

TimesTool

Contents

- Introduction
- Modeling
- Simulation
- Schedulability
- Verification
- Code synthesis
- 예제 모델 설명

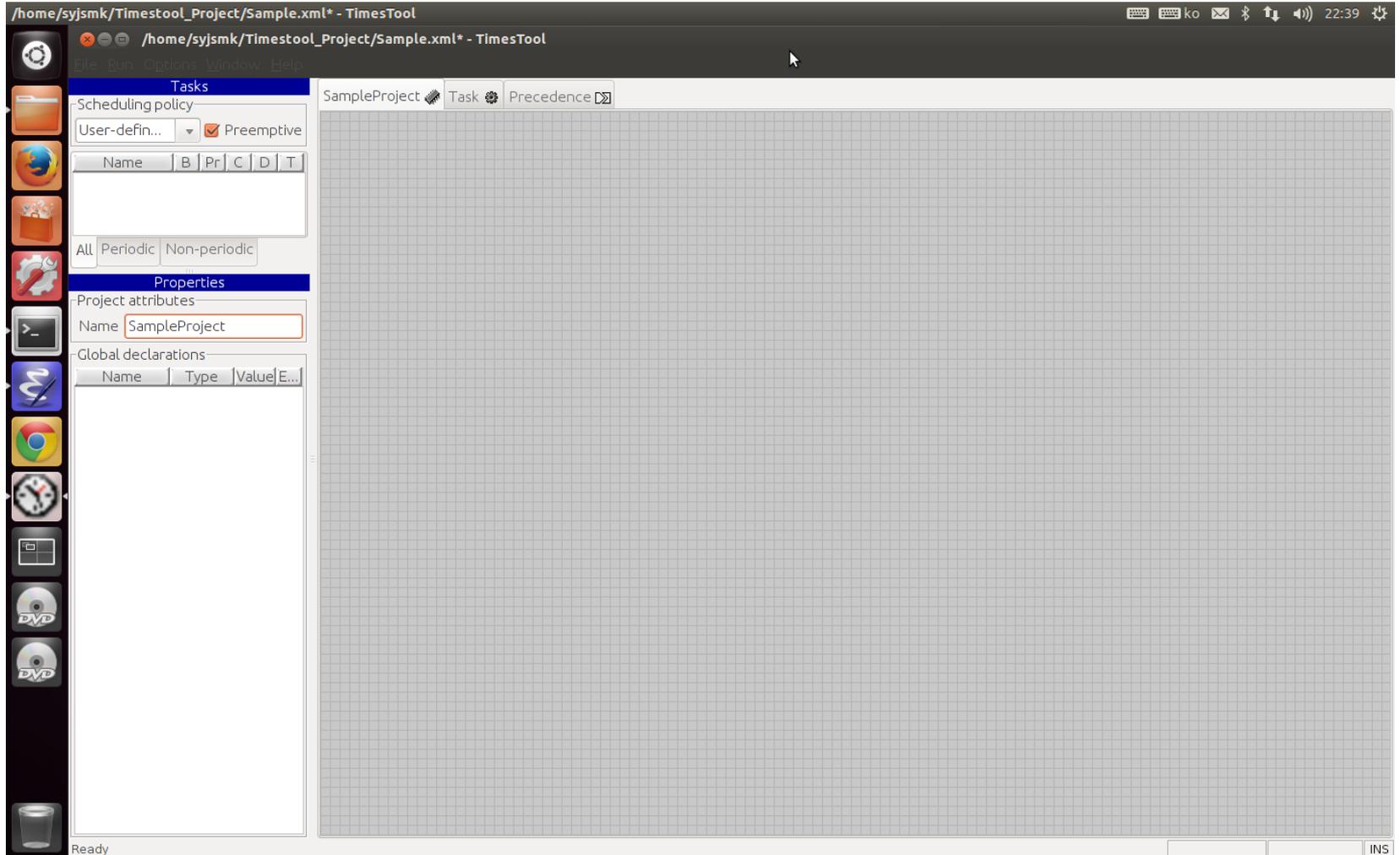
Introduction

Introduction

- TimesTool
 - A Tool for Modeling and Implementation of Embedded Systems.
 - UPPAAL이라는 model checker를 작성한 사람들이 만들었음.
 - Realtime 시스템의 검증을 중요시 함.
 - SMV나 SPIN과는 다르게 GUI 기반으로 모델을 작성.

- 기능
 - Modeling : 원하는 시스템의 모델을 편집.
 - Simulation : 작성한 모델의 동작을 시뮬레이션.
 - Schedulability analysis : 모델이 deadlock같은 문제로 인해 제대로 동작하지 않는지를 확인.
 - Verification : LTL 표기법을 따라 입력한 property를 모델이 만족하는지를 확인.
 - synthesis of schedules and executable code : 모델로부터 C코드 생성.

Introduction



Modeling

Modeling

- Task
 - General real-time scheduling 이론에 따른 코드의 조각.
 - Behavior, Priority, Computing time, Deadline, Period 등의 특성을 가짐.
 - interface에 선언된 코드를 따라 global 변수들에 접근도 가능.

The screenshot shows a software interface for modeling tasks. On the left, the 'Tasks' panel displays a table of tasks and their attributes.

