    On = 0   & Level = 10   & Capacity = 10   & Inflow = 10   & t = 0
```

검증 결과 및 Counter-example

```
On = 0   & Level = 10   & Capacity = 10   & Inflow = 10   & t = 0
```

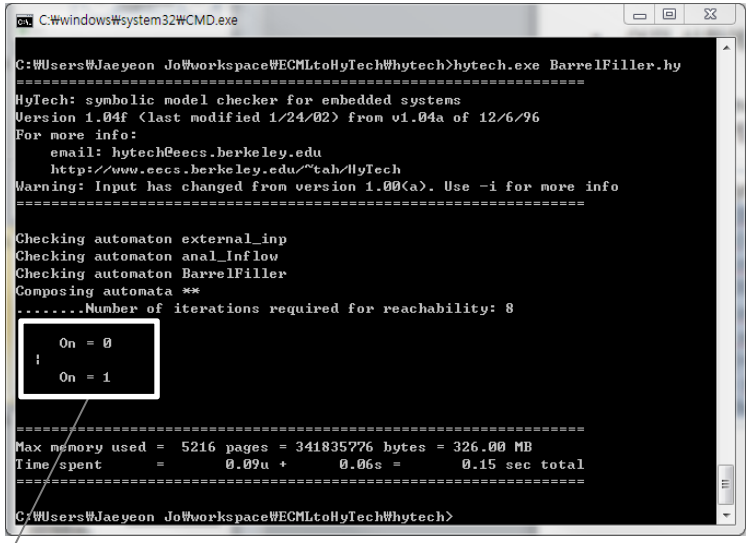
(Level 값이 10이 된 상황)

Parametric Analysis : 어떤 상황에서 특정 변수가 가질 수 있는 범위를 분석

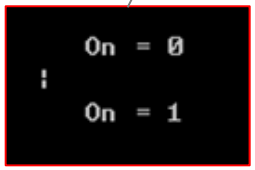
