Time and Processes: Towards Engineering Temporal Requirements

Johann Eder, Marco Franceschetti, Josef Lubas
Alpen-Adria Universität Klagenfurt, Austria
firstname.lastname@aau.at

ICSoft, 6.7.2021

In the beginning -

- there are requirements.

WHAT ARE TEMPORAL REQUIREMENTS?

Examples

Before you can have a PCR test you have to register at the test centers website.

A COVID PCR test certificate is valid for 72 hours.

Access is permitted for up to 72 hours after the PCR Test.

Access is possible for visitors, who received their second vaccination shot more than 21 days ago.

Examples II

A presentation at the conference is 20-25 minutes.

Money transfer between accounts in the EU lasts up to 4 days.

After an online order customers have the right to cancel the order within 2 weeks.

Examples III

The one-time password expires 15 minutes after it was sent to the user

When a wrong password is sent, the account is closed for 30 minutes

Time Failures

- Violation of a temporal constraint
- Consequences
  - Annoyances
  - Penalties
  - Catastrophies
- Difficult or impossible to test
  - Uncertainty
  - Combinatorial explosion
  - Race conditions

Temporal Relations

- Events:
  - Before
  - After
  - At the same time
- Activities:
  - Before
  - After
  - During
  - While
  - Concurrent with

Before you can have a PCR test you have to register at the test centers website.

Quantitative Requirements: Durations

- Duration of activities:

A lecture lasts 90 minutes.

- Minimum and maximum duration:

The talk about temporal requirements lasts between 45 and 60 minutes.

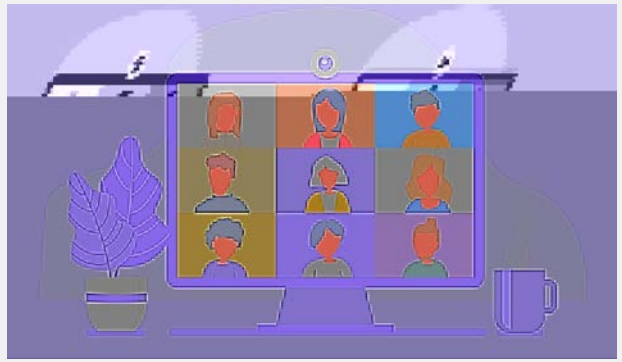
Quantitative Requirements: Durations

CONTINGENT DURATIONS

- Not controllable by executor
- Decide when to start
- Observe when it ends
  - Within known interval

NON-CONTINGENT DURATIONS

- Controllable by executor
- Decide when to start
- Choose when it ends
  - Within known interval
  - Any time during execution



Quantitative Requirements: Bounds

- Lower-bound constraint:
  - Minimum time gap between events

Registration has to be made at least 24 hours before entry.

- Upper-bound constraint:
  - Maximum time gap between events

A COVID-Test certificate must not be older than 72 hours.

Non-functional Temporal Requirements

- Temporal property of a system, which does not influence its function

  - Duration
  - Availability
  - Reaction Time
  - Deadline

Functional Requirements

- The definition of the function contains temporal aspects

If the invoice is paid within 7 days, then 2% can be deducted from the amount.

An order can be cancelled with 14 days.

Descriptive Requirements

- Temporal properties of the environment, of „nature“

Bank transfer lasts up to 4 days.

Access is open from

8:00 – 12:00.

Proscriptive Requirements

- Temporal goals to be achieved

The test certficate should be sent within 24 hours.

Classifying Temporal Requirements

Relational	Quantitative
Controllable (Non-contingent)	Non-controllable (contingent)
Descriptive	Proscriptive
Functional	Non-functional

REPRESENTING TEMPORAL REQUIREMENTS

Temporal Logic Representations of Requirements

```
Workflow WF_Device_Purchase()==[
  %LOC[Role applicant, amanager, ceo, pmanager,
            shopper;
       Task report, signature, permit, audit, order;
       Timer d1=5;
       Var cost;]
  %STM[
     lb=START=>$Olb=s1;
     lb=s1=>$O{(applicant, report)}^$Olb=s2;
    lb=s2=>$O{(amanager, signature)}^$Olb=s3;
    lb=s3^(cost>=1000)=>
        $O{(ceo, permit)}^$Olb=s4;
    lb=s4=>$O{(pmanager, audit)}^$Olb=s5;
    lb=s5=>$O{(shopper, order)}^$Olb=s6;
        lb=s6=>$Olb=STOP]
  ]
```

Source: H. Ma, “A Workflow Model Based on Temporal Logic”, Proc. 8th Int. Conf. Computer Supported Cooperative Work in Design

$$[u_0 \models \varphi_r]_{RV} = \top^p \quad [u_0 \models \varphi_n]_{RV} = \top^p \quad [u_0 \models \varphi_a]_{RV} = \top^p \quad [u_0 \models \varphi_p]_{RV} = \top^p$$

$$[u_1 \models \varphi_r]_{RV} = \top^p \quad [u_1 \models \varphi_n]_{RV} = \top^p \quad [u_1 \models \varphi_a]_{RV} = \bot^p \quad [u_1 \models \varphi_p]_{RV} = \top^p$$

$$[u_2 \models \varphi_r]_{RV} = \top^p \quad [u_2 \models \varphi_n]_{RV} = \top^p \quad [u_2 \models \varphi_a]_{RV} = \top^p \quad [u_2 \models \varphi_p]_{RV} = \top^p$$

