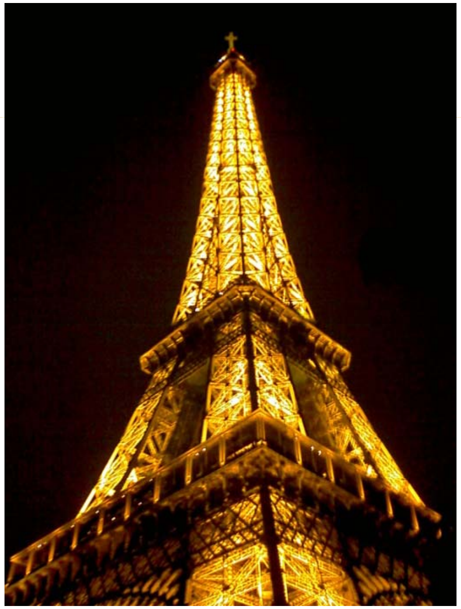
Benefits of the MDE approach for the development of embedded and robotic systems

X. Blanc(*), J. Delatour(**), T. Ziadi(*)

(*)Université Pierre et Marie Curie (UPMC)

Laboratoire d'Informatique de Paris 6 (LIP6)

(**) ESEO



CAR 2007

Plan

- MDE

  raising the level of abstraction of the development life cycle

- AiboDev

  Using MDA concepts for Aibo

- Conclusion

Model Driven Engineering

raising the level of abstraction of the development life cycle

MDE

Models

- *« Modeling is the future, so every company that's working on this I think it's great, and I think there are some real contributions that can be made »* B. Gates
- *« Companies that adopt the MDA gain the ultimate in flexibility: the ability to derive code from a stable model as the underlying infrastructure shifts over time. »* R. Soley

- « Why building models ? At the end, we will write code ? »
- « A good diagram is better than a long speech… but from a UML diagram you can have many speeches ! »

Need of <u>best practices</u> and clear <u>objectives</u>

Practices & Objectives

Best Practices

- Abstraction layers and Viewpoints (Matrix)
- Formalization of layers and viewpoints
- Formalization of abstraction layers / viewpoints relationships