Name	B	Pr	C	D	T
task2	C	2	1	4	X
task1	P	1	2	5	4

Below the table, the 'Attributes' section for 'task2' is shown:

Attribute	Value
Name	task2
Behaviour	Controlled
Priority	2
Computing time	1
Deadline	4
Period	
Offset	
Max # of tasks	1

On the right, the 'Code' editor shows the following code for 'task2':

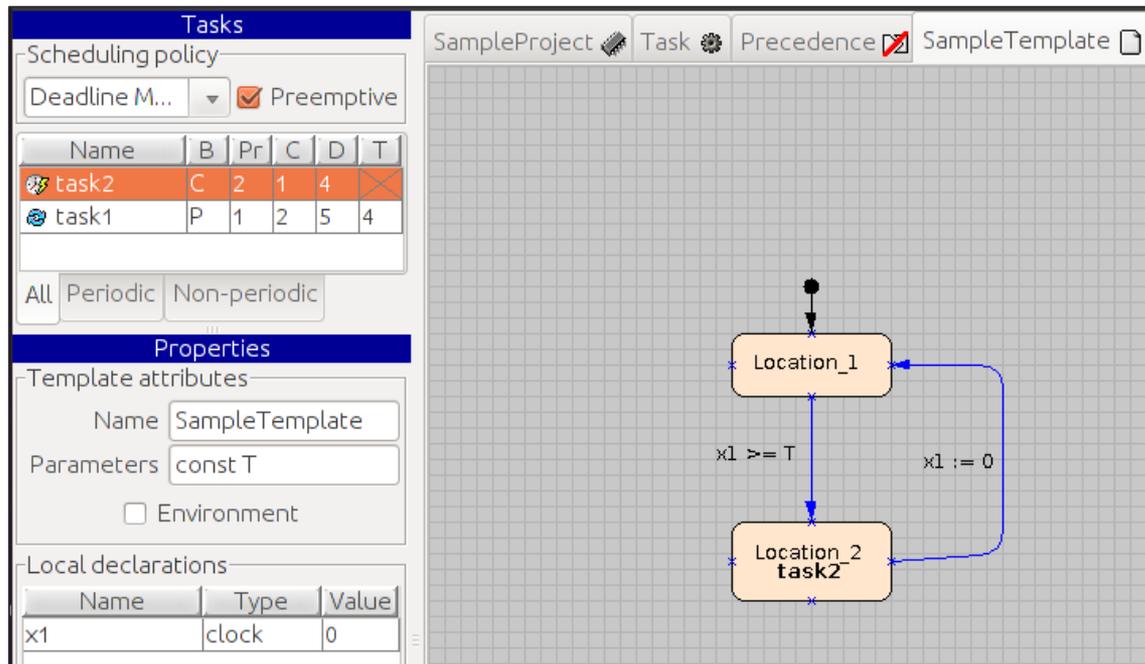
```

// test
if(g_int == 0) {
    g_int = 1;
} else {
    g_int = 0;
}
    
```

Modeling

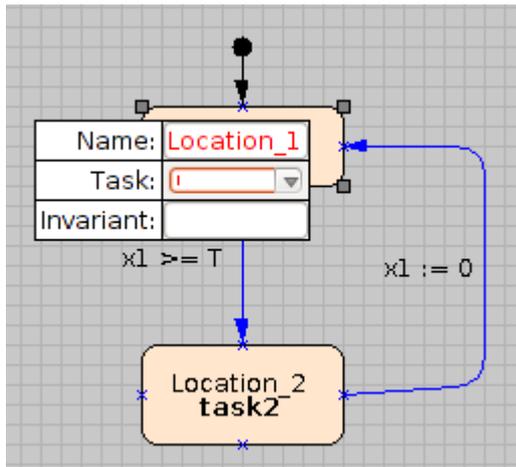
- Automaton Template

- Task들과 parameter를 가지는 timed automaton에 대한 template.
- invariant, guard, synchronization, assignment, probability 등을 이용해 timed automaton을 작성할 수 있음.
- timed automaton의 location에 앞서 작성한 task를 할당해줄 수 있음.
- 각 Automaton별로 가지는 local variable들을 선언 가능.

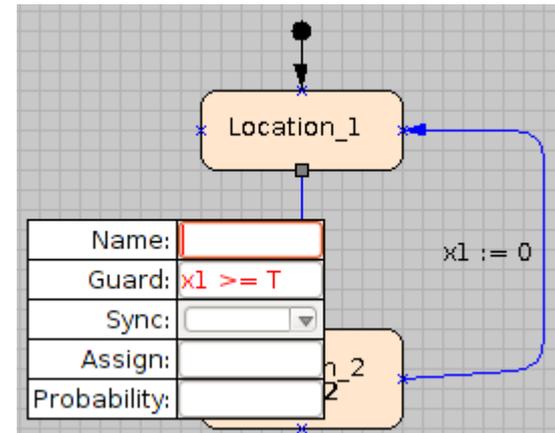


Modeling

- Automaton Template



Location에 Task를 할당하거나 Invariant를 작성 가능



Transition에 Guard, Sync, Assign, Probability등을 작성 가능

Modeling

- Process
 - 시스템을 이루는 Block들.
 - 앞서 작성한 automaton template가 Process에 할당될 수 있음.
 - 전체 시스템의 global 변수를 가질 수 있음.

The screenshot displays a software interface for modeling tasks and processes. On the left, the 'Tasks' panel shows a scheduling policy of 'Deadline M...' with 'Preemptive' checked. A table lists tasks 'task2' and 'task1' with their respective parameters. Below the table are buttons for 'All', 'Periodic', and 'Non-periodic'. The 'Properties' panel shows 'Project attributes' with 'Name' set to 'SampleProject' and 'Global declarations' with a table for 'g_int'.

