

UPPAAL

무인비행체 제어 검증

EC Lab 오향란

UPPAAL

UPPAAL은 실시간 임베디드 시스템을 모델링하고 시뮬레이션 및 검증하는 통합 도구이다. 이 도구는 Timed automata를 기반으로 하며 유한한 제어구조와 실수 값을 가지는 클럭들을 가진 비결정적인 프로세스들의 집합으로 모델될 수 있는 시스템들에 적합하다.

UPPAAL

UPPAAL에서 시스템은 프로세스의 집합으로 이루어져 있다.

각각의 프로세스는 하나의 Timed automata로 표현되고 채널로 연결된다.

각 채널은 송신 이벤트와 수신 이벤트로 구성되어 있다.

하나의 채널로 연결된 두 개의 프로세스는 노드들로 구성되어 있다.

시스템의 동작은 노드들의 전이로 표현된다.

노드의 전이는 동기화 이벤트 외에 한정자(guard)에 의해 영향을 받는다.

UPPAAL

노드의 전이는 동기화 이벤트 외에 한정자(guard)에 의해 영향을 받는다.
한정자는 각 프로세스의 시간 변수에 대해 지연이나
시간 만료와 같은 시간 제약 사항을 표현하는 것이다.
제안된 모델은 시간 제약 사항에 따른 모델 검증이 가능하다.

UPPAAL

UPPAAL v.4 세가지 주요 부분

The model-checker

명세된 시스템의 상태 공간을 탐색해서 시스템이 만족해야 할 속성을 만족하는지 검증한다.

A description language

각 프로세스를 Timed automata로 표현한 것으로 시스템 편집기를 이용해서 도식적으로 시스템을 명세한다.

The simulator

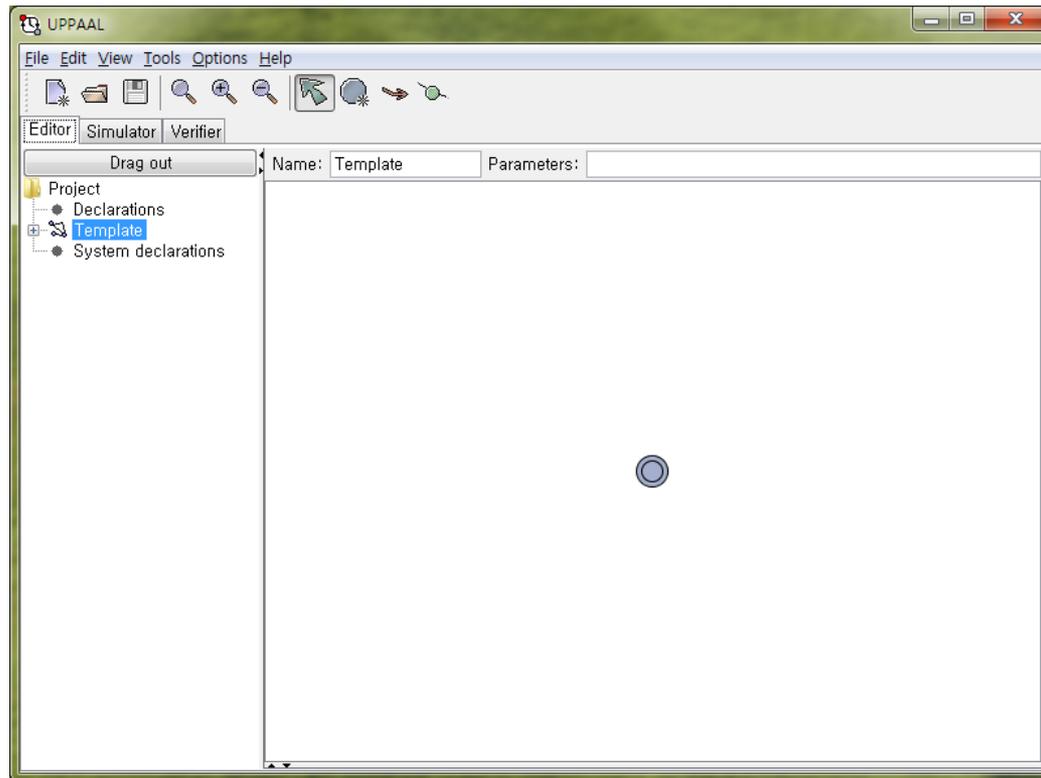
기술언어로 표현된 시스템의 동적인 실행을 시뮬레이션 해서 시스템의 동작을 확인할 수 있게 한다.

