On The Impact of Passive Voice Requirements on Domain Modelling

On The Impact of Passive Voice Requirements on Domain Modelling

Henning Femmer
Technische Universität
München, Germany
femmer@in.tum.de

Jan Kučera
Technische Universität
München, Germany
kucera@in.tum.de

Antonio Vetrò
Technische Universität
München, Germany
vetro@in.tum.de

ABSTRACT

Context: The requirements specification is a central artefact in the software engineering (SE) process, and its quality (might) influence downstream activities like implementation or testing. One quality defect that is often mentioned in standards is the use of passive voice. However, the consequences of this defect are still unclear. **Goal:** We need to understand whether the use of passive voice in requirements has an influence on other activities in SE. In this work we focus on domain modelling. **Method:** We designed an experiment, in which we ask students to draw a domain model from a given set of requirements written in active or passive voice. We compared the completeness of the resulting domain model by counting the number of missing actors, domain objects and their associations with respect to a specified solution. **Results:** While we could not see a difference in the number of missing actors and objects, participants which received passive sentences missed almost twice the associations. **Conclusion:** Our experiment indicates that, against common knowledge, actors and objects in a requirement can often be understood from the context. However, the study also shows that passive sentences complicate understanding how certain domain concepts are interconnected.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specification

General Terms

Requirements Engineering, Quality Assurance, Natural Language

Keywords

Requirements Engineering, Analytical Quality Assurance, Experimentation

ESEM'14 September 18-19, 2014, Torino, Italy.

1. INTRODUCTION

During software development, the artefacts that are produced by Requirements Engineering (RE), such as use cases [8], are usually the central items for communication of stakeholders' needs. Based on these artefacts, developers write source code, testers design test cases and finally the customer accepts or rejects the product.

This central role of RE artefacts suggests that their constitution, i.e. their quality, is important for the rest of the process. This, together with the fact that producing RE artefacts is rarely the goal of the project, implies that we need to understand this notion of requirements quality from a quality-in-use perspective on the software development lifecycle. Hence, the central questions is: Which property of the RE artefact has what kind of influence on activities performed with the artefact?

Such requirements properties are often proposed by common standards (e.g. as *Requirement Language Criteria* in the most relevant RE standard ISO29148 [7], cf. [14]). One of the properties that is often proposed, is the use of passive voice in requirements. The argumentation is usually that in passive sentences it is unclear who is performing a certain action on the system (e.g. [10]). However, when we look at specifications in practice, we nearly always see violations of this property (for examples see [1, 3, 4, 11]). We argue that this is not only because of the possible lack of distribution of the standard in practice, but more due to the fact that we have not yet understood the real impact of passive sentences in textual RE specifications onto downstream SE activities.

An analysis on where textual requirements are used in SE is performed in [8]. There, one of the first steps is to create a domain model that describes the concepts on which a system is working and their interrelations. This domain model, no matter whether it is build explicitly or implicitly, forms the common understanding of the concepts that are used by the involved stakeholders throughout the further system development process. We focus this work on the impact of passive voice requirements on domain modelling for these two reasons, i.e. 1) proximity of the domain modelling activity to requirements specification and 2) usage of domain models on later phases of software development.

2. STUDY

To understand the impact of passive voice in requirements on domain modelling, we must first define requirements and domain models in our context and then identify the activities that are needed to create a domain model.

2.1 Object of study: Requirements

In the following, we understand a *requirement* as a single, textual sentence expressing a stakeholder's need. This is a common format used in many domains and mirrored in common requirements formats such as *ReqIF*[1] and tools such as *DOORS*[2]. One example of a passive voice is from [4]:

Example 2.1. Dependent on which turn signal is set, the arrow showing in the same direction lights up blinking, as long as the turn signal is set.

A version of this example in active voice would be:

Example 2.2. The system shall light up blinking of the arrow showing in the same direction, dependent on which turn the driver sets the turn signal.

2.2 Object of study: Domain model

As a *domain model* we understand a set of domain concepts plus relations between these concepts. A domain model for this requirement is the one represented in Figure 1.

Figure 1: Domain model for the example

2.3 Activites for domain modelling

In order to evaluate the domain models that experiment participants produce, we have to detail how the domain modeling is performed. For this, we took the Unified Process (UP) [8] as a reference to find the detailed subactivities that are part of domain modelling. The advantage of using the UP is that this process provides a comprehensive list of artefacts and corresponding activities: however, most of these activities are not specific to the UP, as they are performed in many forms of processes – either explicitly or implicitly.

