Non-functional aspects management for craft-oriented design

François Mekerke Wolfgang Theurer Joël Champeau

ENSIETA, Brest, France

October 12, 2004

Contents

Outline

Constraints

The problems we encounter are always the same.

- Big systems are always bigger;
- Complex systems are always more complex;
- Critical systems are always more critical;
- And of course, whatever the increase in size or complexity, the budget are not to evolve.

Industrial reality

Industrial projects are often as follows:

- A customer describes its needs to:
- A project manager which distributes the different tasks to:
- Several sub-contractors, which actually realize the different parts of the system.

The responsabilities of the contracting authority are:

- Coordinating the efforts of the sub-contractors so as to achieve:
- Providing a system that complies to the requirements of the customer.

The project manager has to deal with the cross-cutting concerns

Techniques used by the sub-contractors

Modelling, with methodological tools:

- B $\Longrightarrow$ Extensive code generation;
- Shlaer&Mellor $\Longrightarrow$ Object-Oriented Systems Analysis;
- UML $\Longrightarrow$ Graphical Representation;

Validation, through formal methods:

- Model checking;
- Tests;
- Formal languages (Lustre, Esterel...)

None of them are available for the project manager

Just trying to improve our knowledge of the system

Limits of the validation techniques on industrial embedded systems:

- State explosion problems;
- No graphical representation;
- Difficulty of mastering the techniques.

Limits of OO modelling techniques on industrial embedded systems:

- Overloaded diagrams;
- Not fully adapted to industrial habits;
- No formal validation.

Distributed models and refinement

Two cases:

One big model:

- No problem of refinement, relationships are known, but:
- Size actually matters;
- All stakeholders work on the same model.

Distributed model of the system:

- Each specialist describes its own vision of the system in a model, but:
- Relationships among objects through refinement ?

A critical choice

Thus, we have the choice between:

- A huge model of the system $\Longrightarrow$ increasingly difficult to discriminate between objects coming from the different stakeholders;
- A number of models, each dedicated to a domain, but whose relationships have to be redefined at each level of refinement.

Outline

Goal

We would like to allow the following behavior:

- Each specialist working on its own model;
- The project manager having a clear vision of the system;
- Transparent refinement for the project manager.

The structure we would like to implement:

- A distributed model, with one sub-model per domain;
- An abstract model of the system, for the project manager;
- A way to check the models' consistency on the fly.

Facets

What is a facet ?

- A part of the system;
- Related to a sub-contractor;
- The realization of a number of concerns;
- A component.

What a facet is not:

- An aspect;
- A view(-point).

Pivot

What is the pivot ?

- An abstract model of the system for the project manager;
- An interface between sub-contractors' components;
- A reflection of the system's requirements.

What the pivot is not:

- A woven model;
- Refinable.

