Formal methods for batch production systems

Ronan Champagnat
Robert Valette
Hervé Pingaud

This work has been partially supported by the French-Spanish integrated program PICASSO in cooperation with the DIIS-CPS (U. Zaragoza)

Plan

• Introduction

• Hybrid modeling with Petri nets
  – Abstracting continuous behavior by means of time
  – Modeling with colored Petri nets
  – Modeling with hybrid Petri nets
  – Modeling with Petri nets and differential algebraic equations

• Analysis
  – Analysis needs for batch systems
  – Analysis with ordinary Petri nets
  – Analysis with t-time Petri nets
  – Analysis with differential predicate transition nets

• Conclusion

Workshop "Formal Methods for Manufacturing", Zaragoza, sept. 1999
Introduction (1)

What is a batch production system?

- A manufacturing system
  - recipes are part routes, equipments to process raw material, handling/storage units
- Batches of raw material
  - numbers of batches are integers, production management, event driven production
    \[\Rightarrow\text{discrete aspect}\]
  - batches of continuous raw material, batch size and storage capacity are real numbers
    \[\Rightarrow\text{continuous aspect}\]

Hybrid Modeling Issues for Formal Methods

Introduction (2)

What is a hybrid formal model?

- Continuous (dense, real numbers) state variables
- Discrete (finite set of values or integers) state variables
- Continuous dynamics on a dense time (differential algebraic eq.)
- Discrete event dynamics on a discrete time (automata, Petri nets)

A wide range of approaches / proportion of these items
Introduction (3)

In a batch system:

- **Continuous variables:**
  - batch size, quantities of mole in the solvent, temperature, pressure etc.

- **Continuous dynamics:**
  - heat exchange system (cooling device, sterilization of milk), fluid transfers betw. vessels

- **Discrete variables:**
  - vessel and device states, batch states in a recipe, flowsheet configurations (on/off valves)

- **Discrete dynamics:**
  - recipes, cleaning procedures, set up and shut down procedures

---

Hybrid modeling with Petri nets (1)

Abstracting continuous behavior by means of time (1)

- time Petri net, timed Petri nets, stochastic Petri nets

![Petri net diagram]

- start filling (open valve)
- stop filling after $\Delta t$ (the correct pressure is reached)
Hybrid modeling with Petri nets (2)

Abstracting continuous behavior by means of time (1)

• Only one continuous variable: time

• Only the simplest continuous dynamics: \( \frac{dx}{dt} = 1 \)
  – not always possible to predict duration (transfer of variable size batches)

• No restriction about discrete states (places)

• Complex discrete event dynamics: Petri net
  – resource allocation (conflicts)
  – true concurrency (not interleaving, partial order semantics)

Hybrid modeling with Petri nets (3)

Modeling with colored Petri nets (1)

– continuous attributes on tokens
– based on discrete time (sampled system)
Hybrid modeling with Petri nets (4)

Modeling with colored Petri nets (2)
- Continuous variables (token attributes)
- Discrete variables (places)
- Complex discrete event dynamics
- Complex continuous dynamics by means of sampling process?
  → Yes but approximation and cumbersome simulation

Hybrid modeling with Petri nets (5)

Modeling with colored Petri nets (3)

Inconsistency between continuous time and discrete one (events).

For more precision, many samples. Results in cumbersome simulations
Hybrid modeling with Petri nets (6)

Modeling with hybrid Petri nets (1)
- continuous token loads for cont. places,
- continuous firing of cont. transitions
- hourglass (water clock or clepsydra) principle

Hybrid modeling with Petri nets (7)

Modeling with hybrid Petri nets (2)
- Continuous variables (positive?) : cont. place token loads
- Discrete variables : discrete place markings
- Complex discrete event dynamics
- Linear continuous dynamics : dx/dt = t_speed

Ensure consistency between dense time and events
Hybrid modeling with Petri nets (8)

Modeling with hybrid Petri nets (3)

Simulation remains an efficient discrete event one

Only events are considered and precision is ensured

Hybrid modeling with Petri nets (9)

Modeling with hybrid Petri nets - extensions (4)

- negative token loads (zero threshold is no longer an event)
- predefined set of cont. variables and equations (batch Petri nets)
- variable jumps

algebraic constraints?
Hybrid modeling with Petri nets (10)

Modeling with Petri nets and differential algebraic equations (1)
- continuous variables, global or token attributes
- differential algebraic syst. attached to places or markings

Hybrid modeling with Petri nets (11)

Modeling with Petri nets and differential algebraic equations (2)
- General approach for continuous variables and their dynamics
- General approach for discrete event variables and dynamics
- Open to any solver for ensuring consistency between events and integration steps
- But it is necessary to choose an adequate solver for each case
Hybrid modeling with Petri nets (12)

Modeling with Petri nets and differential algebraic equations (2)

The selected solver has to ensure consistency between integration steps and events

Hybrid modeling with Petri nets (13)

Modeling with Petri nets and differential algebraic equations (3)

- purpose of the solver: find the first solution in time for event
- may be done analytically => time colored Petri nets
- can be seen as a generalization of "batch" Petri nets (& hybrid)
- can be seen as an extension to Petri nets of hybrid automata

Requires modularity of continuous and discrete view
Analysis (1)

Analysis needs for batch systems

- In food industry, fine chemical: consumer & environment
- Market competition: satisfy demand on due date

Safety properties
- Critical resource available when required, buffer capacity sufficient (overflow / shortage)

