



**HAL**  
open science

## Metamodels Approach Supporting Models for Manufacturing (MfM) Methodology

Domingo Morales-Palma, Manuel Oliva, Jesus Racero, Ignacio Eguia, Rebeca Arista, Fernando Mas

► **To cite this version:**

Domingo Morales-Palma, Manuel Oliva, Jesus Racero, Ignacio Eguia, Rebeca Arista, et al.. Meta-models Approach Supporting Models for Manufacturing (MfM) Methodology. 18th IFIP International Conference on Product Lifecycle Management (PLM), Jul 2021, Curitiba, Brazil. pp.398-409, 10.1007/978-3-030-94399-8\_29 . hal-04195251

**HAL Id: hal-04195251**

<https://inria.hal.science/hal-04195251v1>

Submitted on 4 Sep 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

# Metamodels approach supporting Models for Manufacturing (MfM) methodology

Domingo Morales-Palma<sup>1</sup>[0000-0001-5816-9528], Manuel Oliva<sup>2</sup>[0000-0001-5945-1162], Jesus Race-ro<sup>1</sup>[0000-0002-7482-0744], Ignacio Eguia<sup>1</sup>[0000-0003-3969-9958], Rebeca Arista<sup>1,3</sup>[0000-0001-9382-1593], Fernando Mas<sup>1,4</sup>[0000-0001-7230-9929]

<sup>1</sup>University of Sevilla, Sevilla, Spain

<sup>2</sup>Airbus, Sevilla, Spain

<sup>3</sup>Airbus, Toulouse, France

<sup>4</sup>M&M Group, Cadiz, Spain

dmpalma@us.es, fmas@us.es

**Abstract.** Models for Manufacturing (MfM) is a novel approach proposed by the authors to apply Ontology-Based Engineering (OBE) concepts to Manufacturing. The methodology is based in a 3-Layers Model (3LM) framework and supported by any modelling tools easy-to-use. The Ontology layer comprises the ontological definition holding the knowledge of the domain. This work presents the introduction of metamodels or surrogate models for supporting MfM methodology with the objective to maintain independence with commercial software tools. This paper introduces the proposed metamodels and relationships, shows the link within the MfM Ontology layer and discusses the further activities.

**Keywords:** Metamodels, Surrogate models, Models for Manufacturing (MfM), Ontology-Based Engineering (OBE), Manufacturing process modelling.

## 1 Introduction

To reduce overall costs in the industry and to secure a short time-to-market, products and industrial systems need to be designed in shorter time often reusing existing products and resources. To achieve these points and to enhance multidisciplinary design and realistic simulation of complex systems, Ontology-Based Engineering (OBE) emerges as a new approach from earlier techniques thanks to the technological advancements in computer science.

Models for Manufacturing (MfM) is an OBE novel methodology, where computer aided graphical modeling authoring tools are used to define and specify data, functions, and behaviors of the industrial systems, where simulation tools will replicate the required system behavior for the design of complex products. It is derived from the concept of MBSE methodology, enabling more robust engineering in terms of models and their associated behavioral abstraction [1].

The authors released several research papers for modelling manufacturing systems in the aerospace industry fundamentally related to assembly in the Final Assembly

Lines (FAL) [2,3], releasing functional and data models, deployed using data structures available in commercial PLM systems [4].

As consequence of the research process, a new approach for modelling manufacturing systems was coined to introduce OBE in the industrial and manufacturing domain. A 3-Layers Model (3LM) architecture based has been defined consisting of a Data layer, an Ontology layer, and a Service layer. The Ontology layer build the core of the 3LM, and it is defined by four models: Scope model, Data model, Behavior model and Semantic model. A preliminary definition of the MfM methodology has been published by the authors in [5].