UPPAAL

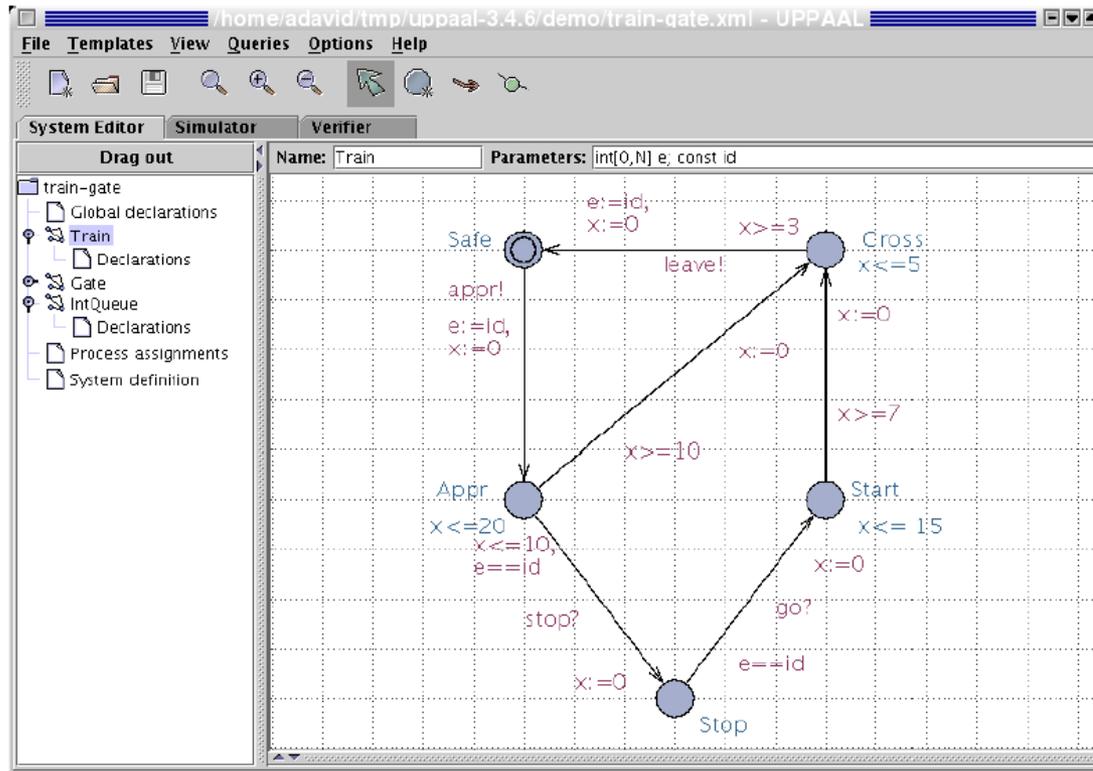
UPPAAL v.4의 주요기능

시스템의 그래픽 설명을 허용하는 그래픽 시스템 편집기
그래픽 시각화를 제공하는 그래픽 시뮬레이터
요구 사항 명세 편집기
자동 검증을 위한 모델 검사기
진단 추적의 생성

UPPAAL



UPPAAL



UPPAAL

The screenshot displays the UPPAAL System Editor interface. The left pane shows a project tree with the following structure:

- train-gate
 - Global declarations
 - Train
 - Declarations
 - Gate
 - IntQueue
 - Declarations
 - Process assignments
 - System definition

The right pane shows the code editor with the following content:

```
/*  
 * For more details about this example, see  
 * "Automatic Verification of Real-Time Communicating Systems by Constraint Solving",  
 * by Wang Yi, Paul Pettersson and Mats Daniels. In Proceedings of the 7th International  
 * Conference on Formal Description Techniques, pages 223-238, North-Holland, 1994.  
 */  
  
const N 5; // # trains + 1  
int [0,N] e1;  
chan appr, stop, go, leave;  
chan empty, notempty, hd, add, ren;  
  
clock x;  
  
int [0,N] list[N], len, i;  
  
Train1:=Train(e1, 1);  
Train2:=Train(e1, 2);  
Train3:=Train(e1, 3);  
Train4:=Train(e1, 4);  
Queue:=IntQueue(e1);  
  
system  
  Train1, Train2, Train3, Train4,  
  Gate, Queue;
```

UPPAAL

The screenshot displays the UPPAAL software interface, which is used for modeling and verifying real-time systems. The window title is "home/adavid/Imp/uppaal-3.4.6/demo/train-gate.xml - UPPAAL".

The interface is divided into several panels:

- System Editor / Simulator / Verifier:** The top navigation bar.
- Drag out:** A panel on the left containing "Enabled Transitions" (e.g., Train1.5, Train2.4.appri, Gate.3.appri?), "Simulation Trace" (a list of events like Stop, Safe, Stop, Stop, Free, Start), and "Trace File" controls (Prev, Next, Replay, Open, Save, Random).
- Variables:** A central panel listing system variables and their values, such as `gl = 1`, `Queue.list[0] = 1`, `Train1.x in [0,15]`, and `Train4.x in [3,35]`.
- State Space Diagrams:** Two diagrams at the top right show the state space of the system. The left diagram shows a state where `x <= 5` and `x >= 10`, with transitions labeled `leave!`, `stop?`, and `go?`. The right diagram shows a state where `x <= 15` and `x >= 10`.
- Gate and Queue:** Two smaller diagrams below the state space diagrams show the internal state of the Gate and Queue components. The Gate diagram includes states for `nonempty?`, `empty?`, and `appri?`. The Queue diagram shows a circular buffer structure with `list` and `len` variables.
- Simulation Trace:** A timeline at the bottom shows the sequence of events for Train1, Train2, Train3, Train4, Gate, and Queue. The trace includes events like `Start`, `Safe`, `Stop`, `Occ`, and `Start`.

UPPAAL