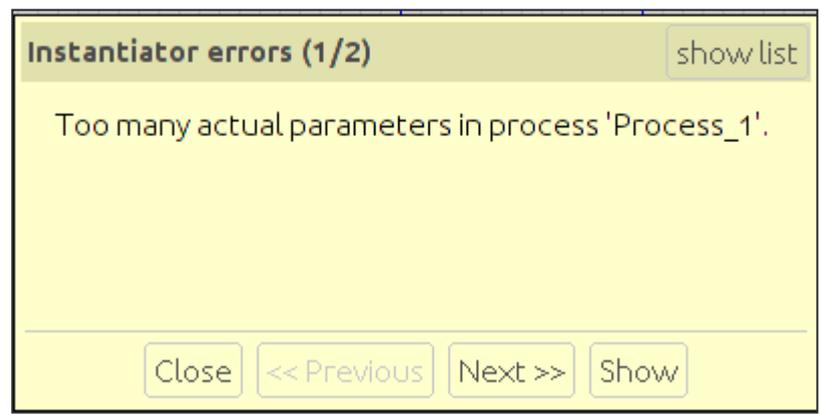
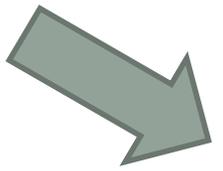
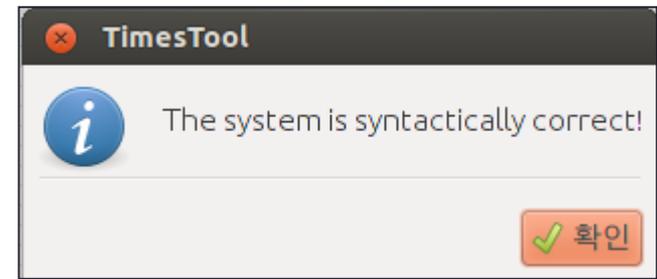
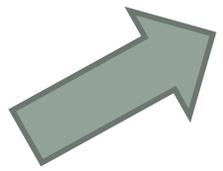
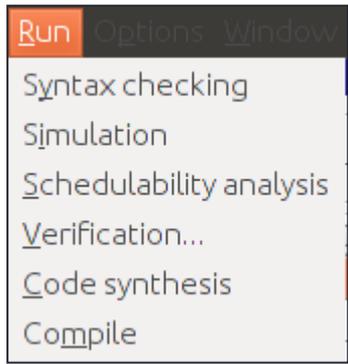
Name	B	Pr	C	D	T
task2	C	2	1	4	X
task1	P	1	2	5	4

Name	Type	Value	E...
g_int	int	0	<input type="checkbox"/>

Process 1
SampleTemplate(7)

Modeling

- Syntax checking
 - 작성한 모델이 문법에 어긋나지 않는지 확인.



Simulation

Simulation

- 작성한 모델을 이용해 시뮬레이션 수행.

The screenshot displays a simulation environment with the following components:

- Enabled transitions:**
 - PERIODIC_TASKS: START->START
 - SCHEDULER: IDLE->ARRIVED_task1
- Status:**
 - Process_1.x1 in [2,4]
 - g_int = 0
- State Transition Diagram:**
 - Process_1:** Location_1 (initial state) leads to START. From START, a transition labeled 'task1' leads to another START state.
 - PERIODIC_TASKS:** START leads to another START state.
 - SCHEDULER:** IDLE leads to ARRIVED_task1, then to RUN_task1, then to FINISHED, and finally back to IDLE.
 - A red arrow labeled 'task1' indicates a transition from the START state of PERIODIC_TASKS to the ARRIVED_task1 state of SCHEDULER.
- Timeline:**
 - Shows the execution of task1 as a red bar on the task1 axis.
 - The idle state is shown as a blue bar on the idle axis.
 - task2 is shown as an empty axis.
 - task1 is shown as an empty axis.

Simulation

- Enabled transitions
 - 시뮬레이션을 한 단계 수행했을 때 다음 상태로 가능한 transitions들.

- Status
 - 시뮬레이션 중 모델에서 선언된 변수의 값 변화를 볼 수 있음.
 - 값이 특정되는게 아닌 순간에는 값의 범위를 [], ()를 이용해서 보여줌.

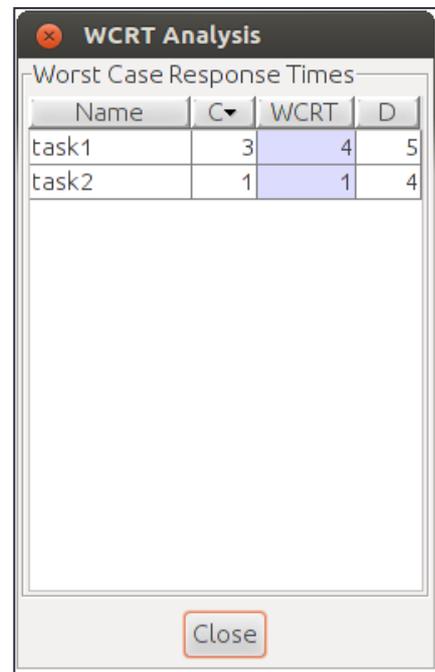
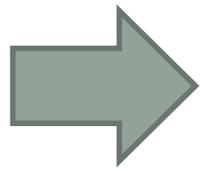
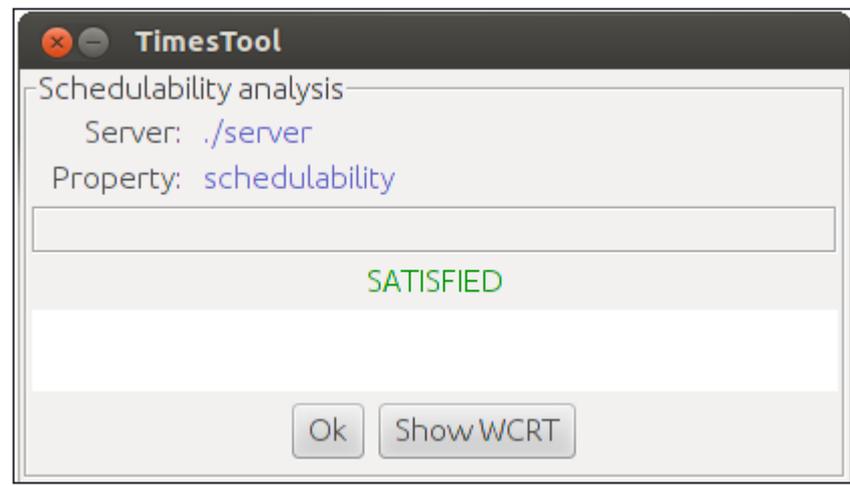
- Message sequence chart
 - 단계별로 진행 상황을 보여줌.

