Requirements Engineering (2024) 29:3–23
https://doi.org/10.1007/s00766-024-00417-2

ORIGINAL ARTICLE

Systematic adaptation and investigation of the understandability of a formal pattern language

Elisabeth Henkel[1] · Nico Hauff[1] · Vincent Langenfeld[1] · Lukas Eber[1] · Andreas Podelski[1]

Received: 14 August 2023 / Accepted: 6 February 2024 / Published online: 4 April 2024

Abstract
Formal pattern languages are used in industry to communicate and analyse requirements, as they are said to be both machine-readable and intuitively understandable for humans. The questions arise to what extent this intuitive understanding of a pattern language is in agreement with its formal semantics and whether this understanding can be increased systematically. We present two consecutive empirical experiments to address these questions. The formal semantics serves as an objective judge on the intuitive understanding. Our experiments confirm the practical usefulness of HANFORPL insofar the intuition matches the formal semantics in most practically relevant cases. They also reveal a number of edge cases where even a prior exposure to formal logic is not a guarantee for correct understanding. We present and validate systematic adjustments to the patterns, leading to several large increases in understandability but come at the cost of new, but less impactful ambiguities. We demonstrate how an inquiry on the alignment of the intuitive and formal semantics of a pattern language can help to understand and improve the language. While results regarding the understandability of HANFORPL are favourable in commonly used cases, there is potential for improvement. The systematic adaption of patterns shows that small modifications may have large effects on the alignment of formal and intuitive semantics, and that modification must be considered with caution in the context of the respective pattern to avoid unintentionally adding new ambiguities. This article is an extension of our published REFSQ paper.

Keywords Pattern languages · Formal requirements · Intuitive understanding · Empirical study

1 Introduction

The formal representation of requirements is supposed to overcome some of the deficiencies of natural language requirements, especially lack of precision and non-machine readability [1–4]. However, if requirements are formulated in a formal logic such as temporal logic, they are accessible to only a restricted group of requirement engineers. To overcome the lack of general accessibility, Konrad and Cheng introduced a pattern language to formulate formal requirements as sentences in a restricted English grammar [5]. The intuitive understanding of these sentences is based on the intuitive understanding of natural language, while the formal semantics is derived through corresponding temporal logic formulas.

For example, we can use its formal semantics to uniquely determine that the requirement below is satisfied by the behaviour depicted in Fig. 1:

> *Globally, it is always the case that if **R** holds, then **S** holds after **1** time unit.*

It would thus seem that with pattern languages, we are in the ideal situation where we can have both, the precision of formal requirements and the accessibility of natural language. However, while the interface to the computer is fixed by the formal semantics, the interface to the human still relies on the intuitive interpretation of natural language.

Elisabeth Henkel, Nico Hauff and Vincent Langenfeld have contributed equally to this work.

✉ Elisabeth Henkel
henkele@informatik.uni-freiburg.de

Nico Hauff
hauffn@informatik.uni-freiburg.de

Vincent Langenfeld
langenfv@informatik.uni-freiburg.de

Andreas Podelski
podelski@informatik.uni-freiburg.de

[1] Department of Computer Science, University of Freiburg, Freiburg, Germany

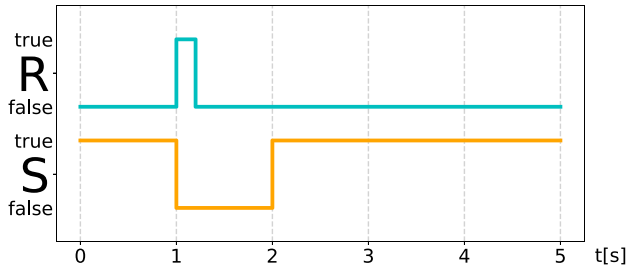


Fig. 1 Example behaviour over the observables ***R*** and ***S***

The question is to what extent we still have the issues of natural language requirements if restricted to the subset of sentences defined in a pattern language. In particular, the question arises to what extent the intuitive understanding of each requirement in the pattern language will be correct.

The existence of a formal semantics for the requirements gives us the unique opportunity to phrase the above question in a mathematically precise sense. We can give a mathematically precise definition of what is the *correct* intuitive understanding of a requirement in the pattern language, namely, through its formal meaning. In contrast, for an informal requirement, it would seem impossible to distinguish one possible intuitive understanding over another one.

