From Textual Scenarios to Message Sequence Charts: Inclusion of Condition Generation and Actor Extraction

Leonid Kof

Fakultät für Informatik, Technische Universität München,
Boltzmannstr. 3, D-85748, Garching bei München, Germany
E-mail: kof@informatik.tu-muenchen.de

Abstract

Natural language is the main presentation means in industrial requirements documents. In such documents, system behavior is specified in the form of scenarios, with every scenario written as a sequence of sentences in natural language. To translate scenarios to executable models, Message Sequence Charts (MSCs), we proposed an approach that analyzes textual scenarios by means of computational linguistics [2]. The presented paper shows that (1) a more differentiated treatment of certain sentence types than in [2] results in better precision of the text-to-MSC translation and (2) it is possible to automate agent identification, performed semiautomatically in [2].

1. Introduction

The majority of requirements documents are written in natural language, as the survey by Mich et al. shows [3]. This results in the fact that the requirements documents are imprecise, incomplete, and inconsistent. Document authors are not always aware of these document defects.

To detect the defects and to validate the documents written in natural language, an approach translating scenarios to MSCs was proposed in [2]. Case studies have shown that this approach suffers from two problems: (1) It cannot properly handle statements about system environment, like "It is raining" in the context of a drive-by-wire system. (2) Correctness of the produced MSCs is highly dependent on the predefined set of actors, which causes the necessity of manual work for glossary construction.

Contribution: This paper presents approaches to tackle the above problems. It investigates heuristics to identify sentences that should not be translated to MSC messages, as well as heuristics to automate agent extraction.

Terminology: For the remainder of the paper, the following terminology is used: A *scenario* is a sequence of natural language sentences. An *MSC* consists of a set *actors*, a sequence of *messages* sent and received by these actors, and a sequence of *conditions* interleaved with the message sequence (see also http://www.sdl-forum.org/MSC2000present/sld003.htm).

2. Sentence Types Translated to Conditions

The original algorithm for translation of scenarios to MSCs translates every sentence to a message. However, some sentences should be translated to conditions, not to messages. By manual analysis of the available scenarios [1], the following candidate sentence types to be translated to conditions were identified: (1) passive sentences (e.g., "the instrument cluster is activated"), (2) sentences without direct object (e.g., "the instrument cluster stays active for 30 seconds"), and (3) sentences where no message sender can be identified (e.g., "it is raining").

All possibilities to generate conditions were evaluated on the same set of scenarios as in [2], taken from the instrument cluster specification [1]. This specification contains 10 use cases, with one main scenario and several error handling scenarios for every use case, a total of 41 scenarios suitable for evaluation.

Evaluation was performed with the same glossary as in [2], to provide comparability with previously conducted case studies. Different sentence types were automatically identified with the aid of computational linguistics. The following rules were applied to evaluate the correctness of the generated MSCs: (1) General statements that are actually irrelevant for the MSC (e.g., "There is no difference between rising and falling temperature values") should be translated to conditions. (2) General statements about the system state (e.g., "The instrument cluster is activated") can be translated both to messages and to conditions. (3) For a statement sequence like "X activates Y", "Y is activated", the first statement should be translated to a message, the second one to a condition. (4) If a statement does not have to

passive sentences translated to conditions	sentences without direct object translated to conditions	sentences without message sender translated to conditions	**correct MSCs, absolute/percentage**	
no	no	no	6	15%
no	no	yes	22	54%
no	yes	no	22	54%
no	yes	yes	**31**	**76%**
yes	no	no	15	37%
yes	no	yes	29	71%
yes	yes	no	25	61%
yes	yes	yes	**31**	**76%**

Table 1. Generation of conditions

be translated to a condition due to one of the above rules, it should be translated to a message. Evaluation results, shown in Table 1, make clear that, at least, sentences without direct object and sentences where no message sender can be identified should be translated to conditions.

3. Identification of Actors

The approach presented in [2] is highly sensitive to proper definition of the set of possible actors (glossary). Furthermore, although semiautomatic tools were used in [2] to construct the initial version of the glossary, the glossary had to be manually post-processed to improve the resulting MSCs.

In order to eliminate manual work necessary to construct the glossary, actors can be extracted directly from scenarios. Manual analysis of the available scenarios showed that several heuristics for the extraction of actors are possible. An actor can be: (1) a subject of an active sentence containing a direct object, like "driver" in "The driver switches on the car", (2) a subject of an active sentence containing no direct object, like "instrument cluster" in "The instrument cluster stays active", (3) a subject of a passive sentence, like "instrument cluster" in "The instrument cluster is activated", (4) a direct object, like "car" in "The driver switches on the car", or (5) the manually specified default sender/receiver (cf. [2]). For the scenarios taken from the instrument cluster case study, used in the presented work, the default sender was always the driver, and the default receiver was the car.

Furthermore, the scope for the extraction of actors can be defined in two different ways: (1) local: for a particular scenario, only actors extracted from this scenario are relevant, or (2) global: for a particular scenario, any actor extracted from any scenario is relevant.

A greedy search was used to compare heuristics with each other and to determine the best set of heuristics to extract actors. The case studies were performed on the same set of 41 scenarios as in Section 2. To generate MSCs, the algorithm variant translating all three condition candidates (passive sentences, sentences without direct object, sentences without message sender) to conditions was used, as the most robust and best performing one.

Evaluated heuristics	**correct MSCs**		**wrong MSCs**		
	absolute number	percentage	wrong, with unnecessary actors	wrong due to missing actors	others
subject of an active sentence containing a direct object, local	31	76%	2	2	6
subject of an active sentence containing no direct object, local	15	37%	7	19	0
subject of an active sentence containing a direct object + subject of a passive sentence, local	28	62%	5	2	6
subject of an active sentence containing a direct object + direct object, local	5	12%	34	2	0
subject of an active sentence containing a direct object + default sender/receiver, local	**33**	**80%**	2	0	6
subject of an active sentence containing a direct object + default sender/receiver, global	**33**	**80%**	4	0	4

Table 2. Heuristics to identify actors

Table 2 shows the evaluation summary. It shows that the best heuristics to extract actors is one of the simplest: just subjects of active sentences containing a direct object, augmented by the default sender/receiver.

4. Poster Presentation

In the course of poster presentation, we intend to explain the process of Text-to-MSC translation. The tool translating scenarios to MSCs should be demonstrated too.

References

[1] K. Buhr, N. Heumesser, F. Houdek, H. Omasreiter, F. Rothermehl, R. Tavakoli, and T. Zink. DaimlerChrysler demonstrator: System specification instrument cluster, 2004. http://www.empress-itea.org/deliverables/D5.1_Appendix_B_v1.0_Public_Version.pdf, accessed 11.01.2007.

[2] L. Kof. Scenarios: Identifying missing objects and actions by means of computational linguistics. In *15th IEEE International Requirements Engineering Conference*, pages 121–130, New Delhi, India, October 15–19 2007. IEEE Computer Society Conference Publishing Services.

[3] L. Mich, M. Franch, and P. Novi Inverardi. Market research on requirements analysis using linguistic tools. *Requirements Engineering*, 9(1):40–56, 2004.