Ex) "Barrel의 Level=10이고, Ready 상태일 때, 입력 변수 On이 가질 수 있는 범위는?"

```
init_reg := loc[BarrelFiller]=Ready  &
            loc[anal_Inflow]=idle1 &
            loc[external_inp]=idle &
            Level=0/10 &
            Capacity=100/10;
reached := reach forward from init_reg endreach;
final_reg := Level=10 & loc[BarrelFiller]=Ready;

print omit all locations hide non_parameters in
            reached & final_reg endhide;
```

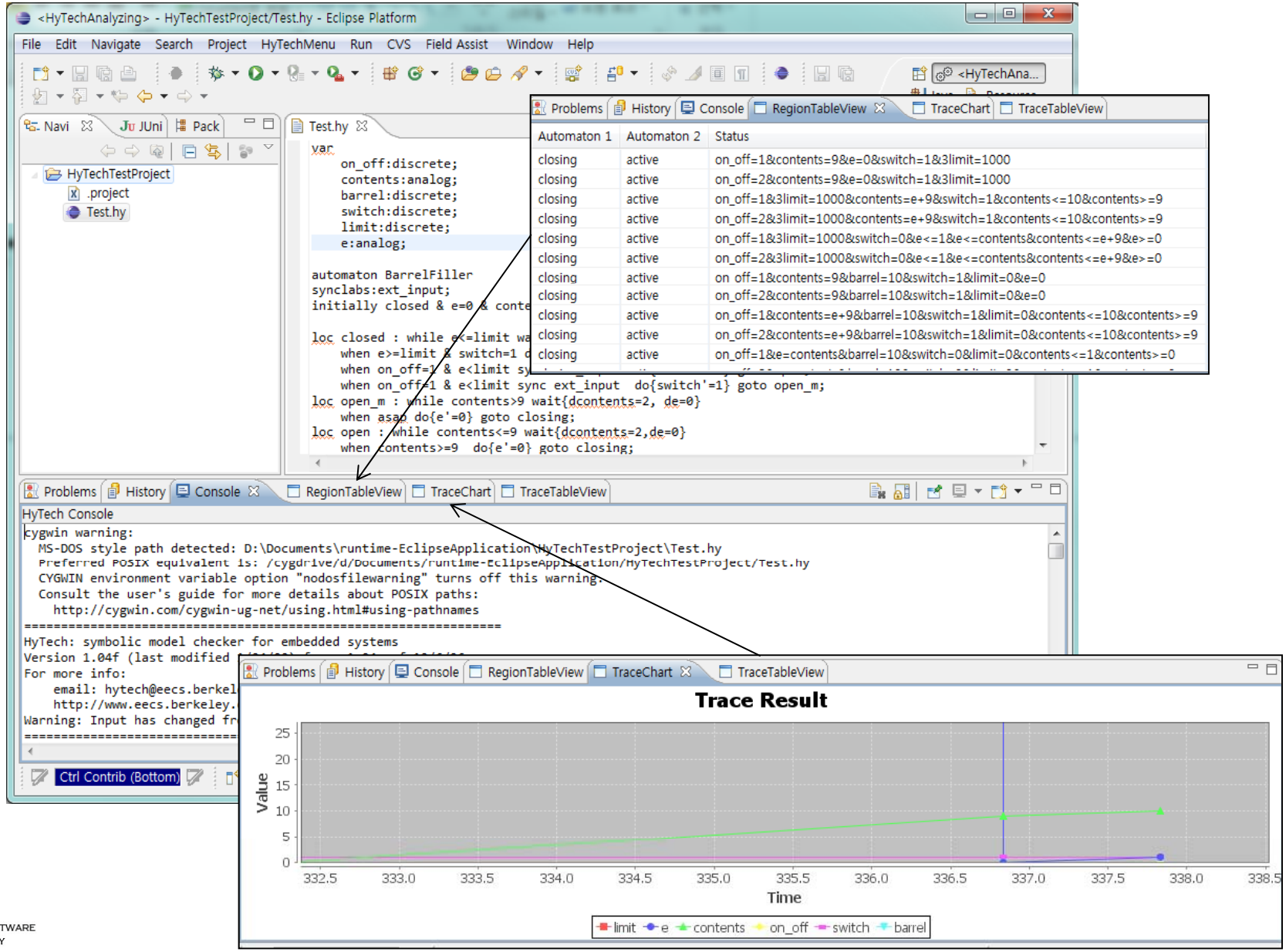
HyTech Analysis Command



Parametric Analysis 결과



HyTech Analyzer : HyTech 사용 지원·자동화 도구



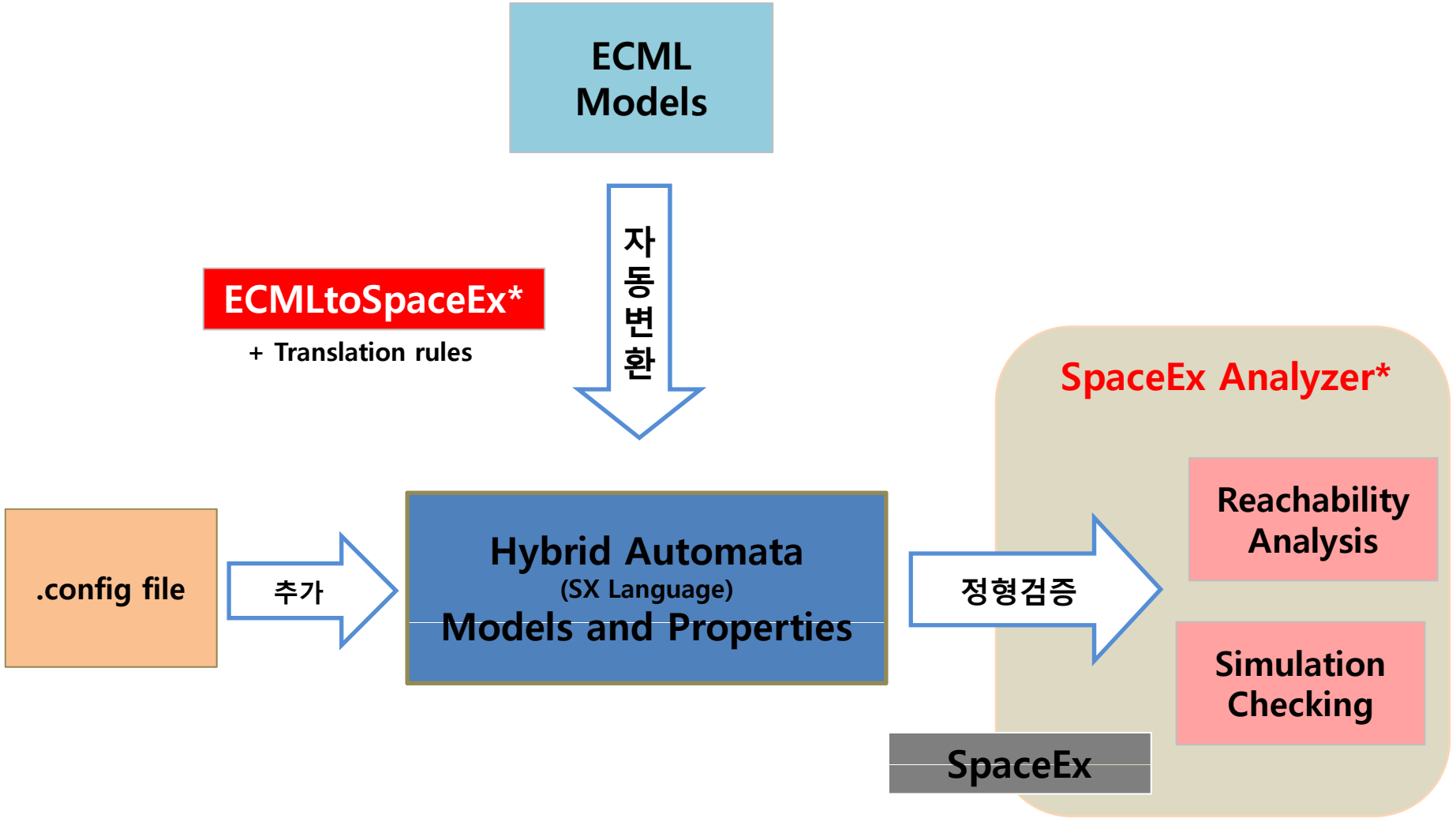