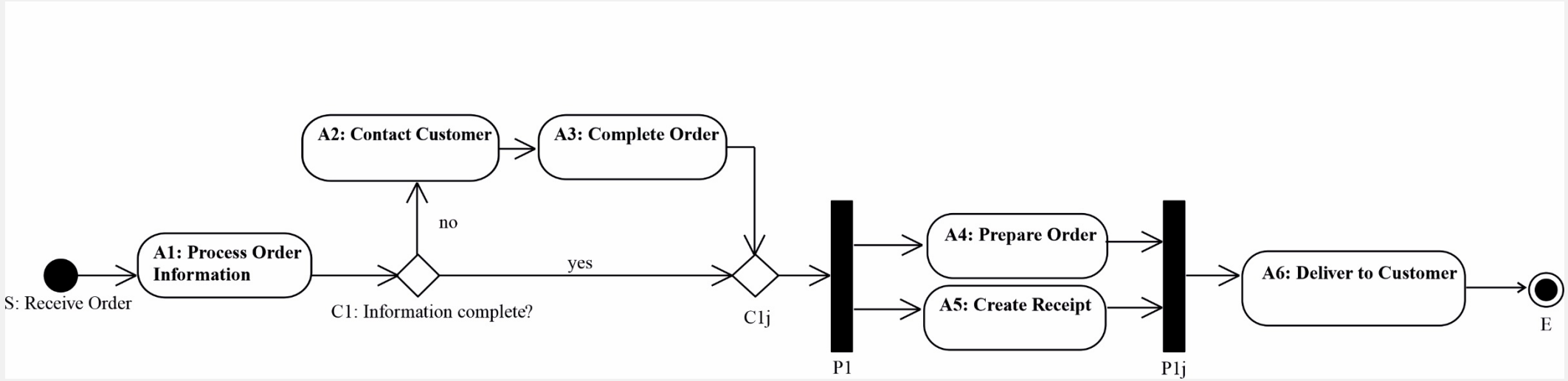
$$[u_3 \models \varphi_r]_{RV} = \top^p \quad [u_3 \models \varphi_n]_{RV} = \bot \quad [u_3 \models \varphi_a]_{RV} = \top^p \quad [u_3 \models \varphi_p]_{RV} = \bot$$

$$[u_4 \models \varphi_r]_{RV} = \top^p \quad [u_4 \models \varphi_n]_{RV} = \bot \quad [u_4 \models \varphi_a]_{RV} = \bot^p \quad [u_4 \models \varphi_p]_{RV} = \bot$$

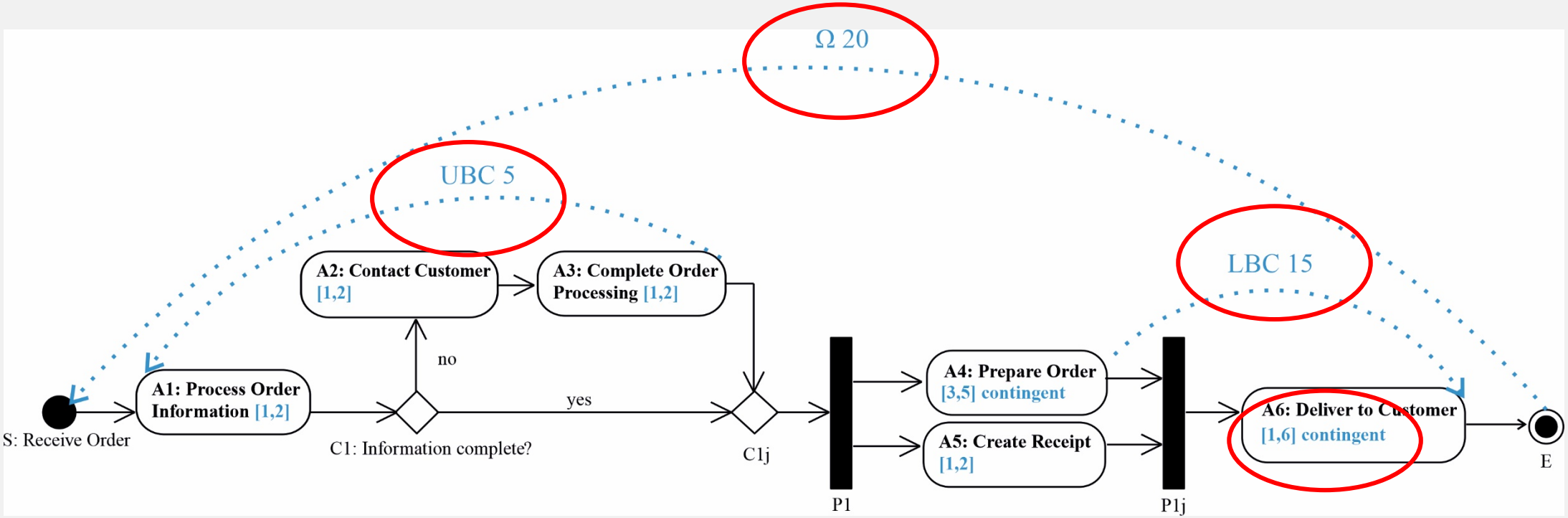
$$[u_f \models \varphi_r]_{RV} = \top \quad [u_f \models \varphi_n]_{RV} = \bot \quad [u_f \models \varphi_a]_{RV} = \bot \quad [u_f \models \varphi_p]_{RV} = \bot$$

Traces representation.
Source: M. Maggi, M. Montali, M. Westergaard and Wil M.P. van der Aalst, “Monitoring Business Constraints with Linear Temporal Logic: An Approach Based on Colored Automata”, Int. Conf. Business Process Management, 2011

Representing Temporal Requirements



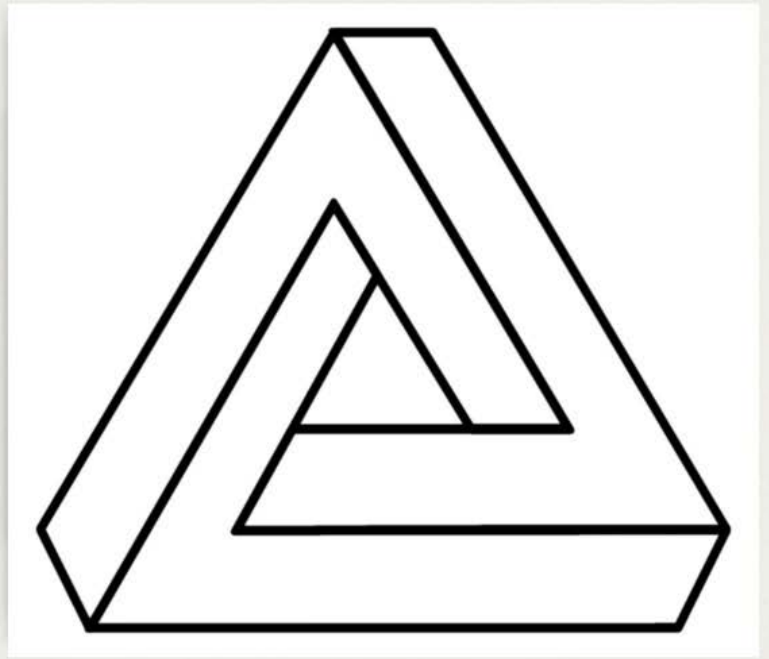
Activity Chart with Temporal Constraints



