SIMULTECH 2022

12th International Conference on Simulation and Modeling Methodologies, Technologies and Applications

Lisbon, Portugal 14 - 16 July, 2022

M&S and MBSE: Individual Challenges and Mutual Opportunities

Andrea D’Ambrogio

Department of Enterprise Engineering
University of Rome Tor Vergata
Rome (Italy)

http://www.sel.uniroma2.it/dambro

Agenda

- Modeling & Simulation (M&S) and Systems Engineering (SE) disciplines
  - Model Based Systems Engineering (MBSE)
- **M&S-based Systems Engineering** (*M&S for SE*)
  - Lifecycle integration
- **Simulation Systems Engineering** (*MBSE for M&S*)
  - Distributed Simulation (DS)
  - Model-based approaches for DS Engineering

Simulation and SE

- Simulation-based and SE-based approaches have been employed ever since mankind started building complex systems
  - More recently, the term *systems engineering* may be traced back to the early 1900s, with examples observed during the Second World War
  - Similarly, what is commonly referred to as *(computer-based) simulation* is traced back to the Second World War
- Both approaches have then evolved into mature disciplines, frequently crossing their paths
  - nevertheless, several projects still experience time and cost overruns or fail to deliver systems that meet the needs of customers, users and other stakeholders

Simulation is just a part of the story..

- A simulation must always have a model and modeling is an essential part of a simulation
- To emphasize the modeling involved in a simulation, simulation should be better referred to as *Modeling and Simulation (M&S)*
- *Modeling* and *simulation* are central activities of our thinking process
  - we think by "*constructing mental models and then simulating them (i.e., executing over time) in order to draw conclusions or make decisions*"

MBSE

- Modeling has also achieved a key role in the SE field with the introduction of *Model Based Systems Engineering (MBSE)*
- MBSE refers to the formalized application of modeling to support the system development, along all the different development phases
- With MBSE, a common model of the system architecture is used to support and drive the engineering process
  - bringing significant advantages over the *document-centric* approach

M&S for SE

- M&S and SE may support each other around the *executable model* concept
- M&S promotes executable models and proposes simulation as the native mechanism to address measures of performance and effectiveness throughout conceptual design, development and later systems life cycle phases
  - facing the well-known SE challenges in terms of budget/time overruns and systems quality

⇒ **M&S Based Systems Engineering**

MBSE for M&S

- As the system complexity increases, the executable system architecture (i.e., the simulation model) may become so complex that it is necessary to assess it not only as a *valid support of SE processes* but also as an *objective of SE efforts*
- Executing an interdisciplinary systems engineering process for developing, maintaining and employing simulations, which enable systems engineers to experiment and gain insights about the systems of interest, is referred to as

⇨ **Simulation Systems Engineering**

M&S Based Systems Engineering

- The integrative approach
- Bridging the gap between M&S and SE lifecycles

M&S Based Systems Engineering

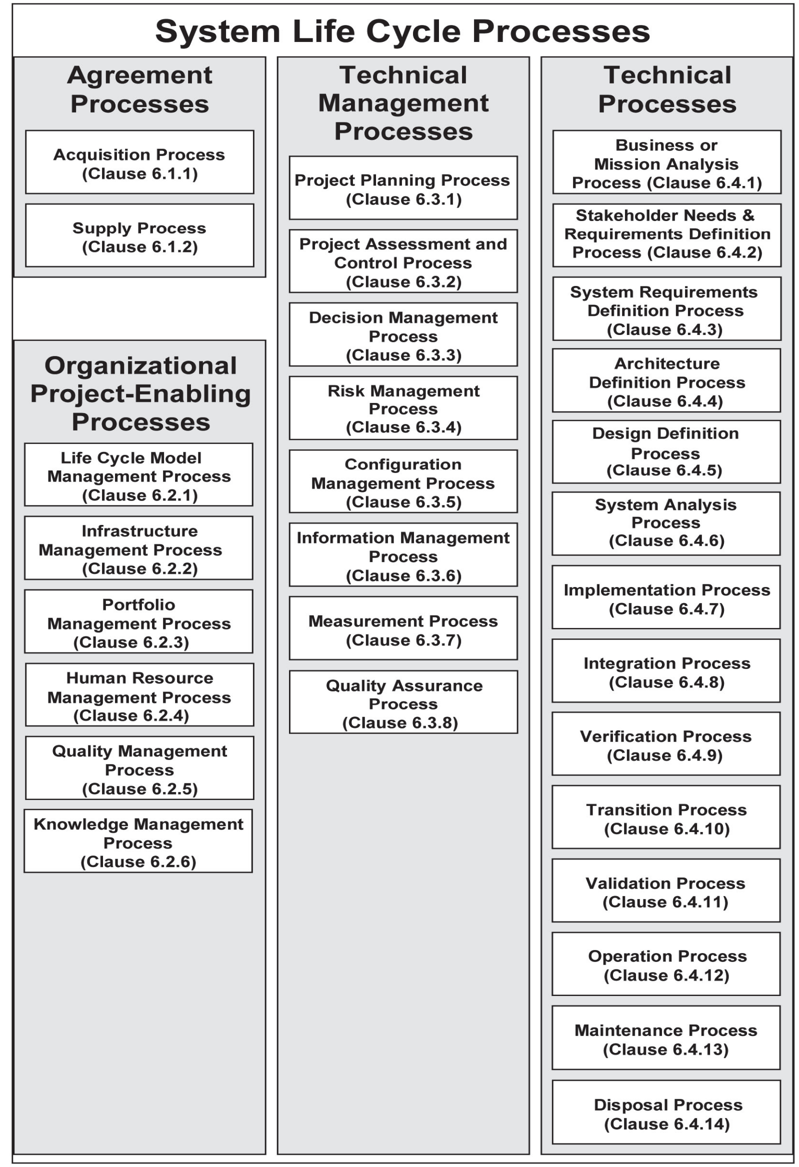
- The adoption of M&S based approaches is widely recognized as:
  - a cost-effective alternative to the development of experimental prototypes
  - a valuable strategy to mitigate the risk of time/cost overrun due to redesign and re-engineering activities
- Nevertheless, their potential and organized implementation in the systems engineering field are not yet fully exploited
  - even though there are plenty of applications in specialty engineering fields, as well as in other fields, such as physical science, social science, computing, medical research, military, business and finance