HyTech을 이용한 ECML 정형검증의 특징

- 자체 개발한 '**ECMLtoLHA 변환기**' 및 '**변환규칙**'
  - ECML을 LHA로 자동 변환
  - 여러 가정과 제약사항 有
    - I/O변수 관련 (Open system → Closed system)
    - Linearity 관련 (Non-linear system → Linear system)
    - Determinism 관련 (Deterministic system → Non-deterministic system)
    - 기타 다수

- 정형검증 결과
  - Model Checking과 유사한 형태(*i.e.,* Counter example)로 제공돼, **유용**하게 사용 가능

- **HyTech Analyzer**
  - HyTech은 GUI가 제공되지 않음
  - 검증 과정 자동화
  - 검증 결과를 효과적으로 확인 가능

2. SpaceEx를 이용한 ECML 정형검증



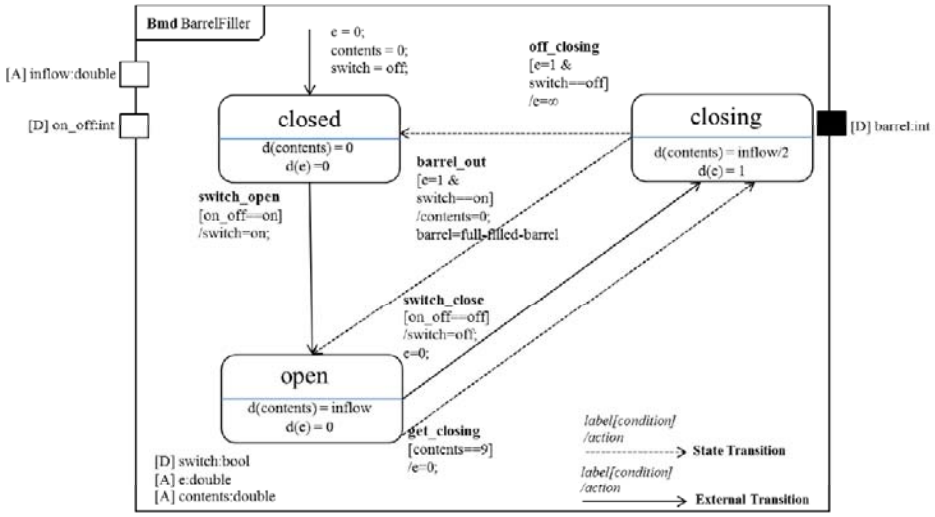


Fig. 1 A barrel-filler model for ECML

자동변환

ECMLtoSpaceEx

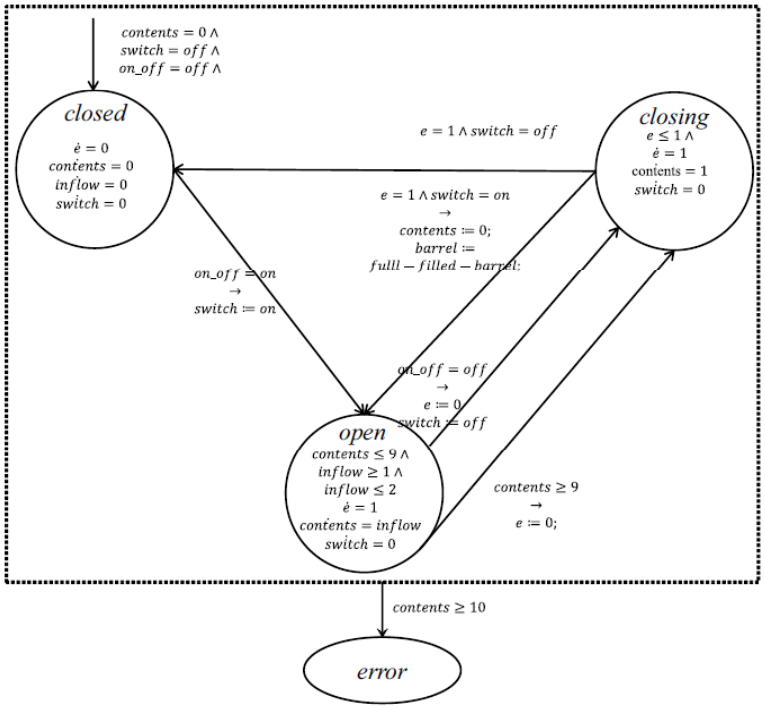