CHECKING TEMPORAL REQUIREMENTS

Conflicting Requirements

- Temporal constraints in conflict
  - Detection
  - Resolution
- Conflicting temporal requirements lead to time failures



Examples for conflicts

- Precedence conflicts:

A <= B, C <= D

D <= A, C = B

- Uncertainty:

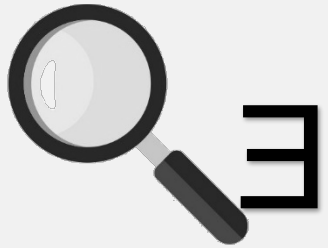
Item should be delivered at the customer within 5 days after the order (but after payment is received).

- Dependence on future events:

Remove the milk from the stove 2 minutes before it bowls over

Satisfiability vs Controllability

- **Satisfiability**:
  Is it at all possible to have a temporally correct execution?



- **Controllability**:
  Is it possible to guarantee for a temporally correct execution under all foreseeable circumstances?

Dynamic Controllability

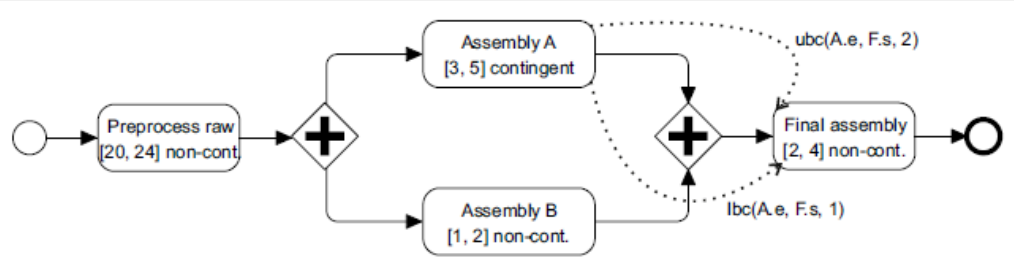
- Dynamic schedule (execution strategy):
  - start time of an activity may depend on all variables whose value is smaller

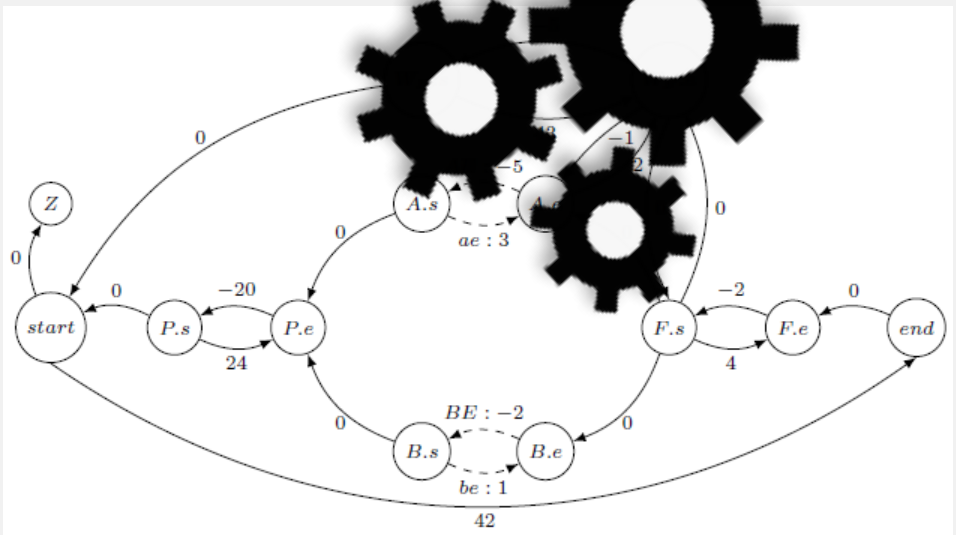
- Dynamically controllable:
  - There exists a dynamic schedule (execution strategy) such that all scenarios over the dynamic schedule are valid

Dynamic Controllability ➔ No conflict between constraints

Checking Dynamic Controllability

- Common approach:
  1. Encode process into Temporal Constraint Network (TCN)
     - STN / STNU/ CSTN / CSTNU / CSTNUD / …
  2. Apply TCN checking procedures