Objectives

- New systems
- Legacy systems
- Master platforms evolutions

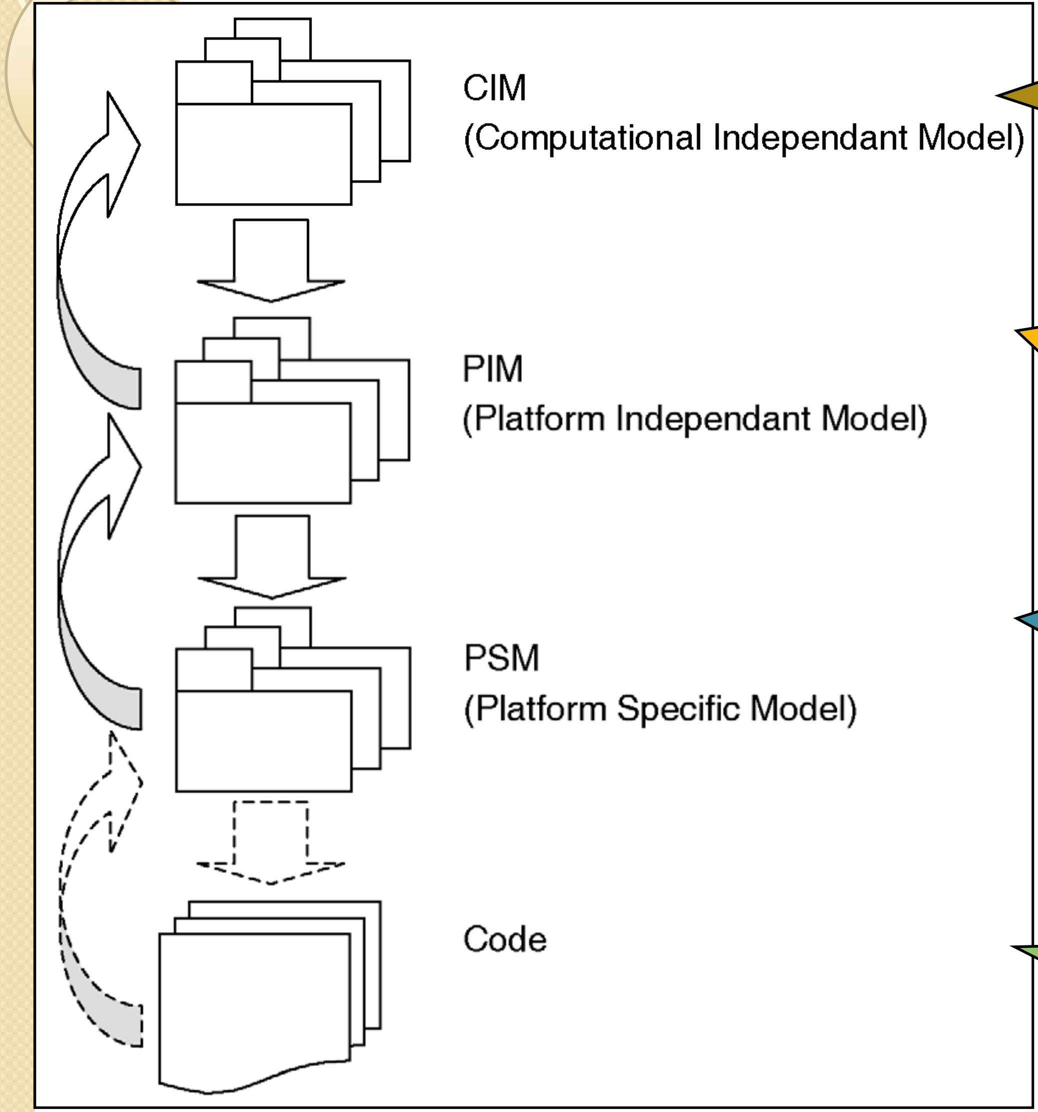
architecte logiciel

MDA en action

Ingénierie logicielle guidée par les modèles

EYROLLES

