Constructing User-Oriented Requirements Specifications

Marco Buijs, student number 3238512

Utrecht University

Group 2

Constructing User-Oriented Requirements Specifications

Introduction
A Software Requirements Specification (SRS) is a precise description of a functionality or capability that a software system has to provide. It can also describe a constraint that the software system has to respect (Wiegers, 2003). In this paper, the term Software Requirements Specification and the term Requirements Specification are used interchangeably. SRSs are often not well understood by the stakeholders (Davis, 1990). The most important stakeholders are considered to be the end-users, but often there are more stakeholders, such as project managers and project financiers. When stakeholders do not understand the SRS, this can lead to a situation in which the stakeholders are presented with a useless system at the very end of the software development process (Overmyer, 1999). This is not only a waste of development effort, but also causes high costs for redevlopment. Therefore, it is essential that stakeholders understand and are able to communicate about the SRS with the party that is responsible for the software development.

Understandability of SRSs is always the responsibility of the writers (Davis et al., 1993). This means that the party that is responsible for the software project has to come up with a representation form that is understandable to all stakeholders. Overmyer proposed a method to increase the understandability of software requirements specifications in 1999. This method, called the Multimedia Requirements Specification Process (MRSP), has been designed to ease communication with stakeholders of software projects about the SRS of that project. The rationale behind the method is that the use of multimedia in requirements specifications could increase the understandability.
The first step of the MRSP is: Perform Requirements Analysis. Overmyer does not elaborate on how this should be performed, but he uses the output of this step: the TECHNICAL REQUIREMENTS SPECIFICATION. The second step is: Establish Electronic Outline for Requirements Specification. In this step, an electronic representation form is established to be able to present multimedia representation forms as well. The third step is to determine for each requirement whether it is useful to make use of multimedia to present this requirement to stakeholders. Step 4 and 5 explain how a TECHNICAL REQUIREMENTS SPECIFICATION can be represented in a way that is understandable for stakeholders. The result of step 4 and 5 is a REQUIREMENTS DOCUMENT. A template of such a REQUIREMENTS DOCUMENT can be found in Appendix A. In step 6 the REQUIREMENTS DOCUMENT is validated with the stakeholders. Figure 1 represents this method in the form of a process model. The central question in the method is whether it is useful to represent a specific requirement specification using multimedia. All the requirements specifications that are ought to be better understandable in multimedia representation than in standard textual representation should be prioritized. After prioritization, the most important ones should be represented using multimedia. Applying this method should result in an SRS that is easy to understand for all stakeholders of a software project.

Overmyer has a broad academic background: first, he gained a Bachelor of Arts degree in the field of Psychology and after that, in 1982 he gained a Master of Science degree in the field of Industrial Engineering at the university of Iowa. During his research at the George Mason University for his Ph.D. in the domain of Information Technology (Info & Software Systems Engineering), he proposed the MRSP. The method is relevant to the fields of Software Requirements Engineering and Software Requirements Analysis, because those fields can make use of the method to ease communication about the system requirements between stakeholders, system engineers, requirements engineers and software engineers.
Example
Imagine that a company called A is contracted by a company called B to develop an online search engine called S that searches for the cheapest holidays at several travel agencies. Following the MRSP model to create SRSs that are understandable to all stakeholders, A would start by performing Requirements Analysis, i.e., step 1. In this case, a few of the resulting requirements are:

1. S must show the cheapest trips that match to the search query entered by the customer. Results should be ordered ascending by price.
2. Customers of S must be able to limit the results of their search by specifying a certain destination and date of departure.
3. When S comes up with results, there should be a transition in the User Interface to make space for the results by reducing the height dedicated to the search form.
4. Transitions in the User Interface of S always have to be smooth and should never be abrupt.

Step 2 of the process is to select a framework and make it operational so that requirements can be stored in textual, visual, phonological or other form. In this case, A chooses to use a high-layer web application framework, because a website is easy to access for the stakeholders and it supports most multimedia formats. In step 3, for each of the four specified requirements, A has to decide whether it would be useful to use multimedia representation instead of only a textual representation. Overmyer mentions three specific properties of requirements to pay attention to during this selection process:

I. The degree of ambiguousness of a requirement.
II. The degree in which timing is important to meet the requirement.
III. The degree of interactivity between the user and the system that the requirement demands.