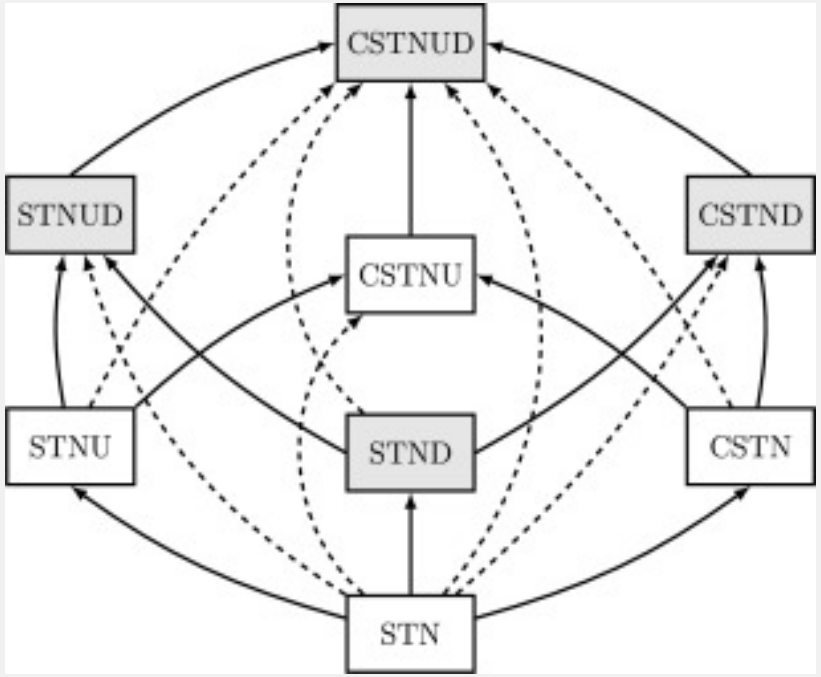


Modeling Languages and Controllability Check

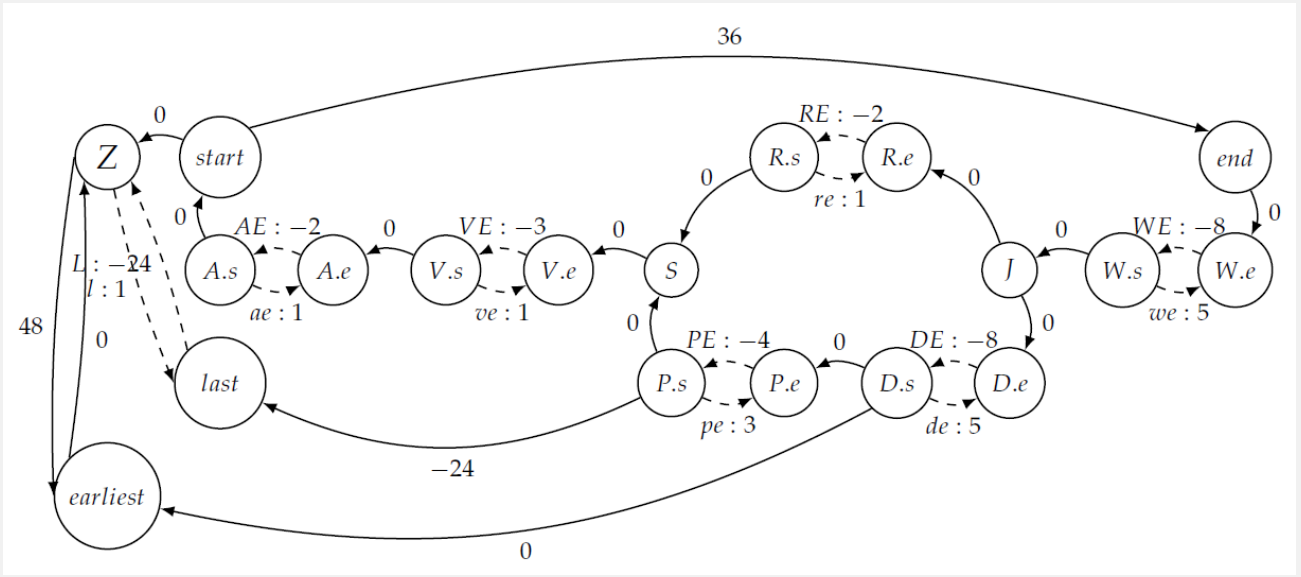
Temporal Expressiveness

Controllability Check

Temporal Constraint Networks

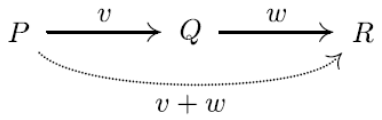
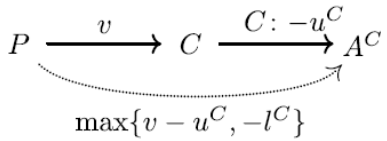
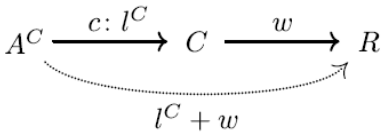


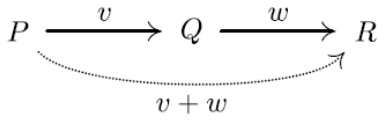
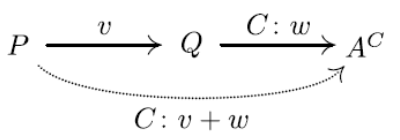
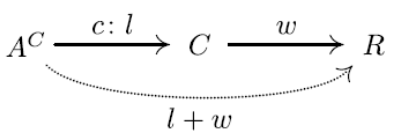
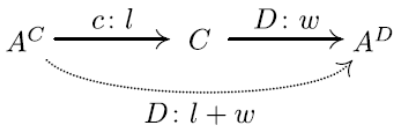
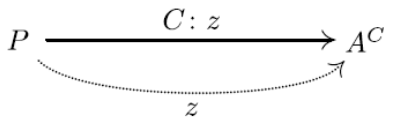
Matteo Zavatteri, Luca Viganò,
Conditional simple temporal networks with uncertainty and decisions,
Theoretical Computer Science,
Volume 797, 2019, Pages 77-101



STNU example

Reasoning: RUL, MMV

Rule	Graph representation	Applicability conditions
RELAX		(none)
UPPER		(none)
LOWER		$w \le 0$ for non-contingent R $w \le u^R$ for contingent $R \not\equiv C$ with upper-bound u^R

Rule	Graph representation	Applicability conditions
No Case		(none)
Upper Case		$P \neq C$
Lower Case		$w \le 0$, $R \neq C$
Cross Case		$w \le 0$, $C \neq D$
Label Removal		$z \ge -l^C$

Massimo Cairo, Romeo Rizzi,
Dynamic controllability of simple temporal networks with uncertainty: Simple rules and fast real-time execution,
Theoretical Computer Science,
Volume 797,2019,Pages 2-16

