Special edition paper

Study of Requirements Analysis Method for Signaling Safety Software Quality Improvement

Signaling safety software becomes complex with the advance of ICT (Information and Communication Technology) in signaling safety systems. Therefore, the importance of the requirement specifications to be incorporated into the signaling safety software is increasing. In this study, we provided a requirements analysis template to prevent definition omission or error in the requirements specifications. Improvement in the quality of requirements specifications is expected by using that requirements analysis template.

Keywords: Signaling safety software, Signaling safety system, Requirements analysis, Requirements acquisition, Requirements definition

1 Introduction

Signaling safety systems are becoming more sophisticated with the advance of Information and Communication Technology (ICT). With that, signaling safety software too is becoming more important, and high quality is demanded. Quality in the software requirements specification phase, an upstream phase in software development, usually greatly affects the quality of downstream phases. For that reason, improvement in quality in the software requirements specification can be expected to lead to improvement in quality of the software as a whole.

In this study, we covered a method for improving quality of software requirements analysis for the software requirements specification phase of the development life cycle (Fig. 1) shown in the IEC 62279 international standard. By improving quality of requirements defined in the software requirements specification phase, we aim to improve quality of signaling safety software.

As a result, we consolidated those as the following issues related to the software requirements specification phase for signaling safety software.

1) Specification definition omission
Requirements for tacit knowledge that is obvious to the user are omitted, and that is discovered in downstream processes.

2) Relationship between specifications is not clear
The relationship between requirements and specifications is not clear, so the range of impact at specification changes cannot be identified.

2.2 Characteristics of Signaling Safety Software

In order to prescribe solutions to issues, brainstorming, confirmation of materials on past consideration of specifications, and experiments to extract specifications from control diagrams and connection diagrams were done at users and manufacturers to sort out characteristics of signaling safety software specifications. The following is the results of classifying the characteristics extracted.

(1) There are many non-functional requirements such as safety and reliability.
(2) Discrepancies in recognition arise at confirmation of control diagrams and connection diagrams, which are existing specifications.
(3) There is variation in level of abstraction of requirements.
(4) Specifications taking into account the movements of various trains are formulated.
(5) There are many derivative models, and there are many specification changes with that.

2.3 Policies for Solving Issues

Fig. 2 shows the relationship between the issues and characteristics of signaling software specifications. From this, we found that characteristics of signaling software specifications are causes of the issues. We thus prescribed policies for solving the issues taking into account the characteristics causing the issues. The following shows the policies for solving issues.

(1) Specification definition omission (Fig. 2, Issue 1)
- Define requirements that are a form of tacit knowledge, non-functional requirements in particular, without fail (Fig. 2, Solution policy 1).
- Analyze current system and state the results of that (Fig. 2, Solution policy 2).
- Define specifications taking into account the movements of various trains and state the results of that (Fig. 2, Solution policy 4).
3 Requirements Analysis Method

3.1 Development of Requirement Analysis Templates

Generally, requirement analysis work is done by verbal meetings, exchange of documents, and the like, the same as is done in ordinary meetings. At that time, missing requirements, variation in level of detail, and discrepancies in understanding were inevitable due to knowledge on duties and amount of experience. We thus decided to consider development of requirement analysis templates with an aim of requirement analysis work being done in set forms.

In requirement analysis templates, we aimed to prevent specification definition omission by stating the perspective by which requirement analysis is done for “specification definition omission (Fig. 2, Issue 1).” And for “unclear relationships of specifications (Fig. 2, Issue 2),” we aimed to clarify the relationship between specifications and requirements that are grounds for deriving those.

We then considered policies for developing requirement analysis templates and existing methods that can be utilized and organized those. Table 1 shows the results of that organization. Based on those results, we came up with the details of requirement analysis templates.

Requirement analysis templates show all of the items that need to be considered at time of requirement analysis, and both the user and manufacturer can enter the results of considering those.