Approach


CIM
(Computational Independant Model)
PIM
(Platform Independant Model)
PSM
(Platform Specific Model)
Code

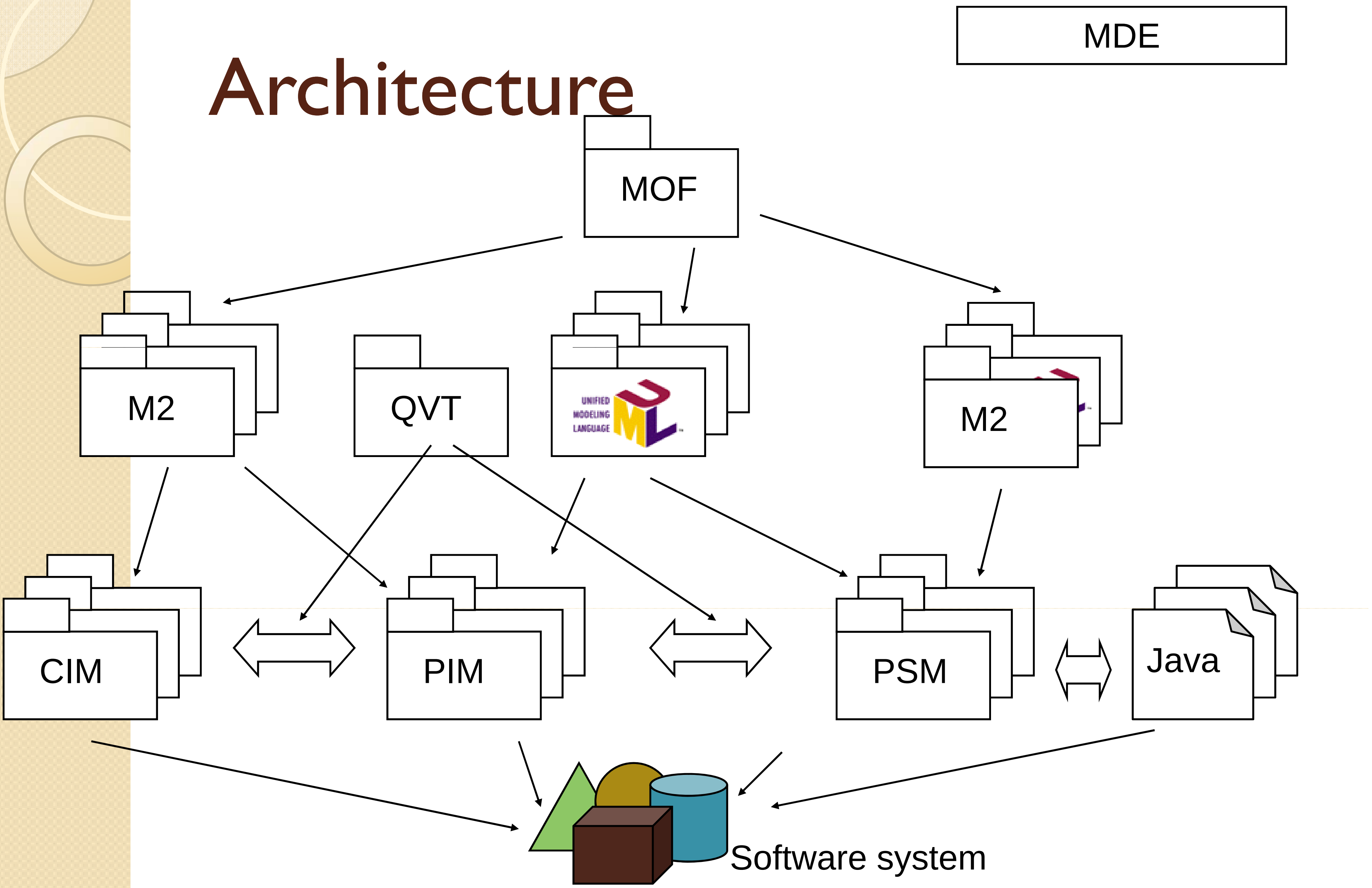
Requirement model: defines the system in its environment.

