



HAL
open science

A metamodel of an informational structure for modelbased technology roadmapping

Ilya Yuskevich, Andreas Makoto Hein, Kahina Amokrane, Abdelkrim
Doufene, Marija Jankovic

► **To cite this version:**

Ilya Yuskevich, Andreas Makoto Hein, Kahina Amokrane, Abdelkrim Doufene, Marija Jankovic. A metamodel of an informational structure for modelbased technology roadmapping. *Technological Forecasting and Social Change*, 2021, 173, pp.121103. 10.1016/j.techfore.2021.121103 . hal-03330657

HAL Id: hal-03330657

<https://hal.science/hal-03330657>

Submitted on 3 Sep 2021

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.

A metamodel of an informational structure for model-based technology roadmapping

Abstract: Recent contributions in the field of technology roadmapping often aim to apply various numerical models and tools to facilitate the roadmapping process and enrich its outcomes. This trend resulted in the emergence of so-called model-based technology roadmapping. We consider it as the future development of the traditional document-based paradigm. One of the general approaches to support the model-based roadmapping is to develop a roadmap's metamodel that would define it independently from the application context and link it to the existing roadmapping literature. In this paper, we attempt to create such a metamodel by generalizing and formalizing existing document-based roadmaps. We validate our metamodel via reproducing three very different roadmaps from the literature, not included in the set of roadmaps from which the metamodel was created.

Introduction

Large enterprises, such as Lockheed, NASA, Motorola, Philips – just to name a few – have been successfully practicing technology roadmapping (TRM) over the previous 40 years (Kerr and Phaal, 2020). Domain experts consider TRM as an efficient technique for long-term planning of technologies, innovations, programs, products, and services.

TRM is a process that produces a technology roadmap. Phaal et al. (2004) define a technology roadmap as a “time-based chart, comprising a number of layers that typically include both commercial and technological perspectives.” In practice, roadmaps serve the internal and external communication of plans, decisions, and responsibilities and help to focus on long-term goals of high-priority (Albright and Kappel, 2003).

Originating from large technological enterprises' practices, TRM now encompasses various applications and organizational contexts. Phaal et al. (2004) distinguish eight classes of roadmaps by the object of planning, such as products, capabilities, strategies, long-range goals, knowledge assets, programs, processes, and operations of complex system integration. Kim et al. (2018) introduce design roadmapping and provide an example of roadmapping applied to a single-product user-experience-focused startup.

A traditional roadmapping process is typically done through a series of workshops and involves a collaboration of multiple experts from different domains. This process is informationally- and labor-intensive, and leads to several known difficulties. Following the digitization trend, researchers have

proposed numerous approaches to support TRM with computer-based tools. Most of the propositions came from the realm of computational linguistics and related to the automated analysis of large textual databases, such as research papers and patents (Kostoff and Schaller, 2001).

Knoll et al. (2018) consider the roadmapping problem from a systems engineering perspective. Indeed, a large number of roadmap elements and relationships between them provide grounds to perceive a roadmap as a system. Following the model-based paradigm in engineering design, Knoll et al. (2018) have proposed the notion of model-based technology roadmapping (MB-TRM). The distinct feature of this MB-TRM is that it uses executable conceptual design models as building blocks of a roadmap. The fact that these models are stored and processed in a computer helps to create, maintain, and visualize the roadmap. Besides, it allows us to define data processing functions within the scope of a digital roadmap that would perform operations useful for decision-makers (e.g., summarizing, verifying, assessing, extrapolating). This MB-TRM was initially proposed for a specific organizational context – a large international corporation that produces complex technical systems and systems-of-systems (Knoll et al., 2018), and no general model for MB-TRM seems to exist. Such a generalized MB-TRM could be applied to all types of contexts – from a startup to a space agency.

This paper proposes such a general model for MB-TRM, which can be used to model domain-specific roadmaps. We call such a general model a metamodel. According to Mellor (2004): “A metamodel is a model of a modeling language. It defines the structure, semantics, and constraints for a family of models.” A

model, according to Mellor (2004), is “a simplification of something <..> It consists of sets of elements that describe some physical, abstract, or hypothetical reality.”

The practical aim of the proposed metamodel is to support the development of data models for domain-specific roadmapping software.

Hence, in view of MB-TRM development, we identify the following research questions:

- RQ1. Can we propose a metamodel compatible with existing definitions of a technology roadmap?
- RQ2. Can we demonstrate that our proposed metamodel is instantiable in various contexts?
- RQ3. Can we propose a metamodel aligned with the principles of MB-TRM?
- RQ4. Can we demonstrate that our metamodel is usable, i.e., internally consistent, concise, and built following best practices?

The rest of the paper is structured as follows. In section two, we discuss the related literature. In section three, we describe our research methodology – the processes of metamodel construction and validation. The results are presented in section four. The final sections are for discussion, future work, and conclusions.

Related literature

In the following, we provide a detailed review of the literature on TRM assisted with digital tools. Then, we discuss the difficulties of current TRM practice to support the arguments for MB-TRM. Finally, we review the actual papers on MB-TRM and highlight the defining principles of the approach we present in the following sections.

Computer-aided roadmapping

Kostoff and Schaller (2001) were apparently the first who classified the roadmapping process into expert-based, computer-based, and hybrid. They define the expert-based roadmapping as an approach in which “a team(s) of experts is convened to identify and develop attributes for the nodes and links of the roadmap,” and computer-based as an approach where “large textual databases that describe science, technology, engineering, and end products are subject to computer analyses.” Kostoff and Schaller (2001) acknowledge that a fully automated process of roadmapping is not feasible at the moment. They, therefore, suggest a hybrid procedure where computer assistance complements the process of expert negotiation.