The integrative approach

- A successful implementation of *M&S based systems engineering* requires an effective integration of M&S and systems engineering life cycles, specifically:
  - the *ISO/IEC/IEEE 15288 (Systems and Software Engineering - System Life Cycle Processes)* standard at the systems engineering end
  - the *IEEE 1730-2010 (Recommended Practice for Distributed Simulation Engineering and Execution Process – DSEEP)* standard at the M&S end

ISO/IEC/IEEE 15288

- Describes a set of 30 processes in terms of purpose, outcomes, activities and tasks
- Processes are grouped in:
  - *agreement* processes
  - *organizational project-enabling* processes
  - *technical management* processes
  - *technical* processes

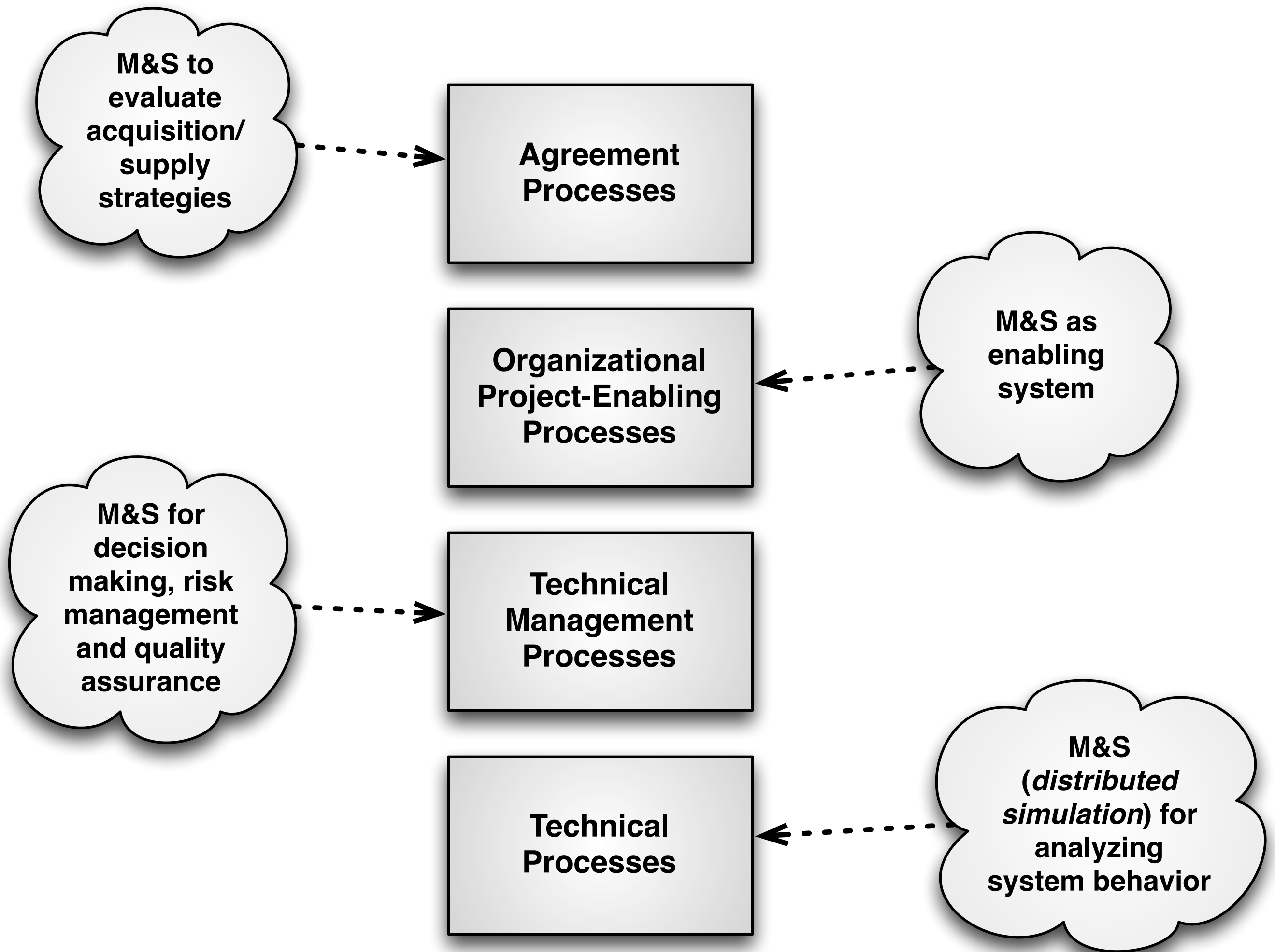


IEEE 1730-2010 (DSEEP)

- Introduces 7 sequential steps, described using *activities*, recommended *tasks*, *inputs*, *outputs* and *roles*