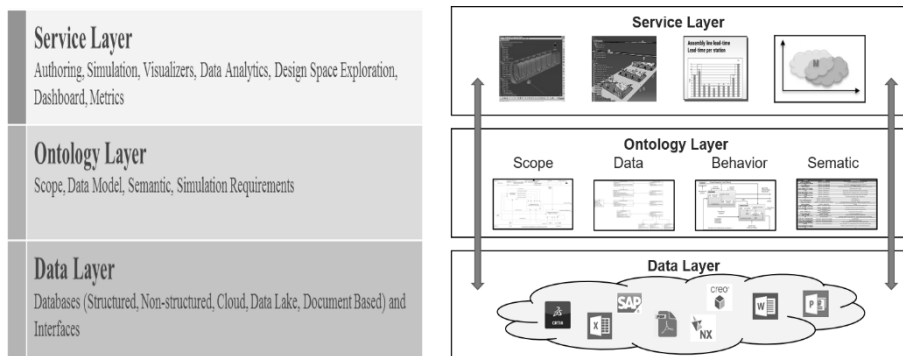
This work presents the introduction of metamodels or surrogate models for supporting MfM methodology with the objective to maintain independence with commercial software tools. This paper introduces the proposed metamodels and relationships, shows the link within the MfM Ontology layer and discusses the further activities.

Following to the rest of the document, Section 2 contains a review of the MfM methodology with references to the industrial use case developed. Section 3 justifies the need of metamodels to support MfM methodology. Section 4 is devoted to present the metamodel, characteristics and details. Section 4 discloses the discussion and conclusions from this work and proposes topics for further research.

## 2 Review of Models for Manufacturing (MfM) Methodology

### 2.1 3-Layers Model (3LM) as framework

Models for Manufacturing methodology is based in a 3-Layers Model framework as shown in Figure 1. The 3LM guarantees the independence between the three layers, holding Service Layer, Data Layer and Ontology Layer isolated each other.



**Figure 1.** 3-Layers Model (3LM) framework.

Each layer collects the same type of items. The Data layer collects all the information in terms of databases and interfaces: legacy databases related to legacy software, commercial software databases, clouds, data lakes and many others. The Ontol-

ogy layer holds the Company knowledge in form of scope, data behavior and semantic models, and will be described in detail in the next subsection. The Service layer holds the software services, legacy or commercial, like authoring tools, simulation tools, visualizers, data analytics services, dashboards, and space design exploration tools.

The MfM methodology follows a set of golden rules:

- MfM methodology is agnostic against tools, do not define any preferred tool.
- MfM methodology encourages tools easy to write models and very easy to read, understand, share, and discuss by skilled engineers on the topic.
- MfM methodology encourages a mechanism to manage the model's lifecycle, configuration and effectivity using PLM tools to fulfill this requirement.

## 2.2 The Ontology layer

The core of the 3LM framework is the Ontology layer, where the knowledge of the Company is created, stored, managed, maintained and used. The Ontology Layer is defined by the Scope model, Data model, Behavior model and Semantic model. Following the golden rules, the methodology is conceived in an agnostic way not linked to any software tool. Surveys of the existing ontology software tools show that a significant number of tools exists on the market which are frequently used [6,7].

The Scope model defines the limits where the model works and contains all main Data model objects and the definition of the functions which will define the behaviors. It is a major model for discussion between experts and should be maintained as simple as possible.

The Data model defines the information handled in the selected scope. Using the data objects from the Scope model, the engineers can enrich, reuse and complete the Data model.

The Behavior model defines the behavior of the system delimited in the Scope model, the simulation or other functions defined in the Scope model and complete the full system description.

The Semantic model maintains connection between models among their lifecycle given digital continuity to the ontologies. The Data model should be instanced with real data coming from the databases (commercial applications, bespoke and legacy systems) and is defined in different ways, different languages or different formats (i.e. Date in American or European format).

## 2.3 Application of MfM to industrial and non-industrial Use Cases

This section describes the most representative research work made by the authors applying MfM methodology and Model Lifecycle Management principles to industrial and non-industrial cases.

Modelling of Single Point Incremental Forming (SPIF) manufacturing technology for sheet metal parts was demonstrated in [8], considering the complexity of the CNC

technology which usually requires a numerical study to validate the process. The ontology models were presented by means of IDEF0 [9] diagrams and concept maps.

Mas et al. [10] presented an ontology for aerospace assembly lines in Airbus, with the introduction of a novel way to characterize the adherence concept in the industrial design process of aerospace assembly lines. The research presented a Scope model and enriching the Data model with attributes and relations. ARAS Innovator was used as MLM system to store the ontology and fulfill MfM requirements to manage lifecycle and configuration of different objects.

