

## ***On-Board Software***

*Jean-Loup Terraillon*

*Software Engineering and Standardisation Section*

# **Technology Domain & Links (1/2)**

## Dossier 0: Space System Software (TD02)

Technology Sub-Domain	Technology Group
A.Advanced Software Technologies	I.XML Technologies
	II.Java and Other Interpreter Technologies
	III.Internet Technologies
	IV.Space Distributed Systems
B.Space Segment Software Systems	I.Methods and Tools for the On-Board Software Engineering Process
	II.Innovative Software Management
	III.Software-Hardware Co-Engineering
	IV.Plug and Play Technologies

## ***Technology Domain & Links (2/2)***

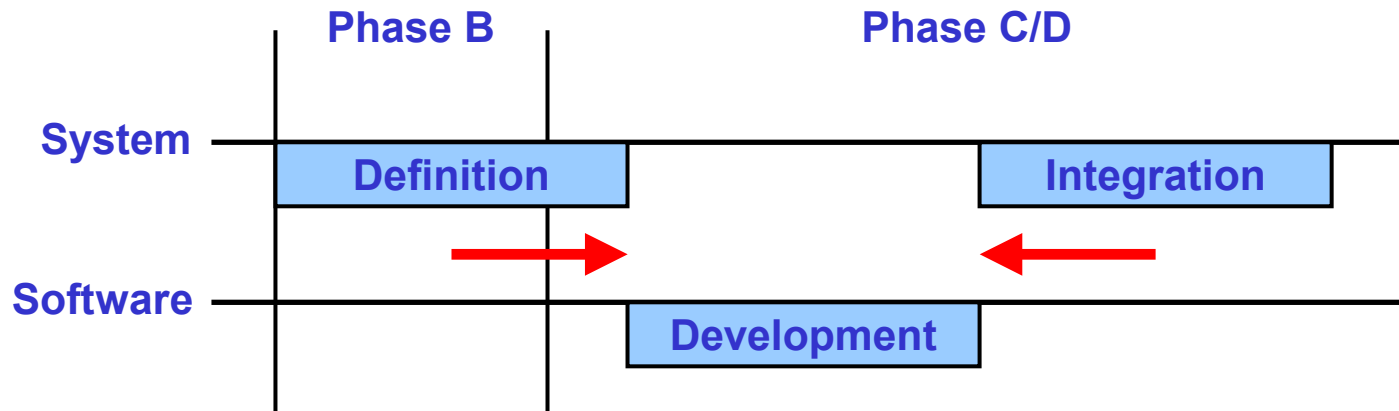
**This dossier has links with the following domains:**

- **On-Board Computer and Data System**
- **On-Board Payload Data Processing System**
- **Microprocessor**
- **System Design and Verification (Specification Methods)**

## ***Missions & Market Needs***

The needs come from:

- Application domains (On-Board Computer and Data System, On-Board Payload Data Processing System, Microprocessor, Robotics)
- Schedule constraints between software and system



## ***Analysis of Needs (1/4)***

- **System/software co-engineering**

Common techniques to describe and verify the properties of the system (i.e. avionics/payload) and the software

- *Avionics Architectural Description language (AADL)*
- *Object-oriented modelling (e.g. HRT-UML)*
- *Behavioural modelling (state machine, data flow diagrams)*
- *Data modelling (XML, ASN.1)*
- *Hardware/software co-design*

## ***Analysis of Needs (2/4)***

- **Better requirements**

Requirement engineering to consolidate and verify the requirements

- *Formal methods*
- *Requirement verification*
- *Requirement management (value analysis, technical data management, traceability, etc)*
- *Fast prototyping*

## ***Analysis of Needs (3/4)***

- **Standardisation and building blocks**
  - *Frameworks, component engineering*
  - *Software repository*
  - *Maintenance industrial organisation*
- **Automation of the life cycle**
  - *Automatic code generation*
  - *Generative programming, aspect oriented programming*
  - *Automatic testing*

