A State-Based Systemic View of Behaviour for Safe Medical Computer Applications

Luca Pazzi, Marco Pradelli
University of Modena and Reggio Emilia
Department of Information Engineering DII-UNIMORE
Via Vignolese 905, I-41125 Modena, Italy
{luca.pazzi,marco.pradelli}@unimore.it

The interconnection issue

Recent directions in the development of computer systems for medical application show a growing interest towards networking medical devices having embedded computers.

The availability of mature interconnecting technologies, typically distributed object middleware, allows indeed a great flexibility in interconnecting control and sensing devices.

The result is a system of interconnected medical devices, which may exercise control toward other devices in the network and which are controllable, on their turn, by human operators.

Opportunities and challenges

- Opportunities:
  - Great flexibility in interconnecting control and sensing devices;
  - Components-off-the-shelf: reduced price, reusability at the component level (apparent);
- Challenges:
  - Development, certification for high confidence medical software resulting from such a heterogeneous system integration;
  - Presence of human operators in the loop;
  - In the medical case, the difficulties inherent system integration are further worsened, since such systems are often assembled in order to support life-critical applications and, in any case, may endanger patient life.

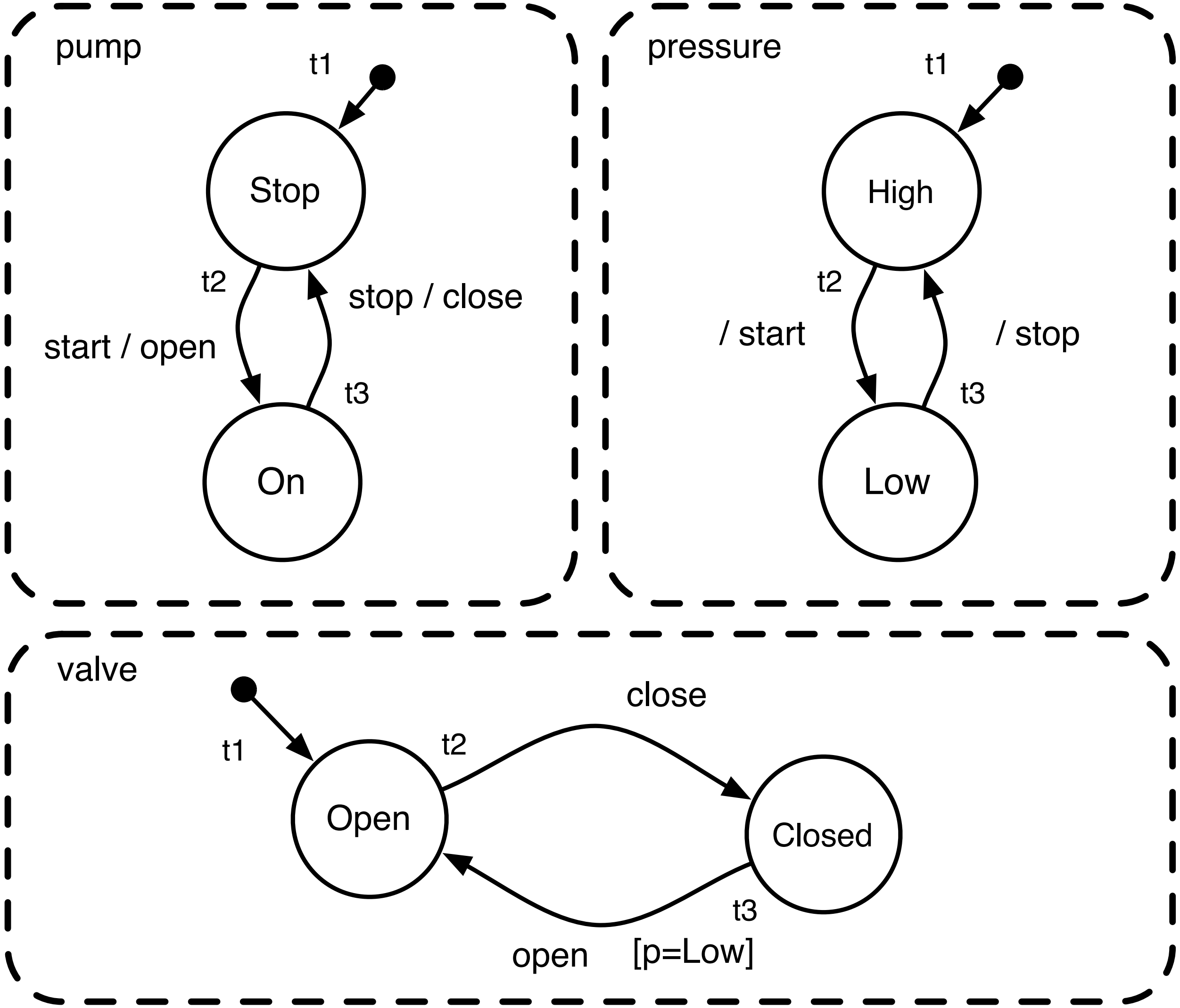
The scenario

- Interconnecting technologies allow to easily assemble network of medical devices in order to obtain an emergent behavior;
- Such a global behavior is typically modelled through nowadays modelling tools and paradigm:
  - Any device **is allowed** to **read** an **modify** the status of any other device in the network;
- **But**, as more devices are added, complexity of the design **growths exponentially, and becomes easily uncontrollable**;
- Aim of this paper is to:
  - Show that nowadays modelling tools are **not effective** in controlling and understanding the **complexity** of the behavior being assembled;
  - Propose the use of different **tools** and **methodologies** for modelling emergent system behavior and to ensure **global safety** and **liveness**;

