



Return on Modelling Effort in an EA context

# Return on Modelling Effort in an EA context

Henderik A. Proper



# Background



Radboud University Nijmegen



# Background

---

My general interest:

- Foundations and applications of domain modelling
- *Modelling practices*

Key application field to *enable* with model-driven systems:

- Modelling practices in an enterprise context
- *EA, ISE, EE, BPM, ...*

# Acknowledgements

---

Martin Op 't Land

Maarten Waage

Giancarlo Guizzardi

Isadora Valle

Jan Schoonderbeek



## Understanding the Variety of Domain Models: Views, Programs, Animations, and Other Models

Henderik A. Proper<sup>1</sup> · Giancarlo Guizzardi<sup>2</sup>

Received: 1 July 2023 / Accepted: 22 July 2024 / Published online: 9 September 2024  
© The Author(s) 2024

### Abstract

Humanity has long since used models, in different shapes and forms, to understand, redesign, communicate about, and shape, the world around us; including many different social, economic, biological, chemical, physical, and digital aspects. This has resulted in a wide range of *modeling practices*. When the models as used in such *modeling practices* have a key role to play in the activities in which these practices are ‘embedded’, the need emerges to consider the effectiveness and efficiency of such processes, and speak about *modeling capabilities*. In the latter situation, it also becomes relevant to develop a thorough understanding of the artifacts involved in modeling practices/capabilities. One context in which models play (an increasingly) important role is *model-driven systems development*, including software engineering, information systems engineering, business process engineering, enterprise engineering, and enterprise architecture management. In such a context, we come across a rich variety of modeling related artifacts, such as views, diagrams, programs, animations, specifications, etc. In this paper, which is actually part of an ongoing ‘journey’ in which we aim to gain deeper insights into the foundations of modeling, we take a fundamental look at the variety of modeling related artifacts as used in the context of model-driven (systems) development, while also presenting an associated *framework for understanding*, synthesizing the insights we obtained during the ‘journey’ so-far. In doing so, we will also argue that the aforementioned artifacts are actually specific kinds of models, albeit for fundamentally different purposes. The provided *framework for understanding* involves definitions of *domain model*, the *Return on Modeling Effort* (RoME), the *conceptual fidelity* of domain models, as well as *views* as a mechanism to manage the complexity of domain models.

**Keywords** Domain modeling · Return on modeling effort · Conceptual fidelity · Views

### Introduction

Whenever we are confronted with complex phenomena, such as the processes we observe in nature, the construction of buildings, the design of information systems, etc., we tend to ‘work with’ an *abstraction* (in our mind) of the actual phenomenon; zooming in on those ‘properties’ of the phenomenon that matter to us, while filtering out all the properties that are not germane to the goals at hand. When we externalize this *abstraction* in terms of some artifact,

then to us, as an individual, this artifact is a model of the observed phenomenon. For such a model to be ‘recognized’ as a model, the artifact needs to be a human understandable representation of said abstraction.

The latter view on models is rooted on (among others) the definitions of *model* as provided by e.g. Apostel [1] and Stachowiak [73]. Here, it is important to already acknowledge the fact that the notion of model as put forward by these scholars does not stipulate any a priori constraints on e.g. the level of completeness, precision, or even (mathematical) formality of the model. Depending on the situation, and the purpose of a model, different requirements can be put on a model [6, 39], including requirements regarding, e.g., its completeness, precision, or formality.

In line with the general notion of *domain* as e.g. provided in the Webster dictionary [79] “a sphere [...] of knowledge, influence, or activity”, we prefer to refer to these models as *domain models*, where the term *domain* refers to the

✉ Henderik A. Proper  
henderik.proper@tuwien.ac.at

✉ Giancarlo Guizzardi  
g.guizzardi@utwente.nl

<sup>1</sup> TU Wien, Vienna, Austria

<sup>2</sup> University of Twente, Enschede, The Netherlands



## Toward an ontology for EA modeling and EA model quality

Jan A. H. Schoonderbeek<sup>1</sup> · Henderik A. Proper<sup>2</sup>

Received: 7 April 2023 / Revised: 15 December 2023 / Accepted: 18 December 2023  
© The Author(s) 2024