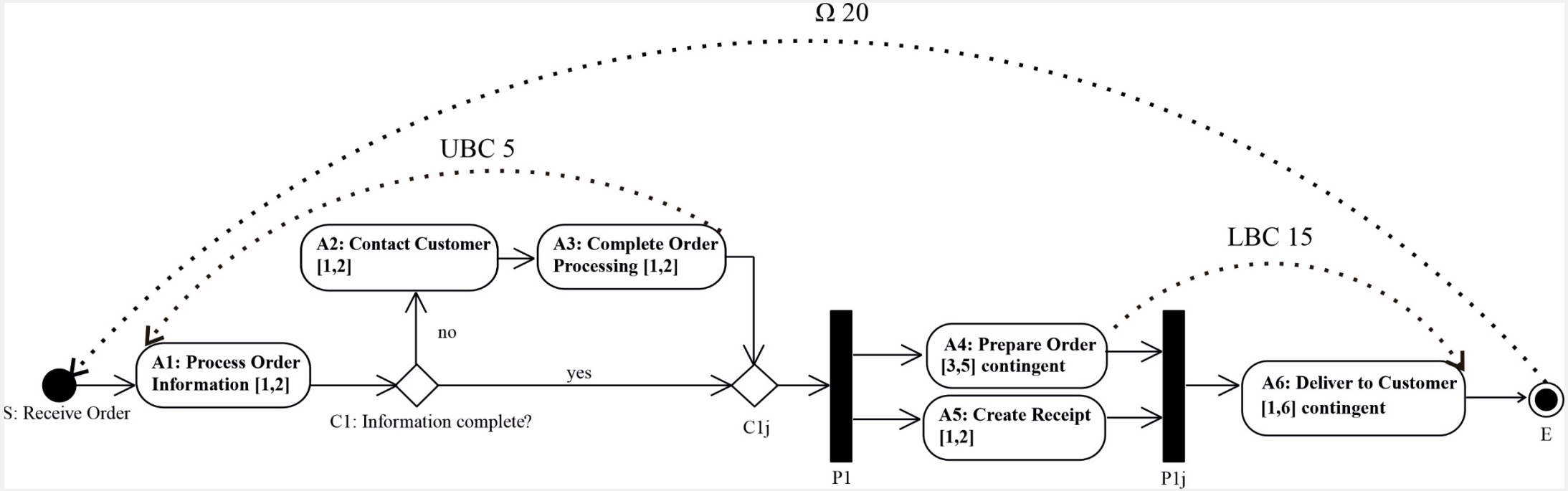
ENGINEERING TEMPORAL REQUIREMENTS

Temporal Requirements Manager

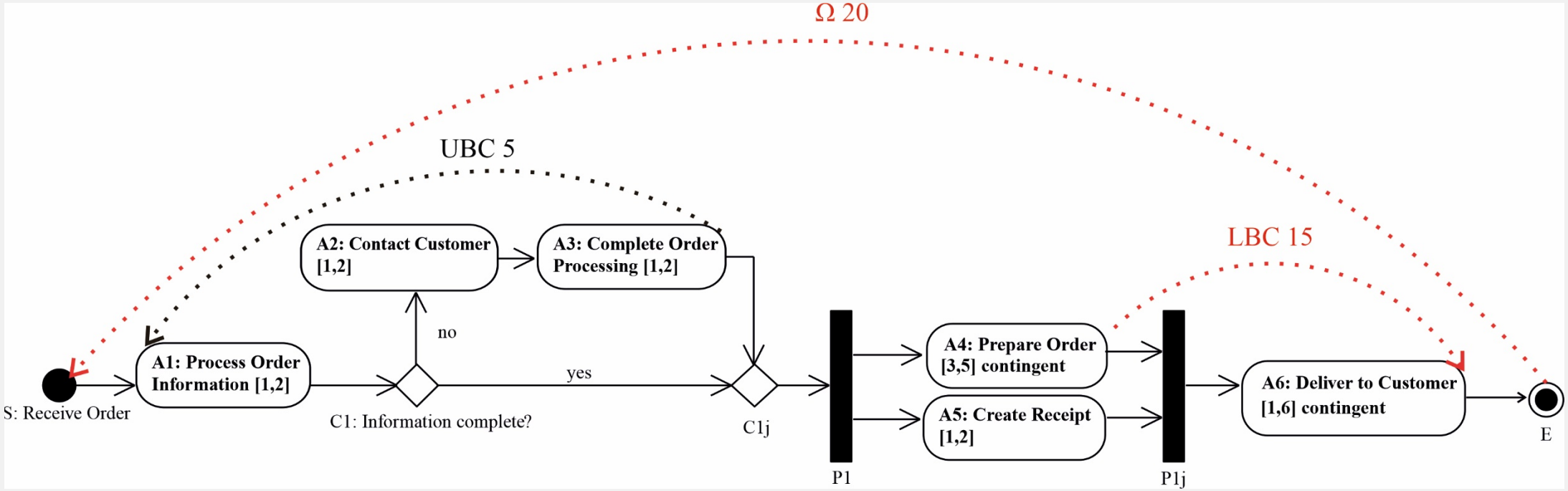
Tool support for managing temporal requirements:

- Is a set of temporal constraints dynamically controllable?
- Which constraints are in conflict?
- Which is the strongest constraint between 2 events that does not create a conflict?
- Incrementally include temporal requirements and check whether they are in conflict with existing ones.

