



Implementing tools for the Cheddar programming language with Platypus

**F. Singhoff, A. Plantec
University of Brest, France
LISYC/EA 3883**

Introduction and project motivations

- **Real time scheduling/queueing system theory :**
 - Provide a way to model and predict real time applications performances. Increase safety, quality, ...
 - First results 30 years ago (Liu & Layland). Stay mostly unapplied. Why ? How to improve suitability ?
- **Cheddar framework, increasing suitability:**
 1. Engineers must be trained (educational tool, runnable survey, 2000).
 2. Lacks of the theory : memory footprint analysis, distributed systems (J. Legrand, Phd 2001/04).
 3. Relationships with design languages (AADL, 2005) and engineering process (embedding expert knowledge).
 4. Lack of analysis tools : flexible (scheduler, domain specific language/DSL, 2003), efficient (large simulations)

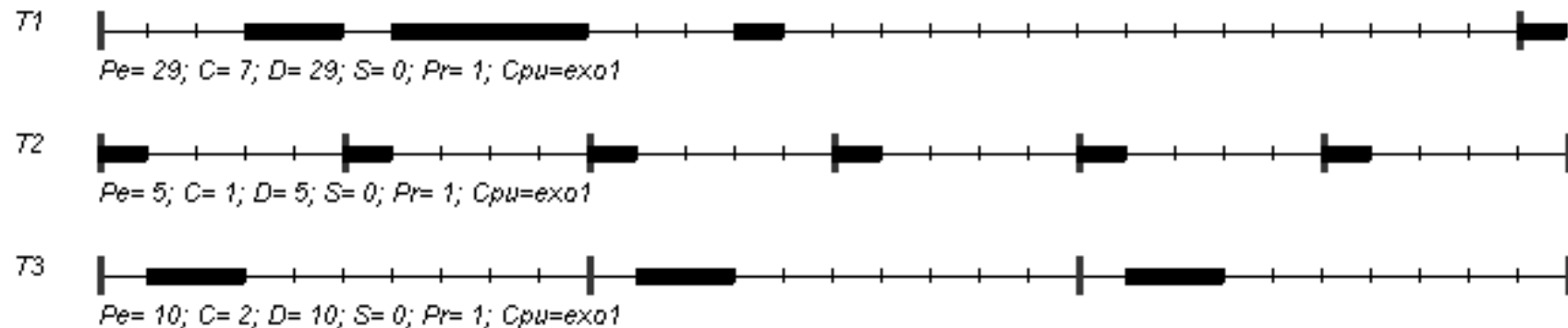
Talk overview



1. Introduction and project motivations
2. Cheddar : a real time systems performance analyzer
 1. Queueing system theory
 2. Real time scheduling theory
3. What we expect from Platypus + Cheddar
4. Conclusion and ongoing works

• Simulation vs analytical tools

- **Simulation** : Rate Monotonic (RM, Liu & Layland 1974), run task with the smallest period



- **Analytical/Feasibility tests example** : the processor utilization factor test

$$\sum_{i=1}^n \frac{C_i}{P_i} \leq n(2^{1/n} - 1) \approx 69\%$$

Cheddar : a real time systems analyzer (1/2)

- **Several levels of modeling and analysis :**
 1. **Built-in** scheduling simulation and analytical tools.
 2. Provides **a domain specific language** (Cheddar programming language) :
 - When no analytical/feasibility test exist.
 - Interpreted user-defined schedulers (Cheddar programs).
 - Cheddar programming language is suitable ... but not enough efficient (industrial case study)
 3. **Extend the framework.** Require to understand the Cheddar framework design.