For a requirement in the pattern language, the formal meaning is defined as the set of system behaviours that satisfy the corresponding temporal logic formula. Thus, we can base the test of the intuitive understanding of a requirement on a set of example behaviours, some of which satisfy the requirement and some of which do not. The existence of a formal semantics allows us to define an objective judge who decides whether the intuitive understanding is correct: the machine. Both, the requirement and the behaviour have a machine representation, and an algorithm exists to decide whether the behaviour satisfies the requirement. Thus, we only compare the intuitive understanding to the algorithmic decision.

In this paper, we report on two consecutive empirical studies to investigate the difference between the formal semantics and the intuitive understanding of requirements in a particular example of a pattern language called HANFORPL. The pattern language comes with a framework to specify requirements and behaviours, and to check whether a behaviour satisfies a requirement [4, 6].

The initial experiment confirms the practical usefulness of HANFORPL. For many cases, especially those used predominantly in industry projects, the intuitive understanding matches the formal semantics when presented to potential stakeholders. However, it also reveals that a number of *phrases of interest* represent critical edge cases, where even a prior exposure to formal logic, which turned out to be a major predictor, is not a guarantee for the correct intuitive understanding.

The second experiment follows up on these results and the mitigation we proposed for points of interest with systematic patterns of misunderstanding. We applied the proposed mitigation to the patterns in the original questionnaire to validate the resulting understandability in a classroom experiment. The control group in this experiment (answering the unmodified questionnaire) also serves as validation of the results from the first experiment.

1.1 Previously published material

This journal article is an extended version of our previously published conference paper [7]. The work is extended in the following aspects: (1) We elaborated pattern modifications in response to the results and feedback obtained in the course of the already published empirical study. (2) We performed a replication of the first study with a notably larger number of participants to increase the confidence in the obtained results and the assumptions we made regarding the understanding of trivial cases. (3) We performed an empirical study to investigate the effect of the suggested pattern modifications on the intuitive understanding of our pattern language.

1.2 Outline

This paper is structured as follows: Sect. 2 introduces the syntax of the HANFOR pattern language. Section 3 reports on the first empirical study investigating the intuitive understanding of this pattern language in an industrial setting. Our approach to modify certain patterns to improve the understanding is presented in Sect. 4. In Sect. 5, we report on the consecutive empirical study that mainly investigates the effect of these pattern modifications on the intuitive understanding of the HANFORpattern language. In Sect. 6, we discuss threats to the validity of the performed studies. Section 7 presents related work, before we finally conclude our work in Sect. 8.

2 HANFOR pattern language

The HANFOR pattern language (HANFORPL) is based on the patterns of Konrad and Cheng [5] and uses the Duration Calculus semantics of Post [8]. In fact, HANFORPL shares a large portion of patterns with the Specification Pattern System (SPS) [8].

Each instantiation of a requirement in HANFORPL is a combination of a *scope* defining the general applicability of a pattern, followed by the *pattern* itself. The scopes can be chosen from the following options *Globally*, *After* ***P***, *After* ***P*** *until* ***Q***, *Before* ***P***, and *Between* ***P*** *and* ***Q***. The resulting patterns are listed in Table 1. During instantiation, placeholders

Table 1 The table shows all patterns of HANFORPL, with their membership to a group describing overall behaviour (the *Order*, the *Occurrence*, or the *Real-Time*), the names of each pattern, and the pattern text. Due to the available space, we use "..." to omit the shared phrase *it is always the case that*. Names of patterns not already part of the SPS [8] are shown in blue colour (Color table online)