## ***Analysis of Needs (4/4)***

- **More efficient Verification and Validation**
  - *Model checking*
  - *Schedulability analysis*
  - *Validation environment to reduce the work on an EQM (virtual satellite, Software Validation Facility, TOPNET, iron bird, etc). From full simulated, introduce computer in the loop and real components when they are ready*
  - *Independent software verification and validation framework*
- **New functions/architectures**
  - *Distributed systems*
  - *Mission/software change*
  - *Security*
  - *Mission management and planning, on-board intelligence, for exploration, robots, etc*
- **Cross development environment**
  - *Porting to LEON*
  - *Maintenance*

## ***Market Perspectives***

- Software developers : good
  - *Increasing number of computers, increasing software size*
- Technology providers : difficult
  - *Space community is a small market*
  - *Reuse commercial technology, adapt, and organise maintenance*
  - *Extend the market to US (e.g. AADL)*
  - *Commonalities with aeronautics (Integrated Project in E.U. FP6)*

**→ Sustain technology providers**

## European Status of Current Technology

- On-Board software development
  - *Primes generally have their SW department*
  - *Several SMEs specialised in O/B SW*

*Note: a SW house needs space system knowledge to make O/B SW*
- Lot of valuable research centres
- European tool vendor sector exists but is fragile:
  - *Specialised on a niche technology (compilers vendor, modelling & verification tools)*
  - *Some enlarge their market but go away from space needs*

*Risk: depend on tools whose license might be regulated*

*Need: ensure that the tool is available and maintained for the long term*

☞ Reinforce European tool vendors

## ***Competitiveness and Benchmarking***

- Competitiveness: assets
  - *a lot of valuable European research are commercialised (PolySpace Verifier, Esterel Studio, Test Composer)*
  - *the European engineering style: top down, formalisation*
    - ➔ ensure that we have tools that support the top down approach,
      - ➔ encourage early verification at each step
- Benchmarking
  - *Spice (Spice for Space), CMM*
  - *European companies are CMM level 2 in average (some targeting 3)*

## ***European Strategic Interest***

- Improve competitiveness in a small market
  - *Adaptation of commercial technology*
  - *Generic architecture and standardised building blocks*
  - *A reference set of open source maintained building blocks compliant with the standardized interface (validate standards, capitalization of experience)*
  - *A number of proprietary implementations according to industrial policy*
- Reuse requirements and adapt the system definition to the existing building blocs by system software co-engineering
- European independence