- Gantt chart
 - 각 task가 수행되면서 시간이 지나는 상태를 보여줌.
 - task별 시간 소모는 task 작성 시 computing time 값.

Schedulability

Schedulability

- 목적
 - 모델이 에러 없이 스케줄링이 가능한지의 여부를 확인
 - Task의 Computing time, Deadline, Max # of tasks 등을 고려하여 계산
- 스케줄링 가능 시



Schedulability

- 스케줄링 불가능 시 (Deadline)
 - task1의 computing time이 deadline(최대 실행 가능 시간)보다 길 경우 deadline에 맞출 수 없기 때문에 NOT SAT

Attributes	
Attribute	Value
Name	task1
Behaviour	Periodic
Priority	1
Computing time	3
Deadline	2
Period	4
Offset	0
Max # of tasks	1

TimesTool

Schedulability analysis

Server: `./server`

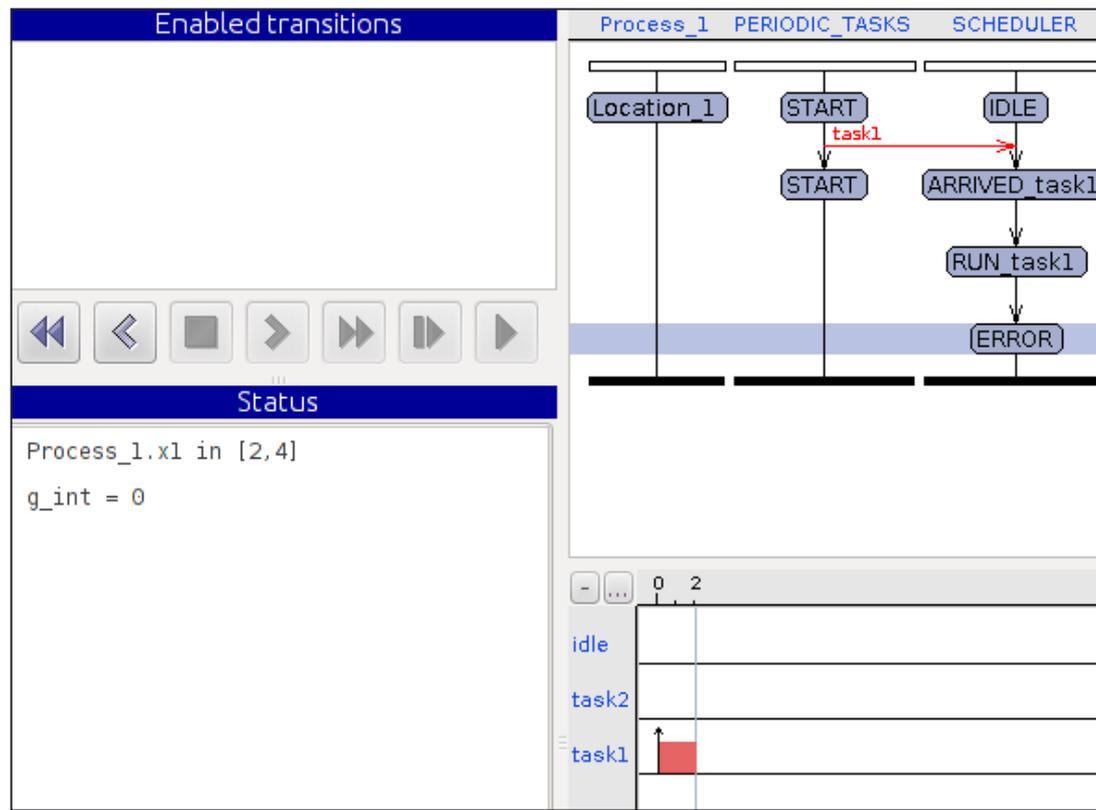
Property: `schedulability`

NOT SATISFIED

The system is not schedulable because the task 'task1' failed to meet its deadline.

Schedulability

- 스케줄링 불가능 시 (Deadline)
 - task1의 computing time이 deadline(최대 실행 가능 시간)보다 길 경우 deadline에 맞출 수 없기 때문에 NOT SAT



Schedulability

- 스케줄링 불가능 시 (Max # of tasks)
 - Period에 따라 task1이 주기적으로 수행되고 있는데 task2의 수행이 사이에 끼면서 먼저 수행된 task1이 종료되지 않아 task1이 하나 더 수행될 수 없기 때문에 NOT SAT

The screenshot shows a software interface with two main panels. The left panel is a table titled 'Attributes' with the following data:

Attribute	Value
Name	task1
Behaviour	Periodic
Priority	1
Computing time	4
Deadline	5
Period	4
Offset	0
Max # of tasks	1