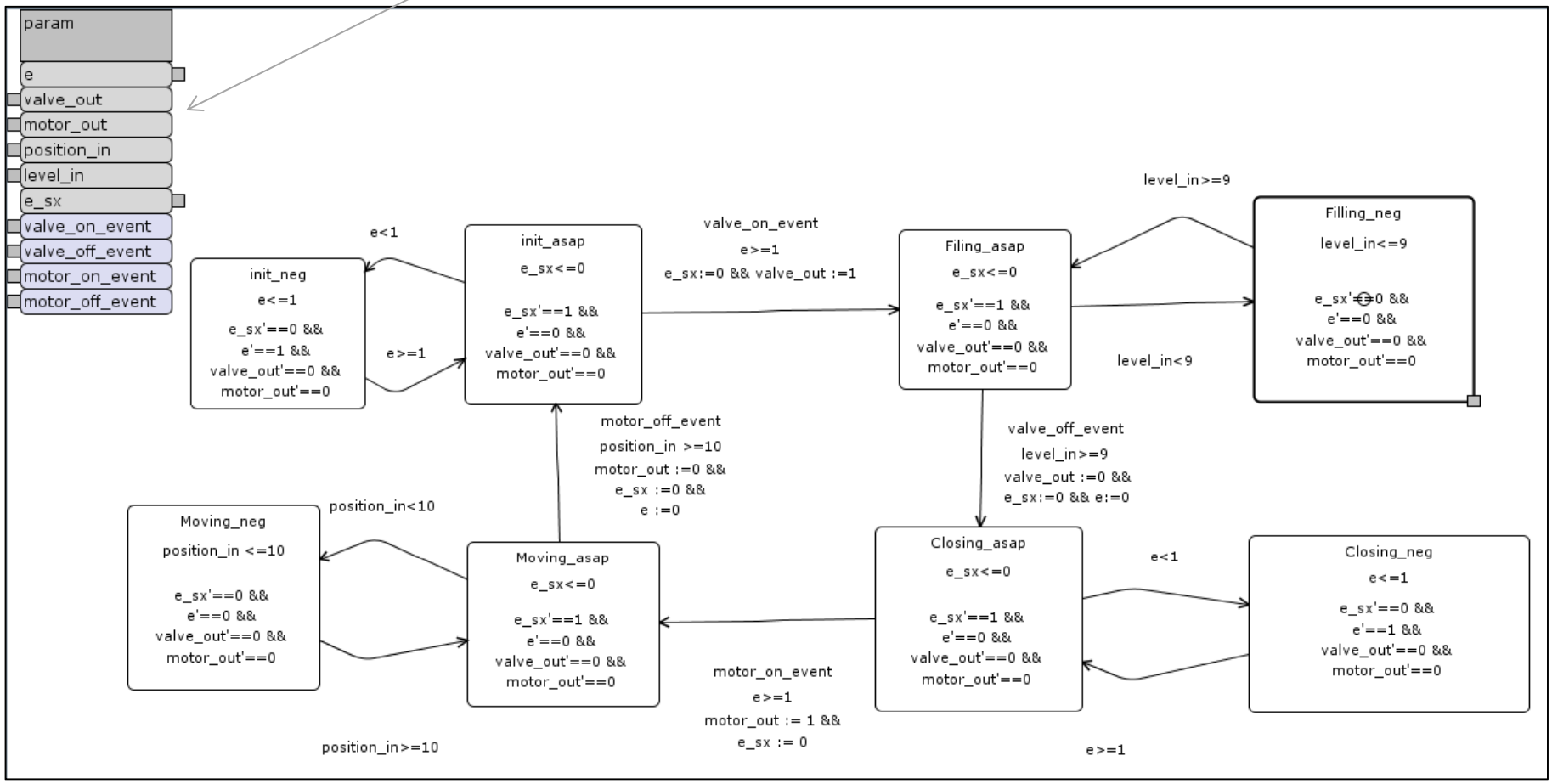
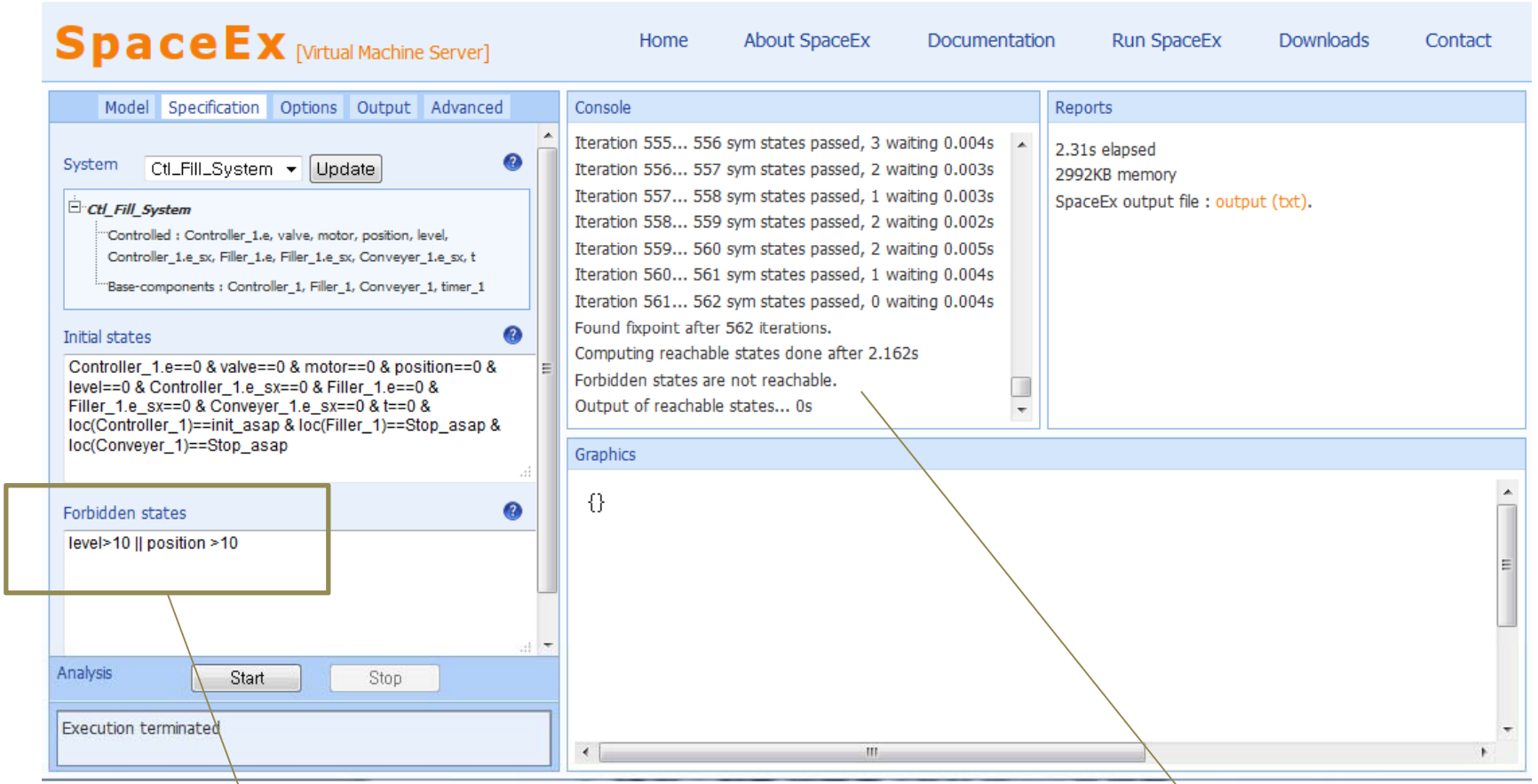


Fig. 3 A barrel-filler model for hybrid automata

Input / Output Variables 이 있음!



A hybrid automaton for the Barrel Filling System (in SpaceEx Model Editor)

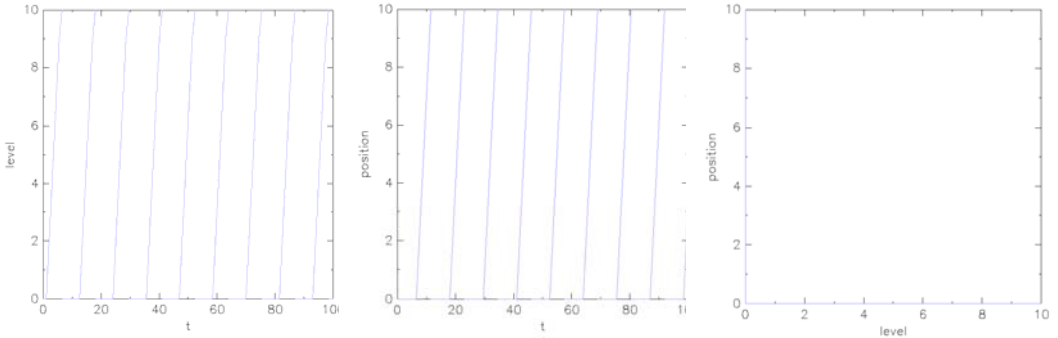


"Level 이나 Position 이 10을 초과하지 않는가?"