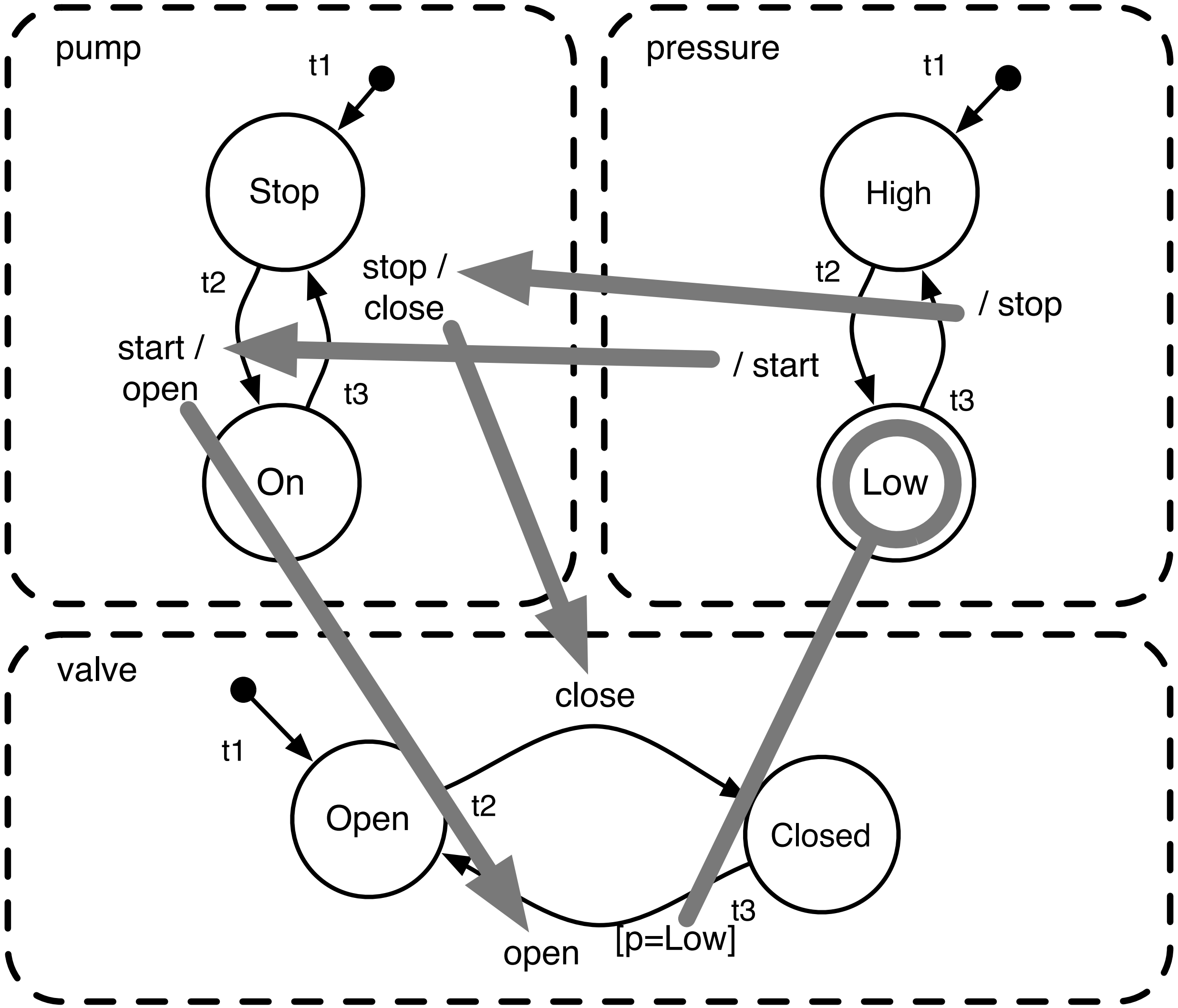
Example

- System made of three components:
  - an infusion pump;
  - an anti-reflux valve;
  - a blood pressure sensor.
- The designer wants to enforce a global behavior such that:
  - The anti-reflux valve has to be opened before the pump starts and to be closed after it stops;
  - It is further required that the blood pressure be always under a certain threshold during blood pump workout; in case blood pressure raises the pump has to be stopped and the valve has to be closed.

The Statecharts Model



Cluttered causal relationships!



Problems with the Statecharts Model

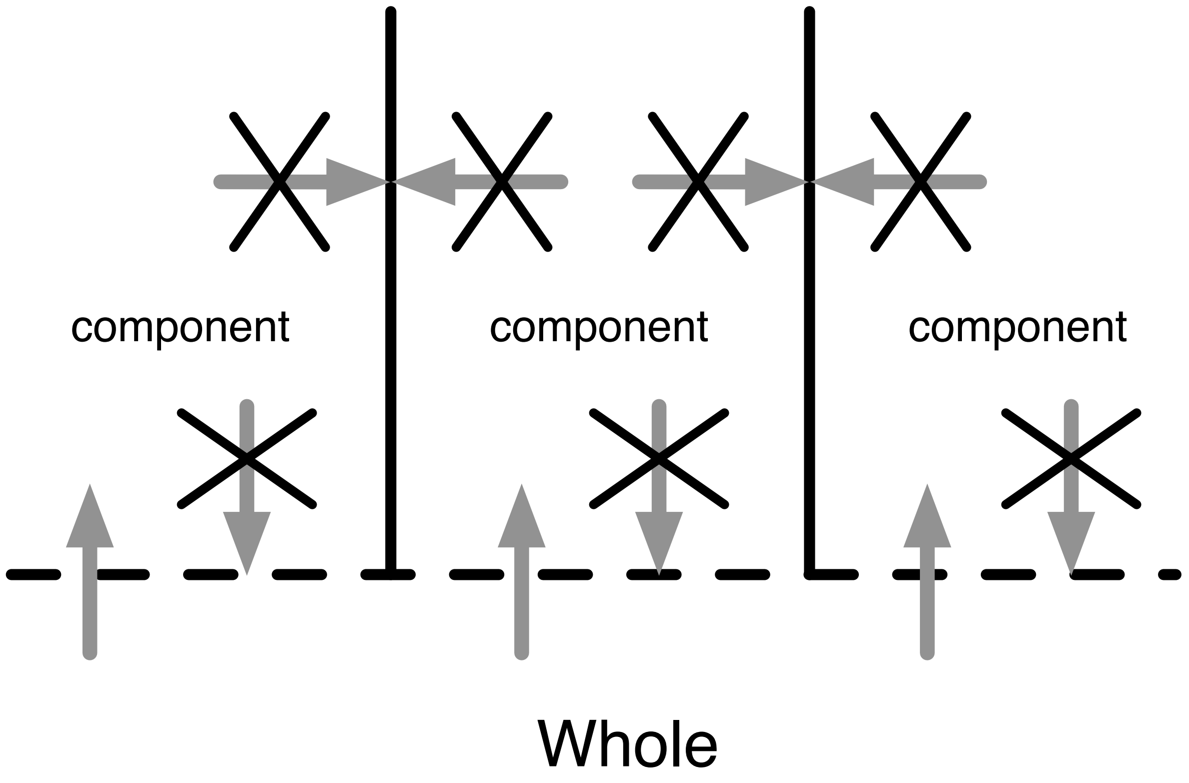
- The intended systemic behavior is difficult to specify, test, understand, modify, exchange due to the intrinsic model of interconnection based on mutual event/command exchange and condition evaluation; two kinds of related problems:
  - On one hand software is not **self-contained** since it depends on the behavior of other machines;
  - On the other hand, **operational problems** are raised, since it is **difficult to predict** which global status will be reached by the networked system, thus jeopardizing the overall safety and liveness of the assembled system;
    - – Model checking not feasible in all cases;
    - – Requires however the designer to manage complex temporal logic formulae without having a complete view of the behavior being modelled.