Zhang et al. (2016) contribute to the hybrid TRM by proposing a TRM framework for technical intelligence, including text mining, bibliometrics, and several qualitative expert-based approaches. The interesting takeaway in the context of our work is that Zhang et al. (2016) designed a structure of a hybrid roadmap. This structure, or, we may say, syntax consists of labels L , implications I , time T , objects $O(L,I,T)$, and the relationships between objects $R(O1,O2)$. It serves then not only as a syntactic pattern for text mining but also as a description language for a roadmap.

Other examples of TRM assisted by text-mining could be found in Kajikawa et al. (2010) and Miao et al. (2020).

Document-based TRM difficulties and foundations of MB-TRM

Kim et al. (2018, 2016) identified several challenges in current roadmapping practices through a series of semi-structured interviews:

- Even though one of the main functions of roadmapping is the communication of plans, there is still a conflict, especially when the communication happens between multidisciplinary teams;
- TRM lacks a feedback loop from final users of a product, so the focus on creating the value for a customer is not followed;
- Lack of flexibility of a roadmap in the fast-changing environment; necessity to keep roadmap ‘alive’ leads to a request of an agile and iterative process;
- Inability to predict the future, hence often the “plan is not followed.”

McMillan (2003) brought up several lessons learned from his practical experience in Rockwell Automation:

- Senior management must demonstrate a commitment to the process;
- Common definitions should be established for elements of a roadmap, but the process should be flexible enough to re-define these elements;
- Business units should at some point “endorse and own the process” of roadmapping;
- Some business units perceive roadmapping as a “mandated exercise to be completed on the command of management” because this process’s benefits were not well understood.

Gradini et al. (2019) suggest that some of these difficulties may be resolved by a major shift from a

document-based to a model-based technology roadmapping.

MB-TRM implies a formal architecture of a roadmap. Phaal and Muller (2009) laid the foundation for it. They divided a roadmap architecture into two realms: informational structure and graphical style, which corresponds to the famed Model-View-Controller pattern from Computer Science. The informational structure of a roadmap essentially consists of hierarchic elements (technologies, resources, components, products, markets, etc.) and relationships between them. Visual style (Kerr and Phaal, 2015) typically reflects an internal structure through a layered chart of a directed graph.

Gradini et al. (2019) and Knoll et al. (2018) present a different view in their MB-TRM framework. They define an architecture of separate roadmap elements (e.g., systems) that must be instantiated, parameterized, and linked to other elements (e.g., technologies) during collaborative design sessions. Linked elements thus turn to nodes of a centralized graph (i.e., a roadmap). Time and layers are implied but do not exist as explicit dimensions.

Several contributions introduce processes that may be integrated into MB-TRM software as useful functions. Geum et al. (2014) incorporated system dynamics simulations to 'linked grid roadmap' to analyze different scenarios of a new service launch. Equations that establish relationships between external influencing factors and service key performances are examples of transfer functions. Lu and You (2018) proposed metrics to assess the importance of nodes (in their case, weapon systems) for technology infusion based on the topology of a graph (an example of a graph traversal function). Yuskevich et al. (2018) proposed an algorithm to assess and forecast the technology rate of change based on historical data (an example of a trend extrapolation function).

First of all, our literature review results highlight the problems of existing TRM practices, which provide the necessary motivation to develop MB-TRM. Second, we have identified a number of approaches to quantitative analysis of the information contained in model-based roadmaps. The third and the most important finding of our literature review is that previous publications on MB-TRM considered domain-specific roadmaps. Therefore, this research aims to address the previously discussed gap by proposing a domain-independent metamodel. The purpose of this metamodel is two-fold. In the theoretical realm, it contributes to a definition of a field by raising the level of abstraction. And from a

practical point of view, it provides a starting point for developers of roadmapping software.

Based on the results of our review, we argue that there are two determining features of MB-TRM. The first is the formal syntax, which, along with the agreed semantics, determines elements of a roadmap and relationships between them. The second is a centralized database allowing for collaborative work and compatible with defined syntax and semantics. Availability of functions, different from basic create, read, update, and delete (CRUD), is beneficial but optional.

The latter differentiates MB-TRM also from what Kostoff and Schaller (2001) address as a computer-based approach, as the definition prescribes the presence of complex computer analysis. MB-TRM can be considered as a special case of hybrid TRM, with a mandatory requirement of formal syntax and defined semantics.

The fact that elements of a model-based roadmap are created, stored, and maintained numerically facilitates all processes and brings flexibility unachievable for document-based roadmapping. We would emphasize here that simple digitization of documents or transferring them to spreadsheets is not an objective. By documents, we mean both paper and digital artifacts.

Research methodology

To validate the proposed metamodel, we will use an ad-hoc procedure inspired by existing methodologies from Model-Driven Engineering (MDE) and Semantic Web domains. We will use a modern language workbench (a software for Domain-Specific Languages (DSLs) design). Our success criterion is to satisfy the objectives of the validation formulated in the following.

Objectives of validation

To address metamodel development and validation, we propose to adapt the approach of metaontology O2 of Gangemi et al. (2006) (Figure 1). On this level, a metamodel expresses the intended conceptualization of some domain by rational agent(s). The intended conceptualization is influenced by pragmatic objectives and semantic space, which, generally speaking, may differ between rational agents. Typically, metamodeling then becomes an exercise in finding a social consensus on the "right" metamodel.

Our objective here is not to find a metamodel for TRM based on social consensus between rational agents.

Instead, our objective is to construct a sufficiently general metamodel for modeling a wide range of existing roadmaps.

