Towards a Model-Driven Engineering Software Development Framework

Julien Delange
Maxime Perrotin
Samir Bennani

Consequences of Software Development issues

- **Increasing cost**
  - Development + certification
- **Late release and impact on other projects**
  - Re-engineering efforts due to testing efforts
- **Remaining bugs leading to potential failures**

Software Development Traps & Pitfalls – tech concerns

- **Heterogeneous notations**
  - Models using different representation
  - Collocation of languages with potential incompatibilities
- **Lack of formalization**
  - Assumptions made by development teams
- **Heterogeneous execution environment**
  - Impact on system execution and software behavior

Software Development Traps & Pitfalls – org. concerns

- **Lack of coordination among teams**
  - Different understanding of system requirements
  - Assumptions based on team experience
- **Human-factor**
  - Introduction of bugs
  - Heterogeneous background

Software Development Traps & Pitfalls – org. concerns

- **Lack of coordination among teams**
  - Different understanding of system requirements
  - Assumptions based on team experience
- **Human-factor**
  - Introduction of bugs
  - Heterogeneous background

Proposed Approach

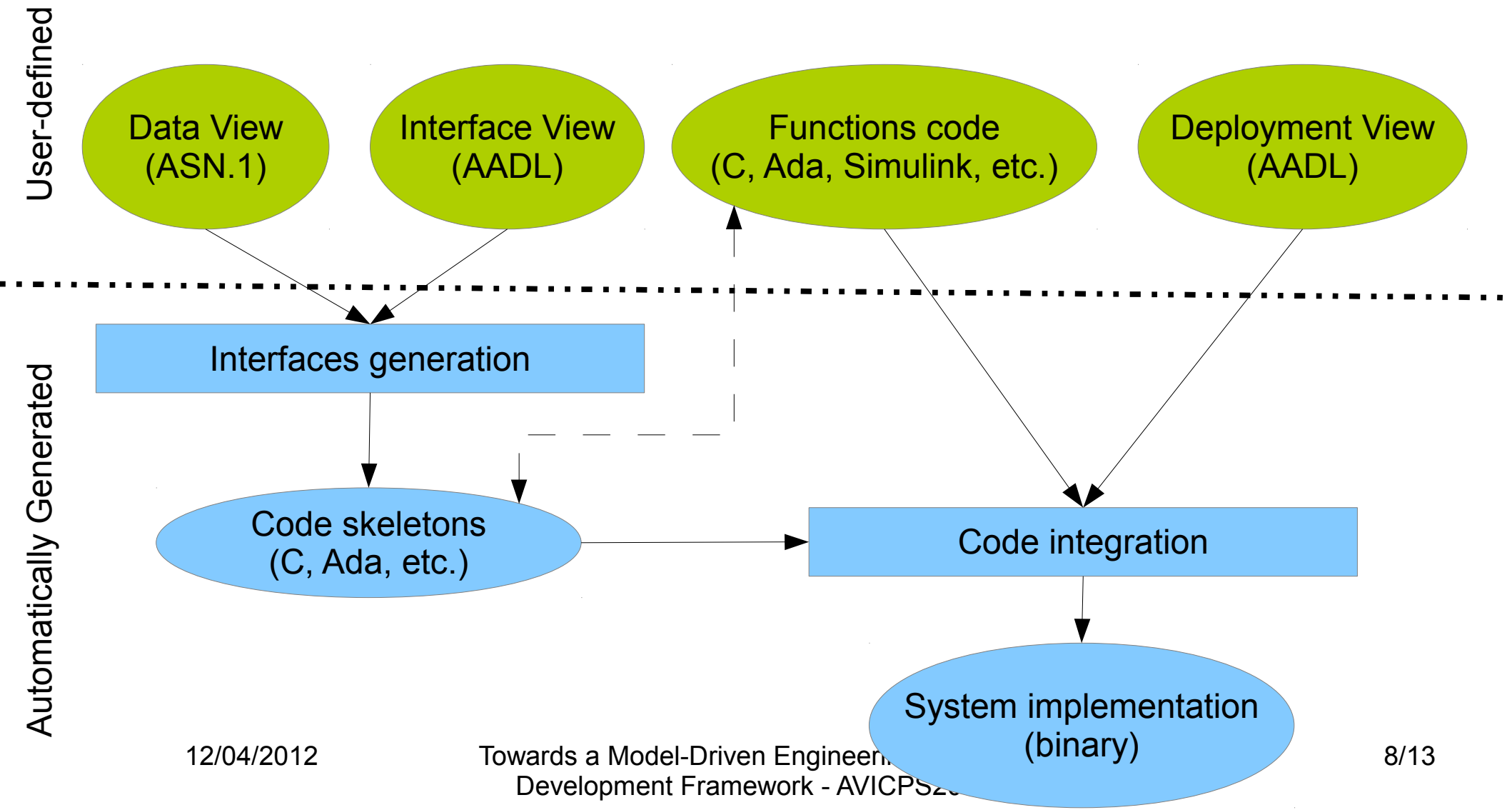
- **Formalize system specifications using models**
  - System functions, interfaces & execution environment
  - Avoid unspecified aspects
- **Automate development aspects**
  - Focus on specialized engineering domain
  - Avoid integration efforts