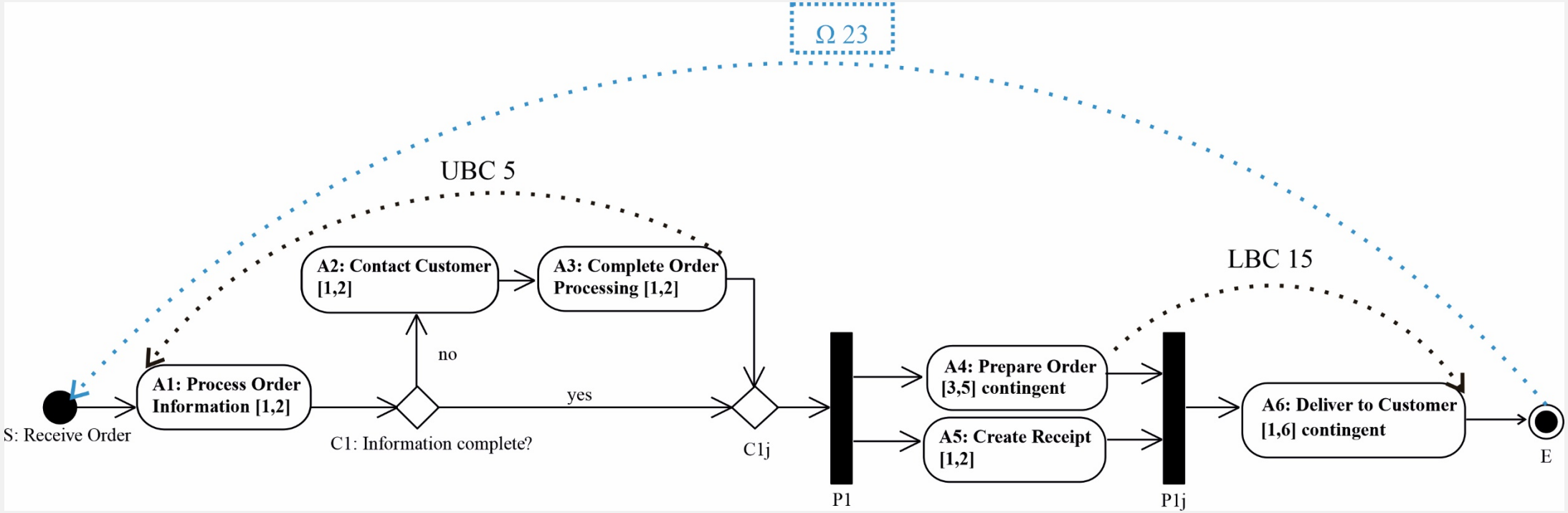
TRM: Temporal Requirements Manager



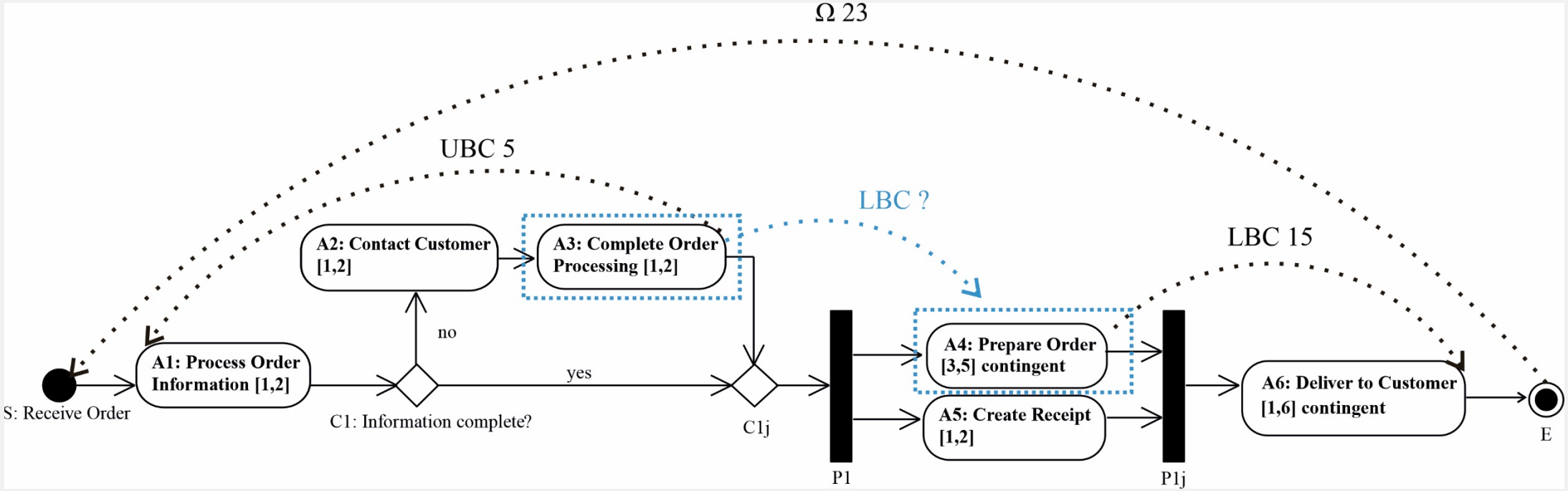
TRM Step 1: Conflict identification



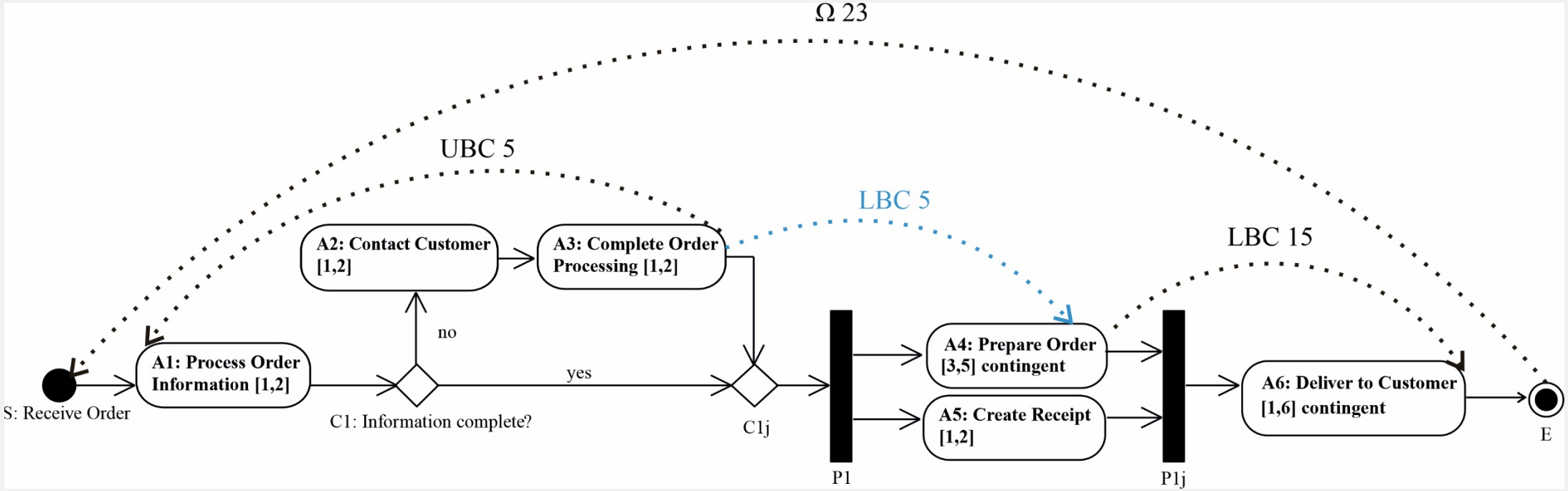