Cheddar : a real time systems analyzer (2/2)

• Example :

start_section :

dynamic_priority : array (tasks_range) of integer;

gen1 : random;

exponential(gen1, 100);

cycle_duration : array (tasks_range) of integer;

end section;

priority_section :

dynamic_priority := tasks.start_time

+ ((tasks.activation_number-1)*tasks.period)

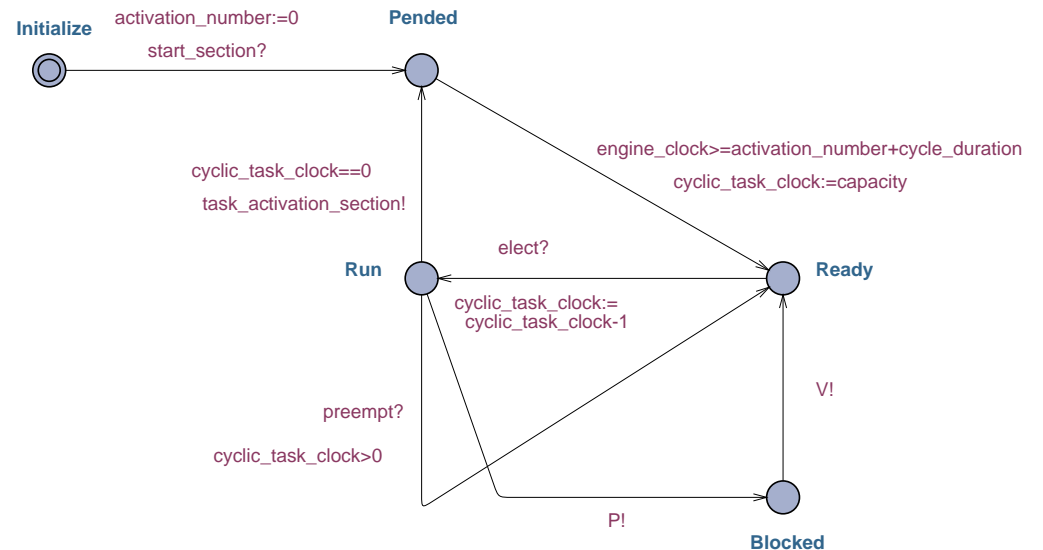
+ tasks.deadline;

end section;

election_section :

return min_to_index(dynamic_priority);

end section;



automaton_section sporadic_model :

Initialize : initial state;

Pended,Ready,Blocked,Run : state:

Initialize --> [,tasks.activation_number:=0 ,start_section?] --> Pended

Ready --> [, cyclic_task_clock:=cyclic_task_clock+1,elect?] --> Run

Blocked --> [, , V!] --> Ready

Run --> [cyclic_task_clock:=0, task_activation_section!] --> Pended

end section;

Talk overview



1. Introduction and project motivations
2. Cheddar : a real time systems performance analyzer
3. What we expect from Platypus + Cheddar
4. Conclusion and ongoing works

Platypus and Cheddar (1/3)

- **Platypus/Cheddar motivations (2005):**

- **First motivation/step :**

1. **Refactoring** of the Cheddar framework (quality, framework readability, configurability)
2. **Modeling** of the Cheddar data model (real time systems meta-model)