	Name	Pattern
Order	ConstrainedChain	… if **R** holds, then **S** eventually holds and is succeeded by **T**, where **U** does not hold between **S** and **T**
	Initialization	… initially **R** holds
	Persistence	… if **R** holds, then it holds persistently
	PrecedenceChain21	… if **R** holds, then **S** previously held and was preceded by **T**
	PrecedenceChain12	… if **R** holds and is succeeded by **S**, then **T** previously held
	Response	… if **R** holds, then **S** eventually holds
	ResponseChain12	… if **R** holds, then **S** eventually holds and is succeeded by **T**
	Precedence	… if **R** holds, then **S** previously held
Occur.	Absence	it is never the case that **R** holds
	ExistenceBoundU	transitions to states in which **R** holds occur at most twice
	Invariance	… if **R** holds, then **S** holds as well
	Universality	… **R** holds
Real-time	DurationBoundL	… once **R** becomes satisfied, it holds for at least **S** time units
	DurationBoundU	… once **R** becomes satisfied, it holds for less than **S** time units
	EdgeResponseBoundU1	… once **R** becomes satisfied and holds for at most **S** time units, then **T** holds afterwards
	EdgeResponseBoundL2	… once **R** becomes satisfied, **S** holds for at least **T** time units
	EdgeResponseDelay	… once **R** becomes satisfied, **S** holds after at most **T** time units
	EdgeResponseDelayBoundL2	… once **R** becomes satisfied, **S** holds after at most **T** time units for at least **U** time units
	InvarianceBoundL2	… if **R** holds, then **S** holds for at least **T** time units
	ReccurrenceBoundL	… **R** holds at least every **S** time units
	ResponseBoundL1	… if **R** holds for at least **S** time units, then **T** holds afterwards
	ResponseBoundL12	… if **R** holds for at least **S** time units, then **T** holds afterwards for at least **U** time units
	ResponseDelay	… if **R** holds, then **S** holds after at most **T** time units
	ResponseDelayBoundL2	… if **R** holds, then **S** holds after at most **T** time units for at least **U** time units
	TriggerResponseBoundL1	… after **R** holds for at least **S** time units and **T** holds, then **U** holds
	TriggerResponseDelayBoundL1	… after **R** holds for at least **S** time units and **T** holds, then **U** holds after at most **V** time units
	UniversalityDelay	… **R** holds after at most **S** time units

(usually P, Q, R, S, T) have to be replaced by Boolean expressions over observables (using $\neg$, $\wedge$ for Boolean and $<$, $=$ for numeric observables).

The semantics of each scope and pattern combination is defined by a logical formula containing the same placeholders. For a more in depth introduction to the formal foundations and the pattern semantics in detail, we kindly refer the reader to the cited work.

3 Experiment 1: intuitive understanding of HANFORPL

In this section, we describe the overall goal of the experiment in our first empirical study, our research questions, and the survey design.

3.1 Goal and research questions

As requirements pattern are used to communicate expected system behaviour, e. g., between customers or different departments, it is necessary that requirements are as understandable as possible to as many stakeholders as possible. That is, the semantics of the pattern defined by formal logics should align with the intuitive understanding of usual stakeholders.

The goal of this experiment is thus to investigate to what extent the intuitive understanding of formal requirements in HANFORPLis correct in the sense that it matches the formal semantics. This is closely related to the question of the practical usefulness of HANFORPL.

Further, we aim to identify possible reasons for misinterpretation in order to improve HANFORPL in the long term.

Based on previous experience [9, 10], we are confident that formally trained people with some training in HANFORPL perform well using the pattern language. With Research Question **R1**, we want to investigate how well participants without any training in HANFORPL understand the patterns.

However, a basic understanding of formal logics and/or requirements engineering in general may serve as a predictor for the performance dealing with edge cases and uncommon concepts (Research Question **R2**).

As the requirements pattern are based on natural language sentences, there may be phrases that allow for several sensible interpretations for complex concepts, e. g., formulations referring to timing constraints and quantification. These phrases of interest are investigated in detail in Research Question **R3**.

R1 How understandable is HANFORPL without former training in the pattern language itself?

R2 Does training in the fields of requirements engineering or formal logics have a positive effect on the understanding of HANFORPL patterns?

a) Requirements engineering
b) Formal logics

R3 How is the understanding of HANFORPL impacted by complex concepts, i. e., formulations referring to timing constraints and quantification?

With regard to the last research question (**R3**), we identified several phrases used within HANFORPL to describe concepts like timing constraints and quantification. In the following, we present a list of these *phrases of interest* (highlighted within the according pattern) together with a description of possible interpretations. Additionally, we state which of the possible interpretations matches the *intended meaning*, i. e., the semantic fixed by the corresponding Duration Calculus formula.