TRM Step 2: Conflict resolution



TRM Step 3: Maximum constraint bounds



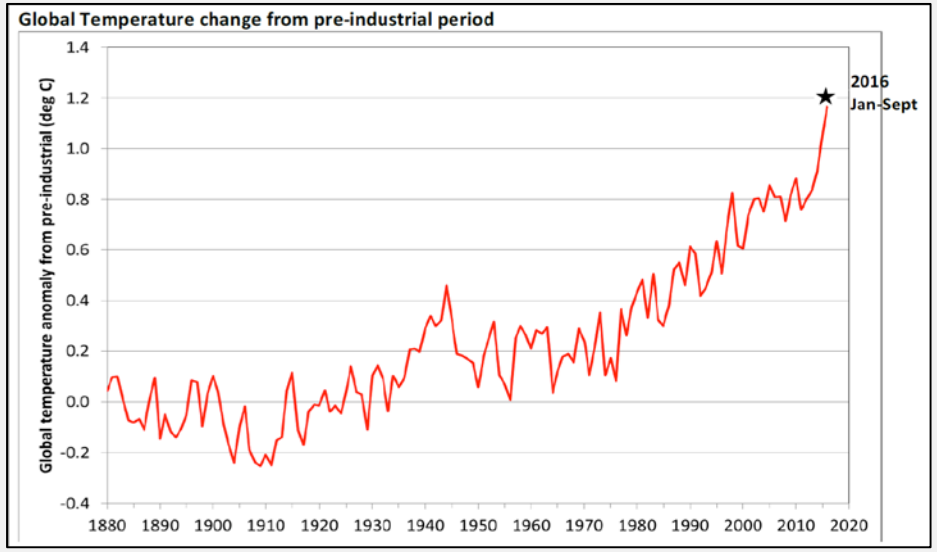
Mockup Step 4: Correct model

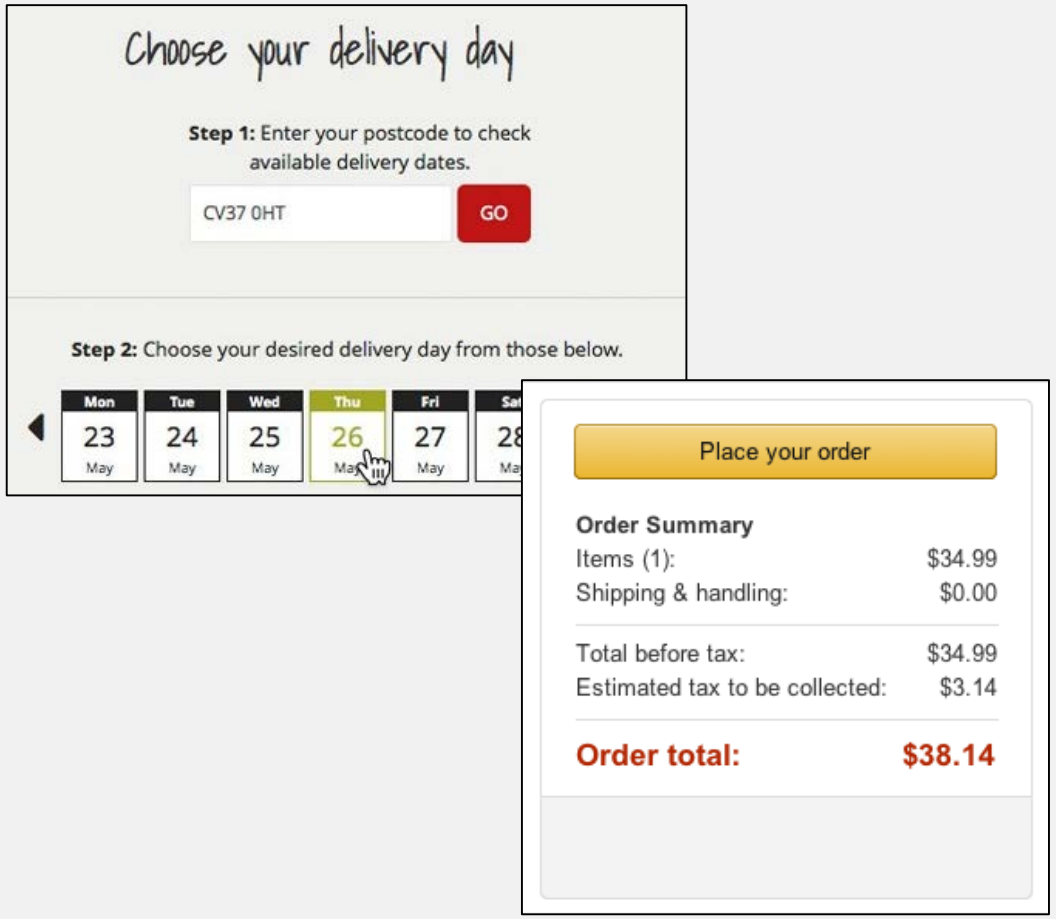


FURTHER ISSUES ...