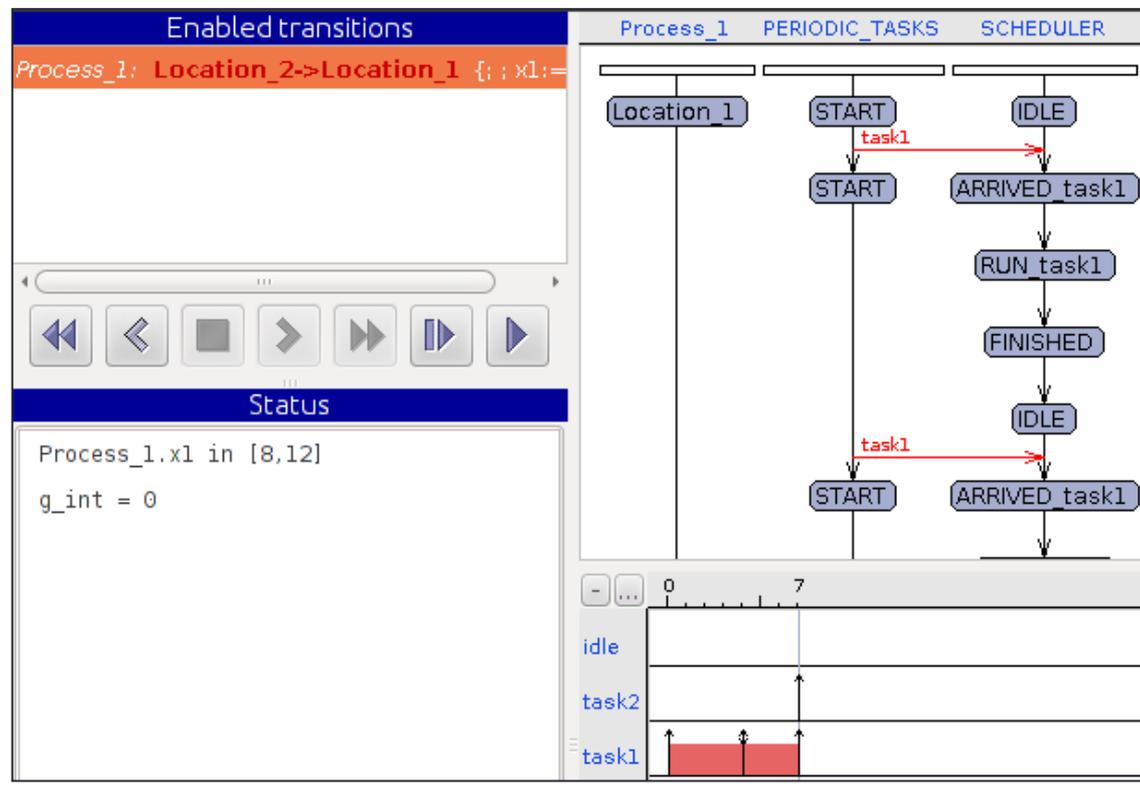
The right panel is a dialog box titled 'TimesTool' with a 'Schedulability analysis' section. It displays the following information:

- Server: ./server
- Property: schedulability

The analysis result is 'NOT SATISFIED' in red text, followed by the message: 'The system is not schedulable because maximum number of 'task1' tasks exceeded.' At the bottom of the dialog are 'Ok' and 'Show trace' buttons.

Schedulability

- 스케줄링 불가능 시 (Max # of tasks)
 - Period에 따라 task1이 주기적으로 수행되고 있는데 task2의 수행이 사이에 끼면서 먼저 수행된 task1이 종료되지 않아 task1이 하나 더 수행될 수 없기 때문에 NOT SAT



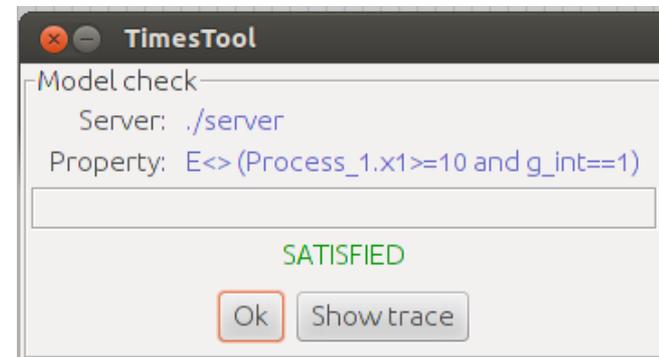
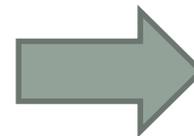
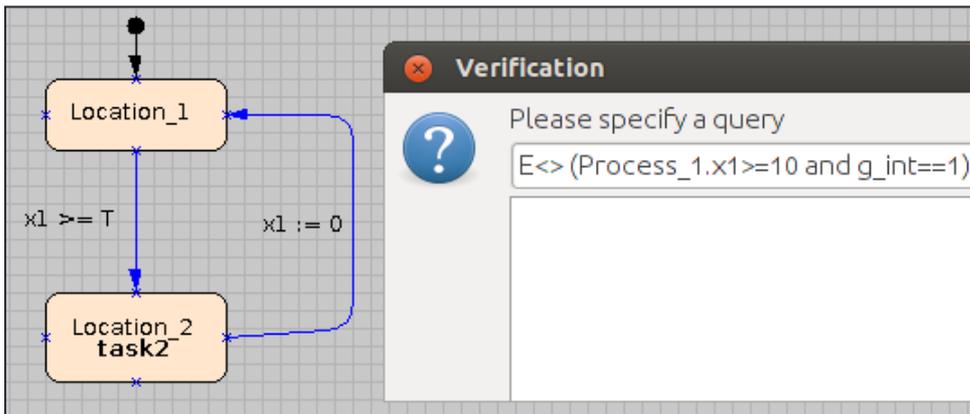
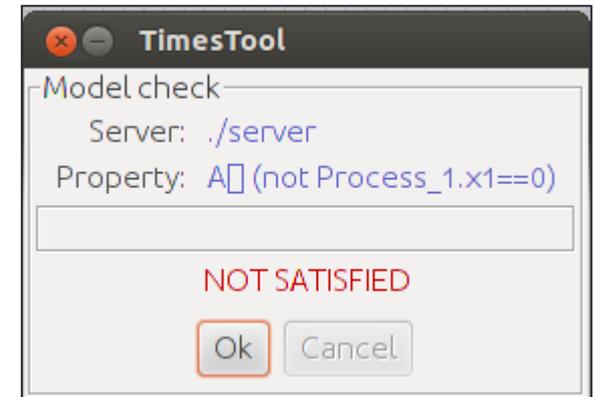
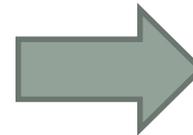
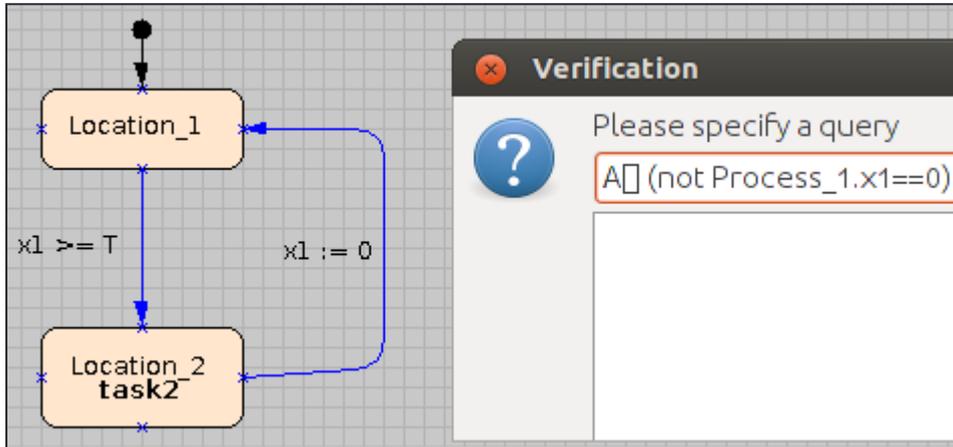
Verification