(prev) *[...] if **R** holds, then **S** previously held* : For this phrasing, we see two possible points for ambiguity. First, the phrase does not specify whether ***S*** has to hold persistently or only for a non-zero time interval before any occurrence of ***R***. And second, it is not specified whether ***S*** has to hold at an arbitrary point in time before the occurrence of ***R*** or directly before ***R*** holds. The intended meaning is the following: Every occurrence of ***R*** must at some point be preceded by a non-zero time interval in which ***S*** held.
(afterw)/(afterw*) *[...] if [...], then **S** holds afterwards* : Analogous to (prev), we identified two possible ambiguities. The phrase does not specify, whether ***S*** has to hold persistently or only for a non-zero time interval (afterw). Additionally, it is not specified, whether ***S*** has to hold directly after the trigger event (the [...]-part) or only at an arbitrary point in time after the triggered event (afterw*). The intended meaning is the following: ***S*** must hold directly after the trigger event for some non-zero time interval.
(aam) *[...] **R** holds after at most **d** seconds* : The phrase does not specify whether ***R*** has to hold persistently after the ***d*** seconds have passed (which is the intended meaning), or only has to hold for a non-zero time interval.
(aam-cond) *[...] if [...], then **S** holds after at most **d** seconds* : This wording is the conditioned version of (aam), i. e., it is dependent on the context of a preceding trigger. Analogous, it is not specified whether ***S*** has to hold persistently or only for a non-zero time interval after ***d*** seconds have passed. The intended meaning is the following: ***S*** has to hold for a non-zero time interval.

However, due to an oversight while extending the pattern language, this interpretation is clearly inconsistent with the intended meaning provided in (aam).
(obs)/(obs+) *[...]* *once* ***R*** *becomes satisfied* *[...]*: We identified two possible ambiguities in this pattern. The first is regarding the meaning of the phrase *becomes satisfied*. It might be unclear, whether a rising edge of ***R*** is strictly required in all cases, or whether this phrase also includes system behaviour where ***R*** initially holds (obs). The second ambiguity concerns the keyword *once*. It might be unclear, whether this means that every occurrence of ***R*** becoming satisfied should be considered or only the first occurrence (obs+). The intended meaning is the following: all occurrences of rising edges of ***R*** should be considered.
(rec) *[...]* ***R*** holds *at least every 2 s* : The intended meaning of this phrase is that the length of intervals in which ***R*** does not hold is at most 2 s. However, this wording might be misinterpreted so to mean, that ***R*** holds at fixed points in time $t_0 = 0, t_1 = 2, t_2 = 4, \ldots, t_n = 2n$.

Remark. Even though some inconsistencies, e. g., the intended meaning of *holds* in (aam) and (aam-cond), were identified while preparing the first experiment, we decided to make no premature changes for two reasons: First, we are interested to know whether such an inconsistency is noticeable in the results. Second, if it is noticeable, which of the different interpretations is the one that most participants agree with.

3.2 Subject selection

Participants for the first experiment were selected via convenience sampling of contacts of the authors and second-degree contacts in an original equipment manufacturer (OEM) in the automotive field. Subjects are mostly computer scientists and requirements engineers from the field of software engineering, automotive engineering and formal methods. The experiment was conducted in the form of an online survey with anonymous participants from the described group. Participants were asked to complete the survey without any help, but there is no control mechanism against actual cheating. At the beginning of the survey, we asked the participants for demographic information including their age group, their experience in requirements engineering, HANFORPL, and formal logics.

3.3 Object selection

This first step into the investigation of the understanding of a pattern language is focused on pattern understanding from reading, as it is the basis for further inquiries, e. g., into the generative task of pattern instantiation during formalisation or requirements elicitation. Therefore, the survey (apart from demographic questions) consists of a single repeated task: to decide if the presented pattern instantiations are fulfilled by timing diagrams of system behaviour. Simply checking phrases in isolation (e. g., What is your understanding of the phrase "*holds after at most 2 s*"?) was rejected as an option, as their interpretation may differ when embedded into the context of a pattern. This can, for example, be seen when comparing the intended meaning of the two phrases of interest (aam) and (aam-cond) within the patterns ***R*** *holds after at most* ***T*** *seconds* and *If* ***R*** *holds, then* ***S*** *holds after at most* ***T*** *seconds*.

Within the survey, we test the participants' understanding of patterns from the HANFORPL. To select a suitable set of patterns, the following criteria were considered: 1) The survey should focus on patterns that are relevant in industrial practice, 2) the survey should include the patterns using phrases of interest, and 3) the survey should be short enough to be filled in without too much interruption to a work day of participants in the industry, i. e., the survey should be completed in about 30 to 40 min.