More specifically, we will focus on the verification of the following statements:

- S1. DSLs instantiated from our metamodel correctly express several publicly available technology roadmaps.
- S2. The roadmaps that we selected for validation belong to significantly different contexts.
- S3. Our metamodel can be used as a prototype of an MB-TRM software system, supporting CRUD-operations and graph-traversal functions.

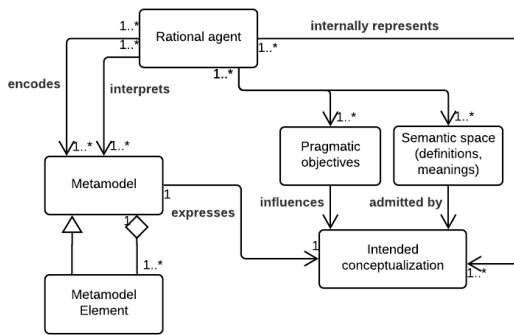


Figure 1 Conceptual model of metamodeling (originated in Gangemi et al. (2006))

If S1 holds, then our metamodel is internally consistent (RQ4) and compatible with definitions of a technology roadmap (RQ1). S2 asserts that our metamodel constitutes a useful generalization for a practice of TRM (RQ2). S3 is a formalized version of RQ3.

Validating the Metamodel

To design our approach to metamodel validation correctly, we propose to review the MDE literature. MDE is a branch of computer science, promoting the further rise of the abstraction level in software, i.e., from general-purpose programming languages to platform-independent models.

We propose to base ourselves on the example-driven metamodel validation procedure from López-Fernández et al. (2015). López-Fernández et al. (2014) highlight two broad classes of validation objectives: “are we building the right metamodel?” and “are we building the metamodel right?” The former aims to validate the correspondence of metamodel concepts to definitions and meanings accepted in the domain of

interest. The latter’s goal is to verify that the metamodel is consistent and meets the standards and best practices of MDE, ensuring that a metamodel is reusable.

López-Fernández et al. (2015, 2014) proposed an example-driven procedure of metamodel construction and validation. They also developed a tool that enables semi-automatic metamodel construction from several model fragments created by domain experts. The resulting metamodel can then be validated by testing it against valid and invalid models.

Since metamodels are built for practical purposes, they can also be validated by direct instantiation for a specific use case. As an example, García-Holgado and García-Peñalvo (2017) performed such validation for learning ecosystems.

The process of metamodel construction and validation is shown in Fig. 2. The process starts with a review of published document-based technology roadmaps. During the review, a metamodel creator (first author of this article) identifies common elements among various documents, conceptualizes and classifies them. The first version of a metamodel (level M2 of MOF architecture) is a product of this step. In the second step, we select several roadmaps with significantly different structures and application domains. Then, for each of these roadmaps, the application domain-specific model (M1) is instantiated from a metamodel (M2). In the third step, each roadmap (user data M0) is instantiated from a corresponding application domain-specific model (M1). Produced data should be identical to the data contained in the selected documents. If we failed to go through all the steps, we can modify a metamodel and repeat the instantiation. The process needs to be repeated iteratively until a metamodel is mature enough.

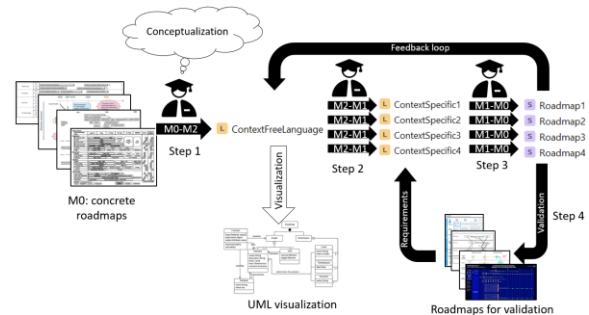


Figure 2 The iterative process of metamodel construction and validation

Software tool for the metamodel validation

To support the procedure, we selected a tool that assists in the creation and composition of DSLs (a language workbench). We have chosen JetBrains MPS – a modern language workbench that allows us to define an abstract and concrete syntax (structure and editor in MPS-terms) of a DSL and write code in a new language (create models) with projectional editing. The projectional editor allows direct manipulation of an abstract syntax tree (AST) of a model. The latter is important for metamodel validation – any expression incompatible with the syntax defined for DSL cannot modify an AST of a model. In this way, we automatically ensure the internal consistency of a metamodel.

[Note: for more information on MPS, visit (“Glossary - Help | MPS,” 2020).]

Language composition is presented in Fig 3. Both the application domain-independent language (metamodel M2) and the application domain-specific language (model M1) are written in the Base Language – a Java-like language for DSL definition in MPS. The operation of metamodel instantiation is equivalent to language extension in JetBrains MPS. User data (in our case, a roadmap) is defined using the concrete and abstract syntax of application domain-specific language (M1).

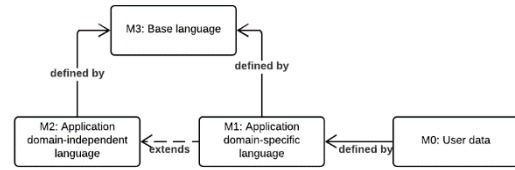


Figure 3 Language composition structure

We chose a language workbench in which an AST is represented in textual form by default. For the sake of clarity, we will present an abstract syntax of the DSL in the form of a UML class diagram (for concepts) augmented with OCL expressions (for constraints); and Mo roadmaps in a textual form aligned with concrete syntax of a corresponding DSL.

Results

Metamodel construction

We conducted an extensive literature review to identify common elements, concepts and establish the classification (see Table 1).