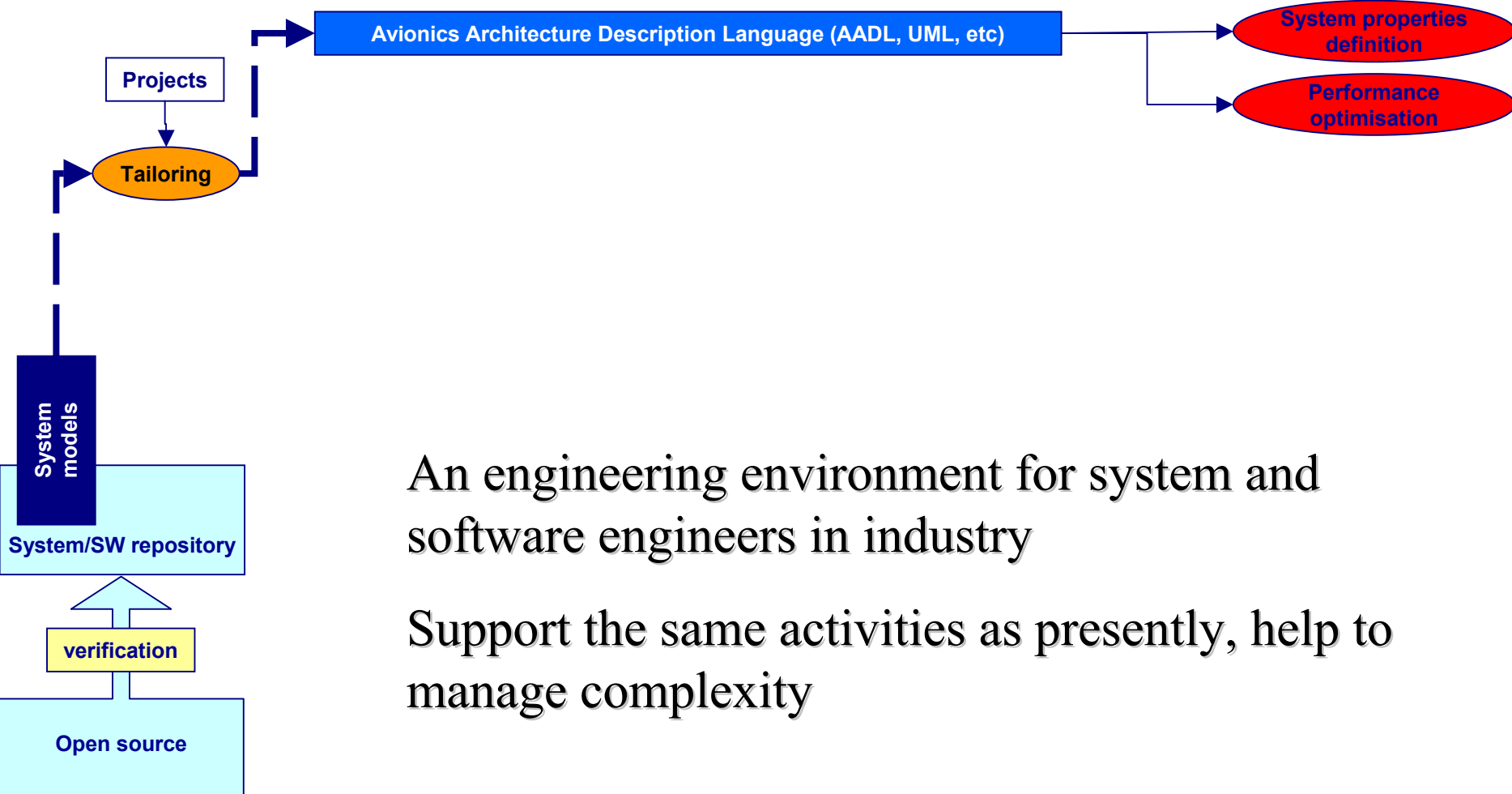
**Eventually...**

## ***Proposed Future Developments***

The future developments addresses software methods and tools to support the global system/software engineering approach....