Table 1 Policy for Development of Requirement Analysis Templates

<table>
<thead>
<tr>
<th>Policy for solving issue</th>
<th>Approach to solving issue</th>
<th>Existing method usable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition of specifications being overlooked</td>
<td>Develop requirement analysis template to prevent overall overlapping of specification definition</td>
<td>IEEE 830-1998 requirements specification template</td>
</tr>
<tr>
<td>Define tacit knowledge requirements, especially non-functional requirements, without omitting or overlooking</td>
<td>Provide section on analyzing non-functional requirements in template, and list specific non-functional requirement perspectives</td>
<td>ISO/IEC 9126, 25010 software quality model</td>
</tr>
<tr>
<td>Analyze current system and state the results</td>
<td>Provide section on analyzing current system in template</td>
<td>UML, DFD, etc. as methods of expression</td>
</tr>
<tr>
<td>Define specifications taking into account movement of various trains and state those</td>
<td>Consider presenting framework for considering specifications in template and consider notation for expressing movement of trains along time axis</td>
<td>Timing chart</td>
</tr>
<tr>
<td>Relationships of specifications is unclear</td>
<td>Classify relationships, and consider notation according to those</td>
<td>Appendix 2, Terms</td>
</tr>
<tr>
<td>State relationships related to grounds for deriving specification</td>
<td>For grounds for deriving specification, note as template format in particular</td>
<td>Goal tree, USD</td>
</tr>
</tbody>
</table>

Fig. 3 shows a configuration of a requirement analysis template. Characteristics of requirement analysis templates are composed of consideration items that are appropriate for signaling software specifications and have a format where the derivation process from requirements to specifications is recorded.

Universal Specification Describing Manner (USDM) had been proposed in the past as a format for expressing the relationship between specifications and requirements that are the grounds for deriving those specifications. With requirement analysis templates, we extended USDM and made that the format for noting the process of considering analysis of requirements and the reason for deciding on specifications (Fig. 4). In this way, the process and reasons for deriving specifications from requirements becomes clear.

Specifically, as the process of requirement analysis, we record questions regarding requirements and describe the reasons for and conclusions on deciding specifications, taking those questions into account. As for the specifications themselves, they are not described in requirement analysis templates; rather, they are expressed as links to function specifications, taking into account duplication of descriptions in function specification documents. Also, the fields entered in by the user and the manufacturer are separated, so the thinking of the individual parties is clearly shown.
We also decided to describe guidelines such as those on content included as characteristics in each chapter of the requirement analysis template and its notation.

3.2 Notation for the Relationships Between Requirements and Specifications

Ordinarily, the relationships between requirements and specifications are the following three types as shown in Fig. 5. However, in conventional signaling safety software development, there was not sufficient consideration for the differences of those relationships.

(1) Requirement – specification relationship
(2) Requirement – requirement relationship
(3) Specification – specification relationship

As notation for relationships between requirements and specifications, the requirement – specification relationship can be written in the USDM extended format by dividing items filled in by the user and the manufacturer in requirement analysis templates. However, the relationship is more than just this. As shown in Fig. 5, there are also requirement – requirement relationships and specification – specification relationships. We thus studied notation for those relationships.

As a result, for requirement – requirement relationships, noting in a goal tree (Fig. 6) used in goal-oriented analysis is appropriate, including for items in a goal – method relationship.

As for specification – specification relationship, notation of the relationship between functions in particular is important. Expressing that by noting in a Data Flow Diagram (DFD) (Fig. 7) used in structured analysis is appropriate.

Summarizing the above, the individual methods used for expressing the relationship between requirements and specifications shown in Table 2 should be used depending on the type of relationship. The notations that can be used were introduced in requirement analysis templates.

3.3. Extended Timing Chart

It is essential for signaling safety system specifications to be verified as being valid in terms of movements of various trains. As a result of verification, notation expressing behavior of signals input and output synchronized with the movements of trains came to be needed.