Accordingly, we identified the following three subactivities where the usage of passive sentences in requirement specifications could potentially have an impact:

1. Find Actors
2. Identify Domain Objects
3. Identify Associations and Aggregations

For the third subactivity, we focused only on associations, because we wanted to keep the study design simple. Furthermore, aggregations can be interpreted as a special case of associations.

2.4 Goal and research questions

The goal of this study (formally defined in Table 1) is to understand whether using passive sentences in requirements has a negative impact on domain modelling activities.

From the goal definition we derive our research question:

RQ1 Is the use of passive sentences in requirements harmful for domain modelling?

The question can be broken down in three subquestions:

RQ1.1 Is the use of passive sentences in requirements harmful for finding actors?

RQ1.2 Is the use of passive sentences in requirements harmful for identifying domain objects?

RQ1.3 Is the use of passive sentences in requirements harmful for identifying associations?

[1] `http://www.omg.org/spec/ReqIF`

[2] `http://www-03.ibm.com/software/products/en/ratidoor`

Table 1: Research goal

Characterize the impact of passives sentences in requirements on domain modelling
with respect to the quality of the artefacts produced from the activities *Find Actors*, *Identify Obvious domain objects*, *Identify Associations*
from the point of view of the software developer (or business or requirements analyst)
in the context of an analysis of requirements from real projects by graduate and undergraduate students in Computer Science

2.5 Design, methodology and metrics

We designed an experiment for university students. Each participant received a set of seven requirements: for one group (P) all requirements were in the original, passive form (requirements are passive examples that we found in the specifications [4] and [11]), for the other group (A) the same requirements were transposed into the corresponding active form (see Table 2). The participants were randomly assigned to one of the two groups of requirements (P,A). They had to perform three activities for modelling the domain: identify the actors, domain objects and associations. Each activity that the participant should perform was previously instrumented by reading material and an example of a solution[3]. To verify our thesis, we observed the artefacts produced by those three activities and compared the artefacts produced in the two groups, by counting the number of *missing actors*, number of *missing domain objects* and number of *missing associations* with respect to the master solution.

i) Number of missing actors: We counted all actors that were identified in the master solution and not recognized by the participants. Additional actors were ignored since missing actors is a more serious error than superfluous ones. Also, we carefully looked for synonyms (e.g., if the master solution identified the actor "realtor", we also accepted "real-estate agent" or even "user" (if there is a clear distinction to other actors).

ii) Number of missing domain objects: The number of missing domain objects were counted similarly to the number of missing actors (i.e. synonyms are accepted). We did not count as a mistake when a participant identified an actor in the first activity and did not write down the actor name when required to draw or list the domain objects. We assumed that the subject correctly identified that actor as an object in this case.

iii) Number of missing associations: We evaluated whether the associations connect to correct domain objects. Although we asked the participants to include also the directions and names of the associations, we didn't evaluate them: these descriptions rather served as a point of assurance for the evaluator that the participant understood the task correctly.

With N_A and N_P being the number of missing actors, missing domain objects, and missing associations respectively in active sentences (A) and passive ones (P), our research questions translate in the following pair of null (1)

[3] All experiment data is available for checks and replication at `http://goo.gl/WlTPE5`

Table 2: Requirements Used in Experiment

ID	Passive voice requirement	Translated into active voice
1	The search results shall be returned no later 30 seconds after the user has entered the search criteria.	The system shall be capable of returning the search results latest 30 seconds after the user has entered the search criteria.
2	The CMA report shall be returned no later 60 seconds after the user has entered the CMA report criteria.	The system shall be capable of returning the CMA report latest 60 seconds after the user has entered the CMA report criteria.
3	The realtor shall be notified of new client appointments after automatic synchronization with office system.	The system shall notify the realtor of new client appointments after automatic synchronization with the office system.
4	All transaction details shall be obtained from the Statement Database.	The system shall obtain all transaction details from the Statement Database.
5	All additions of new users shall be recorded on the User Report.	The system shall record all additions of new users on the User Report.
6	The reliability of the indicator lights and the engine control light shall be tested, whenever the instrument cluster is activated.	The system shall test the reliability of the indicator lights and the engine control light, whenever the system activates the instrument cluster.
7	Dependent on which turn signal is set, the arrow showing in the same direction lights up blinking, as long as the turn signal is set.	The system shall light up blinking of the arrow showing in the same direction, dependent on which turn the driver set the turn signal.

and alternative (2) hypotheses (one pair for each of the three activities):

$$H_0 : N_P \leq N_A \quad (1)$$

$$H_A : N_P > N_A \quad (2)$$

The Mann-Whitney test with 95% confidence interval was used to test the three null hypotheses.