It holds for all of the three properties that the higher the degree, the more benefit you could gain by using a multimedia representation. In this example, requirement 3 and 4 would be the ones that score a high degree on property II and III and would thereby be forwarded to step 4a. Requirements 3 and 4 score high degrees on those properties because timing and interactivity is essential to meet these requirements. In step 4a, requirements are selected that should be represented by multimedia. Both requirements 3 and 4 are selected to represent using multimedia. In step 4b requirement 3 and 4 are prioritized. Requirement 4, which is the most ambiguous, is assigned a higher priority. In step 4c, both requirements are rewritten in multimedia form. In this example, we assume that both requirements are already represented in a prototype designed by A, so the only thing that they have to do is to distil the specific requirements from the prototype and integrate those distilled prototypes into the SRS. After this step, so in step 4d, requirements 1 and 2 are integrated into the website that was setup in step 2. Then, in step 5, the multimedia representations are also integrated into the website. Last but not least, in step 6, the website is used to validate and communicate about the SRS with B and potential customers of S.

Figure 2 gives a detailed example of how requirement 3, represented by text, could be turned into a multimedia representation. This is typically a deliverable of the MRSP: a requirement specification that is understandable to all stakeholders. In this case, an image is used to represent the transition from one screen into another. The arrow indicates that clicking on the search button
leads to the transition from the left screen into the right screen. Furthermore, the textual description remains part of the requirement specification as well. One could argue that another form of representation, such as an animation or prototype, could increase the understandability even more. The hard copy (paper) format in which this paper is published, limits the amount of media that can be used.

![Figure 2: representation of requirement 3 using multimedia: textual and visual representation](image)

**Process description**

Figure 3 shows the Process-Deliverable Diagram (PDD), as defined by Weerd and Brinkkemper (2009), of the MRSP to give an overview of the processes and deliverables that are involved. Table 1 provides an explanation of the different activities in the MRSP and Table 2 describes what the deliverables of the MRSP should consist of. The MRSP consists of three main phases: Determine requirements, Rewrite requirements and Validate requirements (S. P. Overmyer, 1999). The method focuses on rewriting the requirements. The goal of this phase is to create a REQUIREMENTS DOCUMENT that is easy to understand for stakeholders of the software development project. The method does not go into the details of the other two phases. Overmyer (1999) clearly describes the activities that have to be performed when using the method, but he does not always specify exact names for the deliverables of these activities. Therefore, the concepts in the PDD could not always be taken over from the original method. Instead, they are sometimes inferred from the activities that have to be performed. Furthermore, the MRSP does not address who should perform which activity, so the roles are not specified in the PDD. Because the goal of the process is to validate the requirements with the stakeholders, a requirement engineer is probably involved in most of the tasks. It is plausible that an interaction designer is asked to help with the process of rewriting the requirements to a form that is understandable to stakeholders.
There is one rule that cannot be made explicit in the PDD: All TECHNICAL EXHIBITS in the TECHNICAL REQUIREMENTS SPECIFICATION have to be represented in the form of either a MULTIMEDIA EXHIBIT or TRADITIONAL EXHIBIT in the REQUIREMENTS DOCUMENT. So the amount of REQUIREMENTS SPECIFICATIONS in the TECHNICAL REQUIREMENTS SPECIFICATION has to be equal to the amount of REQUIREMENT SPECIFICATIONS in the REQUIREMENTS DOCUMENT. Notice that the transformation of a TECHNICAL EXHIBIT into a MULTIMEDIA EXHIBIT or TRADITIONAL EXHIBIT is sometimes only a matter of copy-pasting.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Sub-activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine requirements</td>
<td></td>
<td>Determining the requirements results in one or more TECHNICAL EXHIBITS. Those TECHNICAL EXHIBITS together form the TECHNICAL REQUIREMENTS SPECIFICATION.</td>
</tr>
<tr>
<td>Rewrite requirements specification</td>
<td>Establish electronic outline</td>
<td>To be able to present and use the MULTIMEDIA EXHIBITS an ELECTRONIC OUTLINE is established. After this step, for each TECHNICAL EXHIBIT it is determined whether it is useful to represent this requirement using multimedia. After the evaluation of all the requirements, the requirements for which multimedia representation is ought to be useful are further evaluated in the</td>
</tr>
</tbody>
</table>
sub-activity “Choose requirements”. The rest of the requirements are further handled in the sub-activity “Describe requirements traditionally”.

