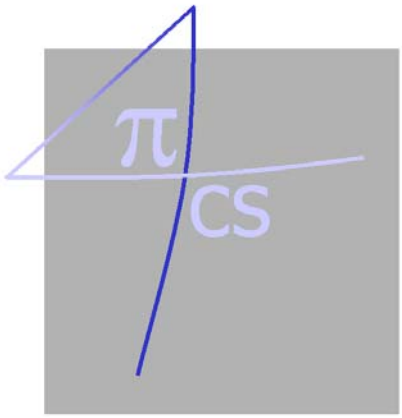
Fraunhofer Institut
Experimentelles
Software Engineering

A Reference model for requirements



Authors:
Thomas Olsson
Christian Denger
Tom Koenig
Michael Eisenbarth
Klaus Schmid

This report was supported by the grant from the autonomous province of Trento (Provincia Autonoma Di Trento), Italy, for the project "ForPICS", PAT resolution n. 171, date 08.02.2002

IESE-Report No. 043.05/E
Version 1.0
June 1, 2005

A publication by IRST and Fraunhofer IESE

Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft.
The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competitive market position.

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach (Executive Director)
Prof. Dr. Peter Liggesmeyer (Director)
Fraunhofer-Platz 1
67663 Kaiserslautern

Abstract

This report describes the reference model underlying the transformation between different models and the traceability between the same.

Keywords: ForPICS, traceability, reference model, requirements engineering

Table of Contents

1 Introduction

Currently requirements specifications, even for very complex systems are described often informally, e.g., using text. Other notations that are frequently used are use cases and certain UML diagrams. But also notations of a formal nature such as SCR [HJL96] or sequence-based specifications [PP03] are discussed. The more formal approaches are usually discussed often in an embedded environment.

During software development several of these notations are used concurrently. Sometimes they are used together to complement each other, sometimes they are used in different steps of the RE process (e.g., starting with high-level requirements sentences, refining them into use cases, detailing this in the form of a sequence-based specification). As the combination of multiple RE techniques without further support is highly error-prone, we developed the *requirements engineering reference model* to provide a semantic basis for the consistent application of multiple requirements specification techniques in a single project.

Besides the mere integration of multiple techniques, the reference model aims at supporting the following activities in the requirements engineering process:

- It enables the systematic checking of consistency among various models that are used together during requirements engineering.
- It provides a semantic basis for the transformation of requirements specifications from a higher level representation to a more detailed specification.
- It supports the mapping of requirements models to other life-cycle models (e.g., architectures and test cases).
- The model provides a semantic basis for traceability.

1.1 Reference Modelling vs. Meta-Modelling

In this report we focus on a *reference model for requirements.* As this can be easily confused with meta-modelling, we will discuss in this section the key differences among the two concepts.

1.1.1 Meta-Modelling

Especially the UML made the concept of meta-modelling rather well-known. The underlying idea of meta-modelling is to model the vocabulary of a lower-level model. From this point of view meta-modelling extends the relationship that exists between a specific model (e.g., for a specification for cruise control systems) and its modelling language (the meta-model) to an infinite number of levels (as it is always possible to find a "higher" abstraction level).

This is shown for three levels in Figure 1. Thus, meta-modelling can be understood as a process of *successive abstractions*.

1.1.2 Reference Modelling

Reference modelling is strongly different to meta-modelling as it does not involve successive abstractions. Rather it aims at providing a common frame of reference for different models (or meta-models) on *the same level*. Thus, a reference model, like the one we propose in this report, can be seen as a form of Esperanto – a common language on the same level of abstraction that can be used to translate basic expressions to and from. Relative to Meta-Modelling our reference model exists not on the meta-meta-model level, but it belongs to the meta-model level.

The key benefit that the existence of the reference model brings to requirements engineering is that it enables to focus on a standard to requirements representation, instead of devising an open-ended number of ways of representation. Thus, while it is of course still possible (and encouraged) to use other means for requirements representation, the availability of the model enables to reduce the number of translations that must be understood from n^2 to n+1 (+1 for the reference model).

Indeed the reference model we describe in this report is more regarded as a means for the translation and mapping of different forms of requirements-related information than as a modelling paradigm in its own right.