2.6 Participants selection

The subjects of the experiment are B.Sc., M.Sc. and Ph.D. students from Technische Universität München (see Table 3). To reduce the threat of participation of students with insufficient knowledge, we removed the 2 participants who achieved less than 70% correct answers in a short test in the field of RE, extracted from [12]. We furthermore excluded one participant who misunderstood the task and one participant who did not finish the experiment.

Table 3: Participants (final number in brackets)

Student's degree	A	P	Sum
B.Sc.	4(2)	1(0)	5(2)
M.Sc.	3(3)	5(5)	8(8)
PhD	3(2)	2(2)	5(4)
unknown	0(0)	1(1)	1(1)
Sum	10(7)	9(8)	19(15)

3. THREATS TO VALIDITY

We report herein the validity issues of our experiment, according to the traditional classification [15]. Regarding internal threats, we recognise the risk that the activities are so simple that no impact could occur in the experiment even though there is some impact in the reality. To reduce this threat we selected seven requirements from real industrial projects. Second, the students in our context had very limited context information. However, this applies for participants with both active and passive voices and thus should not impact the outcome per-se. Nevertheless, it remains open whether the same effect can be observed in practice. We observe a conclusion threat as well: since the activities in the experiment are from an engineering field, there is also a high risk that each participant provides a different solution. We controlled this threat by providing comprehensive instrumentation and defining tolerant criteria for correctness (see Section 2.5).

Finally we report external threats. The first one relates to the generalizability of results to industrial practices, since participants of the experiment were students. The use of students as study subjects has been longly discussed in the software engineering literature and in general considered as suitable under certain conditions, based on generally accepted criteria for validity evaluation of empirical studies [6]. Runeson [13] observed that graduate level students are feasible subjects for revealing improvement trends: since the focus in our study was a comparison and not to find an absolute level of improvements, according to [13] the use of students in our case is suitable. In addition, a study on requirements prioritisation [2] showed that experience and commitment are important factors when using students as study subjects: we balanced this threat by letting the participation of students voluntary and by ensuring an adequate level of experience and skills of participants. Finally, we used requirements from real industrial projects.

4. RESULTS

The box plot in Figure 2 gives an insight on the results and a comparison between active and passive voices requirements. In addition, Table 4 reports the results of the non parametric Mann-Whitney test with 95% confidence interval for the difference between the two groups, and Cliffs' δ as non parametric standardised effect size measure. It was not possible to reject the null hypotheses for finding actors and identifying domain objects. On the contrary, the null hypothesis on identifying associations was rejected in favor of its alternative, i.e. the number of missing associations is higher in requirements with passive sentences (effect estimate: 75% of the time).

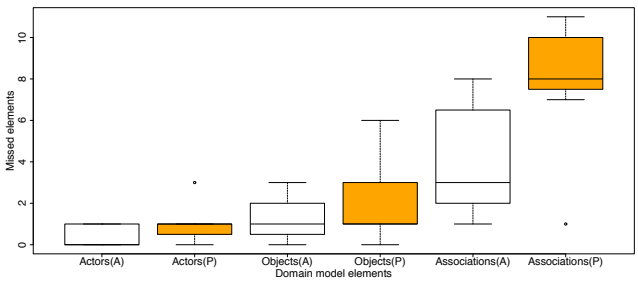

Figure 2: Number of errors for requirements in (A)ctive and (P)assive voice

5. DISCUSSION

First, the participants were able to identify actors in both active and passive voices. However, there was one difference in Requirement 3 (slightly better performance of the passives) and Requirement 7 (heavily worse performance of the passives). For the latter, the participants were unable to

Table 4: Results and descriptive statistics

	Activity	Mean A.	Mean P.	Median A.	Median P.	P value	Conf. Int.	Cliff's δ
missed	actors	0.43	1.00	0	1	0.10	$(0;\infty)$	0.39
	objects	1.29	2.00	1	1	0.25	$(-1;\infty)$	0.25
	assoc.	4.14	7.88	3	8	0.02	$(1;\infty)$	0.75

decide that the "driver" of a car was the main actor. We interpret this as follows: The role of the context is important to find the actor when it is not explicitly present; hence, it may lead to misunderstandings, but in our experiments this was not the case in nearly all examples. While in real situations we assume that more context is present and thus the difference would be even less, we have to analyse the different forms of passives in more depth to understand.