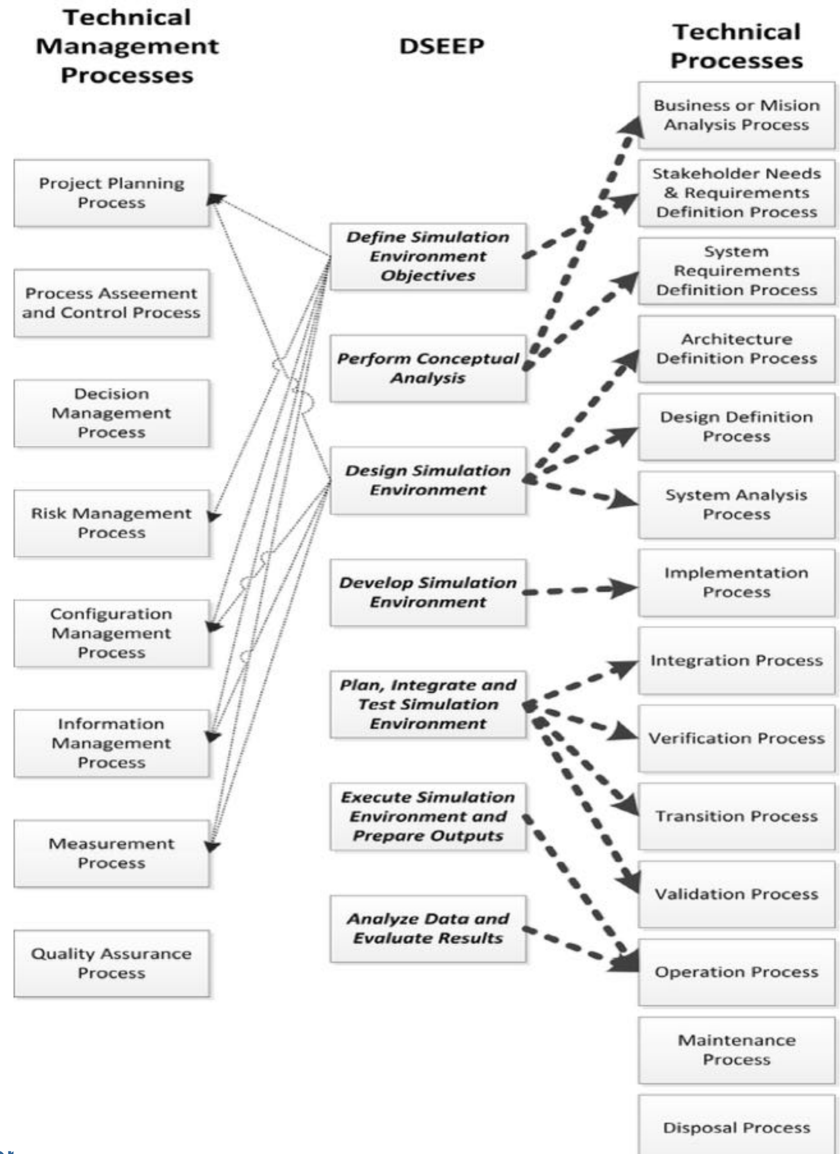
Step	(1) Define Simulation Environment Objectives	(2) Perform Conceptual Analysis	(3) Design Simulation Environment	(4) Develop Simulation Environment	(5) Integrate and Test Simulation Environment	(6) Execute Simulation	(7) Analyze Data and Evaluate Results
Activities	Identify User/Sponsor Needs Develop Objectives Conduct Initial Planning	Develop Scenario Develop Conceptual Model Develop Simulation Environment Requirements	Select Member Applications Design Simulation Environment Prepare Detailed Plan	Develop Simulation Data Exchange Model Establish Simulation Environment Agreements Implement Member Application Designs Implement Simulation Environment Infrastructure	Plan Execution Integrate Simulation Environment Test Simulation Environment	Execute Simulation Prepare Simulation Environment Outputs	Analyze Data Evaluate and Feedback Results

M&S for systems lifecycle processes



Bridging the gap between M&S and SE life cycles

- Integrative approach that allows systems engineers and M&S practitioners to fully catch the benefits of using M&S for systems engineering and vice versa
- Example DSEEP overlay on ISO/IEC 1528 (Systems and Software Engineering - System Life Cycle Processes)



D'Ambrogio, A. and Durak, U. (2016). *Setting systems and simulation life cycle processes side by side*. Proceedings of the 2016 IEEE International Symposium on Systems Engineering (ISSE 2016), October 3-5, Edinburgh, UK.

Simulation Systems Engineering

- Distributed Simulation (DS) in the systems development context
- Model-driven Engineering (MDE): a booster for MBSE
- MDE-based approaches for DS engineering

Distributed Simulation in the system development context

- The validation of complex systems from the early development phases (*lifecycle validation*) can be effort- and time-consuming
- M&S is widely recognized as an effective and powerful tool for lifecycle validation of systems, but:
  - M&S methods must scale with growth and evolution of complex systems and ecosystems (e.g., *SoS* or *ULS*)
- The use of *distributed simulation (DS)* approaches enable M&S methods that take into account the peculiar complexity/scalability/evolvability of complex systems

Useful definitions

- System Under Study (SUS)
  - the system that has to be simulated to get insights into or to predict its behaviour
  - typically specified at development time by use of *system models*
- Simulation Systems Engineering:
  - the set of activities to be carried out first to build a *simulation model* of the SUS and then to implement it into a *simulation system*, i.e., a software system that "executes" the model over time onto a given centralized or distributed platform.
- Local Simulation (LS) System
  - A simulation system deployed onto and executed by a single host
- Distributed Simulation (DS) system
  - A simulation system that consists of a set of sub-systems deployed onto and executed by a set of geographically distributed hosts

Distributed Simulation (DS)

- The term *distributed* is interpreted in the sense of traditional distributed computing (e.g., based on the C/S paradigm)
- *Goal*
  - synchronize and coordinate remote simulation programs
- *Benefits*
  - Geographical distribution
  - Integrating simulators from different manufacturers
  - Reusability
  - Load balancing
  - Fault tolerance

High Level Architecture (HLA)

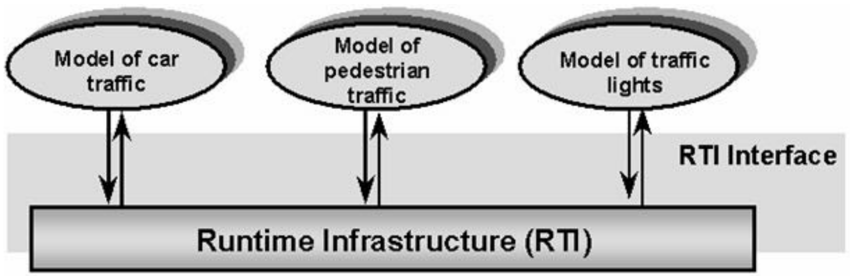
- A set of specifications for DS
- Goal:

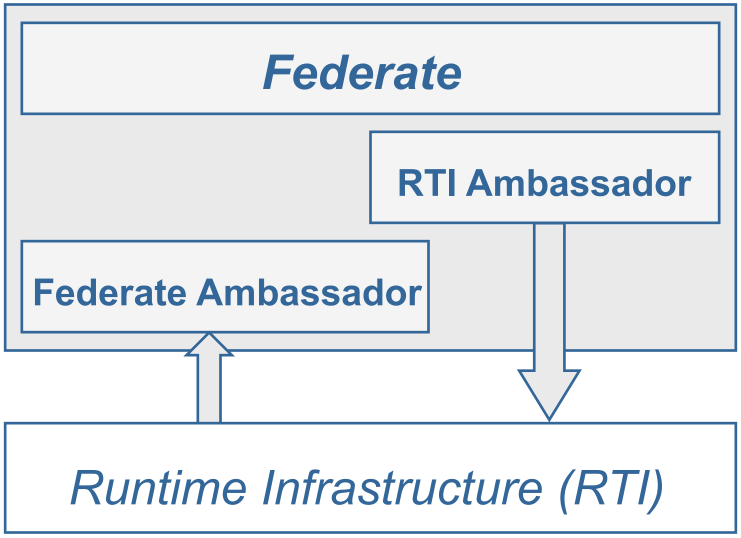
  improving *reusability* and *interoperability* in the simulation field

- Originally introduced by the DoD DMSO
- IEEE standard 1516 since 2000

How HLA simulations work

- *Federate*: a remotely-accessible simulation sub-system
- *Federation*: the overall DS system, composed of a set of Federates
- *RTI*: provides communication and coordination services to the Federates that join into a Federation





What HLA provides

- A distributed simulation framework to be applied, in principle, in different application domains (military, gaming, manufacturing, etc.)
- Simulation-oriented services (time management, information exchange, etc.)
- HLA *does not provide*:
  - Management of federate internal events
  - Services for specific types of simulation (discrete, continuous, etc.)
  - Guidelines for federate implementation

HLA Federation Object Model (FOM)

- The FOM is a data exchange model that describes information that is to be shared between different federates
- Using an information bus, such as the HLA, with a shared information model, such as the FOM, enables *composability*
- A FOM includes the definition of elements according to the HLA Object Model Template (OMT) specification, such as:
  - Shared object classes (i.e., persistent entities) with associated attributes
  - Shared interaction classes (i.e., events) with associated parameters
- A carefully designed FOM facilitates long-term reuse of simulations
- The FOM is part of a federation agreement

MDE: a booster for MBSE

- Model Driven Engineering (MDE) is a set of well-defined principles and practices based on tools that focus on the *productive use of models*
- MDE (Model Driven Engineering) further enhances the advantages of MBSE by increasing the *degree of automation* in the production and operation/maintenance lifecycle stages, by use of:
  - *metamodeling* techniques and
  - *model transformations*

MDE and DS
Opportunities

- MDE is finding increasing acceptance in the development of complex systems
  - enabler of reuse
  - higher degrees of automation
- DS systems are inherently complex
  - intrinsic concurrency
  - required interoperability
  - intricacies of currently available DS platforms
- MDE provides a promising approach for supporting the development of DS systems of higher quality at largely reduced time, effort and cost