We considered patterns that were shown to be used frequently for the formalisation of requirements in the automotive context (criterion 1). We then added patterns containing phrases of interest (criterion 2) if not yet included by the first selection criterion. For patterns whose meaning is inverse to an already added pattern (e. g. *it is always the case that* ***R*** *holds* and *it is never the case that* ***R*** *holds*), we only included the positive formulated pattern in the survey. We do not assume that negative and positive formulations behave similarly and are aware that different formulations may lead to vastly different error counts, as shown by Winter et al. [3]. However, we assume that the use of the negative formulation does not provide any additional insights into the pattern formulation in general. For example using *it is never the case that* ***R*** *holds* does not provide more insights on the phrase ***R*** *holds* than using *it is always the case that* ***R*** *holds* does.

Three patterns adding no unique phrases were dropped due to the timing constraint (criterion 3). The selection process resulted in a list of 17 patterns from the HANFORPL (see Table 2).

3.4 Survey design

The questions should be formulated in a style that avoids errors based on the incomprehensibility of the survey rather than the pattern under investigation. We therefore decided to work with only one type of question, i. e., we asked whether or not a given instantiated requirement in HANFORPL is fulfilled by a given example system behaviour. For each question, the requirement was given as written text, while the example behaviour was depicted as a timing diagram. Skipping a question was not permitted. Figure 2 exemplarily shows the first question that was asked to investigate the understanding

Question 11-A:
Is the requirement "*It is always the case that if* ***R*** *holds, then* ***S*** *holds after at most* ***1*** *second.*" fulfilled in the following example?

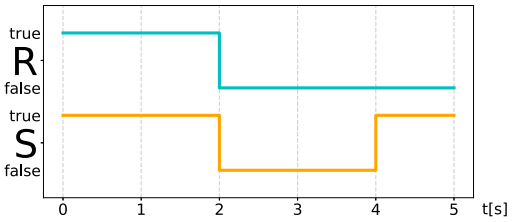


Fig. 2 The first of the four questions to investigate the understanding of the *ResponseDelay* pattern; correct answer: *yes*

of the *ResponseDelay* pattern. Consecutively, we asked the same question for three more timing diagrams (Fig. 3). That is, for each of the selected patterns, participants of the experiment had to match four example system behaviours against an instantiated requirement in HANFORPL, yielding a total of 68 questions.

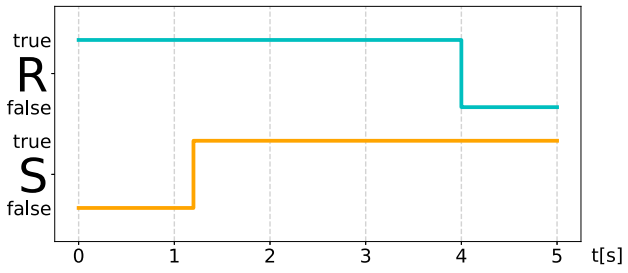
The order of questions in the survey and therefore the order of the requirements presented to the participants was static. Participants should be eased into the language by a controlled encounter with the different features of the language, from one observable, over several observables, timed quantification and so on. Thereby preventing noise within the answers resulting from being overwhelmed by a first occurrence of too many new concepts at once. Apart from the gradual exposure to the language features, we assume that no relevant training effect is present, as no feedback on the correctness of the answers was given.

To make the survey feasible within a time frame of about 30 to 40 min, the survey includes a high number of example behaviours directly targeting the phrases of interest (see Table 3). Correct answers to these questions thus mean, that the general behaviour of the pattern has been understood *and* the phrase of interest was interpreted correctly (with respect to the formal semantics).

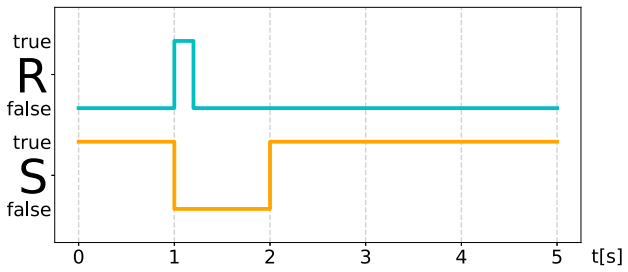
The survey does not investigate the understanding of different scopes. This would introduce another level of complexity and hence require more questions to be asked to infer reasons for possible incorrect answers. We therefore implicitly instantiated all requirements with the scope *globally*.

