Integrating Natural Language Techniques in OO-Method

Isabel Díaz,[1-2] Lidia Moreno,[1] Inmaculada Fuentes,[1] and Oscar Pastor[1]

[1] Universidad Politécnica de Valencia – Dpto. de Sistemas Informáticos y Computación
Camino de Vera s/n, 46022 Valencia, España
[2] Universidad Central de Venezuela
Ciudad Universitaria, Edif. FACES, Piso 2, Caracas 1051, Venezuela
{idiaz,lmoreno,opastor}@dsic.upv.es ; infuegil@inf.upv.es

Abstract. An approach that involves natural language analysis techniques for the treatment of software system functional requirements is described in this paper. This approach is used as the basis for a process developed to generate sequence diagrams automatically from the textual specification of use cases. This facility has been integrated in the Requirements Engineering Phase of OO-Method, an automatic production environment of software. For this purpose, a translator that is based on natural language parser is used. The translator provides grammatical information to each use case sentence and it identifies the corresponding interaction. The automatic transformation is conceived and specified following an orientation that is based on models and patterns. The results of the validation of the transformation patterns are presented.

1 Introduction

The OO-Method is an automatic production environment of object-oriented software that has been created at the Universidad Politécnica de Valencia [1]. It is supported by a tool whose industrial version was given the name OlivaNova Model Execution® (ONME). In the OO-Method, the construction of the *Conceptual Model* plays a leading role from which it is possible to generate the *Execution Model* automatically (Fig. 1). The *Conceptual Model* graphically describes the problem space from a structural, dynamic, and functional perspective, and from the point of view of the presentation. Each piece of the Conceptual Model's graphic information can be automatically transformed into an OASIS concept, an object-oriented formal specification language based on dynamic logic [2]. The OASIS specification is used to generate the Execution Model.

The construction of the Conceptual Model is supported by the models obtained during the OO-Method Requirements Engineering Phase [3]. This phase begins by defining the *Mission Statement* which describes its purpose and the main functionalities of the system. Taking into account the system's possible interactions with its environment, the *Functions Refinement Tree* (FRT) is obtained. The remaining nodes form a hierarchy of the system's functionalities at different abstraction levels. An FRT leaf node is an elementary function that can be activated directly by an actor or as a result of a temporal event. Each one of the ARF