Automatically adapt Cheddar to users need AADL Standards Meeting, Toulouse

A. Plantec⁺, V. Gaudel⁺, S. Rubini⁺, F. Singhoff⁺ P. Dissaux^{*}, J. Legrand^{*}

⁺University of Brest/UBO, LISyC, France *Ellidiss Technologies, France

October 18, 2011









INIVERSITÉ UROPÉENNE IE BRETAGNE





- 2 The Cheddar tool
- 3 The Platypus tool
- 4 Evolution of the Cheddar tool

5 Conclusion

The Cheddar project

Simplify the use of the Real-Time scheduling theory.

- A pragmatic approach
 - Since 2008, partnership with Ellidiss Tech. for open source and industrial support
- A tooled approach: the Cheddar tool
 - Freely available as a standalone tool
 - Can be run with STOOD or with AADLInspector (Ellidiss technology support).

A DESCRIPTION OF A DESC	
	THERE AND AND THE TREE TO AND THE TO AND THE TREE TO AND THE
PROVE A MARKET	
	MARLY I MARLING A SUBJECT AND A DEDUCED AN EMPLOYED.
A STREET AND	
6 IND Keh	maney of factory
	Partner of Corporate Super-
Free properties and a second	
A STATE OF	Parine of Decomponents
11 Bit - BUTTH Londborn finds	Name of two Aspantas
A DECEMPTOR	Party in Party & Gulf
13 Aronal Personnes Banking vi Andrenses (2011 mplane or 001/	
	Terror of Tablet's Assessed have
	Penceges
A RECEIPTER Frank PT	Passar of European
	Native of Pennege RANAURI
Concession international second second	An owner of the second s
	P M • 9 •
11 Manhou Quantum 40 13 mm	Part of the Second Seco
A DEB ROOM DECIDENTS	(FLMAN) / //////////////////////////////////
	has man about the second day of the
	has an element of the second s
A PRODUCT DECEMPTOR THE LAND AND AND A PRODUCT OF A PRODU	TWO has up
	348.88.99 Y
10 10 - DERAD Incidentities	(PLMAN) Inconstitution
[4] DF = TREAD EnvirolMemLy	The second start is a second start of the second start of the second start of the second start of the second st
IN I THREE PLEASAN	NO. FL
C. D. I D. PRINTPOPULATI	(The) Descent Abarah Instand I Descent and A Descharger (1)
10 March 10	343.00 2840
10 1000 ACTING TO	hanniferent/102012 Jone-2860 Ebra 82020
	1975 Reproduction basisful manage. La manuscription



A. Plantec⁺, V. Gaudel⁺, S. Rubini⁺, F. Singhoff⁺P. Dissaux^{*},

Automatically adapt Cheddar to users need

The Cheddar project

A community

- Academic research and educational.
- Industrial utilizations ?

The Cheddar tool is continuously evolving.

Two challenges

- Make Cheddar reusable and adaptable to user's specific requirements.
- Allow Cheddar users to adapt Cheddar themselves.

Validation with a "Design pattern" approach Validation with exhaustive simulations

The Cheddar tool

Engines for real-time architecture evaluation:

- Analysis of AADL models with feasibility tests.
- Exhaustive simulations
 - with classical schedulers: *Rate Monotonic, Earliest Deadline First.*
 - with specific schedulers written into the Cheddar language.

Validation with a "Design pattern" approach Validation with exhaustive simulations

"Design pattern" approach

Define a set of AADL architecture design patterns for real-time systems.

- Models a typical thread communication/synchronization.
- Set of constraints on AADL components/properties.

For each design pattern, all feasibility tests that can be applied according to their applicability assumptions are explicitly declared.

Verification of a real-time system architecture model

- The designer uses a tool that detects which design pattern his architecture is compliant with.
- The designer performs feasibility tests.

Validation with a "Design pattern" approach Validation with exhaustive simulations

Validation with exhaustive simulations

Consists in verifying timing constraints on an AADL architecture. A Scheduling simulation runs during hyper-period.

Two possibilities:

- Use one of the built-in schedulers.
- Provide and use your own scheduler programmed into the Cheddar language.

The Platypus tool

Platypus is a general purpose workbench for the building of target systems with the help of code generators. It allows:

- The specification, the verification and the validation of meta-models.
- The implementation of code generators.

Main idea behind Platypus

Make it possible the very early validation of meta-models.

The Platypus tool

- Platypus is made of an EXPRESS modeling language workbench implemented into Smalltalk.
- A meta model is represented at two levels
 - A set of Smalltalk classes (high level, programmability)
 - A set of EXPRESS entities (Static types, constraints)

Code generators: either into EXPRESS or into Smalltalk.



A. Plantec⁺, V. Gaudel⁺, S. Rubini⁺, F. Singhoff⁺P. Dissaux^{*}, Automatically adapt Cheddar to users need

An iterative design process



Evolution of the architecture meta-models Domain level: adding of a specific scheduler Domain level: adding a new pattern for feasibility tests

Two iteration steps



- Code generation step.
- Ø Meta-models en code generators elaboration steps.



A. Plantec⁺, V. Gaudel⁺, S. Rubini⁺, F. Singhoff⁺P. Dissaux^{*}, Aut

Evolution of the architecture meta-models Domain level: adding of a specific scheduler Domain level: adding a new pattern for feasibility tests

Evolution of the Cheddar tool

Adaptations must be possible at all levels:

• Architecture meta-models

• Very specific needs that imply very fine adaptations of the Cheddar engines.

Cheddar engines

- Specific schedulers.
- New patterns for the selection of feasibility tests.

Evolution of the architecture meta-models Domain level: adding of a specific scheduler Domain level: adding a new pattern for feasibility tests

Evolution of the architecture meta-models

These meta-models constitute the core of Cheddar:

- An evolution requires manual Ada programming to integrate the generated code.
- Example: adding of multi-core/cache management.



- Validation with realistic AADL architectures.
- Example: enforce domain constraints.

A. Plantec⁺, V. Gaudel⁺, S. Rubini⁺, F. Singhoff⁺P. Dissaux^{*}, Automatically adapt Cheddar to users need

Evolution of the architecture meta-models Domain level: adding of a specific scheduler Domain level: adding a new pattern for feasibility tests

Adding of a specific scheduler

The new scheduler is first implemented into the Cheddar language.

- It is first tested with the Cheddar interpreter then, the corresponding Ada code can be generated and integrated into Cheddar.
- The result is a specific version of Cheddar.



Evolution of the architecture meta-models Domain level: adding of a specific scheduler Domain level: adding a new pattern for feasibility tests

Adding a new pattern for feasibility tests

The new pattern is first specified and tested into Platypus.

- Applicability assumptions are designed as additional rules that constraint further the generic layer.
- Rules are translated to Ada, the result is a new version of Cheddar.



Current status Possible developments

Current status

- 30% of the Cheddar Ada code is automatically generated (16500 lines).
- This code is generated from the Architecture and the Cheddar language meta-models (1650 lines).
- We have an example (a running prototype) for the *Synchronous Data Flow* and *Ravenscar* patterns. So, we are ready to build a first version of the generator for the design pattern tool.



Current status Possible developments

Possible developments

- The elaboration capabilities of Platypus are not used enough
 - Some structural issues remain
 - No constraint in the Architecture and Cheddar language meta-models.
- Other part of Cheddar could be generated (User interface, architecture and user input checking...)