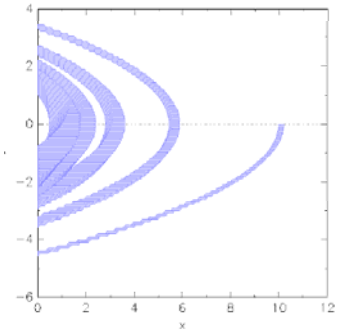
Forbidden states are not reachable.

SpaceEx 검증 결과 출력 방식 (4가지)

- 도달 가능한 변수 2개의 상태를 그래프로 표현하는 2D
- 도달 가능한 변수 3개의 상태를 그래프로 표현하는 3D
- 변수의 Boundary를 표현하는 inv 형식
- 모든 도달 가능한 region을 표현



< 2D 그래프 형식 >



< 3D 그래프 형식 >

```
Bounds on the variables over the entire set:
system.x: [-0,10.205]
system.v: [-4.575,3.43125]

Location-wise bounds on the variables:

Location: loc(ball)==always
system.x: [-0,10.205]
system.v: [-4.575,3.43125]
```

< inv 형식 >

```
{(loc(ball)==always & {[
-9.895,-9.78,-9.655,-9.52,-9.375,-9.22,-9.055,-8.88,-8.695,-8.5,-
-9.995,-9.98,-9.955,-9.92,-9.875,-9.82,-9.755,-9.68,-9.595,-9.5,-
-10,-10.095,-10.18,-10.255,-10.32,-10.375,-10.42,-10.455,-10.48,-
0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1.1,1.2,1.3,1.4,1.5,1.6,1.7
0,-0.1,-0.2,-0.3,-0.4,-0.5,-0.6,-0.7,-0.8,-0.9,-1,-1.1,-1.2,-1.3,