### Abstract

Models have long since been used, in different shapes and forms, to understand, communicate about, and (re)shape, the world around us; including many different social, economic, biological, chemical, physical, and digital aspects. This is also the case in the context of enterprise architecture (EA), where we see a wide range of models in many different shapes and forms being used as well. Researchers in EA modeling usually introduce their own lexicon, and perspective of what a model actually is, while accepting (often implicitly) the accompanying ontological commitments. Similarly, practitioners of EA modeling implicitly also commit to (different) ontologies, resulting in models that have an uncertain ontological standing. This is because, for the subject domain of enterprise architecture models (as opposed to the content of such models), no single ontology has gained major traction. As a result, studies into aspects of enterprise architecture models, such as “model quality” and “return on modeling effort”, are fragmented, and cannot readily be compared or combined. This paper proposes a comprehensive applied ontology, specifically geared to enterprise architecture modeling. Ontologies represent structured knowledge about a particular subject domain. It allows for study into, and reasoning about, that subject domain. Our ontology is derived from a theory of modeling, while clarifying concepts such as “enterprise architecture model”, and introduces novel concepts such as “model audience” and “model objective”. Furthermore, the relevant interrelations between these different concepts are identified and defined. The resulting ontology for enterprise architecture models is represented in OntoUML, and shown to be consistent with the foundational ontology for modeling, Unified Foundational Ontology.

**Keywords** Enterprise architecture · Ontology · Domain model · Enterprise architecture modeling · Enterprise architecture model · Architecture · Model quality

### 1 Introduction

#### 1.1 Models to understand and shape the world

In dealing with complex phenomena, such as processes in nature, the construction of a building, the design of an information system, etc, we tend to ‘work with’ an *abstraction* (in our mind) of the actual phenomenon, while zooming in on those properties of the phenomenon that matter to us, and filtering out all the properties that are not relevant to the goals at hand. When we decide to externalize this *abstraction* in

terms of some artifact, then this artifact is a model (to us, as an individual) of the observed phenomenon [50].

In general, the resulting models may take different shapes and forms, such as sketches, precise drawings, textual specifications, formal specifications, or tangible forms mimicking key physical properties of some original. The suitability of these different shapes and forms of models, depends on the purpose (including the audience) the model should serve [17, 51]. This general view on models is in line with foundational work on the concept of model by, e.g., Apostel [3], and Stachowiak [63]. What is important to stress here, is the fact that the notion of model should not be “framed” into *only* referring to formal models or boxes-and-lines diagrams.

In a development/engineering context, models can be used to, e.g., study, describe, or explain, the current state of affairs, as well as capture (at different levels of precision, detail, and normative strength) guidance toward a future desired state of affairs.

Communicated by Dominik Bork and Sybren de Kinderen.