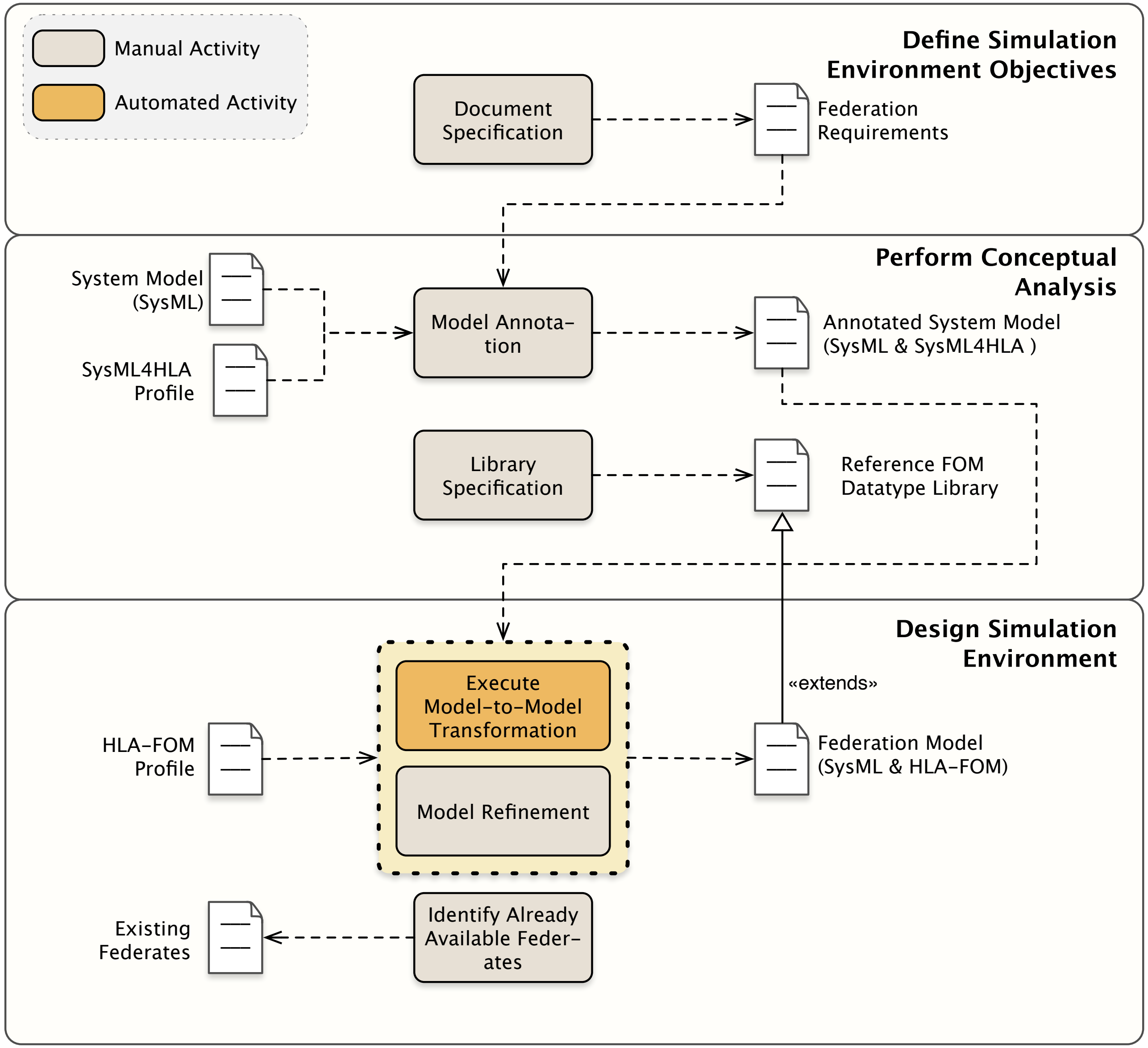
MDE and DS
Challenges

- On the DS side:
  - code-centric approaches
  - development process:
    - often not starting from scratch
    - often requiring the integration of legacy subsystems
  - interoperability is only dealt with at syntactic level
  - support for *simulation-in-the-loop* is limited
- On the MDE side:
  - model-centric approach
  - tool support for defining and orchestrating model transformations is still limited
  - modeling languages strongly influenced by UML

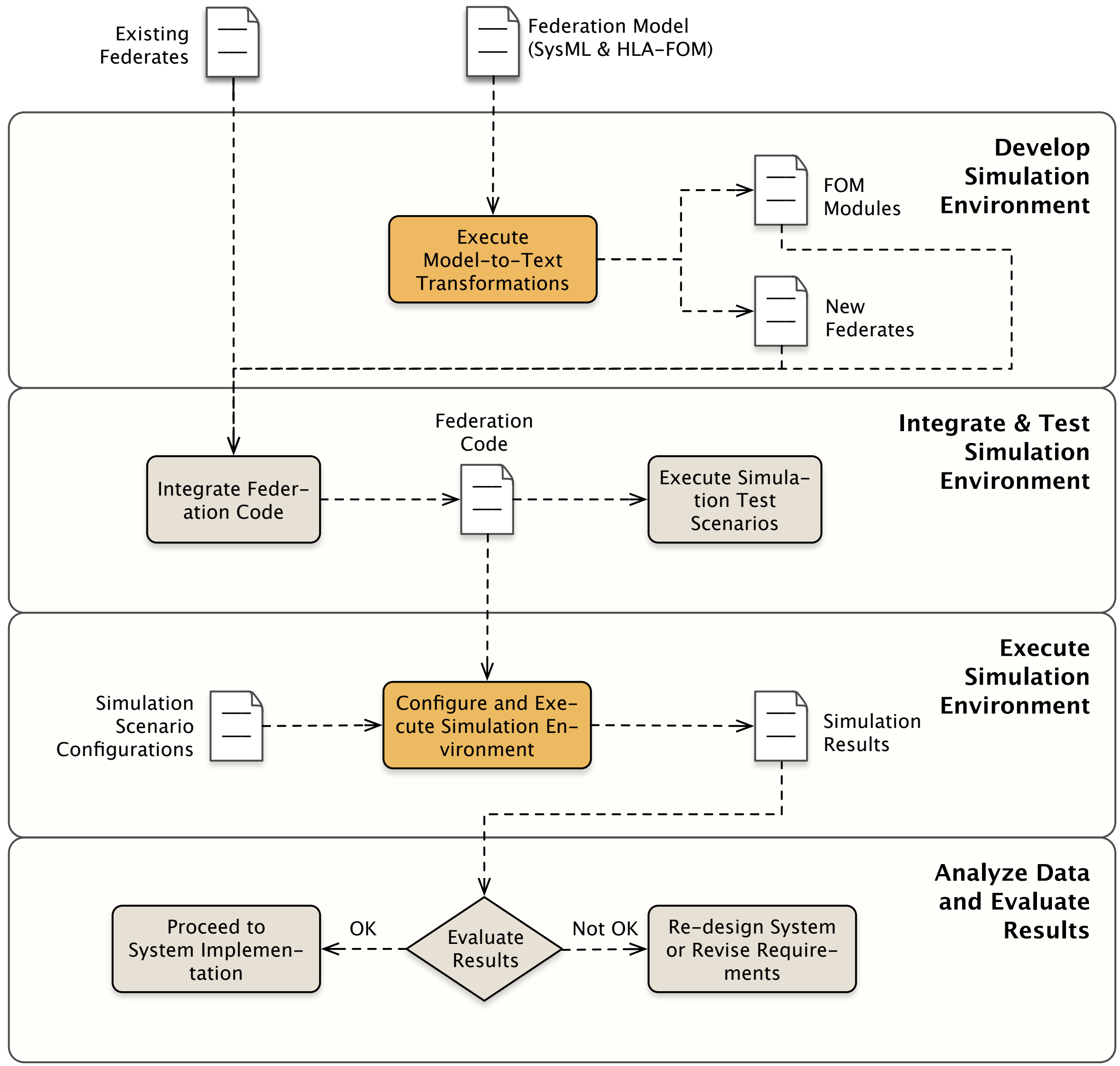
Bridging the gap between MDE and DS

- Approach based on the standard MDA (Model Driven Architecture) process
- Requires:
  - The choice of a specific DS infrastructure (e.g., HLA)
  - The introduction of proper modeling extension (*profiles*) for annotating system (SysML) models with DS infrastructure details
  - The specification of a system model to simulation model *model-to-model* transformation
  - The choice of a given DS implementation
  - The specification of a simulation model to code *model-to-text* transformation

MDA-based DSEEP (top)



MDA-based DSEEP (bottom)