Explicit Modelling by Part-Whole Statecharts

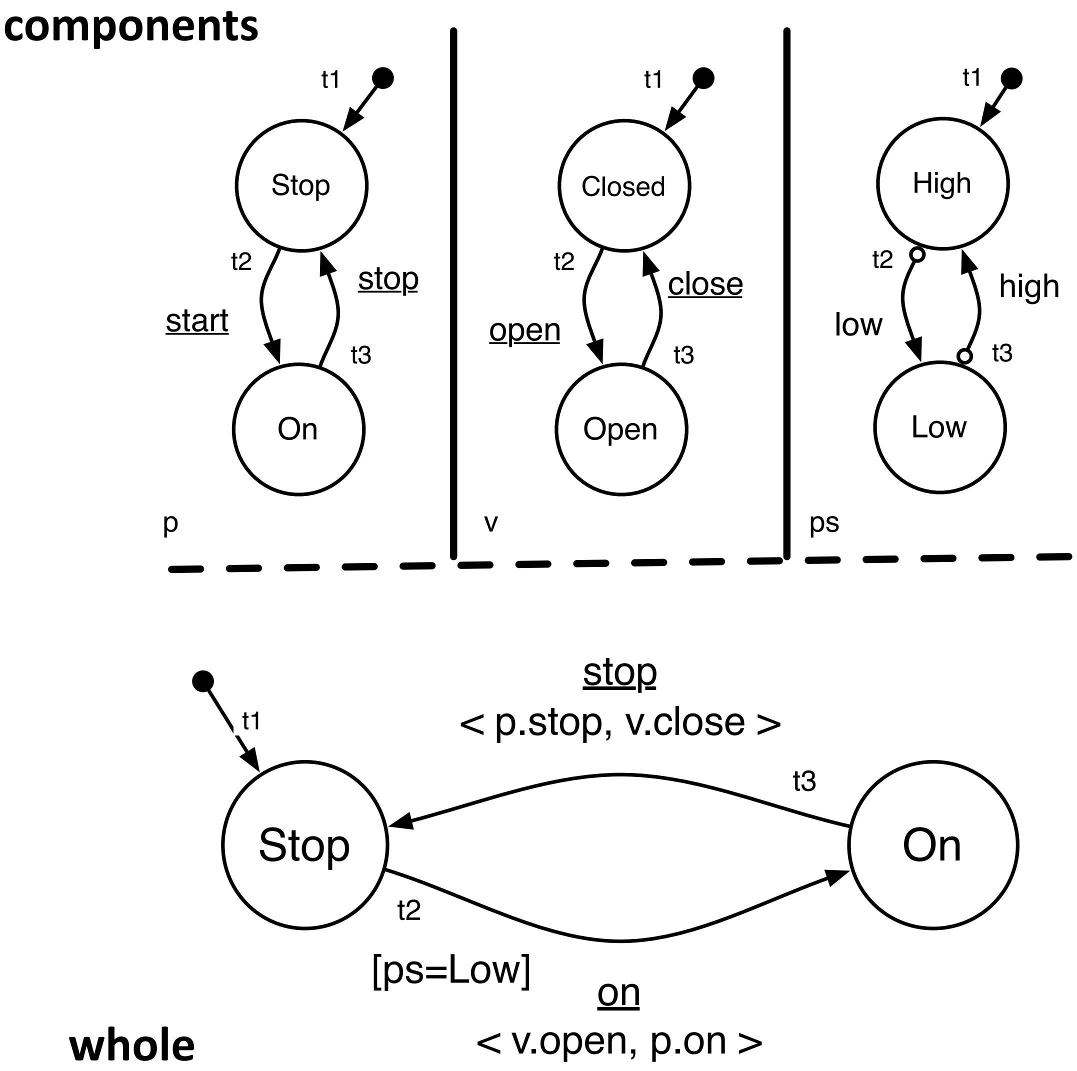
- PW-Statecharts are an evolution of the original Statecharts formalism introduced with the aim of improving software quality of behavioral abstractions;
- Emergent behavior denoted explicitly by the “**whole**” state machine, which reduces coupling among behavioral abstractions; moreover
  - It works as an interface for the system of interacting entities;
  - It embeds the semantics of composition which has been removed from the component behaviors, which are now self-contained;
  - It has a semantics which is computable!