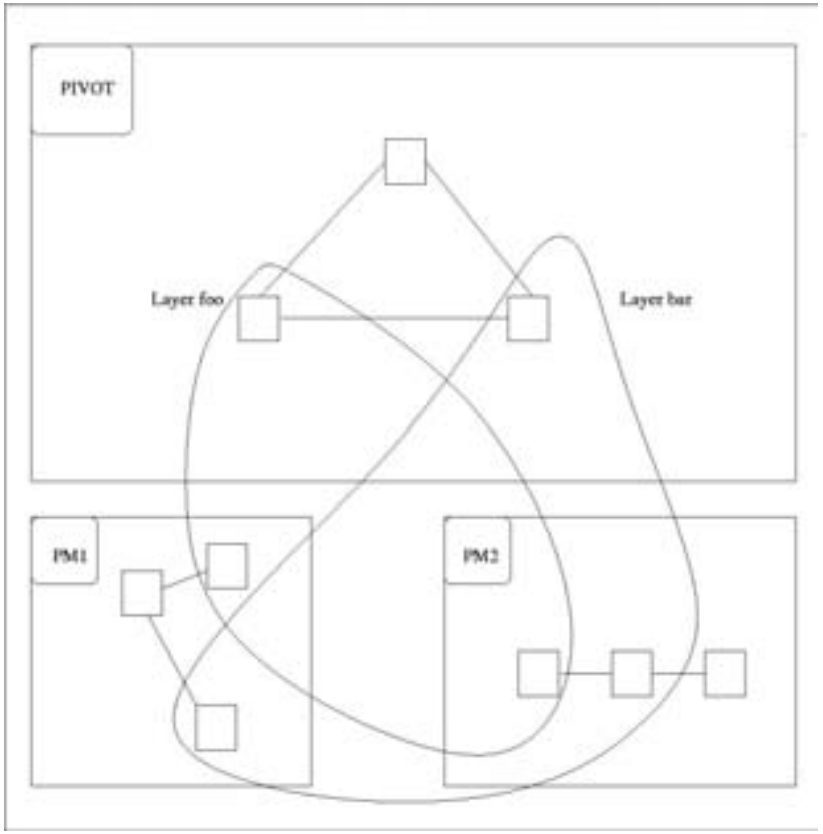
Layers

What is a layer ?

- Related to a cross-cutting concern;
- Provides the pivot with facets' information;
- Evaluates invariants in the pivot.

What a layer is not:

- Only a view of the system;
- Exactly an aspect;
- A part of the system.



Outline

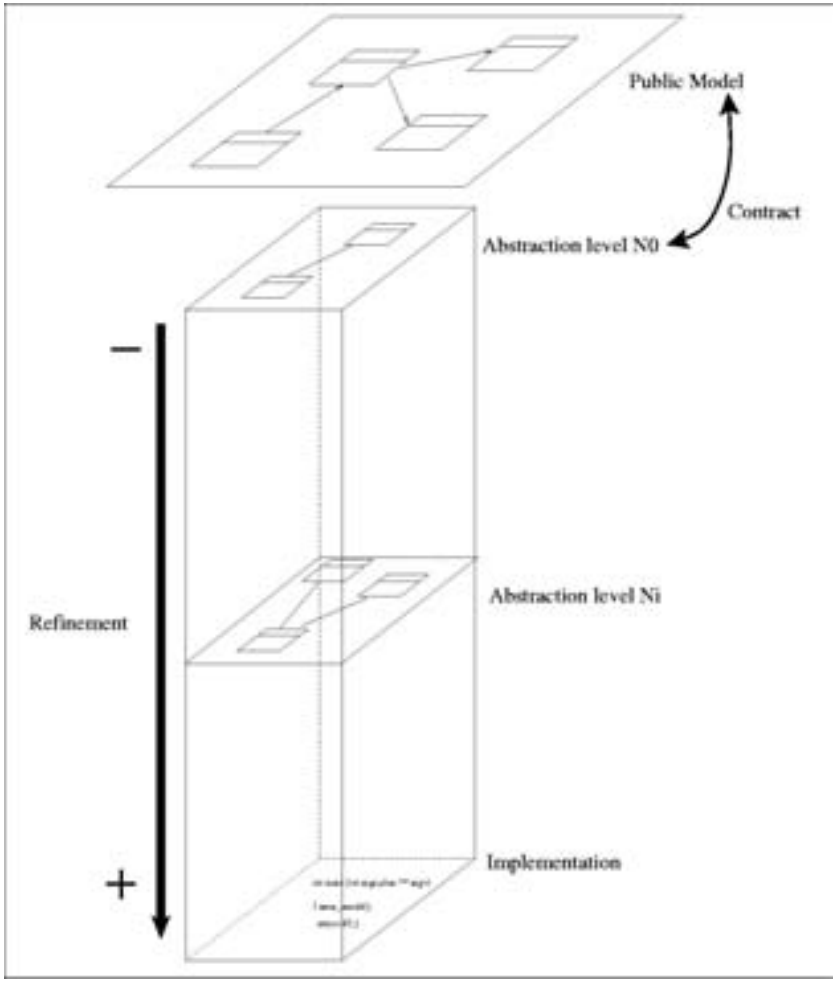
Facets

A facet is structured as follows:

- A public part we call Public Model;
- A private part in which the realization is done.

The Public Model is meant to:

- Act as a contract between the facet and the project manager;
- Provide an abstraction of the objects of the facet to the pivot (FacetOfferedElement);
- Inform the pivot of the facet's requirements concerning the objets from other facets (FacetRequiredElement).