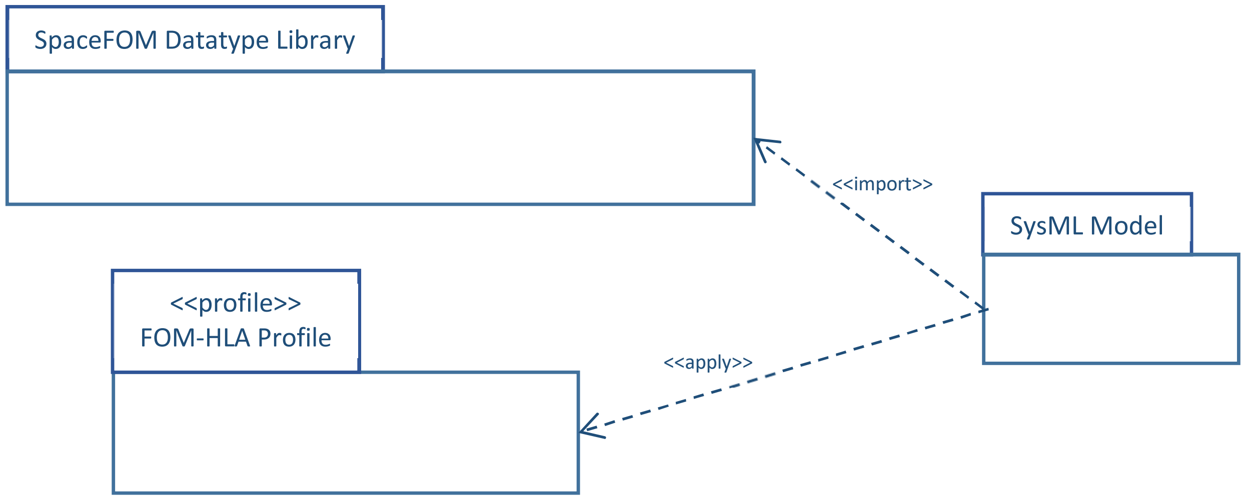


Model Transformations

- M2M transformation
  - `SysML-to-HLA`
    - from *SysML* to *HLA-based SysML*
- M2T transformations
  - `HLA-to-Code`
    - *HLA-based SysML* to *HLA code*
  - `HLA-to-FOM`
    - *HLA-based SysML* to *HLA code*
- Modeling Extensions
  - *SysML4HLA* Profile
    - for *SysML* annotation
  - *HLA-FOM* Profile
    - for *HLA-based SysML* annotation

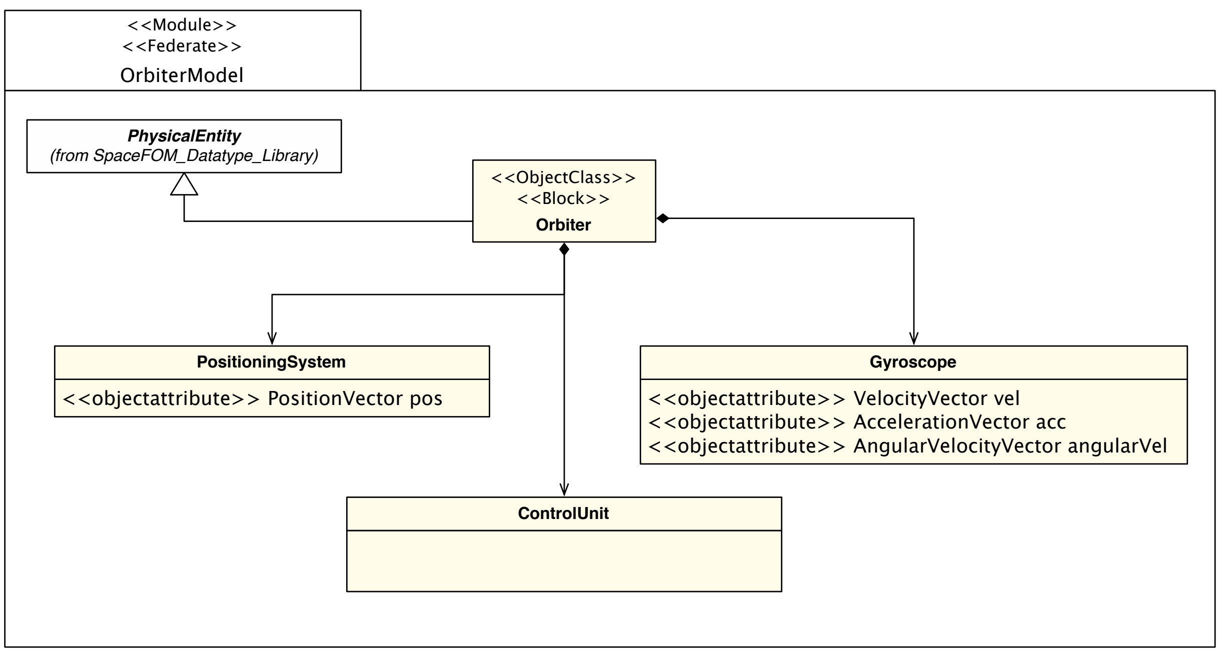
A Tiny Example Application

- An Orbiter federate has been implemented as a component of an *existing Space Federation*, which in turn must conform to the Space Reference FOM
- It is assumed that the SpaceFOM Datatypes Library is available, to enable the specification of federates compliant to the Space Federation