Verification

- Model checking query
 - LTL(Linear temporal logic)을 입력으로 받음.
 - 표기법에 약간 차이 있음.
 - LTL이기 때문에 A, E를 따로 쓰거나 X, G, F를 따로 쓸 수 없음.

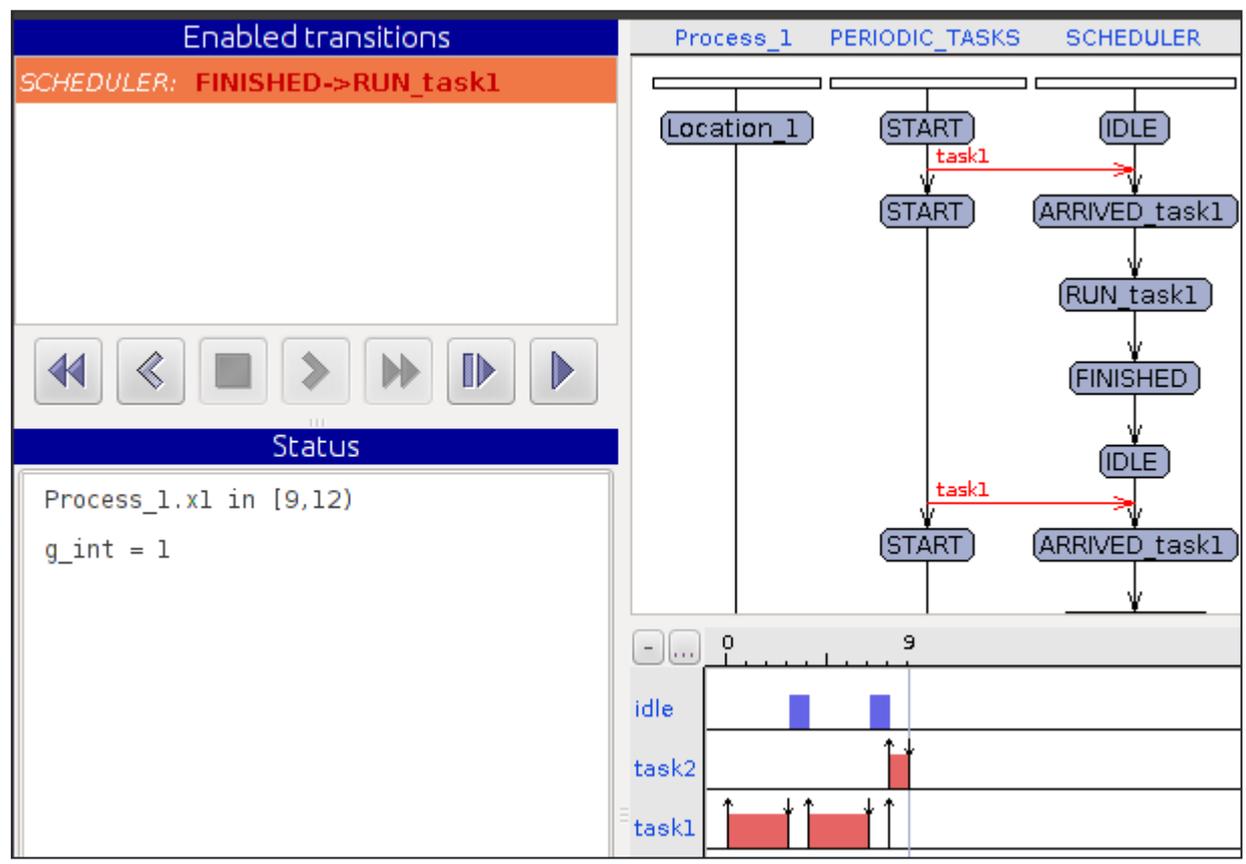
Textual	Symbolict	Explanation
Unary operators:		
X ϕ	$\bigcirc\phi$	neXt: ϕ has to hold at the next state,
G ϕ	$\square\phi$	Globally: ϕ has to hold on the entire subsequent path.
F ϕ	$\diamond\phi$	Finally: ϕ eventually has to hold (somewhere on the subsequent path).