A gender diversity analysis using MfM methodology was presented in [11], with the objective of approaching this complex social problem in a novel way, applying MBSE techniques and proving the applicability of MfM methodology to non-industrial cases. The ontology was composed of a Scope model, a Data model and Behaviors models, generated using several FOSS (Free Open Source Software) tools.

A preliminary ontology to support the collaborative engineering process of industrial system design was presented in Arista et al. [12]. A Scope model and Behavior model was defined and demonstrated through a use case of a global industrial system design for DA08 artifact, introduced by the authors to test aerospace use cases [13].

Some conclusions of these works was that the combination of more than one modelling technique allowed defining a more robust ontology from different perspectives. Also that designing the system scope model among all project participants, is an efficient method to establish the starting point for development of both Ontology and Service layers.

### **3 The need of metamodels to support MfM methodology**

The MfM methodology is being defined and enriched iteratively, fed by the experiences got from its application on different use cases, some of which were presented in the previous section. The ontology layer is therefore being defined and refined progressively. In a first phase, the Scope and Data model types were the ones mainly addressed, as presented in previous works [8,10]. Currently, proposals for the Behavior and Semantic models are worked in [11,12].

Until now, 4 models types have been created using the main concepts of MfM methodology. Other secondary concepts could not be clearly formalized from the beginning, but only after understanding the constraints found in the first applications on real use cases. The goal was set to clarify these concepts at a later stage of the project, and as a result, some ambiguities appeared between models of different use cases. Such ambiguities have not prevented progress in the conceptual development of the MfM methodology, however, they complexified the first implementation tests in a PLM system.

The lack of a clear and precise definition of some concepts in the ontology layer can lead to different interpretations in the team of experts as well as possible confusion with the object schema of the modeling application that is being used. For example, IDEF0 diagrams allow you to define some Controller type objects that are not part of the Scope model. This example shows that it is also necessary to specify how

to use an existing modeling tool, and thus with its own object schema, to implement your own object schema.

The need to formalize all the concepts of the ontology layer has become more evident in the current phase of the project in which a Model Lifecycle Management (MLM) is being developed and implemented. It is not only necessary to formalize the concepts, but also to allow each concept to change and evolve during the development of the MfM methodology. The objective of a MLM is to manage the life cycle, not only of the objects of the different use cases (the models), but also of the formalized concepts of the ontology layer (the metamodel).

The definition of the metamodel has been carried out using the Meta-Object Facility (MOF) standard from the Object Management Group. MOF is a Domain Specific Language (DSL) used to define metamodels [14]. The metamodel developed includes all the concepts of the ontology layer and 4 model types (Scope, Data, Behavior and Semantic), as well as the relationships between them. The metamodel does not include information related to the diagrams visualization (positions of elements, size, etc.).

## 4 Metamodel for the MfM Methodology

### 4.1 Scope Model

The Scope model contains the definition of the main system functions and main data objects. Figure 2 shows the metamodel for the Scope model ('scope' package). The main concepts are the system functions or activities (Activity class), the resources or means used for their realization (Means) and the data objects required and produced by activities (DataObject, from the 'data' package described in the next subsection), as well as the relationships between them.

The root class of the 'scope' package is ScopeModel that is composed of a main 'rootActivity' activity that is decomposed in sub activities.

Some concepts such as activities, means and data objects have been defined by a pair of classes, one of them abstract, with a generalization relationship. See for instance the Activity and AbstractActivity classes in Figure 2. This class structure allows us to define relationships between objects of a same type. Thus, a sequence of activities has been modelled by the relationship between AbstractActivity and Activity and the roles 'prevActivity' and 'nextActivity'.

An Activity class can be a ComposedActivity (see 'activityParent' and 'activityChild' roles) or an ElementalActivity (an activity without children). The latter is a key concept in the Behavior model, as discussed below.

An Activity may require different Means for its realization. Sometimes, the different Means that are used in sub activities can be grouped in a single Means, easier to associate with the parent activity. For example, "CAX" can be used to group "CAD", "CAD/CAM" and "CAPP" systems. This option has been defined with an aggregation relationship between Means and AbstractMeans.