The ASSERT Set of Tools for Engineering

- System description using three views
  - **Data View**: interfaces, data types
  - **Interface View**: system functions and connections
  - **Deployment View**: functions allocation on execution system

- Automatic production of implementation
  - Transform models into executable code
  - Integration with established tools and standards

TASTE Development Process



TASTE ecosystem

Languages

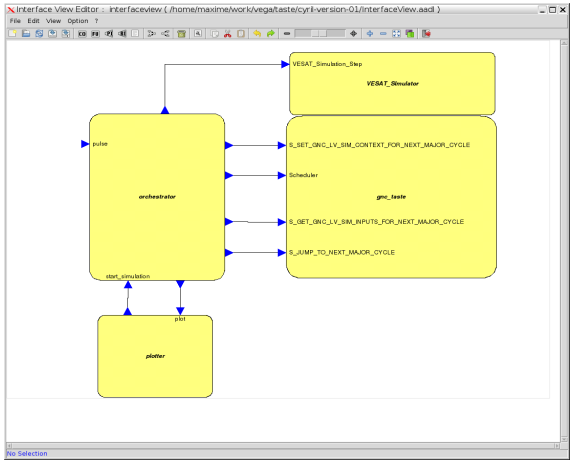
- ASN.1
  - Interfaces & data types
- AADL
  - Functional specification
  - Deployment definition
- Functional modeling languages

Tools

- ASN1Scc (Semantix)
- Ocarina (ISAE)
- TASTE GUI (Ellidiss)
- RTDS (Pragmadev)

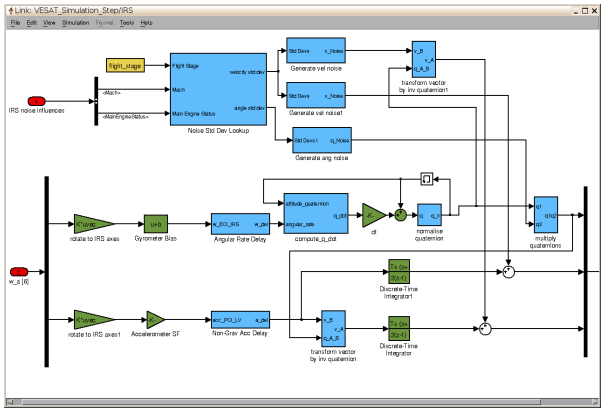
The VEGA case-study

Interface View



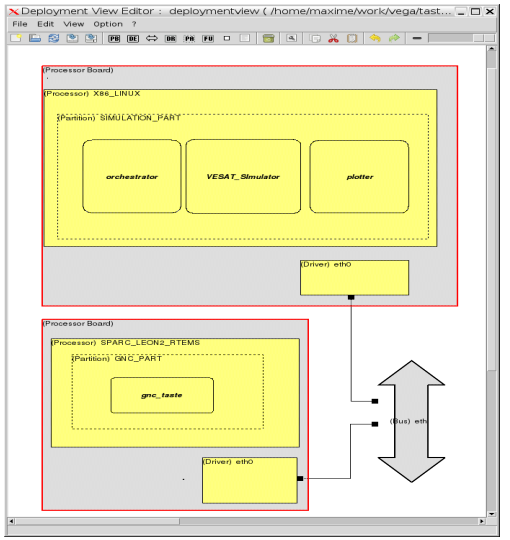
Heterogeneous languages

Functional code
(Guidance and Navigation Control)



Simulink models

Deployment View



Deployment on systems with different architectures

taste

Implementation

Existing projects & feedback

- **Initial use in the European Space Agency**
  - Supporting projects development
  - Abstraction ease system development

- **Extension to other domain**
  - Use in both academic and industrial contexts
  - Demonstrate toolset maturity

Conclusion

- **Pragmatic approach addressing current traps & pitfalls**
  - Strict & formalized system description
  - Automate system production, avoid manual efforts
- **Support for more validation & other domains**
  - Support for validation/certification efforts
  - Extension for other safety-critical systems (e.g. avionics)

Questions ?