✉ Henderik A. Proper  
Henderik.Proper@tuwien.ac.at

Jan A. H. Schoonderbeek  
J.Schoonderbeek@architecting.nl

<sup>1</sup> Maastricht University, Maastricht, The Netherlands

<sup>2</sup> TU Wien, Vienna, Austria



# Agenda

---

Domain modelling practices

Return on Modelling Effort

RoME in an EA context

Break out sessions

# Agenda

---

Domain modelling practices

Return on Modelling Effort

RoME in an EA context

Break out sessions

# Domain model

---

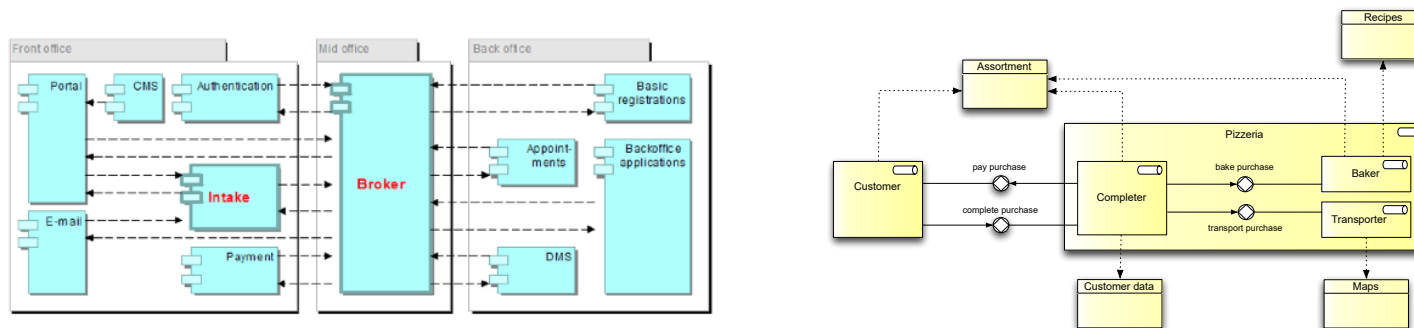
*A social artefact that is*  
*understood, and acknowledged,*  
*by a (collective) human actor*  
*to represent*  
*an abstraction*  
*of some domain*  
*for a particular cognitive purpose*

# Domain model

*A social artefact that is understood, and acknowledged, by a (collective) human actor to represent an abstraction of some domain for a particular cognitive purpose*

Traditional examples:

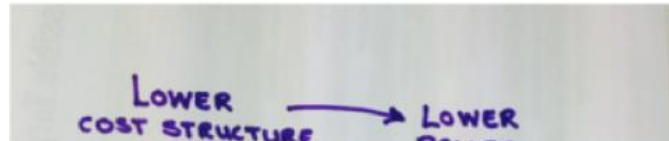
enterprise models, business process models,  
software models, information models, ... .



But ...

# Domain model

*A social artefact that is understood, and acknowledged, by a (collective) human actor to represent an abstraction of some domain for a particular cognitive purpose*



**8. Develop Product Concept**

Based on the product requirements and specifications, multiple product concepts are developed that can potentially satisfy those requirements. Brainstorming and other creativity techniques are used to generate a range of concept alternatives. These concepts are analyzed with respect to the product requirements as well as the existing technology portfolio, company capabilities, and business strategy in order to select the most promising architecture. The architecture is refined and the best aspects of other concepts are synthesized into the concept.

**Tasks**

1. Brainstorm and develop top-level product or system concept
2. Analyze, evaluate and select a preferred product concept on capabilities, development risks, and business strategy.
3. Partition the system into subsystems or modules (and derive requirements)
4. Brainstorm and develop subsystem concepts to satisfy lower-level requirements
5. Analyze, evaluate and select subsystem concepts considering development risks, and business strategy.
6. Identify need for risk-reduction development or investigation
7. Document the concept.

**Inputs**

1. Product requirements document

**Outputs/Deliverables**

1. Product concept block diagram
2. Layout drawing
3. Concept selection matrix

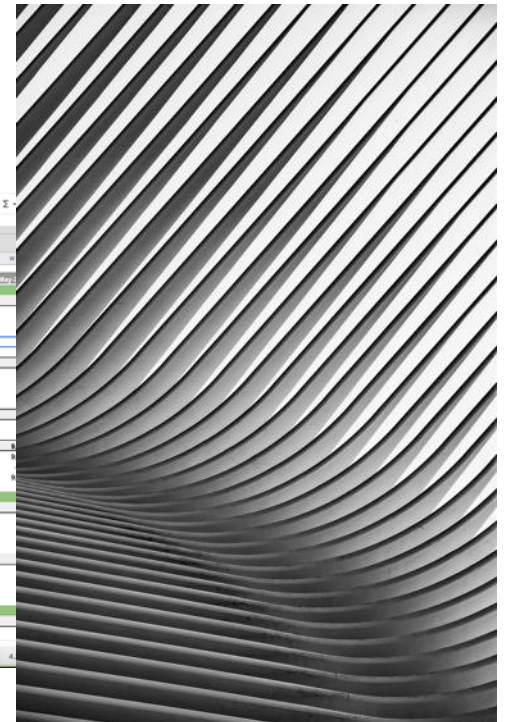
**Personnel Involved**

- Marketing
- Project Manager
- Design Engineers
- Manufacturing Engineer
- Test Engineer
- Supply Management