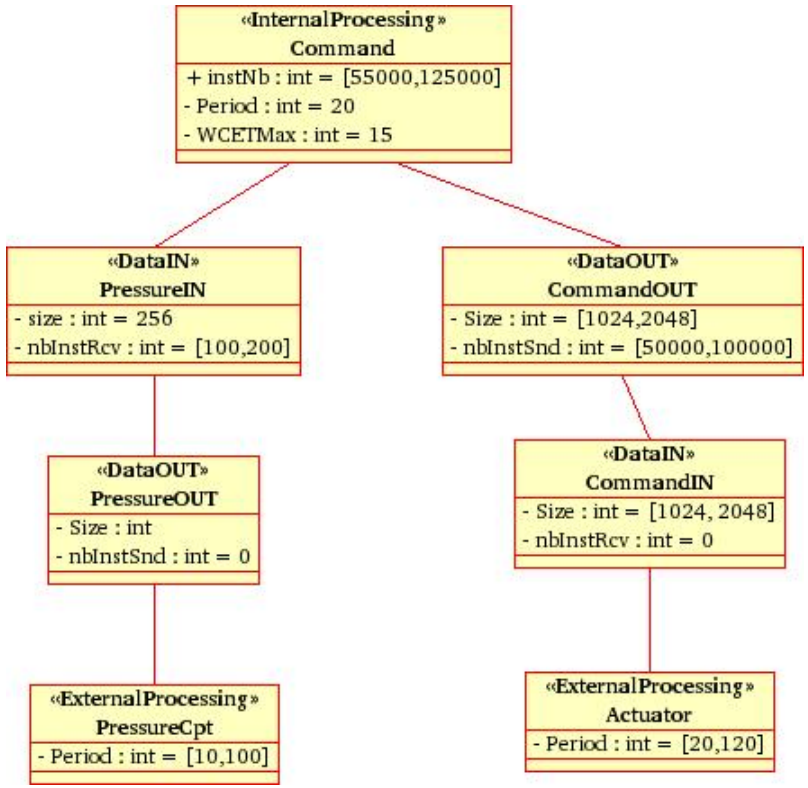
Facet: Example

The Public Model of a facet details its undertaking towards the pivot:

- In terms of structure in the first place;
- In terms of values afterwards.

The example shows the software facet of a hydrographic float:

- Command is the software that compute and emit the command to the actuator from pressure information;
- PressureOUT is the pressure info sent by the sensor;
- PressureIN is the pressure information received by command;
- ...



Pivot

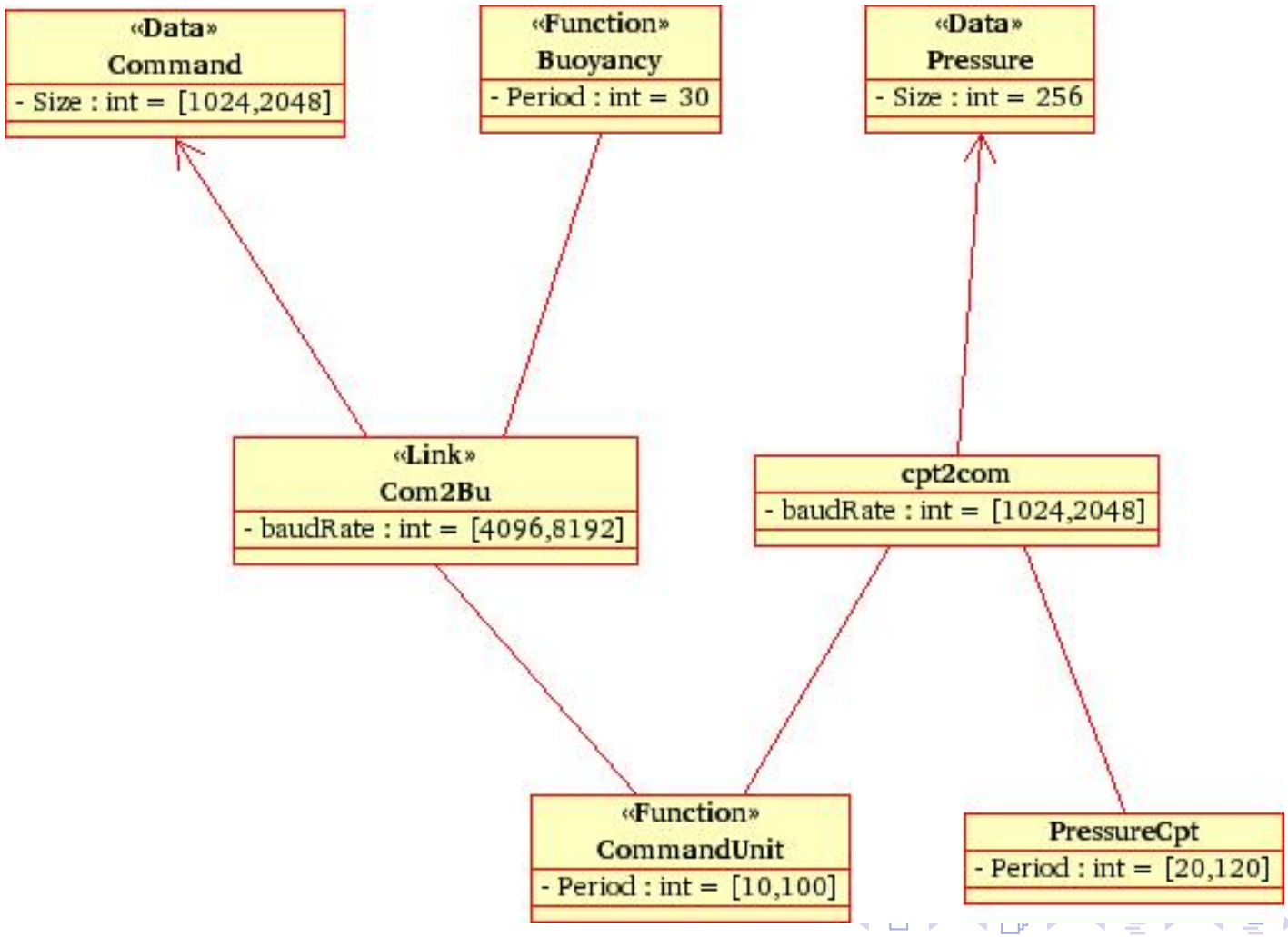
The pivot is a structure meant to:

- Gather all horizontal information;
- Be a storage model;
- Be an analytic description of the system;

Contains all the elements that are concerned by a requirement.

It provides the projet manager with an abstract model of the system that fits its concerns.

Pivot: Example



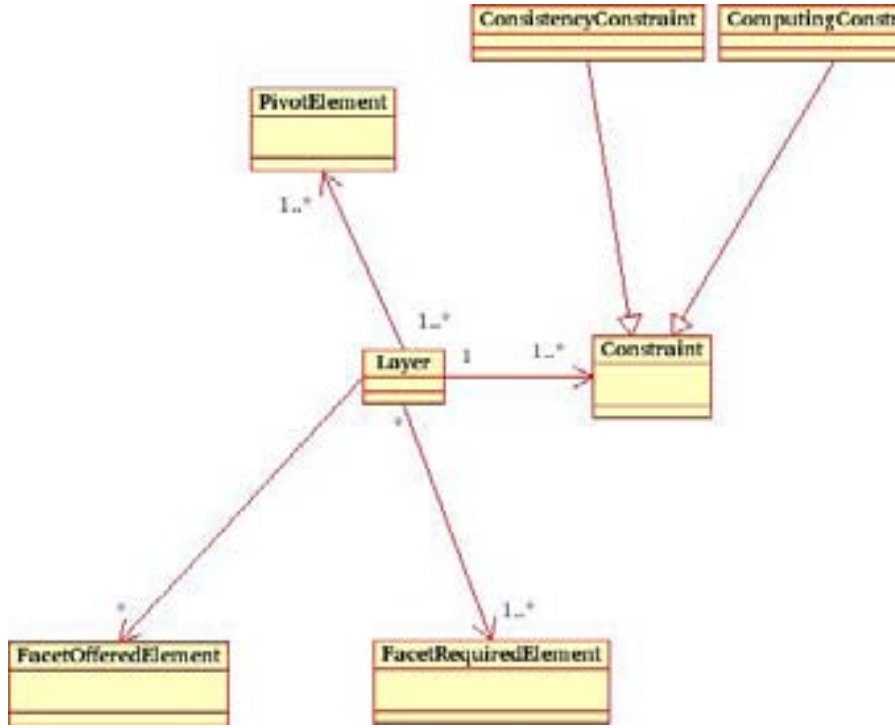
Layer

A layer impacts on a number of pivot elements:

- It extracts and processes information

  From the facets to the pivot.