Conventionally, timing charts (Fig. 8) are used as notation to show signal changes along a time axis. With a timing chart, changes in values of individual signals are shown along the horizontal time axis. In function specifications of current signaling safety system specifications, information on location is superimposed to express behavior corresponding to train movements. In Fig. 8, a train proceeds from left to right, enters the detection section for train detector 1 at time t1, and then signal A of train detector 1 changes from 1 to 0 in conjunction with that. Then, at time t2, the train leaves the detection section for train detector 1 and signal A switches back to 1 from 0, as is shown in the figure. Even with this notation, signal behavior for a single train moving at constant speed can be sufficiently expressed.

However, there are problems with expressing multiple trains, train lengths, and speeds with this method. In Fig. 8, the reason signal A changed from 1 to 0 at time t4 cannot be seen from this diagram. The diagram actually shows the movements of the following trains. Change of signal A at time t4 expresses change due to the following train. Also, signal A of train detector 1 changing from 1 to 0 at time t1 shows that the lead end of the
train entered the detection section. On the other hand, signal A changing from 0 to 1 at time t2 shows that the tail end of the train exited the detection section. These conditions cannot be accurately perceived from Fig. 8.

In order to overcome those problems, we defined an “extended timing chart” (Fig. 9) where the time axis and location axis are clearly separated. The extended timing chart expresses information on location at right angles to the time axis, and train movement is expressed on coordinates of the time axis and location axis. In Fig. 9, the locations of two trains at each time is clearly shown; by visualizing the relationship between the train and signal, we see that the reason signal A changed from 1 to 0 at time t4 was due to the second train.

A certain amount of width is given to the lines expressing trains to show the length of the trains. This way, we can express the timing of the lead end of the train entering the detection section and tail end exiting the detection section. Train speed can also be shown by the inclination of the line expressing the train.

We introduced this extended timing chart to requirements analysis templates as a proposal for notations that can be used.

The following effects are expected from utilizing requirements analysis templates.

The fields entered by the user and the manufacturer are clearly separated in requirements analysis templates, so the thinking of the individual parties is clear. For requirements presented by the user at the beginning of requirements analysis in particular, there previously was not a clear format, and they took diverse forms depending on the developer. But with use of requirements analysis templates, more substantial content can be presented at the beginning, and this can be expected to lead to reduction in rework.

As for the relationships between requirements and specifications, all specifications are described in requirements analysis templates in a way that they associated with requirements, so grounds for deriving specifications are clear. For relationships between individual requirements and between individual specifications, requirements can be visualized by expressions such as goal trees and DFD, and this can be expected to lead to reduction in rework too.

And for extended timing charts, specific train movements can be clearly shown, so results of verification for complex cases can be appropriately visualized.

From these effects, an increase in quality can be expected in the software requirements specification phase.

In this study, we developed “requirement analysis templates,” considered “notation for the relationships between requirements and specifications,” and proposed “extended timing charts” in order to improve the requirement analysis process in the software requirements specification phase. Through those, quality of requirements prescribed in the software requirements specification phase improved greatly, and this is expected to lead to improvement in quality of signaling safety software as a result.

Future issues are closer support for requirement analysis work, mastering of individual notations, and organization of the development process including system requirement analysis.

In the future, we plan to develop specifications using requirement analysis templates on a trial basis and evaluate those as model cases.

**Reference:**

1) IEC 62279 Ed. 1.0: Railway applications - Communications, signalling and processing systems - Software for railway control and protection systems, IEC (2002)
2) IEEE Std 830: Recommended Practice for Software Requirements Specifications, IEEE (1998)
7) Shuichiro Yamamoto, Youkyu Kogaku Kiso Chishiki [in Japanese], (Nagoya University, 2012)
9) IEC 62278 Ed. 1.0: Railway applications - Specification and demonstration of reliability, availability, maintainability and safety (RAMS), IEC (2002)