10.305,10.39,10.465,10.53,10.585,10.63,10.665,10.69,10.705,10.71,
10.205,10.195,10.18,10.155,10.12,10.075,10.02,9.955,9.88,9.795,9.
10.2,10.095,9.98,9.855,9.72,9.575,9.42,9.255,9.08,8.895,8.7,8.495
]]}{variable to dimension map:[x,v]
```

< region 형식 >

SpaceEx를 이용한 ECML 정형검증의 특징

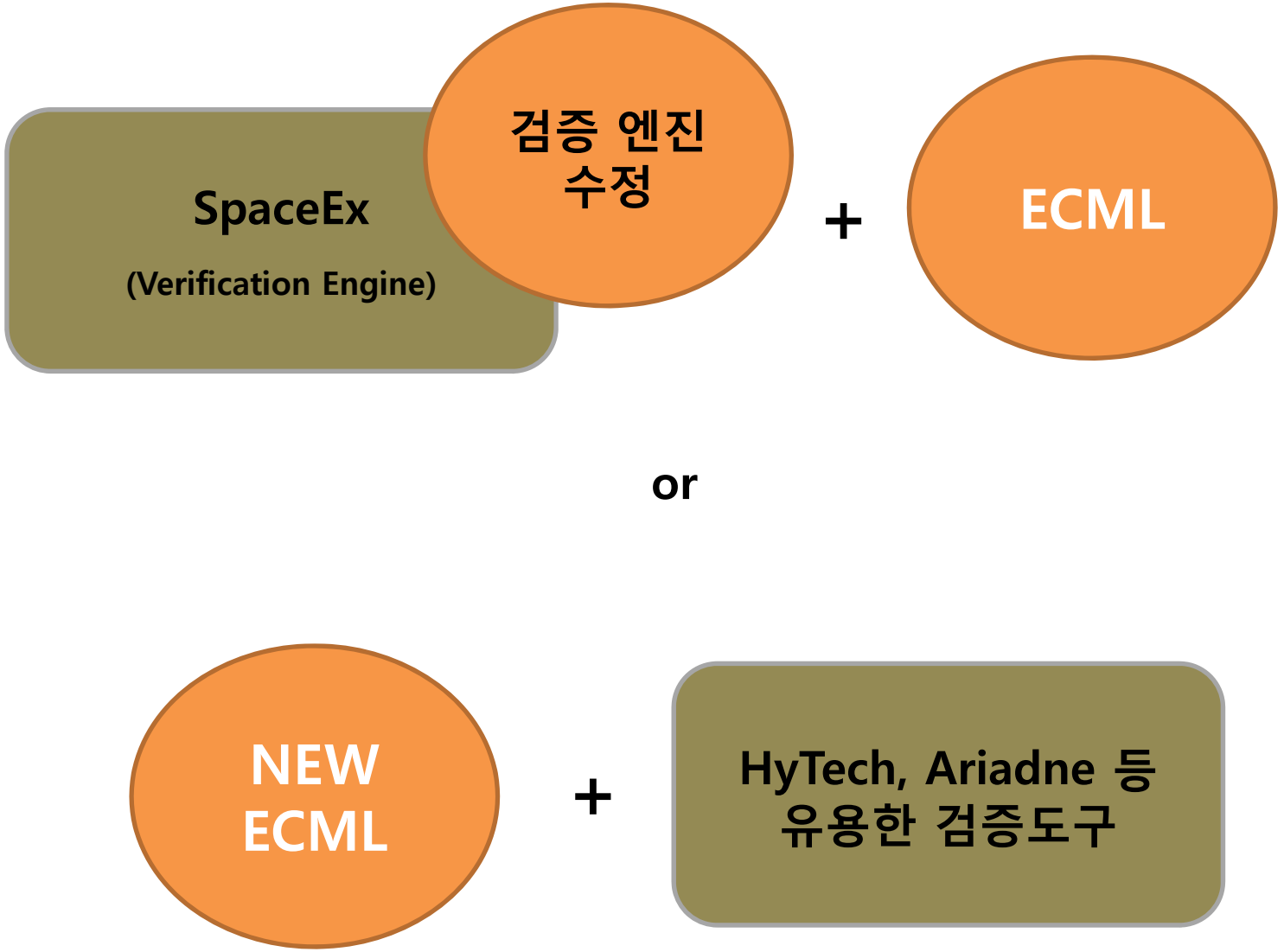
- 자체 개발한 '**SpaceExtoHA 변환기**' 및 '**변환규칙**'
  - 모델링의 가정이나 제약 없이, ECML 모델을 그대로 HA 모델로 변환 가능
  - 현재 '변환 규칙'의 수정 및 보완 中

- 정형검증 결과
  - HyTech과는 다르게 reachability analysis 형태로 표현됨 **(region 중심)**
  - Counter-example과 같은 명확한 형태로 나오지 않아, 정형검증의 유용성이 상당히 감쇄됨

- SpaceEx
  - 유용한 모델링 및 검증결과 출력 용 GUI 제공

- SpaceEx Analyzer
  - region 형태의 검증결과를 보다 compact하게 정리하기 위한 도구
  - ETRI 개발 中

향후 연구 방향


SpaceEx
(Verification Engine)
검증 엔진
수정
+
ECML
or
NEW
ECML
+
HyTech, Ariadne 등
유용한 검증도구

KU KONKUK UNIVERSITY

감사합니다.

http://dslab.konkuk.ac.kr