An Activity is performed by using Means, thus the relationship between both classes is simple and direct. Instead, DataObject can be an input to or an output from

Activity. To define this relationship, an ActivityDataRelation class has been included. This class inherits two attributes ('dataRequired' and 'dataProduced') from an abstract class given that both attributes are also used in the 'behavior' package.

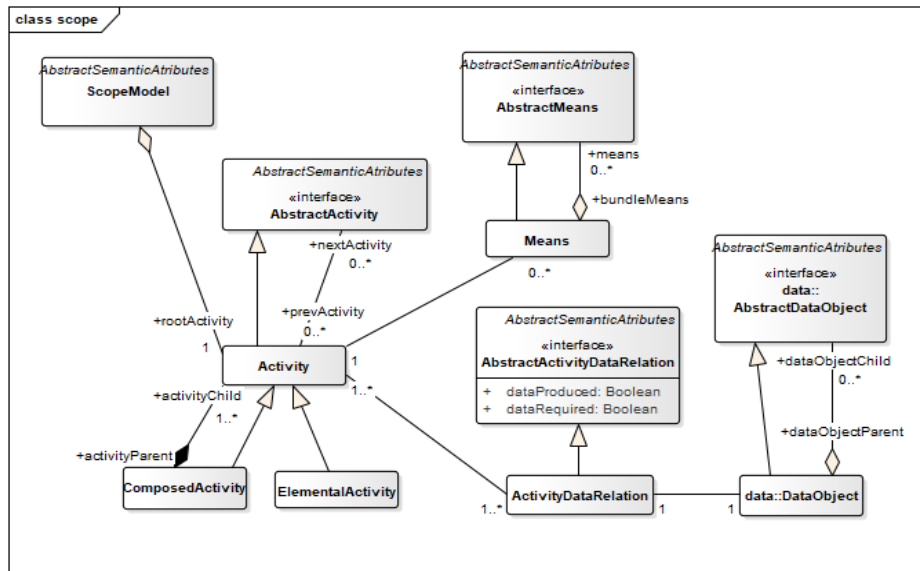


Figure 2. Metamodel of the Scope model.

Similar to Means grouping, data objects can be related to each other by aggregation relationships (see 'dataObjectParent' and 'dataObjectChild' roles in Figure 2).

Currently, for prototyping purposes, IDEF0 is the selected tool for building the Scope model. IDEF0 diagrams are a graphical tool widely used in aerospace since decades and cover the requirement “simple to write, easy to read”. They adjust to the proposed MfM metamodel. Examples of its use in the MfM methodology can be consulted in previous work [8, 10-12].

## 4.2 Data Model

The Data model defines the information to be managed in the selected scope. Since the Scope model already contains the main data objects, these objects are first mapped from the Scope model to the Data model to start building it. Engineers can then enrich, reuse and complete the Data model by adding new data objects and object properties.

Figure 3 presents the metamodel for the Data model. The root class DataModel contains all the data objects. A DataObject can be decomposed into child objects and can have properties. You can also relate two data objects using the DataObjectRelation class, which is used to specify the type of relationship in order to enrich the Data model. A Property is usually (but not limited to) a basic data type of an object and has



two attributes 'dataType' and 'value' (examples: 'float' and '8.0'; class 'Material' and object 'steel\_H240LA').