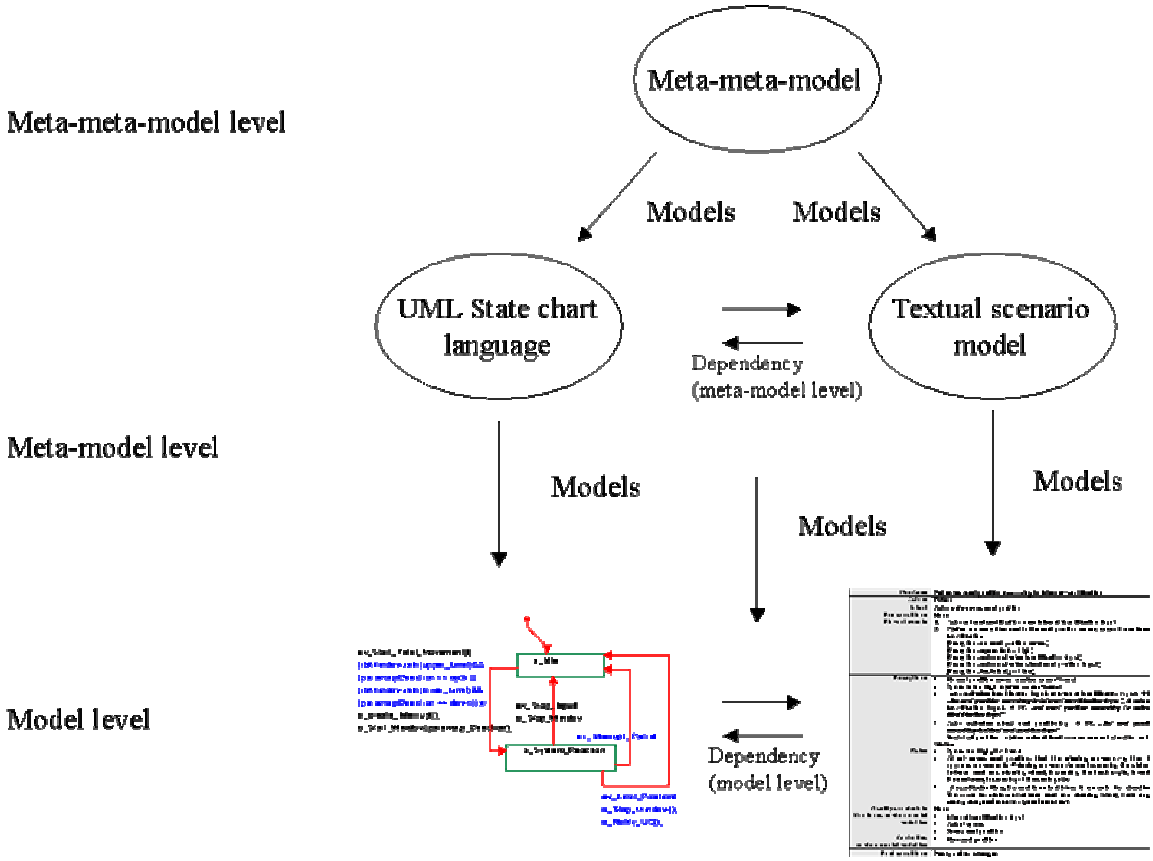


Figure 1 Levels of modeling

2 Related works

In [GGJZ00], a model is presented, defining different kinds of artefacts relevant in the requirements context. Figure 2 depicts the model presented in the reference. Even though this, quote, reference model is useful on a conceptual level, it does not provide further help how to go from the world (where the user belongs) to the program (which is what is being implemented). It should be noted that the reference model in [GGJZ00] is different than of that in this report, and at first sight might be seen as two completely different things. However, they are both reference models, though at a different abstraction level and with a different scope.

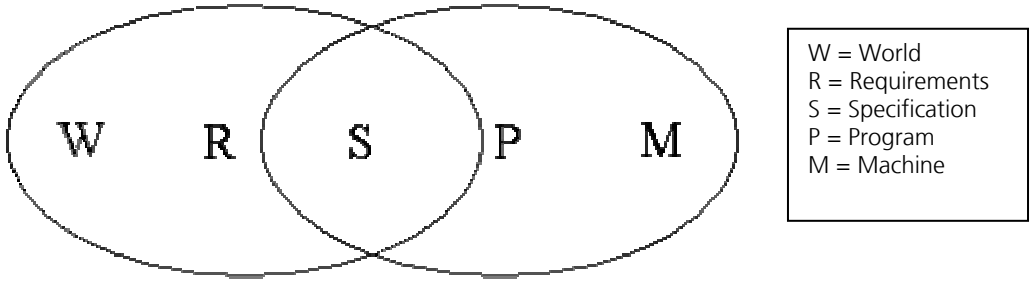


Figure 2 Reference model [GGJZ00]

[GR03] presents a framework based on graph transformation for how to integrate meta-models in a reference model. The general theory on the use of reference models is the same as used for the reference model presented in this report. However, the particular instance is different, as [GR03] focuses on a formal reference model (Labelled Transition Systems, LTS) and graph transformation. Although this gives us a nice formal framework, integrating early artefacts written in natural language is difficult.