Verification



Verification

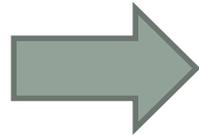
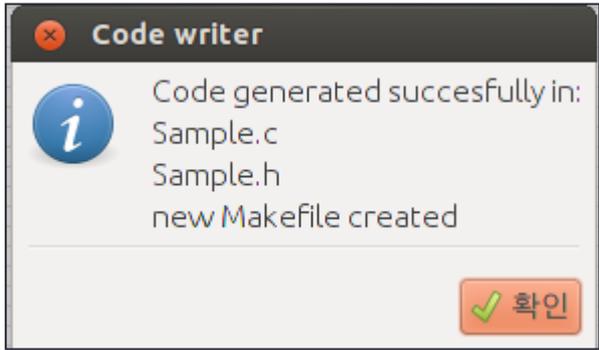
- Trace
 - 해당 경우를 재현 가능할 경우 시뮬레이터를 통해서 볼 수 있음.



Code synthesis

Verification

- 모델에 해당하는 C코드 생성



```
void COPY2LOCAL( int tid, int instance ) {
    switch( tid ) {
    }
};

/**
 * Evaluate guards on transition trn.
 * @param trn Transition id
 * @return true if guard satisfied, false otherwise.
 */
bool eval_guard(int trn) {
    switch(trn) {
    case 1: return (rdClock(Process_1_x1)>=7);
    case 2: return (rdClock(PERIODIC_TASKS_x)==PERIODIC_TASKS_t_task1);
    case 0:
        return true;
    }
    return false;
}

/**
 * Perform assignments on transition trn.
 * @param trn Transition id.
 */
void assign(int trn) {
    switch(trn) {
    case 0:
        setClock(Process_1_x1,0); break;
    case 2:
        setClock(PERIODIC_TASKS_x,0);
        PERIODIC_TASKS_t_task1=4; break;
    }
}

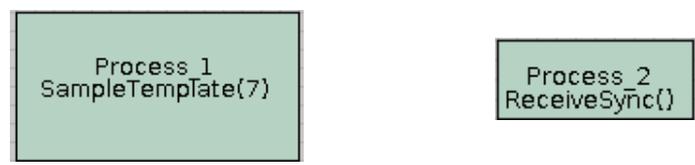
-:--- Sample.c 40% L61 (C/l AC Abbrev)
```

예제 모델 설명

예제 모델 설명

- System

- Process : Process_1, Process_2
 - Process_1 : SampleTemplate라는 timed automaton의 동작을 수행하며 해당 파라미터로 7의 값을 넘겨줌.
 - Process_2 : ReceiveSync라는 timed automaton의 동작을 수행하며 파라미터는 없음.



- Global variable

- g_int, task3_sync : 시스템에서 사용하는 값
- channel : 다른 task끼리 통신하기 위해 사용하는 채널

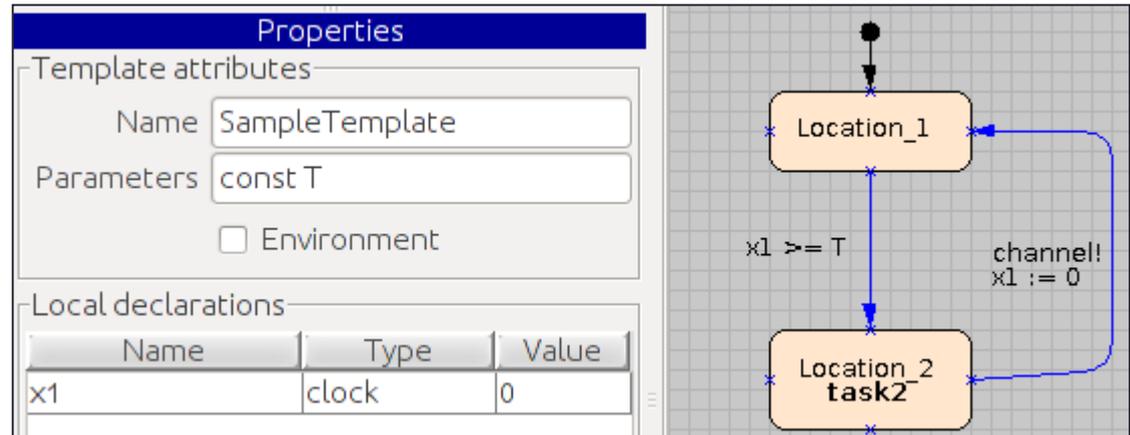
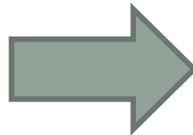
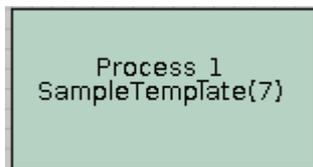
Global declarations			
Name	Type	Value	Env
g_int	int	0	<input type="checkbox"/>
channel	extern chan	 	<input type="checkbox"/>
task3_sync	int	0	<input type="checkbox"/>

예제 모델 설명

- Timed automaton

- SampleTemplate

- Process_1의 동작을 나타냄.
 - 내부 clock인 x1을 가지며 외부에서 입력으로 값 T를 받음.
 - Location_2에서 Location_1로 상태가 변경될 때 내부 x1을 초기화함.
 - Location_2에서 Location_1로 상태가 변경될 때 전역 channel으로 신호를 보냄.

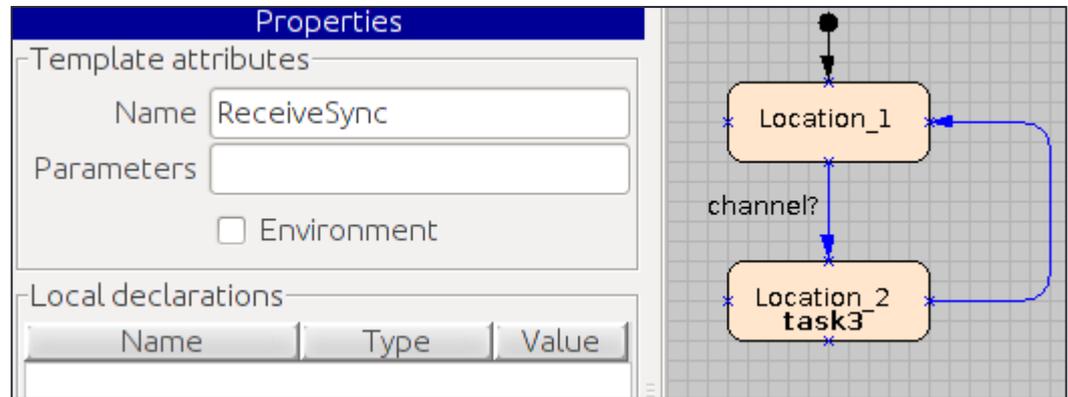
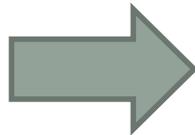