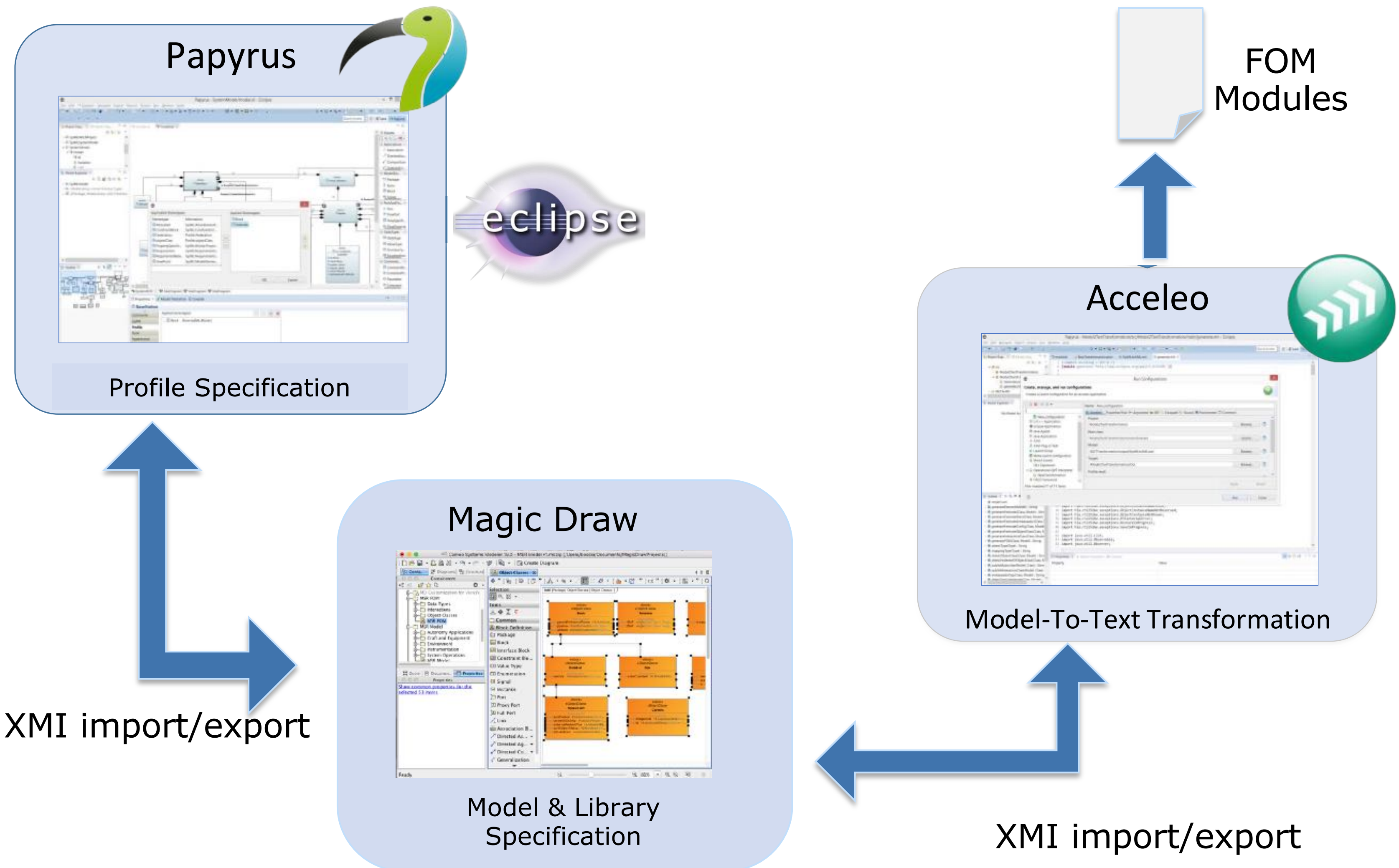


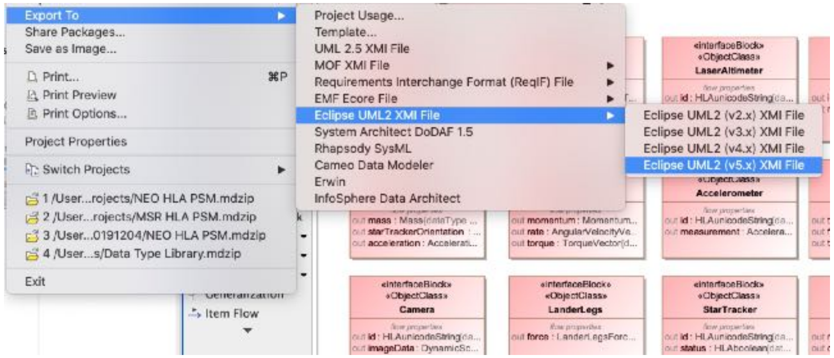
System Model Annotation

- According to the tailored DSEEP, once the simulation objective has been clearly identified, the **Conceptual Analysis** is executed to specify an *abstract model of the Orbiter system*
- At the next step, the **Design Federation Environment**, the model is annotated with stereotypes provided by the HLA-FOM profile

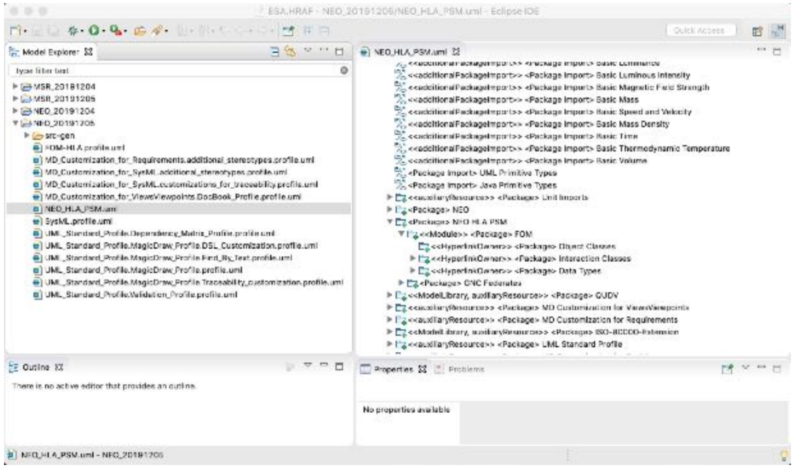


Toolchain


Papyrus
Profile Specification
eclipse
FOM
Modules
Acceleo
Model-To-Text Transformation
Magic Draw
Model & Library
Specification
XMI import/export
XMI import/export



Model XMI-based serialization (from *MagicDraw*)



Model import (into *Eclipse*)