- **Second motivation/step:**

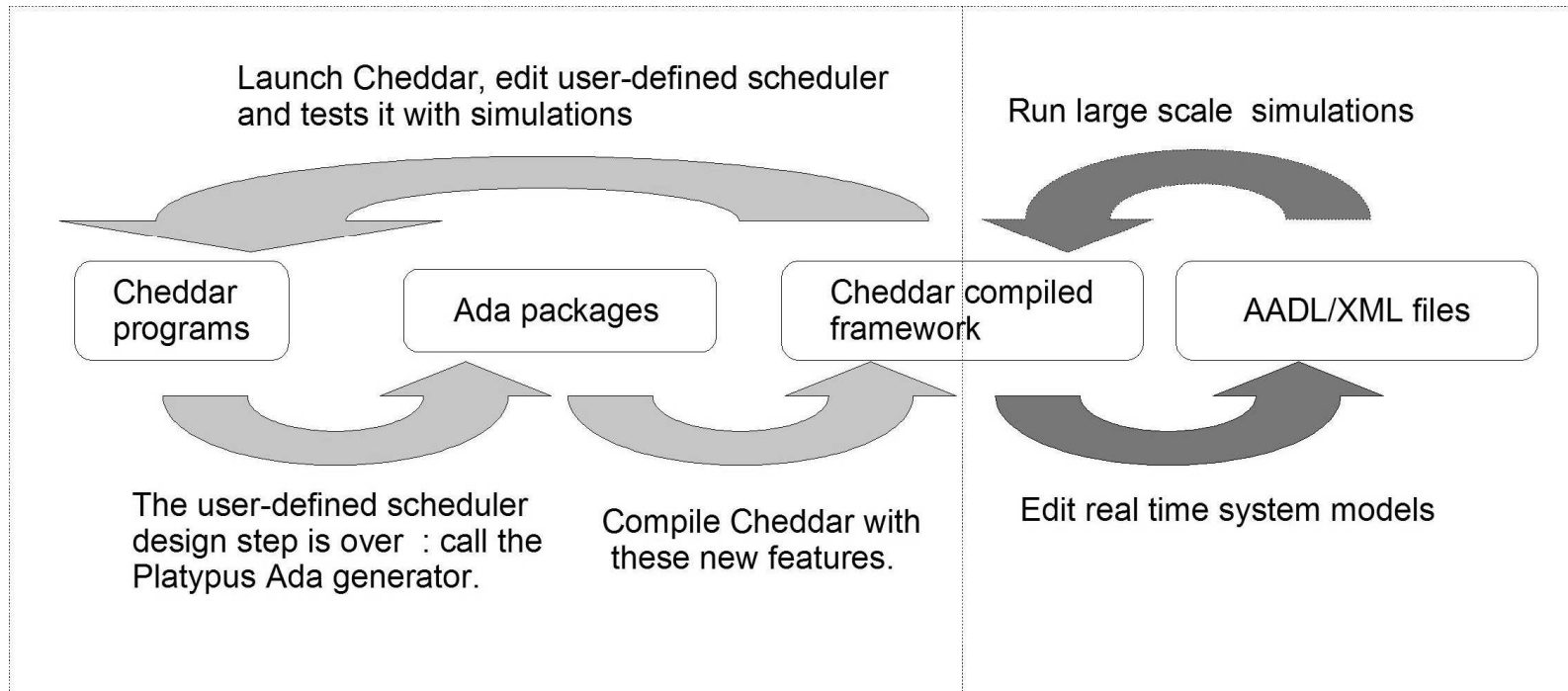
1. **Modeling** of the Cheddar DSL and providing tools for such a DSL (efficiency, large scale simulations).
2. **Platypus enhancing and evaluating.** New investigated domain.

Platypus and Cheddar (2/3)



- **Refactoring to increase flexibility (2006) :**
 - **Ada packages generated by Platypus :**
 - Reduce time to implement new analysis tools (new cheddar data model).
 - Expect to generate 30 % of the framework.
 - **Design/implementation mistakes ;**
 - Removed anti-pattern (open-close principles).
 - Sub-programs moved to the right place
 - Missing sub-programs (CDAI) and renamed identifiers.

Platypus and Cheddar (3/3)



- Model schedulers with the Cheddar programming language.
- Test it with the Cheddar interpreter.
- Run (large) simulations with Ada packages generated by Platypus (from the Cheddar DSL meta-model and the Ada meta-model),

Talk overview



1. Introduction and project motivations
2. Cheddar : a real time systems performance analyzer
3. What we expect from Platypus + Cheddar
4. Conclusion and ongoing works

Conclusion and ongoing works



- **Current status :**

- Cheddar + Platypus = refactoring + DSL tools.
- Refactoring is over, but others packages could be automatically generated (engineering stuff).
- From this refactoring work, a first Cheddar DSL meta-model was designed. First experiments with the DSL interpreter.

- **Ongoing works :**

- DSL compiler and simulation process.
- AADL hierarchical scheduler modeling/analysis (SAE/ADDL wg meeting January 2008)