예제 모델 설명

- Timed automaton

- ReceiveSync

- Process_2의 동작을 나타냄.
- 내부 파라미터, 로컬 변수 없음.
- Location_1에서 Location_2로 상태가 변경되려면 외부에서 extern channel 변수인 channel으로 sync를 보내야함.



예제 모델 설명

- Task
 - Periodical task : Task1
 - Task1
 - 사용자의 입력 없이도 주기적으로 수행됨.
 - 시스템의 값을 변경하지 않음.

Attributes	
Attribute	Value
Name	task1
Behaviour	Periodic
Priority	1
Computing time	3
Deadline	5
Period	4
Offset	0
Max # of tasks	1

예제 모델 설명

- Task
 - Controlled task : Task2, 3
 - Task2
 - 프로세스가 동작할 때 수행됨.
 - 시스템의 전역변수인 g_int의 값을 변경

Attributes	
Attribute	Value
Name	task2
Behaviour	Controlled
Priority	2
Computing time	1
Deadline	4
Period	
Offset	
Max # of tasks	1

task2	
Interface	g_int := (g_int == 0 ? 1 : 0)
Input	
Semaphor...	
Code poin...	/home/syjsmk/Timestool_Project/task2.h

예제 모델 설명

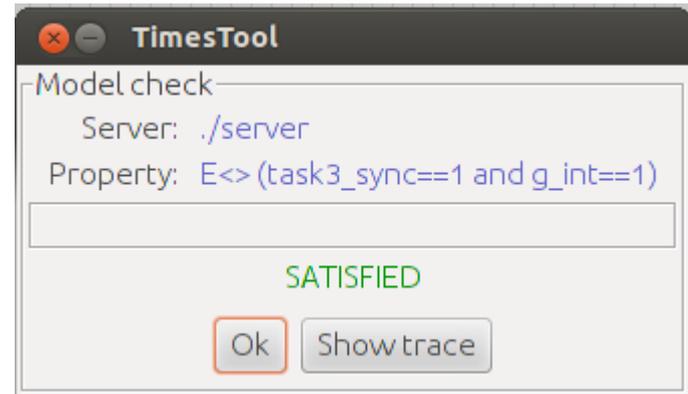
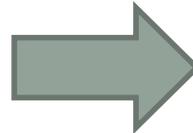
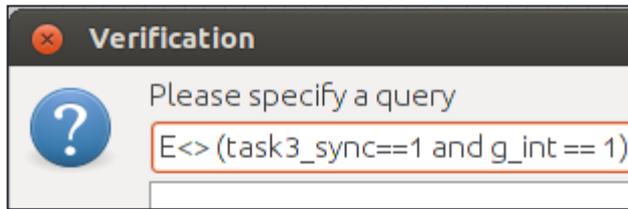
- Task
 - Controlled task : Task2, 3
 - Task3
 - 프로세스가 동작할 때 수행됨.
 - 시스템의 전역변수인 task3_sync의 값을 변경

Attributes	
Attribute	Value
Name	task3
Behaviour	Controlled
Priority	0
Computing time	0
Deadline	4
Period	
Offset	
Max # of tasks	1

task3	
Interface	task3_sync := (task3_sync == 0 ? 1 : 0)
Input	
Semaphor...	
Code poin...	

예제 모델 설명

- Property
 - Global variable인 task3_sync와 g_int가 동시에 1이 되는 경우가 있는지 확인.

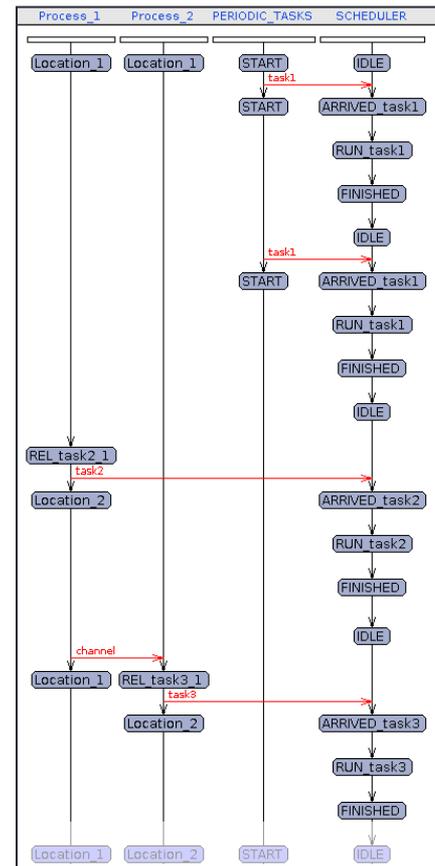
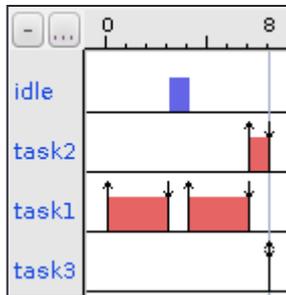


예제 모델 설명

- Property
 - Global variable인 task3_sync와 g_int가 동시에 1이 되는 경우가 있는지 확인.

```

Status
-----
Process_1.x1 = 0
g_int = 1
task3_sync = 1
    
```



예제 모델 설명

- NOT SAT이며 counter example을 보여줄 수 있는 경우

