Use cases written in a natural language are usually employed for specifying of functional requirements. The format of a use case is not standardized but use case sentences traditionally adhere to a simple structure and describe actions which are either communication actions (among actors and a designed system), or internal actions. The natural language is used because it is comprehensible for stakeholders and is universal enough to capture most of the requirements but it is difficult to analyze it in an automated way. Vladimír Mencl employed state-of-the-art linguistic tools to extract a behavior of a system under design from textual use cases. The behavior specification is described in form of pro-cases. His work shows that this is possible but he met several issues. In this thesis, we solve some of the issues. We propose an algorithm based on the Mencl's algorithm which allows to process more use cases than the Mencl's one and we describe a metric which evaluate a quality of parse tree. The metric helps to select the best parse tree of a use case step from parse trees generated by different linguistic parsers. It addresses the issue of eliminating an incorrect parse tree returned by a single parser.