# **Global Engineering Approach**

## **1) System-Software Co-Engineering**

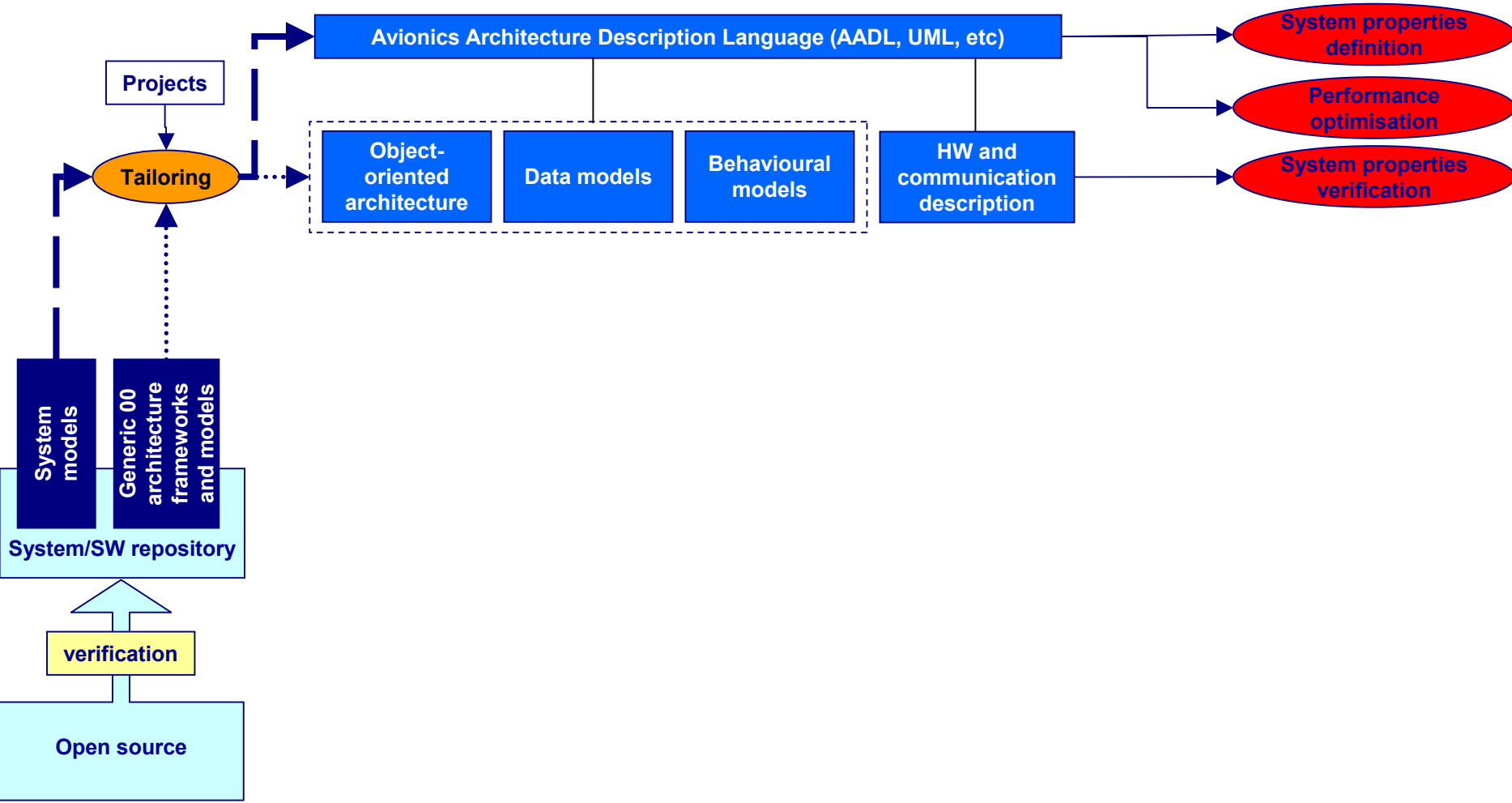


An engineering environment for system and software engineers in industry

Support the same activities as presently, help to manage complexity

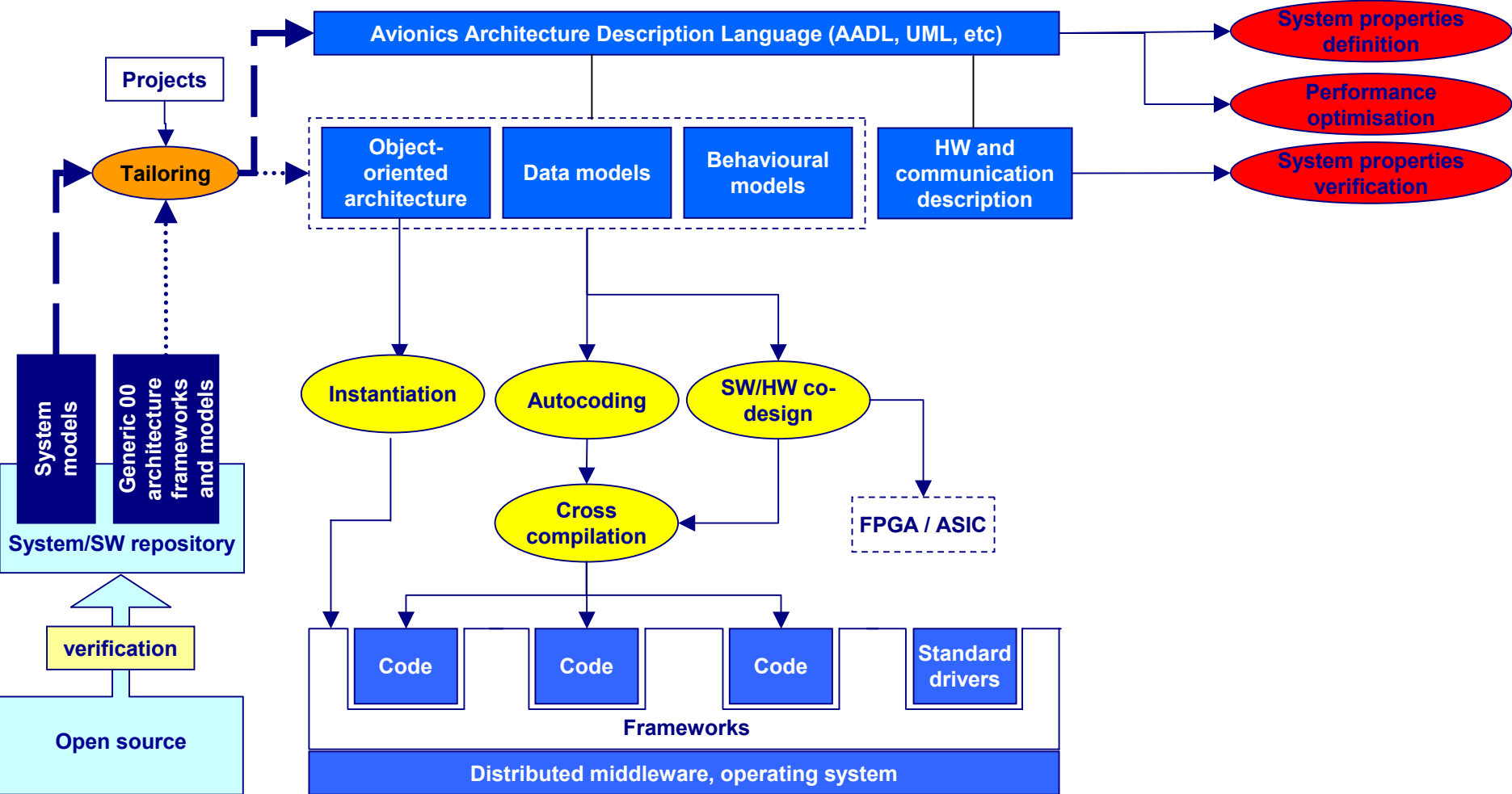
# Global Engineering Approach

## 2) Requirement Engineering



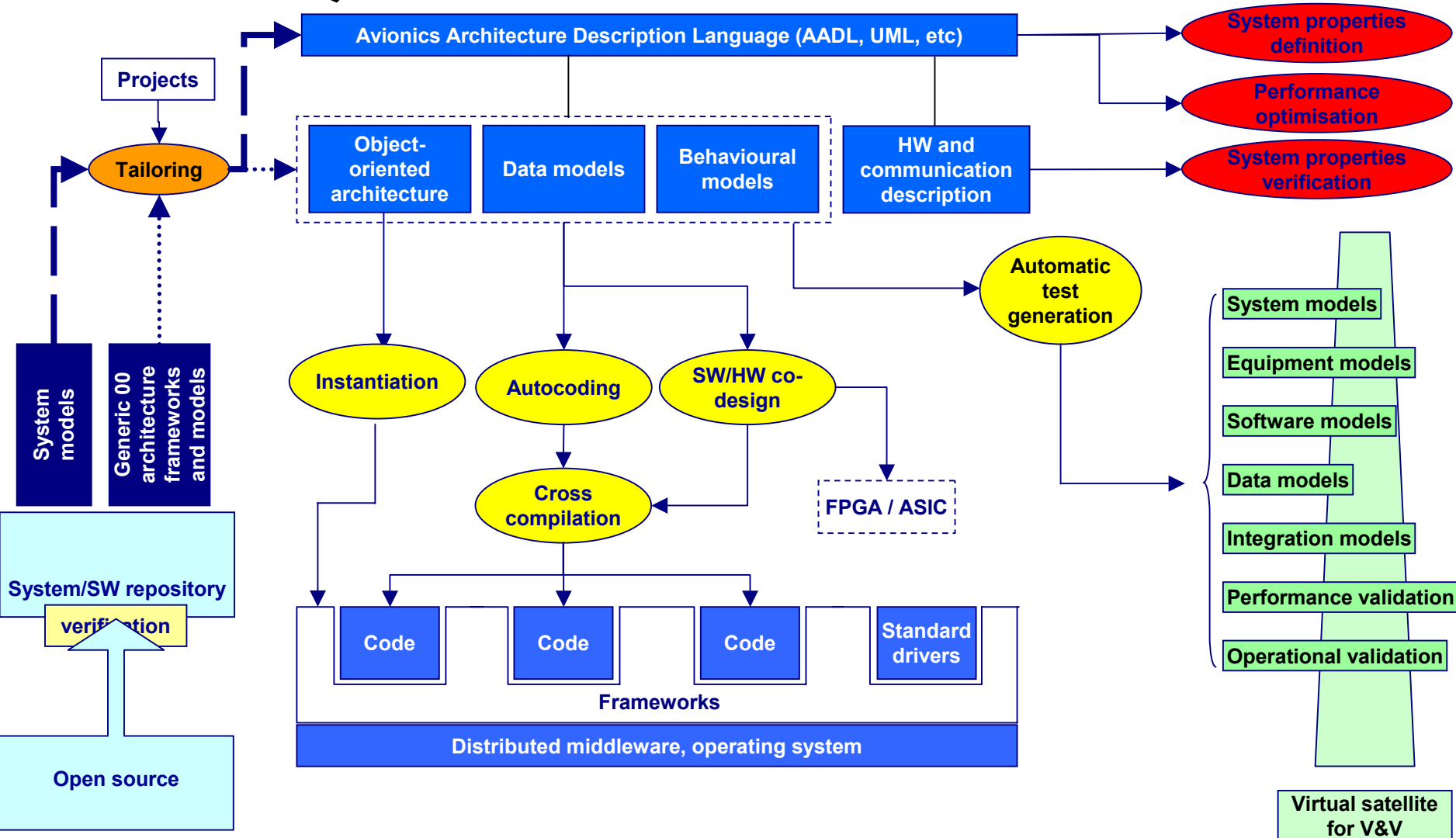
# Global Engineering Approach

## 3) Software Development



# Global Engineering Approach

## 4) Verification & Validation



## **Global Engineering Approach 5) Virtual Satellite**

- The R&D support often stops before a technology is really mature (Ref:D/IMT Procedure for Harmonisation)
- Goal: transfer technology in projects by demonstrating technology in context (and validate the building blocks)
- In orbit demonstration too risky, too much subject to opportunity, too expensive
- Integrate the technology/building block under development into a satellite context is better achieved through a Virtual Satellite
- Bring the technology from readiness level 1, 2, 3 (laboratory, feasibility, predevelopment) to 4, 5 (relevant environment, pre-qualified, proven)

## **Conclusions**

A number of federating themes have been identified according to strategic interest:

- ❑ *System/software co-engineering, requirement engineering*
- ❑ *Support to implementation, validation and maintenance of standardized software building blocks*
- ❑ *Automation of the software life cycle*
- ❑ *Software technology for system new architectures and functions*
- ❑ *Software cross development environments*