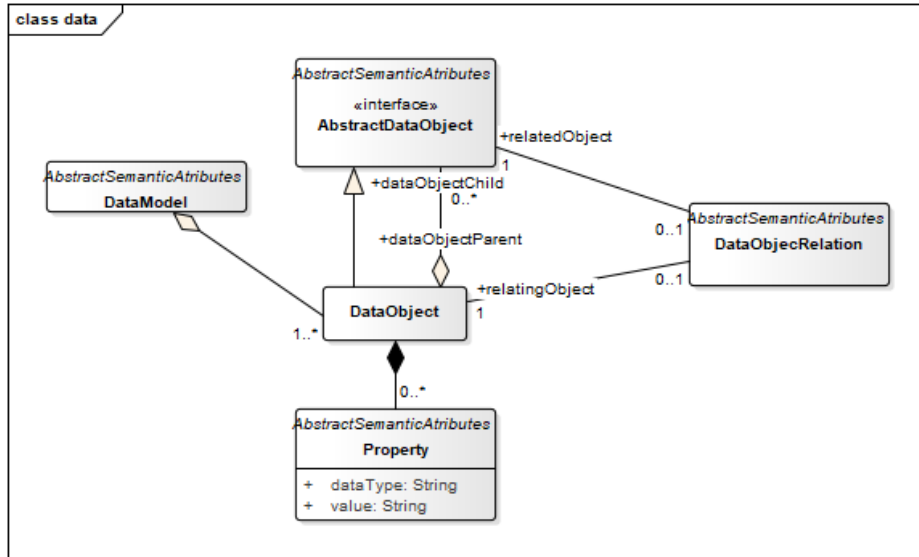


Figure 3. Metamodel of the Data model.

Currently, for prototyping purposes, concept maps are the tool of choice for building the Data model. A concept map is a graphical tool for structuring and representing knowledge. It covers the requirements “simple to write, easy to read” and adjust to the proposed MfM metamodel. Examples of its use in the MfM methodology can be consulted in previous work [8,10-12].

### 4.3 Behavior Model

The Behavior model defines the simulation or other software defined in the selected scope needed to check the full system. The key concepts of the Behavior Model are the set of tasks to carry out each elementary activity previously defined in the Scope model, the rules or procedures associated with the tasks and the constraints associated with processing the data objects and properties previously defined in the Data model.

Figure 4 presents the metamodel for the Behavior model. The root class BehaviorModel is composed of all the ElementalActivity defined in the Scope Model. An ElementalActivity adds at least one Task. Each Means of the ElementalActivity (see 'scope' package above) can be assigned to various Tasks of that ElementalActivity.

Tasks are modelled with the AbstractTask and Task classes. Two tasks can be performed consecutively ('prevTask' and 'nextTask' roles). The rule or procedure to perform a Task is modelled with the Rule class. Each Task has its own Rule. A Rule can have several Constraints. A Constraint is associated with a Property of a data object.