Liveness properties
- No deadlock, recipes can indeed be executed (trajectories from initial to final possible)

Analysis (2)

Analysis with ordinary Petri nets (1): redundant places
- Safety property, critical resource

If for security reasons it is absolutely necessary to have a resource available, then the corresponding place has to be redundant (P10)
Analysis (3)

Analysis with ordinary Petri nets (2)

• More generally, any property which has to be logically ensured
  – logically = does not depend on time or continuous variables

• Deadlock free, liveness, boundedness, reversibility

• p-invariant (and redundant place), siphon, etc for analyzing resource allocation policies

  All the classical Petri net analysis remains valid

Analysis (4)

Analysis with t-time Petri nets (1)

• Operation durations (observed / required) abstract cont. behav.
  – can easily be represented by PN
  – for example observed : enabling transition durations, required : sojourn time on places
  – they are relative time

• Due dates from demand, dates for workload balance
  – they derive from an aggregate point of view
  – they are absolute time points, they are not easily expressed by means of PN

• Logical constraints as Kanban policies, cyclic ones
  – may be described by places and transitions
Analysis (5)

Analysis with t-time Petri nets (2)

• Systematic analysis (state enumeration)

=> class graph construction for t-time and p-time
  – formal proof that a state is reachable or not: safety or liveness
  – decidability (if PN bounded) but state space explosion - relative time

• Minimal and maximal scenario duration

=> diodes (max, +) algebra, linear logic sequent calculus

Analysis (6)

Analysis with t-time Petri nets (3): a resource is idle when required

M = p_1 p_4 p_9

\[ t_5 \text{ fired, resource required only class} 5 \text{ is reached} \]

\[ t_7 \text{ enabled [0,0]} \]

No wait
Analysis (7)

Analysis with differential predicate transition nets (1)

- **No decidability (consistent with hybrid automata)**
  - the number of states is infinite, the number of classes is infinite
  
  **Decidability:**
  - firing date (first solution of DAE system) delimited by a static interval

- **Algebraic properties differ if cont. var. or discrete ones**

- **Region graph approach (Regions: Hyb. Aut., Classes: t-time PN)**
  - local information $\Rightarrow$ exploit concurrency
  - a fragment of marking (i.e. all the markings such that some place token load $> 1$)
  - a set of constraints (invariant) $\Rightarrow$ static delimitation of events (trans. fir.)
  - proof that some state cannot be reached $\Rightarrow$ **Safety Properties**

Analysis (8)

Analysis with differential predicate transition nets (2) : Safety

- To be proved : "there is no overflow of the buffer"

![Diagram of a system with a buffer and reactions](image)
Analysis (9)

Analysis with differential predicate transition nets (3) : Safety

- To be proved: "there is no overflow of the buffer"
- $V_B$ only increases when at least one token in $p_2$
- $p$-invariant implies "one token at most in $p_2$"
- Condition $t_5: V_B < V_{\max} - V_B$
- Invariant $(V_B + V_R = \text{Const})$ and $(V_R > 0)$

$\Rightarrow V_B < V_{\max}$

Proof with one region
encapsulating
all markings with $M(p_2) = 1$
all time points in dense time
By: when go in and out the region

Analysis (10)

Analysis with differential predicate transition nets (4) : Safety

- To be proved: "there is no deadlock"
- The underlying Petri net is live $\Rightarrow$ analyze trans. with conditions
- Condition $t_5: V_B < V_{\max} - V_R$
- $M(p_3) > 1$
- To decrease $V_B : M(p_3) = 1$
- It is not reachable if $M(p_3) = 2$

$\Rightarrow$ the property is not verified
if the region can be reached

Proof with one region
Analysis (11)

Analysis with differential predicate transition nets (5) : Safety
- Example of Differential Predicate Transition net without deadlock

![Diagram](image1)

Analysis (12)

Analysis with differential predicate transition nets (6) : Liveness
- Reachability to be proved : "once enabled, \( t_1 \) may be fired"
- Condition \( t_4 : V_B > V_R \)
- \( M(p_3) > 1 \) and \( M(p_1) = 1 \)
- To increase \( V_B \) : \( M(p_3) = 1 \)
- Fire \( t_1, t_2, t_3 \) possibly several times

\[ \Rightarrow \text{the property is verified} \]

Proof with one token
(remains valid whatever location of second token in \( p_3 \) not in \( p_1 \))

![Diagram](image2)
Conclusion (1)

Modeling

- **Formal methods for modeling batch production systems**
  - A whole range of Petri net based formal methods may be used
  - Ordinary Petri nets (complete abstraction of continuous dynamics, qualitative model)
  - Time Petri nets (encapsulation with time)
  - Hybrid Petri nets (encapsulation with linear differential equations)
  - Differential Predicate Transition nets (representation by means of DAEs)
- **Colored Petri net**
  - Continuous variables but no dense time
  - Analysis is done in a sampling framework with a discrete view

Conclusion (2)

Analysis and formal verification

- **Decidability is lost when continuous dynamics "is not time"**
  - Analysis of colored Petri net with continuous attribute is not decidable
  - Analysis of time Petri net is very complex
- **Validation (does the system correspond to the requirements?)**
  - Complex and not totally formal
  - Formal verification: formal model is an abstraction, requirement is not always formal
  - The elaboration of a formal model and its evaluation (performance) by simulation is the typical way validation is addressed in industry