Information Modeling: the process and the required competencies of its participants

P.J.M. Frederiks[1] and Th.P. van der Weide[2]

[1] paul.frederiks@philips.com, IT-N, Philips Semiconductors Nijmegen, The Netherlands
[2] tvdw@cs.kun.nl, Radboud University Nijmegen, The Netherlands

Abstract. In recent literature it is commonly agreed that the first phase of the software development process is still an area of concern. Furthermore, while software technology has been changed and improved rapidly, the way of working and managing this process have remained behind. In this paper focus is on the process of information modeling, its quality and the required competencies of its participants (domain experts and system analysts). The competencies are discussed and motivated assuming natural language is the main communication vehicle between domain expert and system analyst. As a result, these competencies provide the clue for the effectiveness of the process of information modeling.

1 Introduction

Nowadays many methods exist for the development process of software. A number of examples are: *Iterative development* ([1], [2]), *Evolutionary development*, *Incremental development* ([3]), *Reuse-oriented development* ([4]) and *Formal systems development* ([5]).

As different as all these development processes may be, there are fundamental activities common to all. One of these activities is *requirements engineering* (RE), although this activity has its own rules in each development method. RE is the process of discovering the purpose for which the software system is meant, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, negotiation, decision-making (see [6]) and subsequent implementation. For an extensive overview of the field of RE we refer to [7].

Experts in the area of software engineering do agree that RE is the most import factor for the success of the ultimate solution for reasons that this phase closes the gap between the concrete and abstract way of viewing at phenomena in application domains ([8], [9]). As a consequence, during the RE process, the involved information objects from the Universe of Discourse (UoD) have to be identified and described formally. We will refer to this process as *Information Modeling (IM)*. The resulting model will serve as the common base for understanding and communication, while engineering the requirements.

Where different areas of expertise meet, natural language may be seen as the base mechanism for communication. It is for this reason that each general modeling technique should support this basis for communication to some extent. As a consequence, the quality of the modeling process is bounded by the quality of *concretizing into an informal description* augmented with the quality of *abstracting from this description.*

In this paper focus is on information modeling as an exchange process between a domain expert and a system analyst. Our intention is to describe this exchange process, and the underlying assumptions on its participants. Using natural language in a formalized way can, for example, be seen as a supplement to the concept of *use cases* (see [10]).

Roughly speaking, a domain expert can be characterized as someone with (1) superior detail-knowledge of the UoD but often (2) minor powers of abstraction from that same UoD. The characterization of a system analyst is the direct opposite. We will describe the required skills of both system analysts and domain experts from this strict dichotomy and pay attention to the areas where they (should) meet. Of course, in practice this separation is less strict. Note that as a result of the interaction during the modeling process the participants will learn from each other. The system analyst will become more or less a domain expert, while the domain expert will develop a more abstract view on the UoD in terms of the concepts of the modeling technique. This learning process has a positive influence on effectiveness and efficiency of the modeling process, both qualitative (in terms of the result) and quantitative (in terms of completion time).

According to [11] the IM process is refined into four more detailed phases. Note that this process is not necessarily to be completed before other phases of system development, i.e. design and implementation, can start. At each moment during this process the formal specification may be realized. Still the IM process may be recognized in the various methods for system development as discussed in the beginning of this section.

In order to initiate the information modeling process the *system analyst* must elicit an initial problem specification from *domain experts.* This is referred to as *requirements elicitation.* In this phase domain knowledge and user requirements are gathered in interactive sessions with domain experts and system analysts. Besides traditional techniques, more enhanced techniques may be applied, e.g. *cognitive* or *contextual* techniques. For more elicitation techniques, see [7] or [12].

The requirements elicitation results in an *informal specification*, also referred to as the *requirements document.* As natural language is human's essential vehicle to convey ideas, this requirements document is written in natural language. In case of an evolutionary development, the previous requirements document will be used as a starting point.

In an iterative process of *modeling*, *verification* and *validation* the informal specification evolves to a complete *formal specification*, also referred to as a *conceptual model.* The primary task of the system analyst is to map the sentences of this informal specification onto concepts of the particular conceptual *modeling technique* used. As a side product, a sample population of the concepts derived from the example instances may be obtained. Using the formal syntax rules of the underlying modeling technique, the formal specification can be verified.