This analysis supports the conceptual representation of a technology roadmap as a layered structure with a hierarchy of elements and attributes, connected with causal links and composing a directed graph. This view is well aligned with the architectural framework of

Table 1 Document-based technology roadmap commonalities

Reference	Context	Levels	Elements	Attributes	Links
(Albers et al., 2017)	Strategic roadmap in automotive domain	Market, Product, Technology, Resources	Skill, capability, competency, need, product, technology	-	evolves to, satisfies need, requires resource
(Albright and Kappel, 2003)	Product and technology roadmap in telecom. domain	Market, Product, Technology, Risk	Product/market driver, market trend, market segment, product, technology, capability, event, risk	Market growth, market share, experience curve, priority, price, risk consequence, competitive position	requires, targets market niche, evolves to
(Scalice et al., 2015)	TRM for modular products	Product, market, module	Product, platform, market, technology, module	Differentiation feature, lifecycle stage, lifestyle, benefits, driver importance weight, ranking	-
(Toro-Jarrín et al., 2016)	Business Model Canvas and TRM integrated	Market, product, technology	Political strategy, macroeconomic forces, key trends, market and industry forces, new entrants, substitutes, technology, regulatory, cost trends	Economy growth, employment rate, market volume, features, product size	-
(Al-Ali and Phaal, 2019)	TRM for digital transformation	Digital opportunity, digital initiative, resources	Disruptive/Radical opportunities, internal efficiencies, MVP, legacy system, resources, digital technology	-	Replace, interface, migrate
(Zhang et al., 2016)	TRM for technical intelligence in photovoltaic industry	Material, method, device	Problem, solution, component, implication	Efficiency, stability, losses, etc.	Solve, relate, improve, upgrade

Phaal and Muller (2009). Therefore, we took this structure as a foundation of our metamodel.

The identified concepts represent the basis of the proposed MB-TRM (see Figure 4). The roadmap informational structure is a directed graph consisting of elements and links. Elements of a roadmap exist in the three-dimensional space defined by time, level, and scenario dimensions. Elements may be classified (modeled as “*Element* references to *ElementClass*”); classes can be organized into hierarchies to form taxonomies (each *ElementClass* can be a parent of other *ElementClass*). Each *Element* can have several attributes (*Attribute*) that take quantitative or qualitative values (*AttributeValue*). In some applications, a corresponding element class defines the set of attributes that *Element* can exhibit.

We believe that the majority of document-based TRM can be derived from this structure. To realize the full potential of the MB-TRM approach, we added functions and models. When the set of quantitative or qualitative attributes is not sufficient to characterize elements, we may employ models, e.g., structural, parametric, or stochastic. For their part, functions operate on the entire directed graph, execute models, perform calculations of aggregated metrics, and various checks. Functions and models are the exclusive features of MB-TRM and cannot be derived from document-based roadmaps.

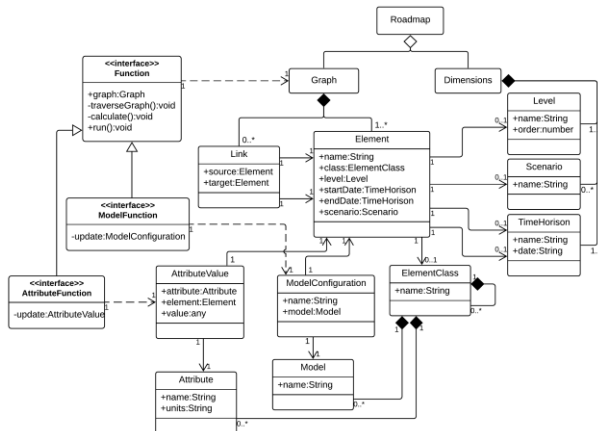


Figure 4 Metamodel for MB-TRM

The geometric interpretation of the metamodel is shown in Fig. 5 and 6. Points represent *Elements*, i.e., markets, systems, and technologies. Horizontal causal links represent the evolution of elements. Cross-level links model interdependencies.

Metamodel validation

To validate the metamodel, we additionally selected three roadmaps (see table 2) to check whether these roadmaps can be modeled based on the metamodel. We created an abstract and concrete syntax of a DSL and user data layer, precisely representing targeted roadmaps for each of them.

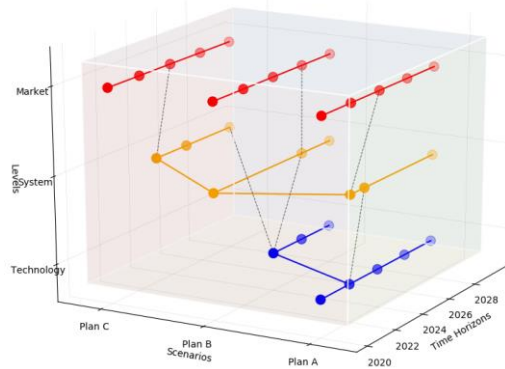


Figure 5 Geometric interpretation

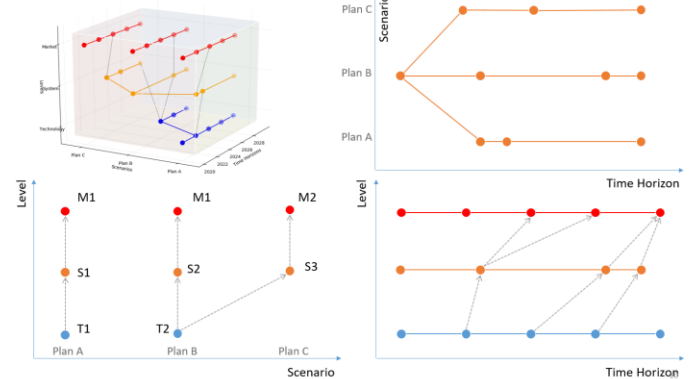


Figure 6 Geometric interpretation – cross-sections

Table 2 – Roadmaps selected for validation

Reference	Context	Structural challenge
<p>“2015 NASA Technology Roadmaps (Archive),” 2015)</p>	<p>Technology roadmap of a space agency. Horizon of planning – 30 years.</p>	<p>Relatively complex taxonomy of elements. An enormous amount of data – more than 2000 pages.</p>
<p>(Kim et al., 2018)</p>	<p>UX-centered product, design, and technology roadmap of a single-product startup. Horizon of planning – 3 years.</p>	<p>Simple structure, but data represent subtle and qualitative concepts – user needs and experiences.</p>

