3 Specification of Requirements Models

Ricardo J. Machado, Isabel Ramos and João M. Fernandes

Abstract: The main aim of this chapter is to present and discuss a set of modeling and specification techniques, in what concerns their ontology and support in the requirements representation of computer-based systems. A systematic classification of meta-models, also called models of computation, is presented. This topic is highly relevant since it supports the definition of sound specification methodologies in relation to the semantic definition of the modeling views to adopt for a given system. The usage and applicability of Unified Modeling Language (UML) diagrams is also related to their corresponding meta-models. A set of desirable characteristics for the specification methodologies is presented and justified to allow system designers and requirements engineers to more consciously define or choose a particular specification methodology. A heuristic-based approach to support the transformation of user into system requirements is suggested, with some graphical examples in UML notation.

Keywords: Modeling, Specification, Meta-Models, Requirements, Model transformation.

3.1 Introduction

Computer-based systems integrate, as information processing sub-systems, one or more computing systems able to capture, store, process, transfer, present and manage information. Within the design of computer-based systems, this justifies the need for the incorporation of several technological entities: (1) software, firmware, and (analog and digital) hardware, to process and store information; (2) communication network services to transport information; (3) sensors and actuators to interact with the physical environment; and (4) human-machine interfaces to exchange information with human operators. Although computer-based systems can be strictly based on computer technologies, they normally include other entities such as human operators, organizational subsystems, documentation, and manuals.

Since computer-based systems are, by nature, heterogeneous, modeling and specifying their requirements demands a holistic approach.

A requirement can be defined as “something that a client needs.” From the point of view of the system designer or the requirements engineer, a requirement could also be defined as “something that must be designed.” The IEEE 610 standard [21] defines a requirement as: (1) a condition or capability needed by a user to solve a problem or achieve an objective; (2) a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documents; (3) a documented representation of a condition or capability as in (1) or (2).
Clients and developers (system designers and requirements engineers) have, naturally, different points of view towards requirements, which imply that requirements can be divided into two different categories: user and system requirements.

User requirements result directly from the requirements elicitation task (see Chap. 2 for further details on requirements elicitation techniques), as an effort to understand the clients’ needs. They are, typically, described in natural language and with informal diagrams, at a relatively low level of detail. User requirements are focused in the problem domain and are the main communication medium between the clients and the developers, at the analysis phase. System requirements result from the developers’ efforts to organize the user requirements at the solution domain. They, typically, comprise abstract models of the system, at a relatively high level of detail, and constitute the first system representation to be used at the beginning of the design phase. The correct derivation of system requirements from user requirements is an important objective because it assures that the design phase is based on the effective clients’ needs. This also guarantees that no misjudgment is arbitrarily introduced by the developers during the process of system requirements specification.

The aim of this chapter is to present and discuss a set of modeling and specification techniques, in what concerns their ontology and support in the requirements representation of computer-based systems. This chapter is not intended to be used as an exhaustive survey and summary of existing modeling approaches. It provides some guidelines to system designers and requirements engineers so that they select the modeling approach that best fits their problems. The intended audience of this chapter is system designers and requirements engineers who wish to expand their background knowledge on meta-modeling and improve their development strategy options.

Section 3.2 discusses the differences between the modeling and the specification activities. In this chapter, specification is only related to models, and not to other possible forms. Sect. 3.3 presents a systematic classification of meta-models as a key issue for the semantic definition of the modeling views to adopt for a given system. Some authors use the term “modeling techniques”, instead of “meta-models”. Sect. 3.4 describes a set of desirable characteristics for specification methodologies, so that system designers and requirements engineers can more consciously define or choose a particular specification methodology. Sect. 3.5 briefly describes a heuristic based approach to support the transformation of user into system requirements. This section shows that model continuity is a key issue and highlights the importance of having a well defined process to relate, map and transform requirements models.

### 3.2 Modeling vs. Specification

The first decision of developers, when they want to specify a system, is to select which part of the system they wish to take into account. The selection of that part defines the system view, i.e., the system perspective that needs to be represented