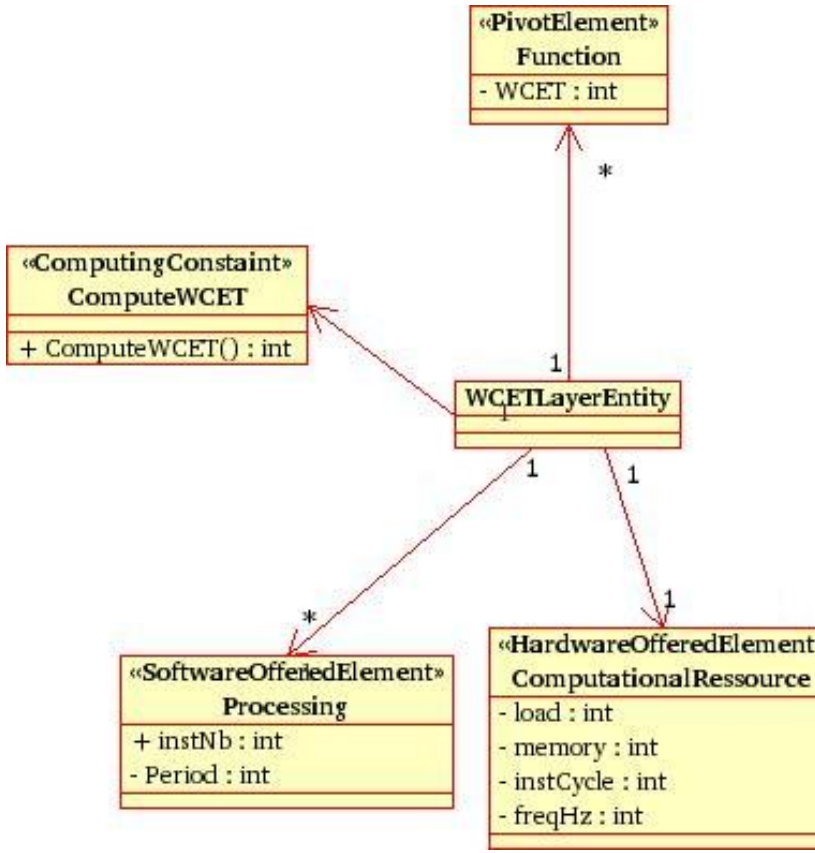
- And then evaluates a number of constraints.



Layer: Example

In the example, we have:

- A layer that computes the WCET of all the functions associated to a particular computational ressource through:
- A computing constraint ComputeWCET that computes WCET for all the functions using the ComputationalRessource.



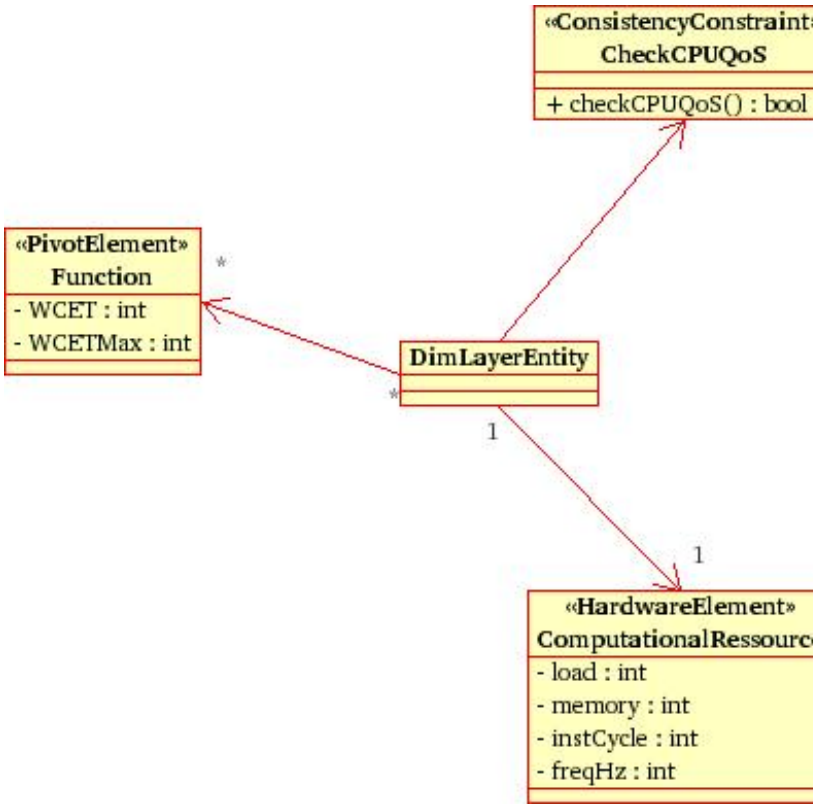
Layer: Example

In the example, we have:

- A layer that checks that each function associated to a computational ressource respects its real time specs
- CheckCPUQoS():

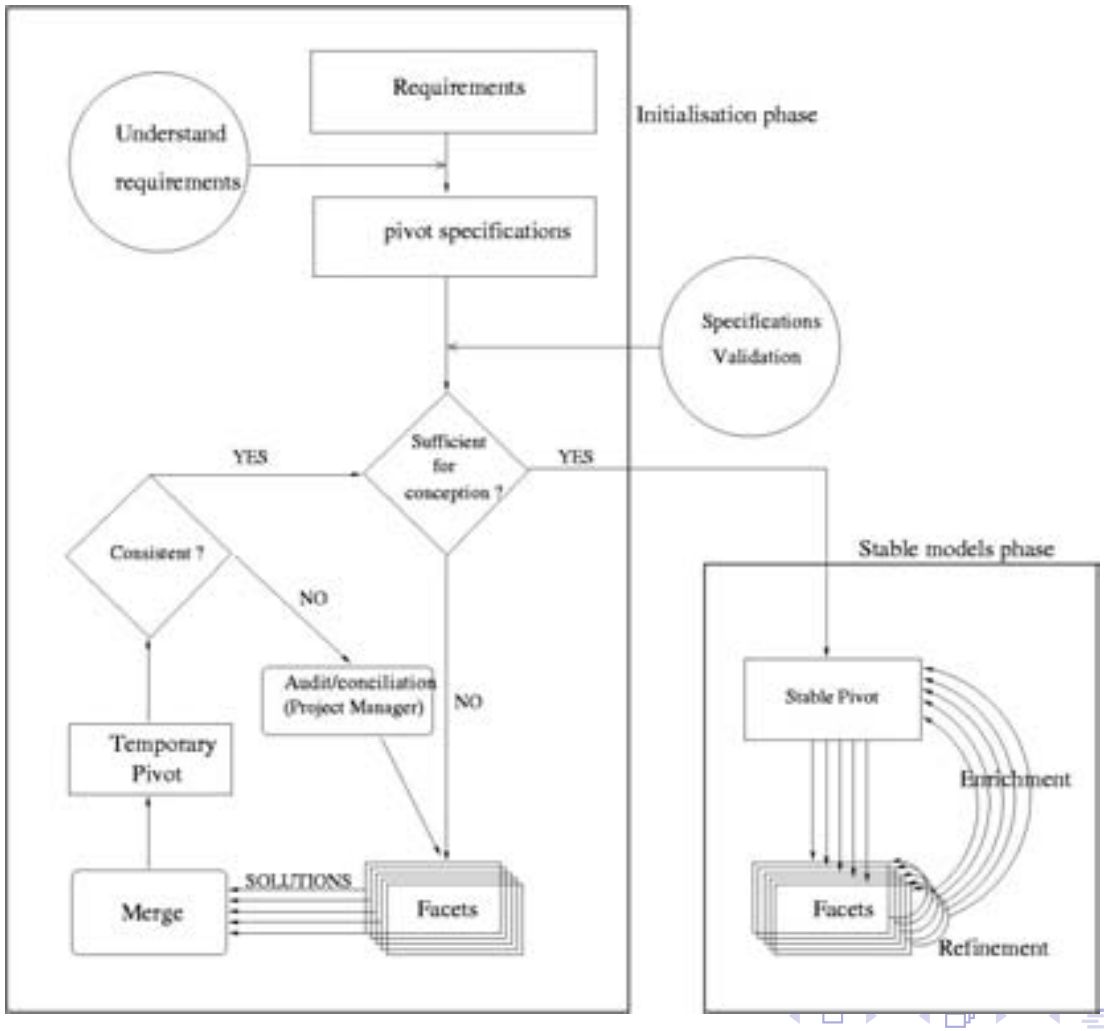
$$\forall f \in DimLayerEntity.Function,$$

$$f.WCET < f.WCETMax$$



Outline

Associated development process



Outline

Conclusion

We presented a methodology that addresses:

- Refinement problems among distributed models;
- Consistency checking among distributed models.

To do so, we have chosen the following structure:

- Facets, which reflect existing industrial habits;
- A pivot, meant to help the project manager;
- Layers, which contains their relationships.

Last but not least...