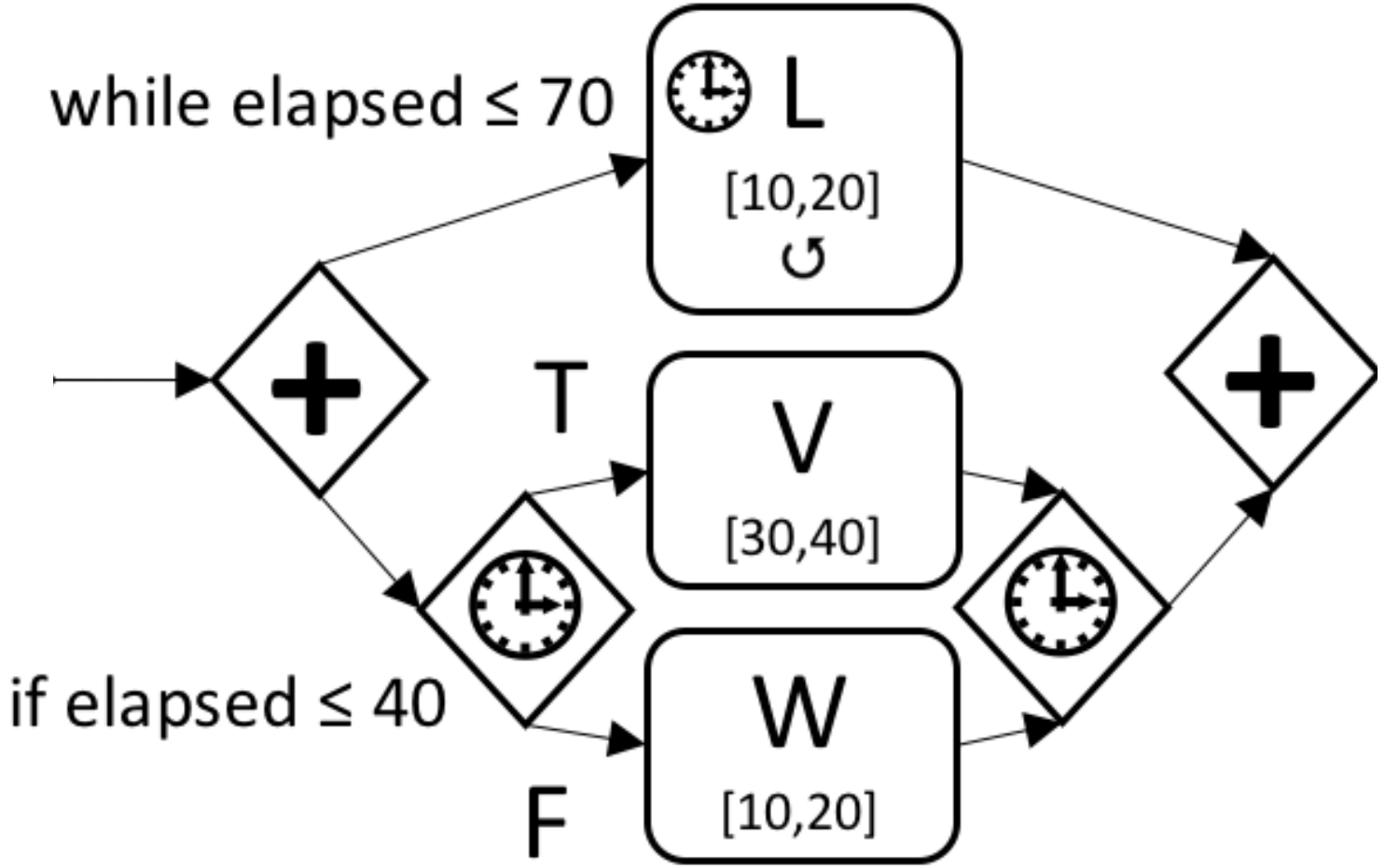
Challenges: Temporal Data

- Data of type date/timestamp
- History of data
- Data versions





Challenge: Temporal Control Structures

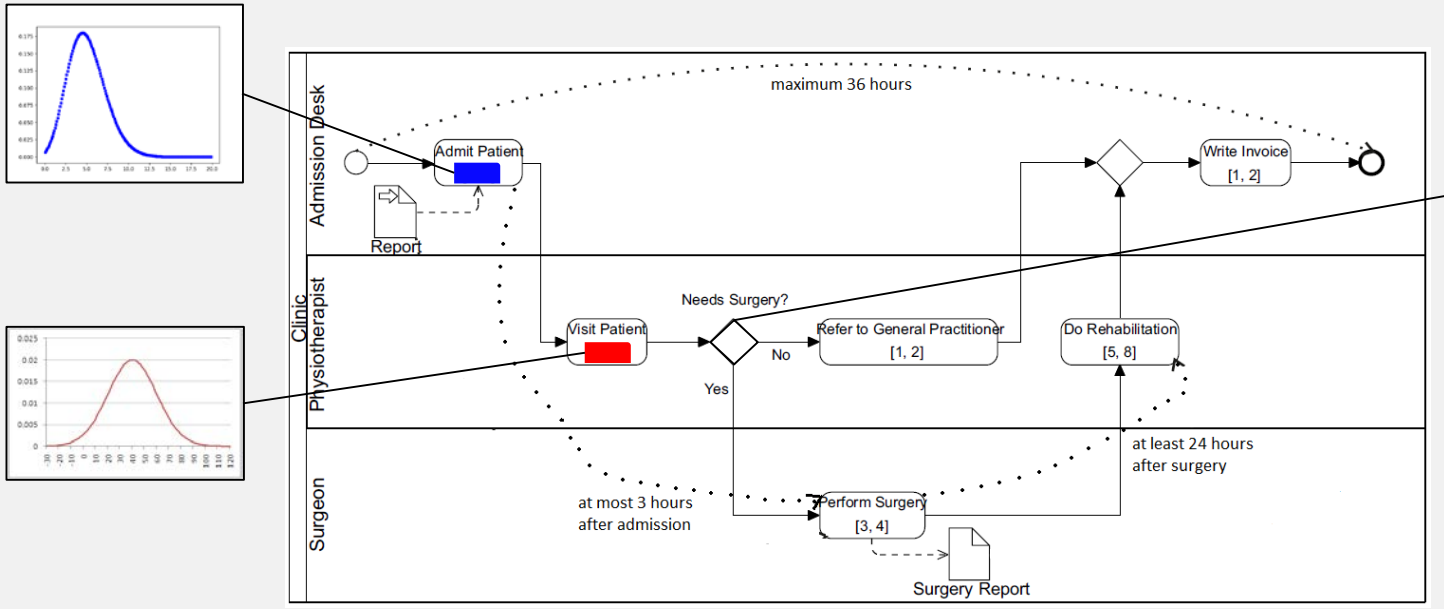


Challenges: Probabilistic Controllability

- Non-binary property
- In real world applications, risk is accepted and taken
- Probability of time failure in general
- Probability that a particular constraint is violated
  - e.g. risk to deliver product later than promised < 5%
  - e.g. risk to miss train < 10%
  - e.g. risk to miss the flight < 1%

Probabilistic uncertainty

- Distribution function for duration
- Branching probability at XOR splits
- Distribution function for number of iterations



Patient Needs Surgery:
85% of the time NO
15% of the time YES

Conclusions: TIME MATTERS

- Temporal requirements are everywhere
- Temporal constraints are a simple way of representing temporal requirements in activity charts
- Dynamic Controllability is a good definition for „free of conflicts“
- Theory delivers effective checking procedures for conflicting temporal requirements
- Tools support representing and checking temporal requirements