Second, for the identification of domain concepts, the results are not really different between the two groups. One could argue that it is possible to identify all relevant domain objects from the text, regardless of the form.

Last, the identification of associations shows a statistically significant difference. We see three different reasons for this: Either, this difference only shows up as significant because the activity is harder. In fact, the average number of errors is higher. But, this is the same for both active and passive voice. Or, the reason is that requirements in passive voice contain less information. But this would contradict the fact that there is no significant difference between the active and passive during identification of objects. Hence, we argue that, even though the information about the existing concepts is there, the passive complicates understanding how these concepts are linked. An additional observation supports this hypothesis: 3 out of the 9 participants in the group of passive voices (P) were not even able to draw relations, even though they quite correctly identified actors and objects.

6. RELATED WORK

There are various approaches that aim at detecting passive voice in requirements (as one form of ambiguity), with the assumption of bad impacts (e.g. [5] or [9]). Other approaches such as [10] go even further and aim at compensating the missing actors by deducing them from the context.

However, we are not aware of any study that focuses on understanding the impact of this property of requirements onto activities of the software engineering lifecycle.

7. CONCLUSION AND FUTURE WORK

In this paper, we described an experiment conducted to characterise the impact of requirements in passive voice onto domain modelling, as one activity of the SE lifecycle.

The results indicate that while the commonly discussed danger of missing actors did not show to be substantial, there is a statistically significant gap in the understanding of how concepts are related in a sentence in passive voice.

We provided a link to the experiment instrumentation for the sake of exact replication of this study. Additionally, we especially need replications with other requirements to analyse whether there is a difference in outcome when using different forms of passive voice.

On a methodical level, we are working on an evidence-based approach towards understanding quality for requirements engineering in an activity-based manner. This study is one step indicating that experiments can provide such evidence on property-impact-activity relationships.

Acknowledgments

We would like to thank Maximilian Junker, Daniel Méndez Fernández and Andreas Vogelsang for their reviews.

8. REFERENCES

[1] Canal monitoring and control system. Technical report, MoDRE, 2011.

[2] P. Berander. Using students as subjects in requirements prioritization. In *ISESE*, Aug 2004.

[3] Boston Scientific. Pacemaker system specification. Technical report, 2007.

[4] K. Buhr, N. Heumesser, F. Houdek, H. Omasreiter, F. Rothermel, R. Tavakoli, and Z. T. DaimlerChrysler demonstrator: System specification. Technical report, EMPRESS Project, 2003.

[5] G. Génova, J. M. Fuentes, J. Llorens, O. Hurtado, and V. Moreno. A framework to measure and improve the quality of textual requirements. *Requirements Engineering*, 2011.

[6] M. Höst, B. Regnell, and C. Wohlin. Using students as subjects - a comparative study of students and professionals in lead-time impact assessment. *Empirical Softw. Eng.*, 2000.

[7] ISO, IEC, and IEEE. 29148:2011 - Systems and software engineering - Requirements engineering. Technical report, 2011.

[8] I. Jacobson, G. Booch, and J. E. Rumbaugh. *The unified software development process*. Addison-Wesley, 1999.

[9] N. Kiyavitskaya, N. Zeni, L. Mich, and D. M. Berry. Requirements for tools for ambiguity identification and measurement in natural language requirements specifications. *Requirements Engineering*, 2008.

[10] L. Kof. Treatment of passive voice and conjunctions in use case documents. *NLDB*, 2007.

[11] T. Menzies, B. Caglayan, Z. He, E. Kocaguneli, J. Krall, F. Peters, and B. Turhan. The promise repository of empirical software engineering data, 2012.

[12] R. S. Pressman and D. Ince. *Software engineering: a practitioner's approach*. McGraw-Hill, 1992.

[13] P. Runeson. Using Students as Experiment Subjects - An Analysis on Graduate and Freshmen Student Data. In *EASE*, 2003.

[14] F. Schneider and B. Berenbach. A Literature Survey on International Standards for Systems Requirements Engineering. In *CSER*, 2013.

[15] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering*. Springer, 2012.