PW Statecharts Rationale

- Events are not allowed to travel among components; components are only allowed to communicate with the whole;



The PW Statecharts Model



We have now a state machine, called the **whole**, in place of the cluster of mutual relationships among **component modules**;

Such a state machine represents the whole behavior of the system and is itself a modular unit which is able to reused, tested, extended, certified and so on.

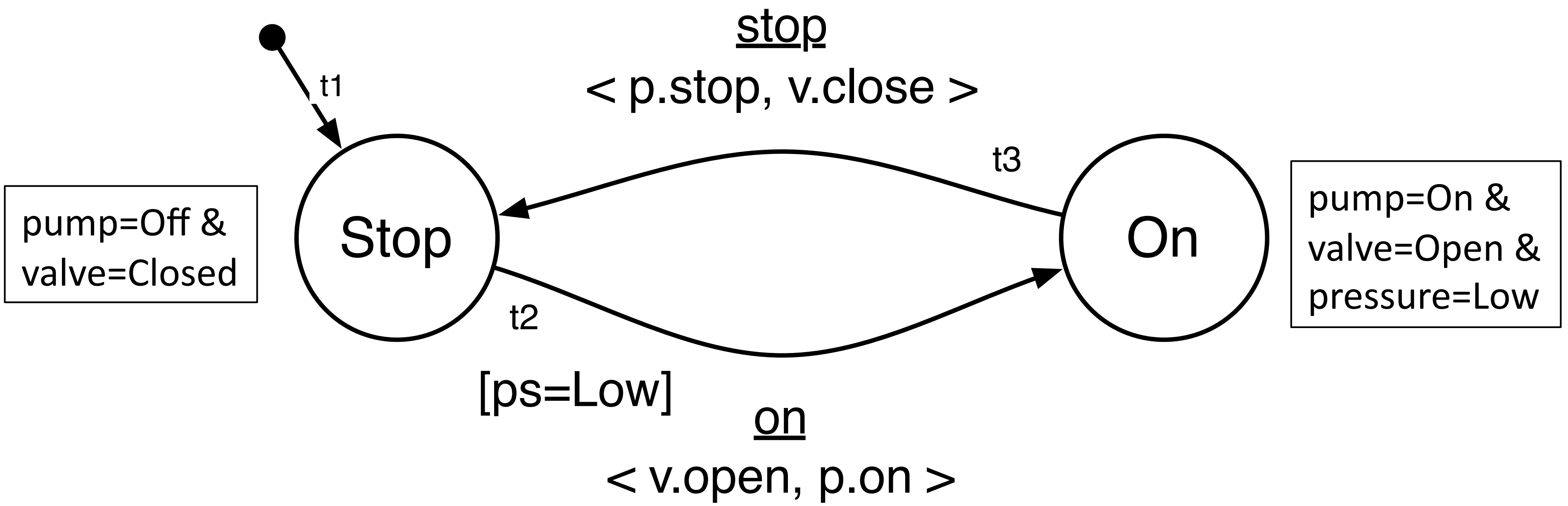
Component modules have been freed on their turn from mutual references which hampered self-containment;

Advantages of PWS modelling I

- Full reusability of components behavior;
- Full reusability of system behavior (whole);
- ➢ Semantics of composition **directly computable at design time**:
  - – Explicit view of the global behavior being modelled;
  - – No need to employ model checking techniques;
  - – States in the whole section may be constrained in fact to have a user defined semantics **which is enforced at design time**:
  - – **On** ⇔ pump=On & valve=Open & pressure=Low
  - – **Stop** ⇔ pump=Off & valve=Closed