(Geum et al., 2014)	Emerging car-sharing business, heavily dependent on the turbulent external environment. Horizon of planning – 30 months.	Alternative realities (scenarios). The abundance of KPIs. A dynamic model of KPIs complements a roadmap.
----------------------------	--------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------

The instantiation of a metamodel for the NASA roadmap is shown in figure 7. *TechnologyArea*, *TechnologyFamily*, *TechnologyClass*, *Technology*, and *Capability* are instances of *ElementClass*. They represent NASA’s technology taxonomy. *Technology* and *Capability* define the set of attributes that concrete elements of a roadmap – *TechnologyCurrent*, *TechnologyTarget*, *CapabilityCurrent*, and *CapabilityTarget* – can take. Another specificity of NASA roadmap is that these elements do not refer to the time dimension directly (note cardinality 0..1 between *Element* and *TimeHorizon* in the metamodel). Instead, they are connected with multiple *MissionNeeds* by *Enables* and *Enhances* links. This is how different due dates for the same performance target are modeled.

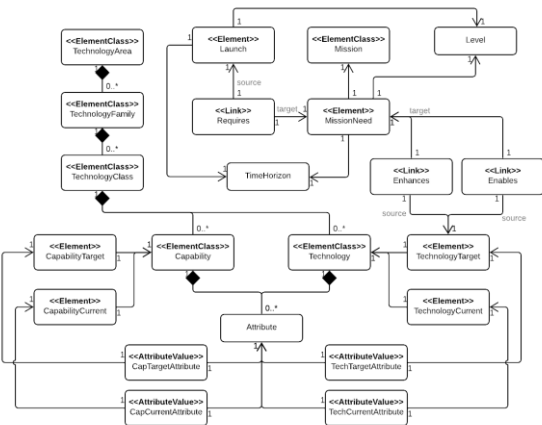


Figure 7 Instantiation of the metamodel for NASA TRM

Concrete syntax created for this usecase is given in Figure 8-10. For obvious reasons, we modeled only a small portion of the NASA roadmap. However, we can safely generalize our results on the entire document, thanks to the NASA roadmap's formal structure.

The ‘code completion’ (e.g., Design, Product and Technology levels on Figure 12) and the ‘intention actions’ (e.g., ‘Technology Parameter’ on Figure 9) dialogs are shown on these and the following figures. This is done to illustrate that in MPS we do not write code to be parsed and compiled but directly manipulate AST, which is the major premise for the validation:

strict compliance of every written expression to defined abstract syntax is ensured.

```

Technology Area 1 Launch Propulsion Systems
  Technology Family 1.1 Solid Rocket Propulsion Systems
    Technology Class 1.1.1 Propellants
    Technology Class 1.1.2 Case Material
    Technology Class 1.1.3 Nozzle System
    Technology Class 1.1.4 Hybrid Rocket
    Technology Class 1.1.5 Fundamental Solid Propulsion Technologie
    Technology Class 1.1.6 Integrated Solid Motor Systems
    Technology Class 1.1.7 Liner and Insulation
  Technology Family 1.2 Liquid Rocket Propulsion Systems
  ...
  Technology Family 1.3 Air-Breathing Propulsion Systems
  ...
  Technology Family 1.4 Ancillary Propulsion Systems
  ...
  Technology Family 1.5 Unconventional and Other Propulsion Systems
  ...
  Technology Family 1.6 Balloon Systems
  ...
Technology Area 2 In-Space Propulsion Technologies

```

Figure 8 Concrete syntax for NASA TRM – classifier

```

Insulating/Ablative Sprayable Liner is a Technology
with attributes:
  Current case material weigh units: %
Technology Description:
  This is a liner capable of insulative properties similar to fiber-filled nitrile butadiene rubi
Technology Challenge: Producing an acceptable sprayable liner is a technical challenge.
Technology Development Dependent Upon Basic Research: false
Technology Class: 1.1.7 Liner and Insulation
Classifier: 1.1.7.2
Technology State of the Art:
  This technology is under development in Space Launch System (SLS) Booster project. Current issues
TRL: 5
define attribute values:
  [redacted]
Tech Technology Parameter (AttributeValue in M2_AbstractLanguage) Lation thickness
TRL: 9
define attribute values:
  Current case material weigh 25 %
A sprayable liner is a Capability
with attributes:
  Current case material weigh units: %
Capability Description:
  The sprayable liner can maintain the solid rocket booster motor case internal temperature below

```

Figure 9 Concrete syntax for NASA TRM – technology editor

```

DRMS Asteroid Redirect - Robotic spacecraft is a Mission

create Launch Launch_DRMS
( start date: 2018 level: Human exploration and Operations scenario: <no scenario> )

create Need for DRMS Asteroid Redirect - Robotic spacecraft Need_DRMS
( epoch: 2015 level: Human exploration and Operations scenario: <no scenario> )

1.1.7.1 Performance Goal enables Need_DRMS
1.1.7.2 Performance Goal enables Need_DRMS

Press [Enter] to define enhancer links

Launch_DRMS requires Need_DRMS

```

Figure 10 Concrete syntax for NASA TRM – roadmap editor

The instantiation of the metamodel for the design roadmap is shown on Figure 11. The structure consists of 9 *Element* instances and 3 *Link* instances. The meaning of each *Element* is characterized by a textual description.