Financial Plan Template - Creandum

		Dec-21A	Jan-22F	Feb-22F	Mar-22F	Apr-22F	May-22F
<b>Financial metrics</b>							
Revenue	€	26,000	33,009	43,315	57,668	66,329	66,329
Recurring revenue Enterprise (ARR/MRR)	€	10,000	15,833	23,671	35,825	39,583	39,583
Growth from previous period	%	13.84%	58.33%	49.90%	80.90%	11.91%	11.91%
Recurring revenue SME (ARR/MRR)	€	13,000	17,205	19,644	22,043	26,746	26,746
Growth from previous period	%	19.48%	24.82%	14.04%	12.21%	21.33%	21.33%
<b>Cost of revenue</b>							
Cost of revenue	€	17,000	15,825	15,825	15,825	26,250	26,250
Customer support & account management	€	5,000	3,125	3,125	3,125	11,250	11,250
Hosting & Service Delivery	€	10,000	12,500	12,500	12,500	15,000	15,000
Other Operations	€	2,000	1,500	1,500	1,500	2,000	2,000
Gross profit	€	11,000	17,434	27,690	42,043	40,079	40,079
Gross Margin	%	39.3%	52.7%	63.9%	72.9%	60.4%	60.4%
<b>Cash Balance</b>							
Cash Balance	€	325,000	10,363,038	10,251,332	10,133,879	9,917,359	9,917,359
Cash beginning of period	€	645,000	625,000	10,363,038	10,251,332	10,133,879	9,917,359
Net Burn	€	-123,000	9,858,038	-131,706	-117,303	-216,629	-216,629
Cash end of period	€	525,000	10,363,038	10,251,332	10,133,879	9,917,359	9,917,359
<b>Customer metrics</b>							
Enterprise clients	#	4	6	9	10	11	11
Average revenue per enterprise customer	€	2,500	2,648	2,645	3,517	3,501	3,501
CLV	years	15.8	16.7	16.7	16.7	16.7	16.7
CLTV	€	450,000	529,543	528,951	703,332	700,259	700,259
SME clients	#	22	27	30	34	40	40
Average revenue per SME customer	€	600	646	646	646	667	667
CLV	years	3.5	4.2	4.2	4.2	4.2	4.2
CLTV	€	25,200	32,262	32,262	32,262	33,303	33,303
<b>Customer acquisition costs</b>							
CAC SME (marketing)	€/client	5,167	1,967	1,899	1,879	2,661	2,661
SME customer acquisition costs	€	10,333	8,125	8,125	8,125	17,750	17,750



# For Enterprise Architects ... ;-)

---

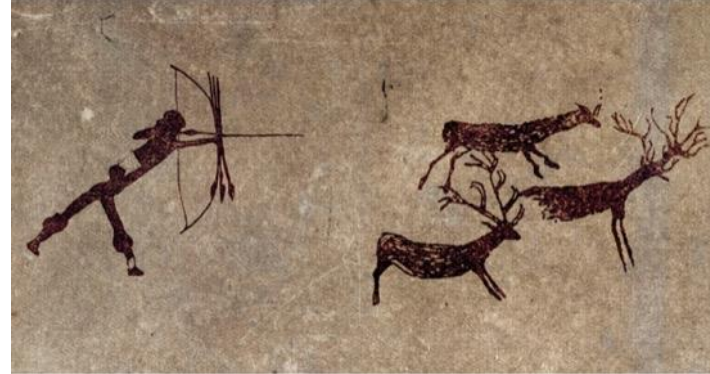
*A view is a model that is derived from another model*

It may:

- have a stricter focus
- further abstract from the original domain
- use other symbols & language for its representation
- use another medium for its representation

# Modelling practices

Modelling is natural



Modelling practices emerge naturally

When modelling becomes critical ...

then we should concern ourselves with *modelling capabilities*

# Agenda

---

Domain modelling practices

Return on Modelling Effort

RoME in an EA context

Break out sessions



# Return on Modelling Effort (RoME)

---

Modelling involves different (potential) efforts

# Return on Modelling Effort (RoME)

---

Modelling involves different (potential) efforts

Beforehand:

- Learn a language/conventions
- Acquire needed tools and infrastructures
- ...