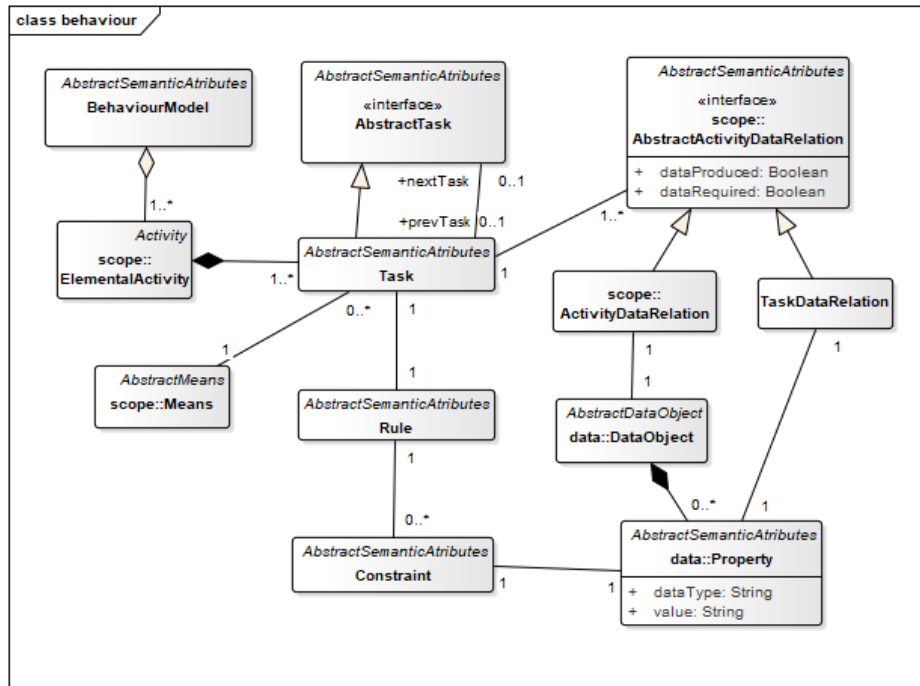


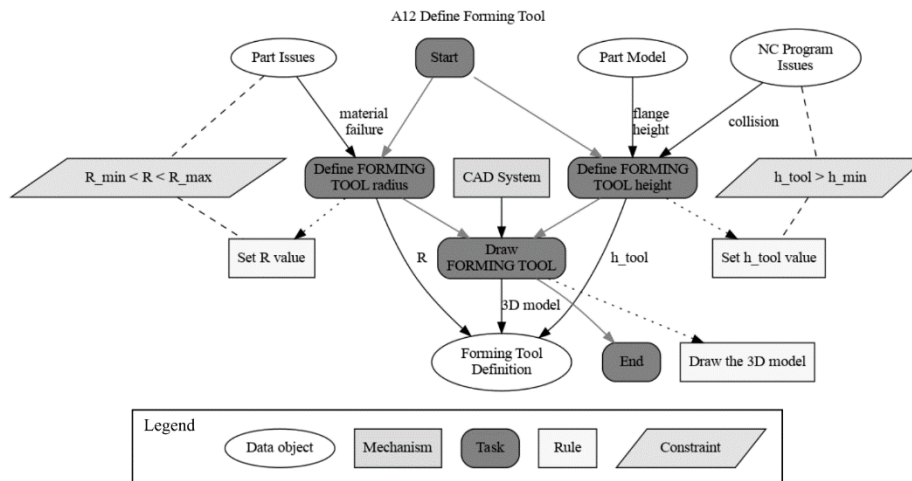
Figure 4 Metamodel of the Behavior model.

All the data objects associated with the ElementalActivity are now assigned to the corresponding Tasks. This can be done in two different ways. The first way simply specifies which task in the elementary activity each data object uses. It is identical to how it is done in the Scope model with the ActivityDataRelation class, substituting the ElementalActivity for the Task. The second way is to use the TaskDataRelation class to relate the Task to a specific Property of the data object, thus adding more detailed information. Both ActivityDataRelation and TaskDataRelation classes inherit the 'dataRequired' and 'dataProduced' attributes from the AbstractActivityDataRelation class.

Currently, for prototyping purposes, an ad-hoc diagram type designed to build Behavior models based on the proposed metamodel is used. These diagrams are built by using the open source software GraphViz in *dot* format [15]. They cover the requirements "simple to write, easy to read" and adjust to the proposed MfM metamodel.

Figure 5 has an example of Behavior model for use case "Manufacturing a part by SPIF". This example describes the system behavior for the activity "A21 Define Forming Tool". There are tasks to define the tool radius and height and to draw the tool in a CAD (Computer Aided-Design) system, following the indicated flow of tasks from Start to End. Each task has a rule associated with it, which in this case are described by short phrases such as "Set R value", "Set h\_tool value" and "Draw the 3D model", respectively. Nonetheless, the content of the Rules must describe in sufficient detail how the tasks are performed. On the other hand, some data objects such as

"Part Issues", "Part Model" and "NC Program Issues" are required to perform the tasks. These inputs are represented by an arrow pointing to the task. The text labels accompanying the arrow indicates the attribute of the data object required to perform the task. Similarly, the data objects that are generated (outputs) and their attributes are represented by arrows from tasks to objects and text labels accompanying the arrows, respectively. The example also shows constraints for the definition of the R and h\_tool parameters. For instance, the constraint " $R_{min} < R < R_{max}$ " allows avoiding possible issues in the part to be manufactured (the input object), in particular a failure of the material being deformed (attribute), with the procedure to determine the R value (the rule).



**Figure 5** Example of a Behavior model using GraphViz.

#### 4.4 Semantic Model

The Data model in the ontology should be instanced with real data and run real use cases. Information from databases (commercial applications, bespoke and legacy systems) can be defined in different ways, different languages or different formats. The Semantic model aims to maintain connection between models among the lifecycle given digital continuity to the ontologies.