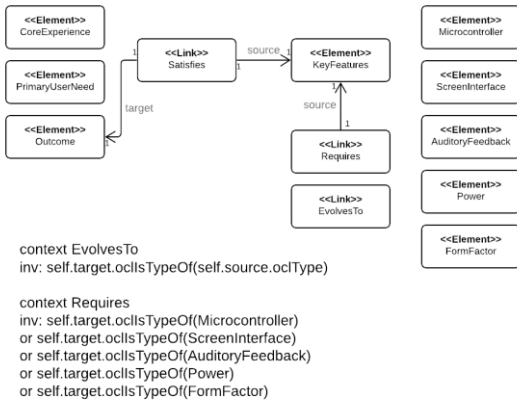


Figure 11 Instantiation of the metamodel for Design roadmap

Some of the classes are not connected (Figure 11) to others by association links. It does not mean that they are unrelated to other parts of the model. The model of level M1 inherits the relationships defined by the metamodel (level M2). These general relationships may be additionally specified on the level M1 with OCL-expressions. For instance, a target and a source of the link *EvolvesTo* should be of the same concept, as it signifies the evolution of needs/products/technologies. Furthermore, link *Requires* connects *KeyFeatures* and any *Element* that models technology. At the same time, link *Satisfies* can go strictly from *KeyFeatures* to *Outcomes*. Fragments of a concrete syntax for the design roadmap are given in Figures 12-13.

```

Core experiences:
create Core Experience Core experience 1
  ( start date: 2012 level: Design Roadmap scenario: <no scenario> )
  description: Make learning about diabetes fun through game play
create Core Experience Core experience 2
  ( start date: 2013 level: Design Roadmap scenario: <no scenario> )
  description: Learn about diabetes through a combo of free and guided play
create Core Experience Core experience 3
  ( start date: 2015 level: Design Roadmap scenario: <no scenario> )
  description: Learn about c
    * Design Roadmap ^levels (M1_2_DesignRoadmap.null)
    * Product Generation ^levels (M1_2_DesignRoadmap.null)
    * Technology Development ^levels (M1_2_DesignRoadmap.null)
Primary User Needs:
create Primary User Need User Need 1
  ( start date: 2012 level: Design Roadmap scenario: <no scenario> )
  description: Learning about diabetes should be fun and easy
create Primary User Need User need 2

```

Figure 12 Concrete syntax for Design roadmap – Elements

```

Form-factor:
create Form Factor A Bear shell-1 (Teddy Bear)
  ( start date: 2012 level: Technology Development scenario: <no scenario> )
  description:
  create Form Factor A Bear shell-2 (Teddy Bear)
  ( start date: 2013 level: Technology Development scenario: <no scenario> )
  description: easy to clean Asthma-friendly fabric

Links:
Generation 1 satisfies Outcome 1
Generation 1 requires Arduino
Generation 2 satisfies Co
Generation 3 requires Co
Generation 2 requires Co
Generation 1 evolves to
Generation 2 evolves to

```

Figure 13 Concrete syntax for Design roadmap – Links

Finally, the instantiation of a metamodel for the emerging car-sharing business (Geum et al., 2014) is shown in figure 14. This is an example of a roadmap modeled in a three-dimensional space. Casual links connect *Business Drivers*, *Products*, *Services*, and *Technologies* either within the same scenario, or produce scenario branching by connecting two *Elements* placed in distinct scenario planes. In the original paper of Geum et al. (2014), such a structure was used to launch a dynamic simulation to calculate business KPI's for pessimistic/neutral/optimistic scenarios. The interdependencies between KPI's and elements of a roadmap were defined by 'causal and loop diagram.' Our metamodel is compatible with such a problem setting: a mutual influence between roadmap elements and KPI's is established by the *Influence* link, the transfer function of each KPI – in the *KpiModel*, and initial values – in the *KpiModelValues*. The simulation is executed by *DynamicSimulation* function that traverses graphs and iteratively updates *KpiModelValues*.

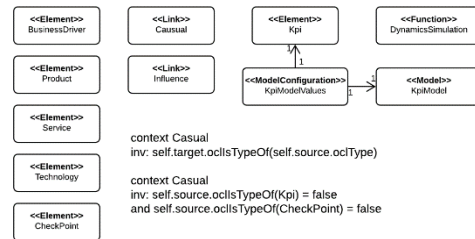


Figure 14 Instantiation of the metamodel for emerging car-sharing business roadmap

The examples of a concrete syntax for this usecase are in figures 15-17.

```

Add Product Delete
create Product Development of environmental-friendly car
  ( start date: Jan-2010 end date: Mar-2011 level: Product scenario: Neutral )
create Product Automatic payment function
  ( start date: Jul-2011 end date: Jan-2012 level: Product scenario: Neutral )
create Product Car for ergonomics
  ( start date: Jul-2010 end date: May-2011 level: Product scenario: Neutral )
create Product Car for multiple customers
  ( start date: Jan-2010 end date: Jul-2010 level: Product scenario: Neutral )
Add Service Delete
create Service e-Tracking service
  ( start date: Jul-2011 end date: Apr-2012 level:
    * Neutral ^scenarios (M1_4_ScenarioPlanning)
    * Optimistic ^scenarios (M1_4_ScenarioPlanning)
    * Pessimistic ^scenarios (M1_4_ScenarioPlanning)
  )
create Service Automatic return service
  ( start date: May-2011 end date: Nov-2011 level: Service scenario: Neutral )
create Service Public Transp. notification

