A Methodological Template for Model Driven Systems Engineering

P. Bocciarelli, A. D’Ambrogio, E. Caponi, A. Giglio and E. Paglia.
Dept. Enterprise Engineering, University of Rome “Tor Vergata”.

Rome, November 25, 2014
Table of contents

- Introduction
- Background
- Methodological Template for MDSE
- MDSE Example Application
- Conclusions & Future Work
Introduction

- **Systems Engineering (SE)** focuses on producing and maintaining successful systems that meet user’s requirements and development objectives.
- **Architectural Frameworks** use models to represent aspects, perspectives and views of the system (i.e. DODAF, TOGAF).
- **SysML (Systems Modeling Language)** (INCOSE 2001) is a systems engineering extension to the well known and widely used Unified Modeling Language (UML).

**Model Based Systems Engineering (MBSE)**

- Architectural frameworks
- SysML
- UML
Introduction

Model Based Systems Engineering (MBSE)

**Basic Idea:** a model evolves over the system development life-cycle, and it is iteratively used during the analysis and design phases.

**Limitations:** lack of practices and tools that automate the models development.
Introduction

• Model Driven Engineering (MDE) supports systems development activities through the automated transformation of abstract models to operational components and applications.

• Model Driven Architecture (MDA) is the most used incarnation of model-driven engineering provided by the Object Management Group (OMG).

We introduce a methodological template for applying MDE principles and standards to MBSE approaches:

Model Driven Systems Engineering (MDSE)

MDE + MBSE
Background

Model Based Systems Engineering (MBSE), a definition:

„formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases“

Methodology:

• Process (P) – a logical sequence of tasks
• Method (M) – a set of techniques for performing a task
• Tool (T) – an instrument that can enhance the efficiency of a task
• Environment (E) – the surroundings, conditions or factors that influence the actions of an object
Background

Model Driven Engineering (MDE): an approach to software design and implementation that addresses the raising complexity of execution platforms, by focusing on the use of formal models.

Model Driven Architecture (MDA): OMG's incarnation of MDE

MDA Standards:

• *Meta Object Facility (MOF)* – for specifying technology neutral metamodels

• *XML Metadata Interchange (XMI)* – for serializing MOF metamodels and models into XML-based schemas and documents

• *Query/View/Transformation (QVT)* – for specifying model transformations
Overview of MDA Standards:
Methodological Template for MDSE

• Strongly based on the productive use of models;

• Defined as a quintuple:

<Em, Pm, Mtm, Tm, Lm>:

• Environment (Em)
• Process (Pm)
• Method (Mm)
• Technique (Tm)
• Language (Lm)
The set of semi-formal models that assist the team participants in grasping the abstract concepts related to SE
How much broad and deep is the knowledge for a systems engineer?
Each system can be subdivided into five levels:

5-Level Hierarchical Model:
- **System (L1)** – a set of interrelated elements that work together to achieve a specific objective;
- **Sub-Systems (L2)** – the major portions of the system that perform a closely related subset of the overall system functions;
- **Components (L3)** – the middle-level entities of system that perform a specific functionality;
- **Sub-Components (L4)** – which perform elementary functions and are composed of several parts;
- **Parts (L5)** – which perform elementary functions and are composed of several parts.
The set of the processes that describes the evolution of a particular new system. The stepwise evolution of each process is referred to as the *system life-cycle model*, which subdivides the system’s development process into a set of basic steps and phases.
Methodological Template for MDSE

- Method (Mm)

<Em, Pm, Mtm, Tm, Lm>:

- Environment (Em)
- Process (Pm)
- Method (Mm)
- Technique (Tm)
- Language (Lm)

The set of the activities that are iterated in each phase of the process. We refer to this dimension with the term *systems engineering method model*. 
Methodological Template for MDSE

System Engineering Method Model:

- Requirement Analysis
- Functional Analysis
- Physical Definition
- Design Validation

Such four activities are carried out in each phase of SE life-cycle model and vary in their specifics depending on the type of the system and the phase of its development.
Methodological Template for MDSE

The set of standards that supports process and methods.

<Em, Pm, Mtm, Tm, Lm>:

- Environment (Em)
- Process (Pm)
- Method (Mm)
- Technique (Tm)
- Language (Lm)
Methodological Template for MDSE

**Model:** the primary artifact to support the construction of complex systems.

**Metamodeling:** architectural abstraction that provides the foundations for construction, manipulation and validation of models.

**Model Transformation:** is related to a fundamental task associated with the productive and automated use of models.

**OMG MDA:** a family of technology standards.
Methodological Template for MDSE

- Environment (Em)
- Process (Pm)
- Method (Mm)
- Technique (Tm)
- Language (Lm)

<Em, Pm, Mm, Tm, Lm>:

- Environment (Em)
- Process (Pm)
- Method (Mm)
- Technique (Tm)
- Language (Lm)

The set of languages that ensures the formal and correct manipulation of the models.
Methodological Template for MDSE

**UML**: general purpose language that allows the construction of hardware and software models.

**SysML**: lightweight UML extension which is the standard modeling language in the SE domain.

**QVT (Query/View/Transformation)**: MDA Standard that deals with the specification of model transformations.

**XMI (XML Metadata Interchange)**: used for defining, exchanging and processing data objects in XML format.

**OCL (Object Constraint Language)**: formal specification language that allows the definition of constraints over model elements through a syntax based on the first-order predicate calculus.
Critical Issues:
1. Manual modeling activities are effort and time consuming;
2. The derivation of software models from system models can be error-prone and not easily traceable.

Solution Adopted:
The specification of a *SysML/UML bridge* based on an automated PIM-to-PSM transformation coded in the QVT/Operational Mappings language.

Example rule:

```plaintext
mapping Block::getComponentOperations() : Set(Operation){
    init{ result := self.ownedOperation.map operation2operation() asSet() ->
        union(self.ownedPort>select(isStereotypeWidth('PublishPort')).map
            getPublisherOperation() asSet() ->
        union(self.ownedPort>select(isStereotypeWidth('SubscribePort')).map
            getSubscriberOperation() asSet()); }
```
Conclusions & Future Work

• The analysis and the production of a complex system must be supported by tools and mature technologies.

• Each phase of the development process must be highly structured, so it is necessary to define the documents that must be produced or the tests that are to be planned and executed.

• It is essential to adopt solutions that can be widely recognized.
Conclusions & Future Work

• The model-driven approach has been applied in two stages, but can also be extended to the engineering step.
• The object-oriented method alone is not sufficient and also functional analysis and predictive performance analytics should be carried out.
• MDA techniques alone are not sufficient, as they focus their efforts on aspects of design models. Techniques for model-checking correlated with the validation of the models can be used.

Interesting challenges may be faced as future work: the evolution of the template in a methodological framework that can fully support the production activities.