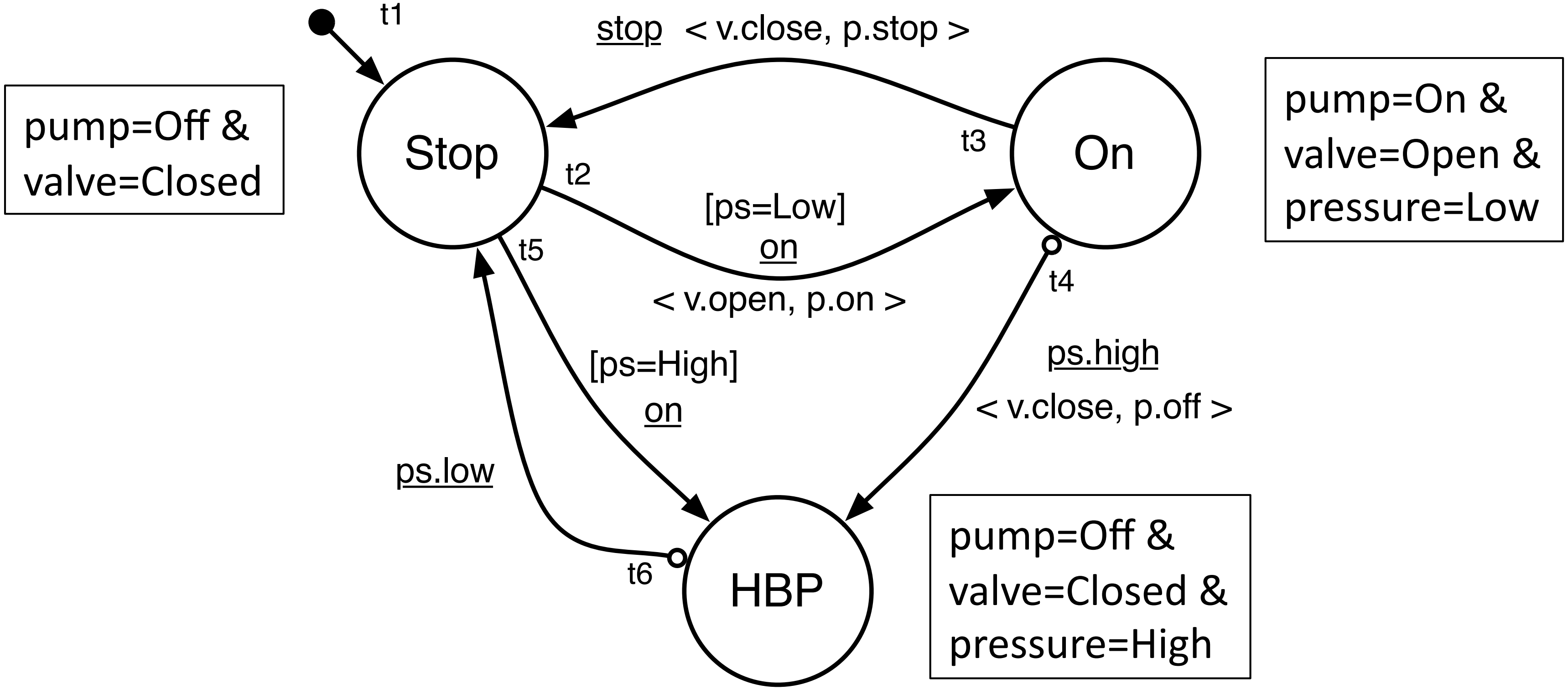
Advantages of PWS modelling II

- A novel methodology may be employed in order to verify consistency of the design with respect to user defined state constraints;
  - **What if** we try to turn the infusion pump on when blood pressure is high?
  - **What if** blood pressure raises when the infusion pump is on?



Final System Design

- The methodology forces the designer to insert a new state HBP (**H**igh **B**lood **P**ressure) in the whole section together with related state transitions:
  - HBP ⇔ pump=Off & valve=Closed & pressure=High;



Conclusions

- We showed how *ad hoc* interconnections among medical devices may easily conceal the global behavioral view of the system being modelled; **hidden behavior** may **endanger** safety & liveness of the system being assembled;
- PW Statecharts (PWS) allow a systemic view of the global behavior being modelled;
- It is also possible to assign user defined properties to the global states of the system as well as to check for them **at design time** by a novel patent pending methodology (PCT/EP2008/051300) applied to PWS modelled systems which:
  - Ensures user specified safety & liveness properties;
  - Allow to **certify** the behavior of an assembled system;

  …without resorting to model checking techniques an/or long and not always exhaustive test cases.

Thank you!