In the current preliminary approach, the Semantic model has been defined in a very simple way by using a data dictionary, i.e. maintaining a detailed description of all objects in the models. The data dictionary aims to help in the future construction of interfaces for databases. This is modelled by adding common attributes such as 'name' and 'description' to all classes in the above 'scope', 'data' and 'behavior' packages. These attributes are defined in the class `AbstractSemanticAttributes` which is inherited by the rest of classes, as depicted in Figure 6.

Figure 6 presents the dependency between packages of the metamodel. The `AbstractMLMAttributes` class in the 'mlm' (model lifecycle management) package contains a series of common attributes for managing the lifecycle of models.

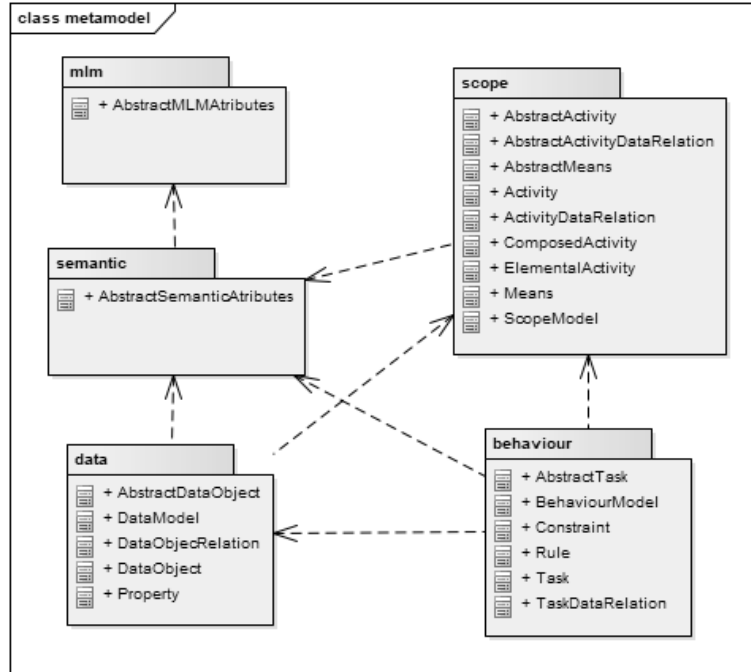


Figure 6 Metamodel packages and their dependencies.

## 5 Discussion and Further Work

Models for Manufacturing methodology aims to be an agnostic methodology successfully applied in different industrial and non-industrial cases. With the definition of the metamodels presented in this paper, MfM defines an independent link with software modellers. Any modelling software covering as minimum MfM metamodels is a candidate.

The metamodel is designed to represent the full set of ontology concepts as simply as possible. In this sense, the simpler the metamodel, the greater the potential modeling software candidates. As said before, IDEF0 diagrams and concept maps have been used to model the Scope and Data ontology models, respectively. Both diagram types fit the metamodel requirements since they have enough modeling elements to cover all the concepts and relationships of the metamodel. The use of these modeling tools have been specified to implement the ontology layer. For example, the Controller type object in IDEF0 diagrams, which is not part of the Scope model, is processed as a DataObject.

To support the lifecycle of the MfM models and to establish a collaborative environment, the authors are currently working in a Model Lifecycle Management (MLM) implementation to demonstrate different use cases and proofs of concept [16,17]. As a consequence of the interfaces with the MLM, authors are now considering decoration and cosmetics within the different software as an option.

Further work will focus on deriving behaviour and simulation from the Scope model to the Behaviour model, address the collaborative use of MfM techniques during the conceptual ontology design, improving the model's lifecycle management, and building a virtual manufacturing environment to simulate MfM prototypes and use cases.

## 6 Acknowledgement

The authors would like to thank to Sevilla University colleagues and AIRBUS and M&M Group colleagues for their support and contribution during the development of this work.