Analysis and design model: defines the system architecture.

Realization model: defines how the system is built.

Code of the system and configuration artifacts.

Architecture



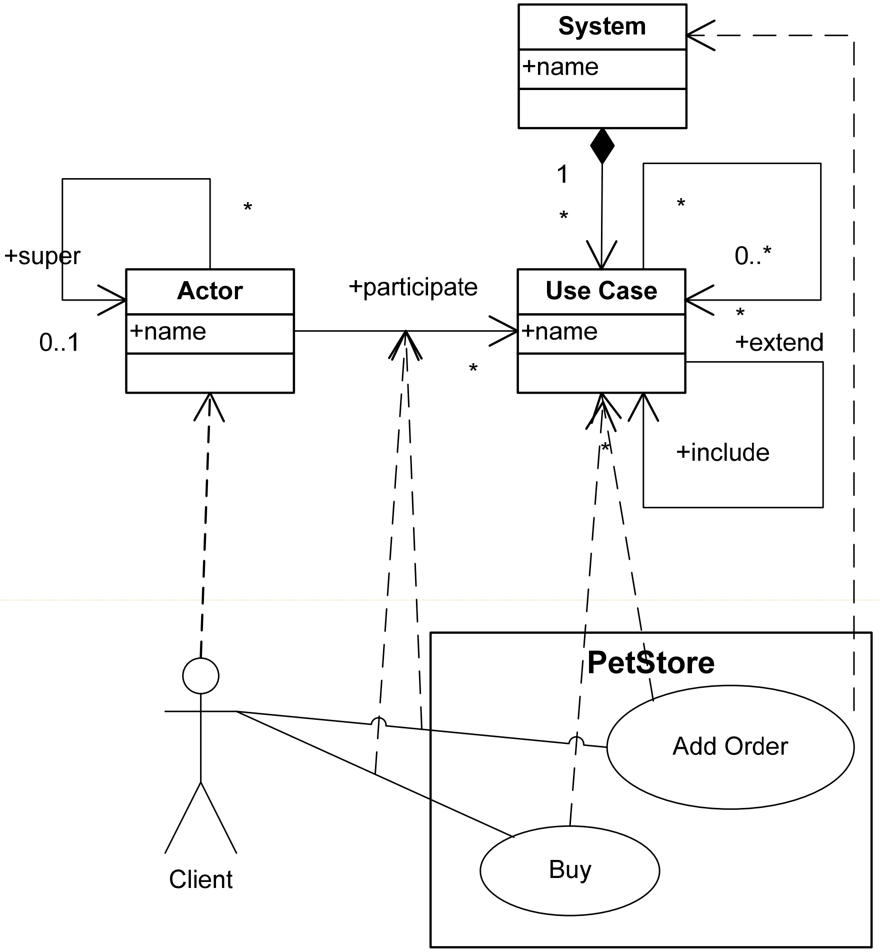
Meta-model

A meta-model defines concepts and their relationships thanks to a class diagram.