```

Figure 15 Elements of a roadmap definition

```

Policy for Green Growth Korea causes Law for sustainable traffic development [ Neutral ->Optimistic ]
Policy for Green Growth Korea [ Neutral ] causes Policy for reducing traffic volume [ Neutral ]
Social movement toward sustainability causes Law for sustainable traffic development [ Neutral ->Optimistic ]
Social movement toward sustainability [ Neutral ] causes Policy for reducing traffic volume [ Neutral ]
Policy for reducing traffic volume [ Optimistic ] causes Policy for improving parking lot [ Optimistic ]
Law for sustainable traffic development [ Optimistic ] causes Law for tax deduction for car-sharing [ Optimistic ]
Law for tax deduction for car-sharing ^elements (M1_4_ScenarioPlanning.sandbox.Optimistic )

```

Figure 16 – Causal links definition

```

create Key Performance Indicator Use of Car-sharing service model: F1
create Key Performance Indicator Total revenue model: F2
create Key Performance Indicator Energy Consumption model: F3
Social movement toward sustainability influence Use of Car-sharing service
Level Of Company Motivation influence Use of Car-sharing service
Use of Car-sharing service influence Improvement of traffic environment
Use of Car-sharing service influence Total revenue
Use of Car-sharing service
Use of Car-sharing service
Use of Car-sharing service
  (M) Public Transp. notification
  (M) Reduction of Environmental burden
  (M) Seat customization service for multi-users
  (M) Service for individuals
  (M) Social movement toward sustainability
  (M) Total revenue
  (M) Use of Car-sharing service
  (M) Weak Policy for Green Growth Korea

```

Figure 17 – Dynamic model set up

Discussion

During the validation, we were able to assert all three of our statements (S1-S3).

S1: we successfully created three DSLs that could precisely represent the information contained in the three publicly available roadmaps. By doing so, the internal consistency of our metamodel is validated. Second, we demonstrated the ability of the proposed abstract syntax to describe real roadmaps correctly.

S2: we deliberately selected extremely different roadmaps: the roadmap of the space agency versus the roadmaps of the startups; the descriptive roadmaps used for external and internal communication versus the executable roadmap used for simulation. By illustrating different case studies, we aimed to show that our metamodel applies to a wide range of contexts.

S3: our metamodel complies with MB-TRM principles and introduces concepts that extend conventional technology roadmaps: element models and graph traversal functions.

However, several research limitations can be identified and discussed. Being not a natural phenomenon but a kind of collective pragmatic knowledge, TRM's notion can be subject to different interpretations. Therefore, we have opted for the analysis of the scientific literature instead of seeking expert opinion. Based on that, we are convinced that our metamodel corresponds to what most academics mean by this notion. At the same time, we are aware of the possibility of alternative views.

There is a large variety of TRMs, which additionally integrate other methods of strategic management. We

References

- 2015 NASA Technology Roadmaps (Archive) [WWW Document], 2015. . NASA. URL <http://www.nasa.gov/offices/oct/home/roadmaps/index.html> (accessed 7.24.20).
- Al-Ali, A.G., Phaal, R., 2019. Design Sprints for Roadmapping an Agile Digital Transformation, in: 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC). Presented at the 2019

do not claim to have presented an all-encompassing metamodel. However, we do believe that the proposed metamodel is applicable to a broad set of TRMs.

Finally, focusing on the informational properties, we have entirely ignored the part of the architecture responsible for the visualization of all this information. It is a notable omission as roadmapping is essentially a visual approach that needs to be addressed in the future.

Conclusions and future work

In this paper, we propose a metamodel for the MB-TRM, a new concept in the field of technology management and strategic planning. We demonstrate that our metamodel can be instantiated to three roadmaps, sufficiently different in terms of the context and primary objectives. The latter indicates that our metamodel can be used in practice as a generalization for the development of new software tools.

Moreover, our proposal helps to bridge the gap between the technology roadmapping domain and the MDE domain. Furthermore, we propose an original method to validate a metamodel using a modern language workbench. This can be considered as a methodological contribution and might be beneficial for researchers dealing with metamodeling in the field of systems engineering and design science.

We plan to further extend the metamodeling to product planning for user-centered design in the automotive industry in future work.

Acknowledgments

The first author would like to thank Prof Alessandro Golkar, Prof Olivier de Weck, Prof Clement Fortin, Prof Rob Vingerhoeds, and Dr. Dominic Knoll for the insightful discussions and supervision that led to the idea of this paper.

- IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), IEEE, Valbonne Sophia-Antipolis, France, pp. 1–6. <https://doi.org/10.1109/ICE.2019.8792597>
- Albers, A., Krämer, L., Arslan, M., 2017. Roadmap-based Methodology for the Forecasting of Competences within Automotive Product Development, in: Sanchez, R., Heene, A., Polat, S., Asan, U. (Eds.), *Mid-Range Management Theory: Competence Perspectives on Modularity and Dynamic Capabilities*, Research in Competence-Based Management. Emerald Publishing Limited, pp. 109–130. <https://doi.org/10.1108/S1744-21172017000008005>
- Albright, R.E., Kappel, T.A., 2003. Roadmapping In the Corporation. *Research-Technology Management* 46, 31–40. <https://doi.org/10.1080/08956308.2003.11671552>
- Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J., 2006. Modelling Ontology Evaluation and Validation, in: Sure, Y., Domingue, J. (Eds.), *The Semantic Web: Research and Applications*, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 140–154. https://doi.org/10.1007/11762256_13
- García-Holgado, A., García-Peñalvo, F.J., 2017. Preliminary validation of the metamodel for developing learning ecosystems, in: *Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality, TEEM 2017*. Association for Computing Machinery, Cádiz, Spain, pp. 1–9. <https://doi.org/10.1145/3144826.3145439>
- Geum, Y., Lee, S., Park, Y., 2014. Combining technology roadmap and system dynamics simulation to support scenario-planning: A case of car-sharing service. *Computers & Industrial Engineering* 71, 37–49. <https://doi.org/10.1016/j.cie.2014.02.007>
- Glossary - Help | MPS [WWW Document], 2020. . MPS 2020.1. URL <https://www.jetbrains.com/help/mps/glossary.html> (accessed 7.23.20).
- Gradini, R., Salgueiro, S., Budideti, J., Golkar, A., 2019. Model-based technology roadmapping: potential and challenges ahead, in: *Proceedings of the 2019 Summer Simulation Conference, SummerSim '19*. Society for Computer Simulation International, Berlin, Germany, pp. 1–11.
- Kajikawa, Y., Takeda, Y., Matsushima, K., 2010. Computer-assisted roadmapping: a case study in energy research. *Foresight* 12, 4–15. <https://doi.org/10.1108/14636681011035726>
- Kerr, C., Phaal, R., 2020. Technology roadmapping: Industrial roots, forgotten history and unknown origins. *Technological Forecasting and Social Change* 155, 119967. <https://doi.org/10.1016/j.techfore.2020.119967>
- Kerr, C., Phaal, R., 2015. Visualizing Roadmaps: A Design-Driven Approach. *Research-Technology Management* 58, 45–54. <https://doi.org/10.5437/08956308X5804253>
- Kim, E., Beckman, S.L., Agogino, A., 2018. Design Roadmapping in an Uncertain World: Implementing a Customer-Experience-Focused Strategy: California Management Review. <https://doi.org/10.1177/0008125618796489>
- Kim, E., Chung, J., Beckman, S., Agogino, A.M., 2016. Design Roadmapping: A Framework and Case Study on Planning Development of High-Tech Products in Silicon Valley. *J. Mech. Des* 138. <https://doi.org/10.1115/1.4034221>
- Knoll, D., Golkar, A., de Weck, O., 2018. A concurrent design approach for model-based technology roadmapping, in: *2018 Annual IEEE International Systems Conference (SysCon)*. Presented at the 2018 Annual IEEE International Systems Conference (SysCon), pp. 1–6. <https://doi.org/10.1109/SYSCON.2018.8369527>
- Kostoff, R.N., Schaller, R.R., 2001. Science and technology roadmaps. *IEEE Transactions on Engineering Management* 48, 132–143. <https://doi.org/10.1109/17.922473>
- López-Fernández, J.J., Cuadrado, J.S., Guerra, E., de Lara, J., 2015. Example-driven meta-model development. *Softw Syst Model* 14, 1323–1347. <https://doi.org/10.1007/s10270-013-0392-y>
- López-Fernández, J.J., Guerra, E., de Lara, J., 2014. Metamodel validation and verification with MetaBest, in: *Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering, ASE '14*. Association for Computing Machinery, Vasteras, Sweden, pp. 831–834. <https://doi.org/10.1145/2642937.2648617>
- Lu, H., You, H., 2018. Roadmap Modeling and Assessment Approach for Defense Technology System of Systems. *Applied Sciences* 8, 908. <https://doi.org/10.3390/app8060908>
- McMillan, A., 2003. Roadmapping—Agent of Change. *Research-Technology Management* 46, 40–47. <https://doi.org/10.1080/08956308.2003.11671553>
- Mellor, S.J. (Ed.), 2004. *MDA distilled: principles of model-driven architecture*, Addison-Wesley object technology series. Addison-Wesley, Boston.
- Miao, H., Wang, Y., Li, X., Wu, F., 2020. Integrating Technology-Relationship-Technology Semantic Analysis and Technology Roadmapping Method: A Case of Elderly Smart Wear Technology. *IEEE Transactions on Engineering Management* 1–17. <https://doi.org/10.1109/TEM.2020.2970972>
- Phaal, R., Farrukh, C.J.P., Probert, D.R., 2004. Technology roadmapping—A planning framework for evolution and revolution. *Technological Forecasting and Social Change* 71, 5–26. [https://doi.org/10.1016/S0040-1625\(03\)00072-6](https://doi.org/10.1016/S0040-1625(03)00072-6)
- Phaal, R., Muller, G., 2009. An architectural framework for roadmapping: Towards visual strategy. *Technological Forecasting and Social Change, Knowledge Driven Planning Tools for Emerging and Converging Technologies* 76, 39–49. <https://doi.org/10.1016/j.techfore.2008.03.018>
- Scalice, R.K., da Silva, J.O., Ostetto, J.N., de Paula, G.A., 2015. Modular deployment using TRM and function analysis. *Technological Forecasting and Social Change* 92, 1–11. <https://doi.org/10.1016/j.techfore.2014.10.018>

- Toro-Jarrín, M.A., Ponce-Jaramillo, I.E., Güemes-Castorena, D., 2016. Methodology for the Building Process Integration of Business Model Canvas and Technological Roadmap. *Technological Forecasting and Social Change* 110, 213–225. <https://doi.org/10.1016/j.techfore.2016.01.009>
- Yuskevich, I., Vingerhoeds, R., Golkar, A., 2018. Two-dimensional Pareto frontier forecasting for technology planning and roadmapping, in: 2018 Annual IEEE International Systems Conference (SysCon). Presented at the 2018 Annual IEEE International Systems Conference (SysCon), pp. 1–7. <https://doi.org/10.1109/SYSCON.2018.8369565>
- Zhang, Y., Robinson, D.K.R., Porter, A.L., Zhu, D., Zhang, G., Lu, J., 2016. Technology roadmapping for competitive technical intelligence. *Technological Forecasting and Social Change* 110, 175–186. <https://doi.org/10.1016/j.techfore.2015.11.029>