## References

1. O. Holland, "Model-Based Systems Engineering". Loper M. (eds). Modeling and Simulation in the Systems Engineering Life Cycle. Simulation Foundations, Methods and Applications. 2015. Springer, London (2015). doi: 10.1007/978-1-4471-5634-5\_23
2. Mas, F., Rios, J., Menendez, J.L. and Gomez, A., "A process-oriented approach to modeling the conceptual design of aircraft assembly lines", 2013, The International Journal of Advanced Manufacturing Technology, 67 (1-4), pp. 771-784.
3. Mas, F. Oliva, M., Rios, J., Gomez, A. et al., "PLM Based Approach to the Industrialization of Aeronautical Assemblies", 2015, Procedia Engineering 132, pp. 1045-1052.
4. Gomez, A., Rios, J., Mas, F. and Vizan, A., "Method and software application to assist in the conceptual design of aircraft final assembly lines", 2016, Journal of Manufacturing Systems 40, pp. 37-53.
5. Mas F., Racero J., Oliva M., Morales-Palma D. (2018) "A Preliminary Methodological Approach to Models for Manufacturing (MfM)". In: Chiabert P., Bouras A., Noël F., Ríos J. (eds) Product Lifecycle Management to Support Industry 4.0. PLM 2018. IFIP Advances in Information and Communication Technology, vol 540. Springer, Cham. [https://doi.org/10.1007/978-3-030-01614-2\\_25](https://doi.org/10.1007/978-3-030-01614-2_25)
6. Thabet Slimani, "Ontology Development: A Comparing Study on Tools, Languages and Formalisms", Indian Journal of Science and Technology, Vol 8(24), 2015.
7. Lu, Jinzhi, "MBSE EPFL in 15 minutes", <https://youtu.be/kTxaLVdu-pw>, 2021 Last accessed April 2021
8. D. Morales-Palma, F. Mas, Jesús Racero, C. Vallengano, "A preliminary study of Models for Manufacturing (MfM) applied to Incremental Sheet Forming", IFIP International Conference on Product Lifecycle Management PLM18 Proceedings, pp. 284-293 (2018). DOI: 10.1007/978-3-030-01614-2\_26.
9. NIST: "Integration Definition for Function Modeling (IDEF0)". Computer Systems Laboratory of the National Institute of Standards and Technology, December 1993. <http://www.idef.com/wp-content/uploads/2016/02/idef0.pdf>. Last accessed March 2021.
10. F. Mas, J. Racero, M. Oliva, D. Morales-Palma, "Preliminary ontology definition for aerospace assembly lines in Airbus using Models for Manufacturing methodology", Proceedings of 7th International conference on Changeable, Agile, Reconfigurable and Virtual Production CARV18 (2018). DOI: 10.1007/978-3-030-01614-2\_26.

11. R. Arista, F. Mas, “A preliminary model-based approach for Gender analysis of Airbus Research Organization”, Proceeding of the IEEE International Conference on Engineering, Technology and Innovation (2018). DOI: 10.1109/ICE.2018.8436319.
12. Arista, R., Mas, F., Vallellano, C., Morales-Palma, D., Oliva, M. (2020), “Towards Manufacturing Ontologies for Resources Management in the Aerospace Industry”, International Conference on Interoperability for Enterprise Systems and Applications I-ESA 2020, Tarbes, France. In-Press
13. Mas Morate F., Rios J., Menendez J.L, Hernandez J. C., Vizan A. (2009). *Information Model for Assembly Line Design at Conceptual Phase*. Proceedings of 3rd Manufacturing Engineering Society Int. Conference, pp. 288-295.
14. Meta Object Facility, Object Management Group, <https://www.omg.org/spec/MOF/> (accessed April 28<sup>th</sup>, 2021).
15. Gansner, E.R. and North, S.C. (2000), “An open graph visualization system and its applications to software engineering”. *Software Practice Expertise*, 30: 1203-1233.
16. R. Arista, F. Mas, M. Oliva, J. Racero, D. Morales-Palma, (2019) “Framework to support Models for Manufacturing (MfM) methodology”, *IFAC-PapersOnLine*, Volume 52, Issue 13, 2019, Pages 1584-1589, ISSN 2405-8963, <https://doi.org/10.1016/j.ifacol.2019.11.426>.
17. Racero, J., Oliva, M., Eguia, I., Morale-Palma, D., Arista, R., Mas, F., “Using Model Lifecycle Management to support Models for Manufacturing Methodology”. (2021), *Engineering, Technology and Innovation (ICE)*, 2021 International ICE Conference., In Press.