A meta-model only defines structure (no semantic).

A model is an instance of a meta-model if it respects the structure defined by the meta-model.

The UML meta-model defines the structure that all UML models must have.



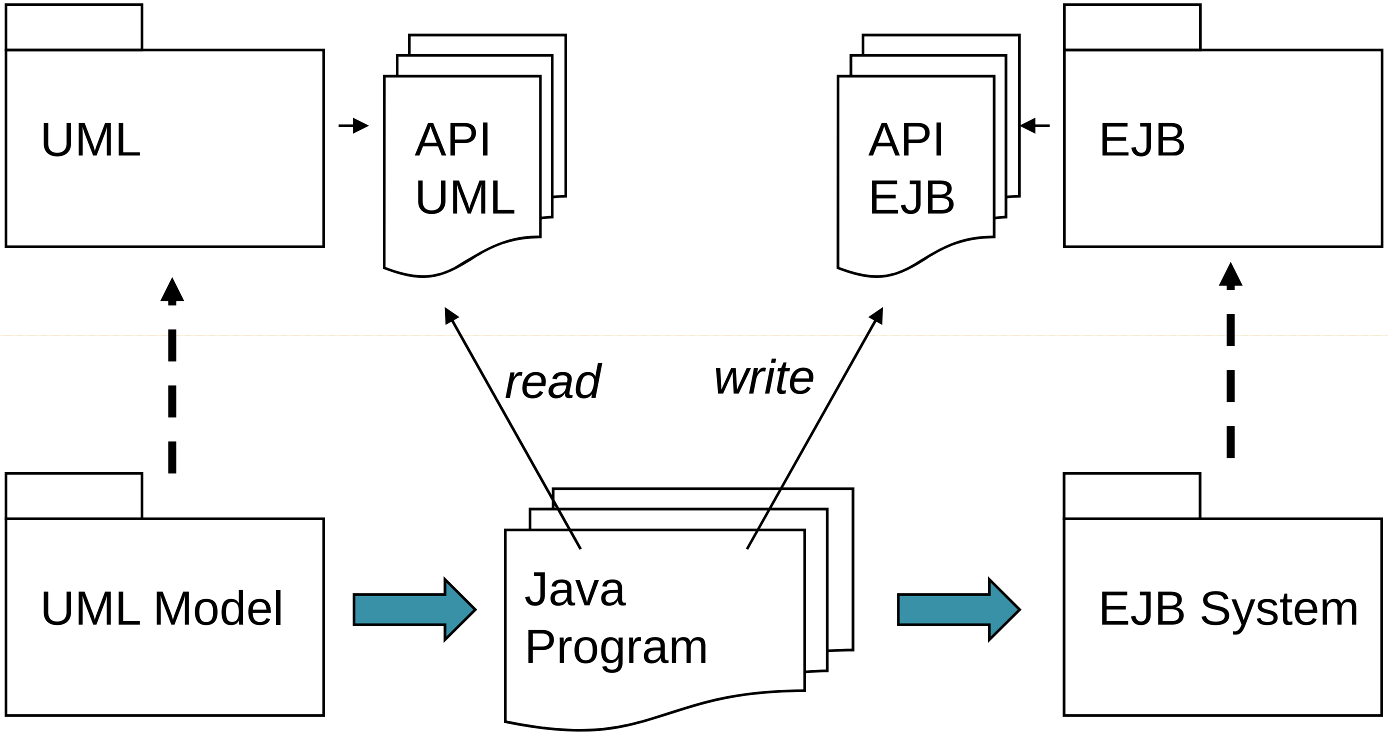
Model Transformations

- Model transformations are a central part of MDE
  - CIM to PIM, PIM to PSM, PSM to code .
- Model transformation are based on meta-models
  - Any UML component gives an EJB component.
- Platform providers should provide model transformation for their platform
  - UML to EJB
- Companies should be able to customize those transformations
  - Ex: Do not use entity bean!