# Return on Modelling Effort (RoME)

---

Modelling involves different (potential) efforts

During model creation:

- Gather relevant sources
- Creation of the model
- ...

# Return on Modelling Effort (RoME)

---

Modelling involves different (potential) efforts

After model creation:

- Management of the model
- Utilization of the model
- Interpreting the model
- ...

# Return on Modelling Effort (RoME)

---

Modelling is done for a reason ... its purpose

As such, it presumably adds value

Value-in-Action (ViA):

- Value-in-Use of a model
- Value-in-Creation of a model

# Return on Modelling Effort (RoME)

---

Modelling is done for a reason ... its purpose

As such, it presumably adds value

Value-in-Action (ViA):

- Value-in-Use of a model
- Value-in-Creation of a model

{ understand, assess, diagnose, design, realise, operate, regulate, ... }

# Return on Modelling Effort (RoME)

---

Modelling is done for a reason ... its purpose

As such, it presumably adds value

Value-in-Action (ViA):

- Value-in-Use of a model
- Value-in-Creation of a model

{ collaborative, individual } x { appreciation, ideation, ... }

# Agenda

---

Domain modelling practices

Return on Modelling Effort

RoME in an EA context

Break out sessions



# Enterprise models

---

A domain model pertaining to an enterprise

Could pertain to:

- different aspects (business to IT) and their relations
- its past, current, and desired affairs

# Enterprise models

---

A domain model pertaining to an enterprise

Can be:

- Extensional models (e.g. expressed in ArchiMate)
- Intensional models (e.g. in terms of architecture principles)

# Role of Enterprise (Architecture) Models

---

Understand

Assess

Diagnose

Plan

Realise

Operate

Regulate

# Role of Enterprise (Architecture) Models

---

Understand

*Understand the working of an existing enterprise and / or its environment*

Assess

Diagnose

Plan

Realise

Operate

Regulate



# Role of Enterprise (Architecture) Models

---

Understand

Assess

*Assess an existing enterprise in relation to e.g. a benchmark or a reference model*

Diagnose

Plan

Realise

Operate

Regulate

# Role of Enterprise (Architecture) Models

---

Understand

Assess

Diagnose

*Diagnose the causes of an identified problem in an enterprise and / or its environment*

Plan

Realise

Operate

Regulate



# Role of Enterprise (Architecture) Models

---

Understand

Assess

Diagnose

Plan

*Express and analyse different designs and roadmaps, for the future affairs;  
Possibly guided by reference models*

Realise

Operate

Regulate

# Role of Enterprise (Architecture) Models

---

Understand

Assess

Diagnose

Plan

Realise

Operate

Regulate

*Guidance, specification, or explanation during the realisation of a (future) design of an enterprise*



# Role of Enterprise (Architecture) Models

---

Understand

Assess

Diagnose

Plan

Realise

Operate

Regulate

*Guidance, specification, or explanation for the actors involved **in** the operations of an enterprise*



# Role of Enterprise (Architecture) Models

---

Understand

Assess

Diagnose

Plan

Realise

Operate

Regulate

*Externally formulated regulation on the operational behaviour of an enterprise*



# Agenda

---

Domain modelling practices

Return on Modelling Effort

RoME in an EA context

Break out sessions



# Role of Enterprise (Architecture) Models

---

Understand

Assess

Diagnose

Plan

Realise

Operate

Regulate

ROME!



# Agenda

---

Domain modelling practices

Return on Modelling Effort

RoME in an EA context

Break out sessions



# Break out sessions

---

## Topics:

1. Powerpoint and stories are an enterprise architect's best modelling tool  
Moderator: Marijke
2. The board has the responsibility to understand architectural models  
Moderator: Erik