The MDA community has similar goals as the approach presented here, see for example the special issue of IEEE Software on MDA [Software], or at OMGs initiative [OMG].

3 Requirements Reference Model

The purpose of the requirements reference model is to provide the foundation for systematic development in the real world, i.e. when faced with a set of models, not just one. This section presents the general idea of a reference model and how it is interpreted in this approach and specifically the proposed reference models for requirements. In Section 1.1, the general concept of a reference model is presented. An introduction of the Requirements Reference Model is found in Section 3.1. The entities in the reference model are presented in Section 3.2.

3.1 Introduction

The requirements reference model can be seen in Figure 3. The boxes are the entities in the reference models and the lines the relationships between the entities. The reference model is described in class diagram syntax.

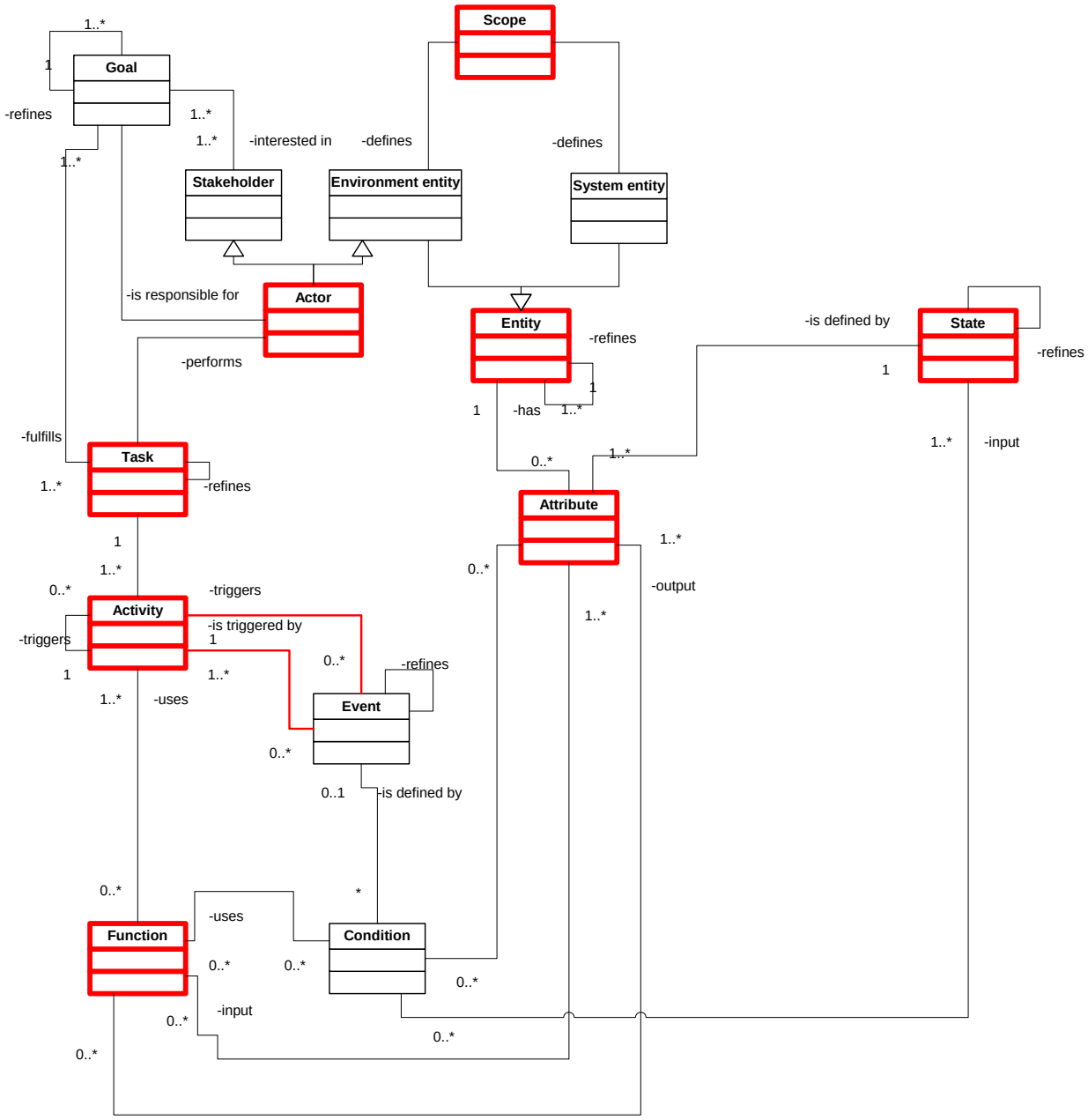


Figure 3 Requirements reference model

3.2 Requirements reference model entities

The entities in the requirements reference model, Figure 3, are described in this section. The entities are intended to be the union of the meta-modelling entities for all relevant requirements models. Hence, each model used for requirements should be translatable into this reference model, though not necessarily one-to-one.