- Today, there are three approaches for writing model transformations
  - Program, Template, Model

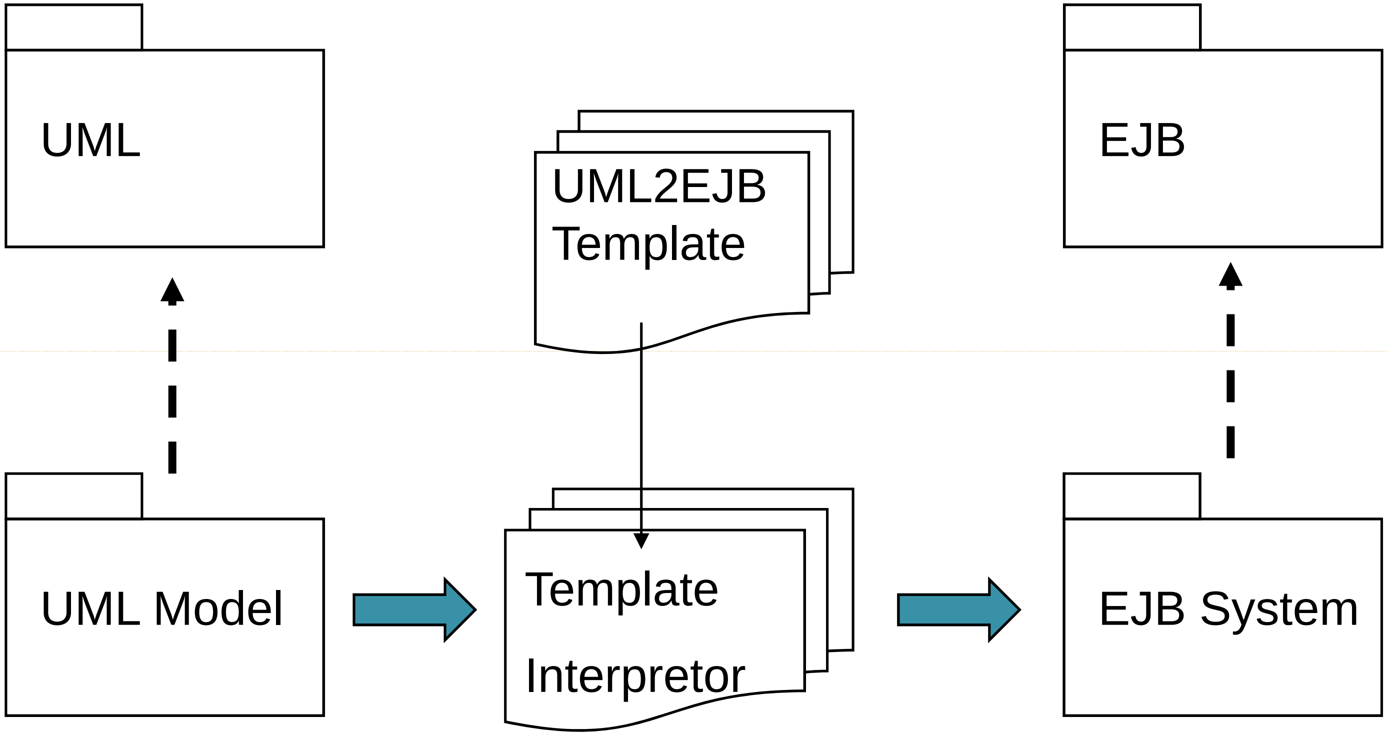
Program

- Model transformation is a program that makes use of model manipulation APIs



Template

- Model transformation is a template written in a dedicated language



AiboDev

Using MDA concepts for Aibo

Aibo

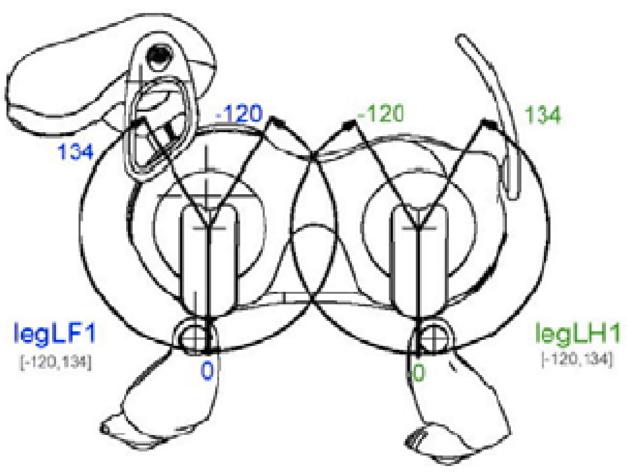
AiboDev 2006

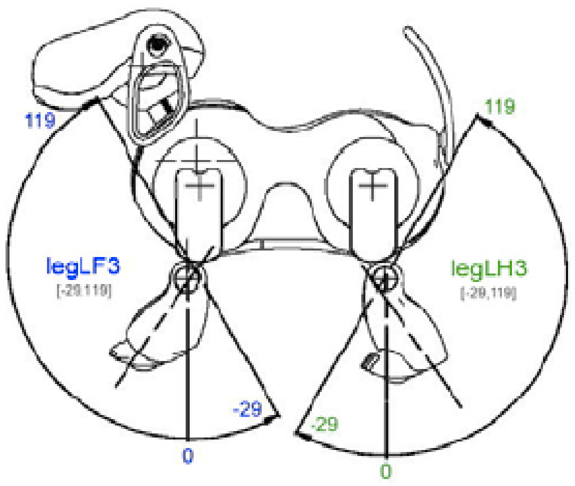
- Model Environment for Aibo
- Behavioral models
  - Dance, Guard, Escape
- 100% Code generation
- Validation

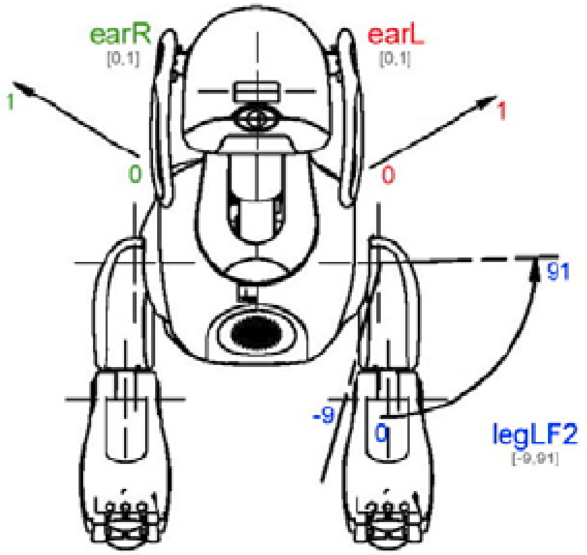
- Master Students
- 7 weeks
- 3 industrial partners