**Choose requirements**  
From the TECHNICAL EXHIBITS of which multimedia specification is ought to be useful, a first selection is made based on the degree in which multimedia representation increases the understandability by stakeholders.

**Prioritize requirements**  
The TECHNICAL EXHIBITS in the CANDIDATE LIST are prioritized based on multimedia representation importance. From this prioritized list, a final selection is made based on costs and time schedule. This results in a FINAL LIST.

**Create multimedia exhibits**  
This task is concerned with the conversion of TECHNICAL EXHIBITS into MULTIMEDIA EXHIBITS.

**Describe requirements traditionally**  
Requirements are described using tables, figures and text, resulting in TRADITIONAL EXHIBITS. In this step they are also integrated with the ELECTRONIC OUTLINE.

**Integrate with electronic outline**  
The MULTIMEDIA EXHIBITS are integrated with the ELECTRONIC OUTLINE. After this step, the REQUIREMENTS DOCUMENT is finished.

**Validate requirements**  
The requirements are reviewed and validated with the stakeholders using the REQUIREMENTS DOCUMENT. The result is a REQUIREMENTS REVIEW REPORT

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TECHNICAL REQUIREMENTS SPECIFICATION</td>
<td>A TECHNICAL REQUIREMENTS SPECIFICATION consists of one or more TECHNICAL EXHIBITS and is written by either software engineers, programmers or analysts (Sage &amp; Palmer, 1990).</td>
</tr>
<tr>
<td>TECHNICAL EXHIBIT</td>
<td>A TECHNICAL EXHIBIT is a REQUIREMENT SPECIFICATION that is represented in a form that is understandable and usable for the software developers of the project. Such a TECHNICAL EXHIBIT is often not well understood by stakeholders.</td>
</tr>
<tr>
<td>ELECTRONIC OUTLINE</td>
<td>An ELECTRONIC OUTLINE is a framework that is used to store and access one or more REQUIREMENTS DOCUMENTS. Although usually only one REQUIREMENTS DOCUMENT is created during the MRSP, it is possible to host more than one REQUIREMENTS DOCUMENT on the ELECTRONIC OUTLINE.</td>
</tr>
<tr>
<td>CANDIDATE EXHIBIT</td>
<td>A CANDIDATE EXHIBIT is a TECHNICAL EXHIBIT that would be clearer to stakeholders when it is transformed into a MULTIMEDIA EXHIBIT.</td>
</tr>
<tr>
<td>CANDIDATE LIST</td>
<td>A CANDIDATE LIST is a list that contains all CANDIDATE EXHIBITS. In other words, it consists of zero or more CANDIDATE EXHIBITS.</td>
</tr>
<tr>
<td>REQUIREMENT SPECIFICATION</td>
<td>A REQUIREMENT SPECIFICATION is a precise description of a functionality or capability that a software system has to provide. It can also describe a constraint that the software system has to respect (Wiegers, 2003).</td>
</tr>
<tr>
<td>SELECTED EXHIBIT</td>
<td>A SELECTED EXHIBIT is a CANDIDATE EXHIBIT that is selected to be represented with a MULTIMEDIA EXHIBIT. Furthermore, it has been assigned a priority.</td>
</tr>
<tr>
<td>FINAL LIST</td>
<td>A FINAL LIST is a list that contains zero or more SELECTED EXHIBITS.</td>
</tr>
<tr>
<td>MULTIMEDIA EXHIBIT</td>
<td>A MULTIMEDIA EXHIBIT is a REQUIREMENT SPECIFICATION that is represented using a computer screen or other electronic device. Examples are animations and interactive prototypes.</td>
</tr>
</tbody>
</table>
Constructing User-Oriented …