The conceptual model in turn can be translated into a comprehensible format. For some purposes a prototype is the preferable format, for other purposes a description is better suited. In this paper we restrict ourselves to a description in terms of natural language sentences that is to be validated by the domain expert. The example population serves as a source when instantiated sentences are to be constructed, thereby creating a feedback loop in the IM process. This translation process is called *paraphrasing*.

Basically, the conceptual model may be seen as a generative device (grammar) capable to generate not only the informal specification, but also all other feasible states of the UoD.

The *correctness* of this way of working depends on whether the formal specification is a proper derivate of the informal specification which in its turn must be a true reflection of the UoD. Being a proper derivate is also referred to as the *falsification principle*, which states that the derived formal specification is deemed to be correct as long as it does not conflict with the informal specification. Being a true reflection is referred to as the *completeness principle*. It is falsified when a possible state of the UoD is not captured, or when the grammar can derive an unintended description of a UoD state.

These two principles require the participants of the modeling process to have some specific competencies. For instance, a domain expert should be able to come up with significant sample information objects. On the other hand a system analyst should be able to make a general model out of the sample information objects such that the model describes all other samples. Section 3 discusses these competencies in more detail.

The *effectiveness* of this way of working depends on how well its participants can accomplish their share, i.e. (1) how well can a domain expert provide a domain description, (2) how well can a domain expert validate a paraphrased description, (3) how well can a system analyst map sentences onto concepts, and (4) how well can a system analyst evaluate a validation. At least one iteration of the modeling loop is required, a bound on the maximal number of iterations is not available in this simple model. Usually modeling techniques tend to focus on modeling concepts and an associated tooling but less on the process of modeling, i.e. the way of working. To minimize and to control the number of iterations, this way of working requires methods that support this highly interactive process between domain experts and system analysts. Furthermore, guarantees for the quality of this process and the required competencies of the participants involved during RE are insufficiently addressed by most common methods.

In this paper focus is on the IM process, its quality and the required competencies of the participants in this process. Section 2 describes the IM process in more detail. The competencies of the participants of the IM process is subject of discussion in section 3. In section 4 the correctness of this way of working is verified and sketched.

2 The information modeling process

In order to make a more fundamental statement about the quality of IM, we describe in figure 1 the modeling process in more depth by further elaborating the activities of *elicitation*, *modeling* and *validation*. We will also make more explicit what elements can be distinguished in a *formal specification*. In the next section we will make explicit what competencies are required from its participants, and use these competencies to verify this process.

The processes represented by the arrows labelled 4 upto 8 are suitable for automation support. A formal theory for these processes and corresponding models is elaborated in [13] and may be used as the underlying framework for building a supporting tool.

2.1 Elicitation

The stage *elicitation* is refined in figure 1 into the following substages (the numbers refer to the numbers in the figure):

1. Collect significant information objects from the application domain.
2. Verbalize these information objects in a common language.
3. Reformulate the initial specification into a unifying format.

The communication in the UoD may employ all kinds of information objects, for example text, graphics, etc. However, a textual description serves as a unifying format for all different media. Therefore the so-called *principle of universal linguistic expressibility* ([14]) is a presupposition for this modeling process:

> *All relevant information can and must be made explicit in a verbal way.*

The way of working in this phase may benefit from linguistic tools, see [11], for example to detect similarities and ambiguities between sentences.

Independent of what modeling technique used, the (sentences in the) initial specification are to be reformulated in accordance with some unifying formatting strategy, leading to the informal specification. At this point the modeling process may start, during which the involvement of the domain expert is not required.

Only few conceptual modeling techniques provide the domain expert and system analyst with clues and rules which can be applied on the initial specification, such that the resulting informal specification can actually be used in the modeling process. Examples of such modeling techniques are NIAM ([15]), Object-Role Modeling ([16]) and PSM ([17]).

2.2 Modeling

The intention of the modeling phase is to transform an informal specification into a formal specification. This phase can be decomposed into the following substages (see figure 1):