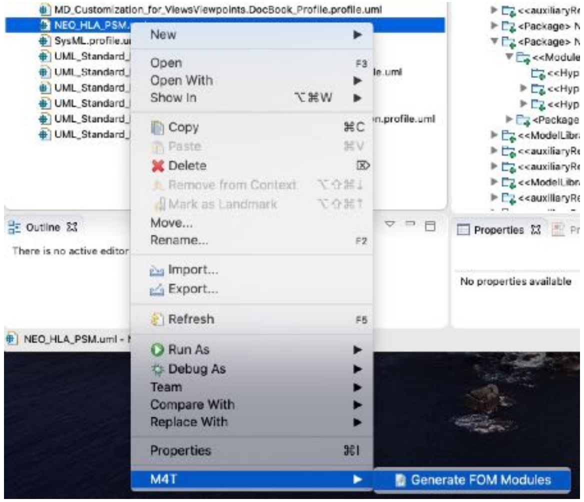
Listing 1: Fragment of Space Reference FOM

```
<?xml version="1.0" encoding="UTF-8"
   standalone="yes"?>
<objectModel xsi:schemaLocation="http://
   standards.ieee.org/IEEE1516-2010
...
<dataTypes>
<simpleData>
 <name>Velocity</name>
 <representation>HLAfloat64LE
 </representation>
...
<arrayData>
 <name>VelocityVector</name>
 <dataType>Velocity</dataType>
 <cardinality>3</cardinality>
 <encoding>HLAfixedArray
 </encoding>
...
</arrayData>
</dataTypes>
<objectClass>
 <name>HLAobjectRoot</name>
 <objectClass>
 <name>PhysicalEntity</name>
 <sharing>PublishSubscribe
 </sharing>
 <semantics>A PhysicalEntity is...</
    semantics>
...
</objectModel>
```

Listing 2: Fragment of Orbiter_FOM Module

```
<?xml version="1.0" encoding="UTF-8"
   standalone="yes"?>
<objectModel>
<modelIdentification>
        <name>Orbiter_FOM Module</name>
        <type>FOM</type>
        <version>0.1</version>
...
</modelIdentification>
...
<objects>
<objectClass>
 <name>HLAobjectRoot</name>
 <objectClass>
  <name>PhysicalEntity</name>
  <objectClass>
   <name>Orbiter</name>
   <sharing>PublishSubscribe</sharing>
   <attribute>
    <name>vel</name>
    <dataType>VelocityVector
    </dataType>
    <updateType>Conditional
    </updateType>
    <sharing>PublishSubscribe
    </sharing>
    <transportation>HLAreliable
    </transportation>
 ...
</objectModel>
```

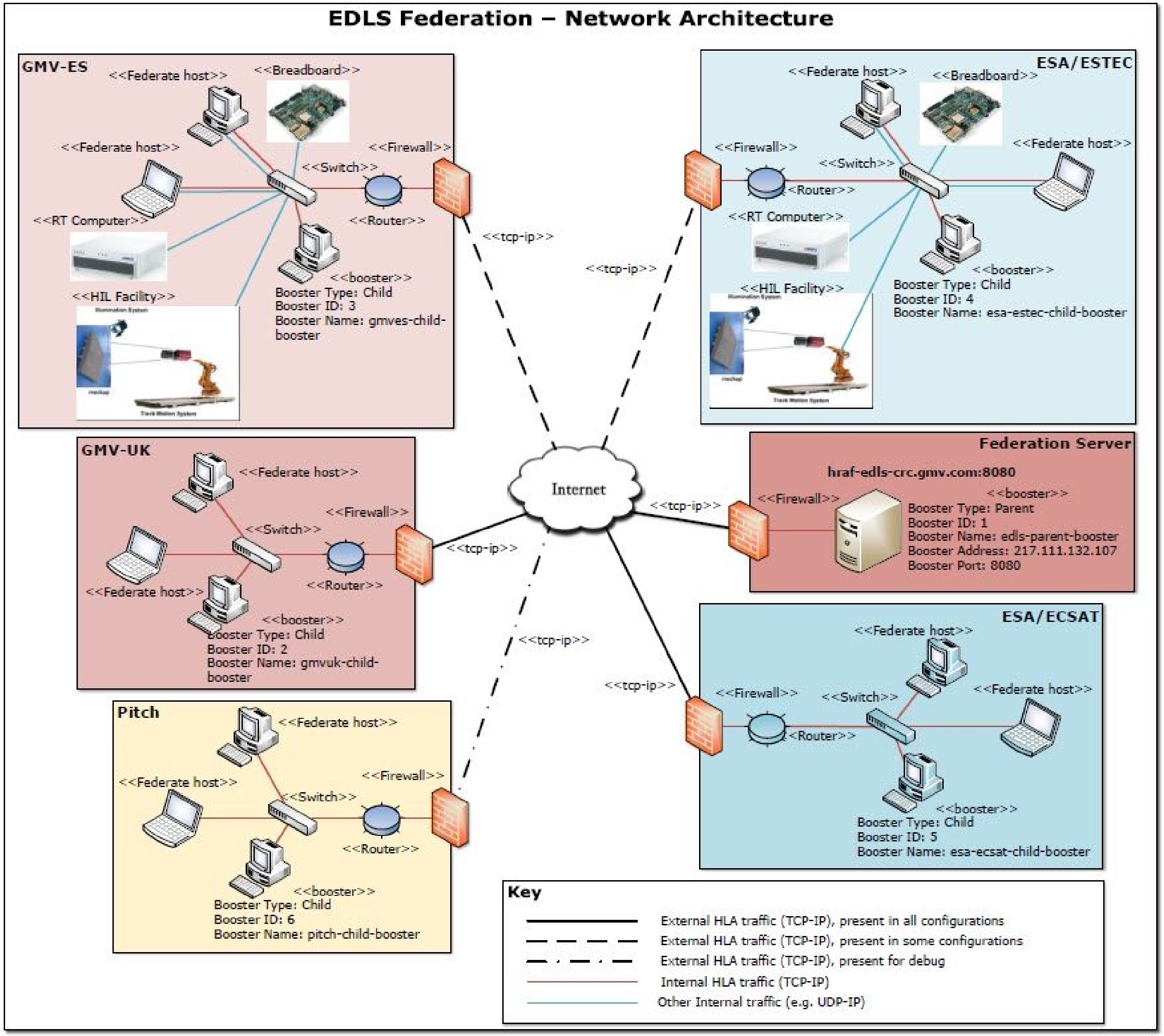
M2T transformation output



M2T transformation execution

dambro@uniroma2.it

ESA HRAF Project



Conclusions

- A successful synergy between M&S and systems engineering disciplines can be enacted by addressing the mutual support in terms of:
  - *M&S based systems engineering*
  - *Simulation systems engineering*
- In this respect, a significant effort has been spent at educational, theoretical, methodological and professional levels
- However, additional research and development effort is needed to further investigate and fully exploit the potential of cross-domain solutions

Modeling and Simulation-based Systems Engineering Handbook