Aibo

AIBO Motors





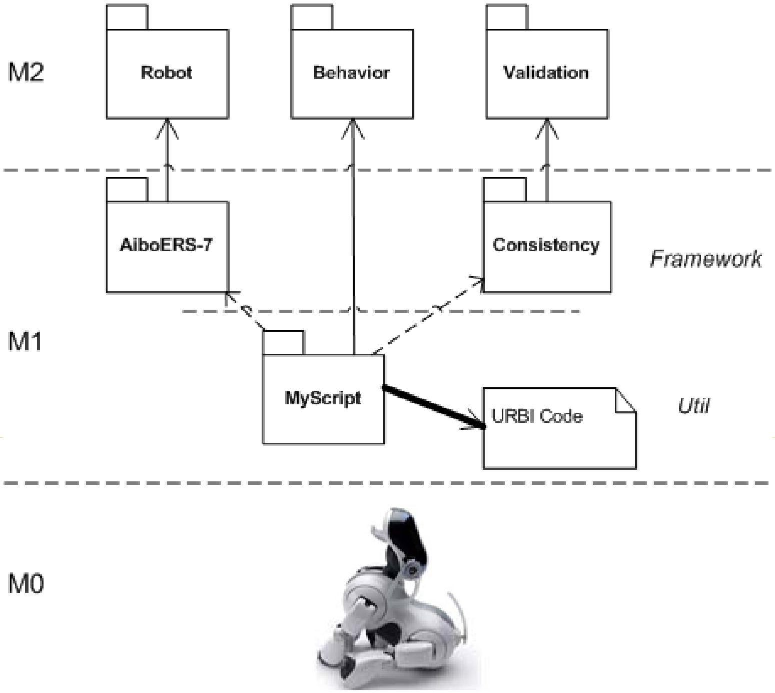


Aibo

Approach

- Robot metamodel
- Behavioral metamodel
- Validation metamodel

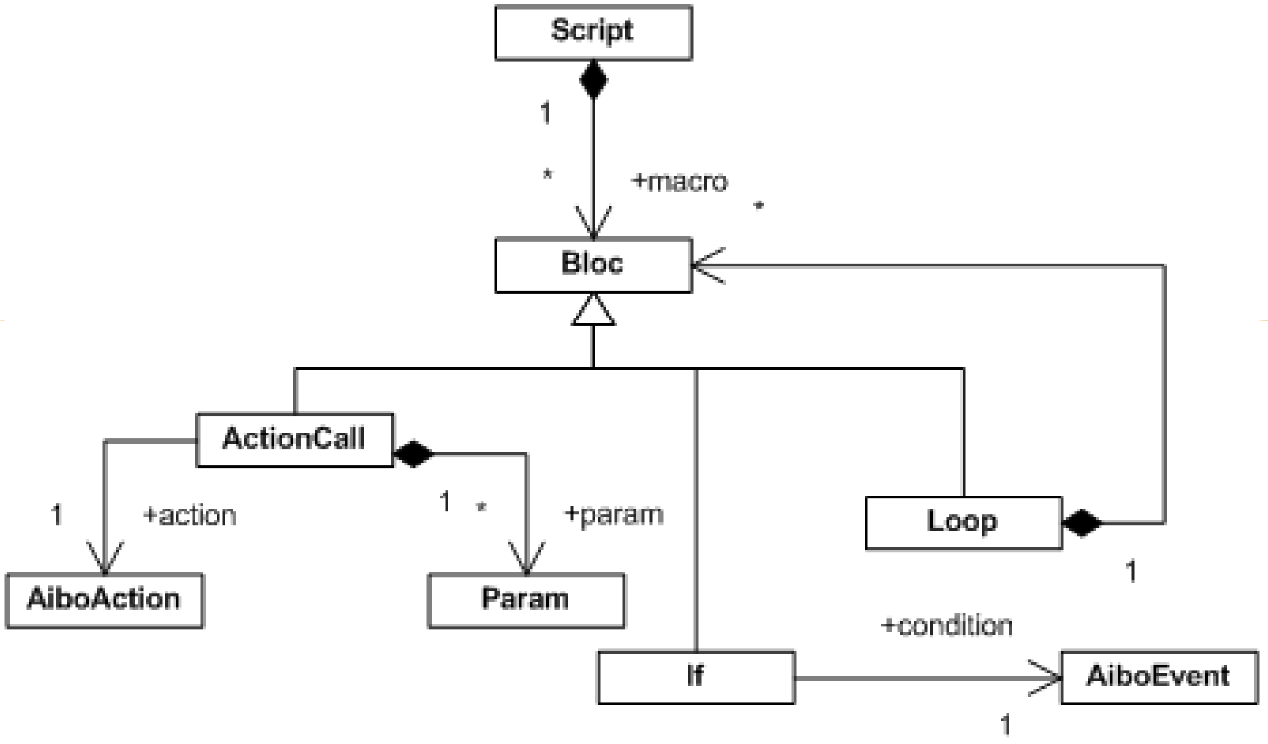
- Validation = Inclusion
- Code generation = template



Aibo

Behavior Model

- UML2 Activity oriented (vs StateMachine)
- Events are conditions



Aibo

Automated Operations

Code Generation

- Template Approach
- URBI as a programming language

Validation

- Inclusion of models
- Java
- Not Formal

Conclusion

Advantages

- Usable (industrial)
  - Metamodel, Validation, Code Generation
- Multi-View

Limits

- « Good » Meta-model ?

http://www-src.lip6.fr/homepages/Xavier.Blanc/courses/CAR/AiboDev2006/