The core concepts can be found in Table 1.

Table 1 Model entities

Entity	Description	Example
Goal	A general condition that shall be achieved in the future or should still be supported.	All receipts must be processed within 24 hours No non-authorized person should gain access
Scope	A derived entity from all system and environment entities.	ATM: The ATM hardware, with the interface to customers (environment entity) and keyboard and display (system entities)
State	A (sub-)system is at any time in a state. Two states are distinguishable, if the same input leads to different system behaviour.	ATM: Ready to accept card Missile: Weapon armed
Event	An observable change of monitored variables	The weather is changing (e.g., lightning bolt) A request for payment is made
Condition	A logical expression over the vocabulary provided by the specification language (implies only that complex combinations can be made) (includes restriction)	Condition on state (if state is ready and previous state is booting).. Condition on inputs (if person=klaus and occupation = Fraunhofer...)
Function	Mapping between input and output information (typically attributes) (similar to the mathematical definition of a function)	ATM: determine payment (types of bills) - this is timeless ABS: incrementally increase breaking (restricted by amount of time - gradient)

Environment entity	Anything that is outside of what we are describing but has a relationship to what we are describing. Environmental component is always never a human.	ATM: Display ABS: the brakes
Task	A series of at least two function, events, states, steps, sub-tasks, etc, and that involves an interaction between the system and environment components or stakeholders. A task is performed to achieve a goal	ATM: Withdraw money Missile: Destroy target ATM: Insert card is not a task, at it is not goal directed by itself
Activity	An individual step or part of a task.	ATM: Insert card
Stakeholder	Any human that have a direct or indirect relationship to what we are describing	User, card owner, driver, captain, developer, buyer, customer
Actor	Special kind of Stakeholder that is interacting directly with the system.	User, maintainer, ...
System entity	Any entity/component which is within what we are describing.	Operating system: Scheduler ABS: signal converter / output register"
Attribute	Anything observable or changeable about an environmental component or a system entity	ABS: break pressure (controlled about brakes) wheel rotation (measured about wheels)

4 Summary

This report presents an attempt to develop a reference model for requirements engineering models. The goal of the reference model is to provide the support for transformation and analysis across different models and modelling approaches.

Future work includes elaborating on the relationships between individual elements and a more formal foundation of the reference model, specifically to support automation. The former is as of yet not in the reference model at all. The latter is specifically of interest en considering tool support.

References

[OD04a] Thomas Olsson, Christian Denger, Tom Koenig, Michael Eisenbarth, Klaus Schmid. "Applying the IESE Requirements Reference model". IESE Report.

[OD04b] Thomas Olsson, Christian Denger, Tom Koenig, Michael Eisenbarth, Klaus Schmid. "Mapping and tracing requirements using the IESE requirements reference model". IESE Report.

[BH00] R. Bharadwaj and C. Heitmeyer, Developing High Assurance Avionics Systems With the SCR Requirements Method, In Proc. 19th Digital Avionics Systems Conference, 7-13 October 2000, Philadelphia, PA

[GR03] M. Große-Rhode, Semantic Integration of Heterogeneous Software Specifications. Berlin: Springer-Verlag, 2003.

[SK03] S. Sendell and W. Kozanczynski, "Model transformation: The heart and soul of model-driven software development," IEEE Software, vol. 20, no. 5, pp. 42-45, 2003.

[GGJZ00] C. A. Gunter, E. L. Gunter, M. Jackson, and P. Zave, "A Reference Model for Requirements and Specifications," IEEE Software, vol. 17, no. 3, pp. 37-43, 2000.

[Selic03] B. Selic, "The Pragmatics of Model-Driven Development," IEEE Software, vol. 20, no. 5, pp. 19-25, 2003.

[Software] IEEE Software, special issue on MDA, vol. 20, no. 5, 2003.

[OMG] MDA initiative from OMG, www.omg.com/mda.

[Seid03] E. Seidewitz, "What models mean," IEEE Software, vol. 20, no. 5, pp. 26-32, 2003.

[HJL96] C.L. Heitmeyer, R.D. Jeffords, B.G. Labaw, Automated consistency checking of requirements specifications, ACM Transactions on Software Engineering and Methodology 5 (3) (1996) 231–261.

[PP03] Prowell, S.J. and Poore, J.H. "Foundations of sequence-based software specification", IEEE Transactions on Software Engineering, 29(5), 416-429, 2003.

Document Information

Title:	A Reference model for requirements
Date:	June 1, 2005
Report:	IESE-043.05/E
Status:	Final
Distribution:	Public