<table>
<thead>
<tr>
<th>TRADITIONAL EXHIBIT</th>
<th>A TRADITIONAL EXHIBIT is a REQUIREMENT SPECIFICATION that is represented in the form of tables, figures and text.</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUIREMENTS DOCUMENT</td>
<td>A REQUIREMENTS DOCUMENT contains all the requirements of the project. The representation form is such that it is understandable for all stakeholders. It consists of zero or more MULTIMEDIA EXHIBITS and zero or more TRADITIONAL EXHIBITS. However, it makes no sense to create a REQUIREMENTS DOCUMENT without any requirements. So you could argue that a REQUIREMENTS DOCUMENT should contain at least one TRADITIONAL EXHIBIT or MULTIMEDIA EXHIBIT. The REQUIREMENTS DOCUMENT is validated by one or more REQUIREMENTS REVIEW REPORTS. A template of a REQUIREMENTS DOCUMENT can be found in Appendix A.</td>
</tr>
<tr>
<td>REQUIREMENTS REVIEW REPORT</td>
<td>A REQUIREMENTS REVIEW REPORT is a report that states the opinions of the stakeholders about the content of a REQUIREMENTS DOCUMENT. In other words, it validates one REQUIREMENTS DOCUMENT.</td>
</tr>
</tbody>
</table>

Table 2: Explanation of concepts

Related literature

The MRSP has been created from a user-oriented view on SRSs development. Shi (2010) gives another view on SRSs development: he describes how SRSs should be documented from a business-oriented perspective. As mentioned before, Overmyer has proposed the MRSP in 1999. The problem that he describes regarding the miscommunication with stakeholders is based on research from Davis (1990) and Sage and Palmer (1990). Overmyer’s research is supported by and builds on the work of Aiken (1989) and Palmer and Aiken (1990). They state that SRSs cannot be represented properly by using only text.

To be able to put the MRSP in a broader perspective, I refer to Pressman (2009) who elaborates on widely used Software Engineering methods. One of the sub-disciplines of Software Engineering is Software Requirements Analysis. Software Requirements Analysis consists of three main tasks: gather, evaluate and document requirements (Wiegers, 2003). Robertson and Robertson (2006) explain in detail how requirements should be gathered and evaluated during the process of Software Requirements Analysis. The MRSP has been developed as an extension to the activities in the documentation task to improve the understandability. Overmyer (1999) conducted one case study to test the effectiveness of the MRSP. This is the only case study that has been performed to validate the effectiveness of the MRSP. More than a decade after the publish date of Overmyer’s work, there is one noticeable research paper which refers to his work: Ratchev, Urwin, Muller, Pawar and Moulek (2003) wrote a research paper about the Knowledge Acquisition and sharing for Requirement Engineering (KARE) approach to requirements analysis. That research was part of the European Strategic Program on Research in Information Technology, an initiative of the European Union. They did not go into the method that Overmyer proposed, but used his work to emphasize the importance of unambiguous requirements specifications. It follows that the method is not widely used, probably because nowadays the use of multimedia to increase understandability is ought to be common sense (Mayer, 2003).
References


Appendix A

This is a template of a REQUIREMENTS DOCUMENT. Clicking a link next to a requirement opens a new window representing either a MULTIMEDIA EXHIBIT or TRADITIONAL EXHIBIT. Of course, the MULTIMEDIA EXHIBIT can also contain textual information. The REQUIREMENTS DOCUMENT is digital and the exhibits contained in it as well. Therefore, the relation between these concepts is a link. It is plausible that a REQUIREMENTS DOCUMENT contains attributes, such as a name and identifier, but it is not necessary according to the description of the method by Overmyer (1999). The same holds for the exhibits.

An example of what a specific requirement can look like is presented in figure 2. Notice that it is not possible to create a template on requirement level because a requirement specification is generic: it can be of any multimedia form.

### Requirements overview

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Type</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement 1</td>
<td>MULTIMEDIA EXHIBIT</td>
<td>requirement1.jpg</td>
</tr>
<tr>
<td>Requirement 2</td>
<td>TRADITIONAL EXHIBIT</td>
<td>requirement2.html</td>
</tr>
<tr>
<td>Requirement 3</td>
<td>TRADITIONAL EXHIBIT</td>
<td>requirement3.php</td>
</tr>
<tr>
<td>Requirement …</td>
<td>TRADITIONAL EXHIBIT</td>
<td>requirement…html</td>
</tr>
<tr>
<td>Requirement …</td>
<td>MULTIMEDIA EXHIBIT</td>
<td>requirement…avi</td>
</tr>
<tr>
<td>Requirement n</td>
<td>MULTIMEDIA EXHIBIT</td>
<td>requirementnn.html</td>
</tr>
</tbody>
</table>

Below, the template is filled in with some dummy data.