3.5 Study results

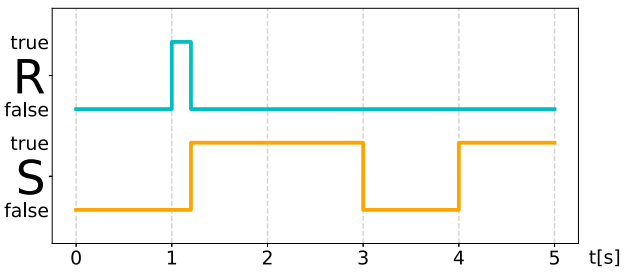
The survey was completed by 37 participants with an average experience in requirements engineering of 3.3, in HANFORPL of 1.8, and in formal logics of 3.9 on a self assessment scale of 1 (not experienced at all) to 5 (very experienced). The median age group was *41 to 50*. One participant



(a) Correct answer: *no*.



(b) Correct answer: *yes*.



(c) Correct answer: *yes*.

Fig. 3 Timing diagrams used to investigate the understanding of the *ResponseDelay* pattern (Questions 11 B - D)

indicated that they clearly misunderstood the given task as part of a feedback email. The described answer set (all *false*) was clearly identifiable, and the participant was removed as an outlier. Table 2 shows the detailed performance of all participants over all patterns and questions.

Participants had to rate their familiarity with HANFORPL in the beginning of the survey (see Fig. 4). To investigate Research Question **R1**, we separate the participants into two groups: The 26 participants being untrained in HANFORPL (answering 1 in the related self assessment question) answered with 75% accuracy (on average 51.1 of 68 questions answered correctly). The 10 participants that received former training in HANFORPL (answering > 1 in the related self assessment question) answered with 79% accuracy (on average 53.7 of 68 questions answered correctly).

Table 2 Survey results of the first experiment, per pattern (listed in the order they occur in the survey) and question (columns A,B,C,D) (Color table online)

Pattern Name	Average of correct answers (%)				
	A	B	C	D	Total
Universality	94	100	100	100	99
Invariance	67	97	64	89	79
Initialization	94	100	100	83	94
Persistence	75	100	86	100	90
Precedence	97	53	78	89	79
DurationBoundL	14	100	92	81	72
DurationBoundU	89	14	92	97	73
ReccurrenceBoundL	97	83	86	92	90
UniversalityDelay	53	92	53	86	71
InvarianceBoundL2	64	58	92	61	69
ResponseDelay	47	89	81	89	76
ResponseDelayBoundL1	58	75	92	53	69
ResponseBoundL1	39	94	53	53	60
ResponseBoundL12	50	100	92	83	81
EdgeResponseBoundL2	97	56	17	86	64
EdgeResponseBoundU1	42	72	86	11	53
EdgeResponseDelayBoundL2	100	78	75	56	77

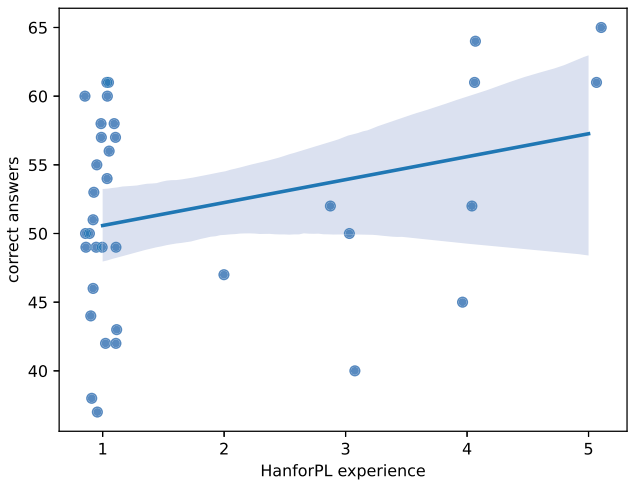


Fig. 4 The influence of former training in HANFORPL (x-axis) on the number of correct answers given in experiment one (y-axis)

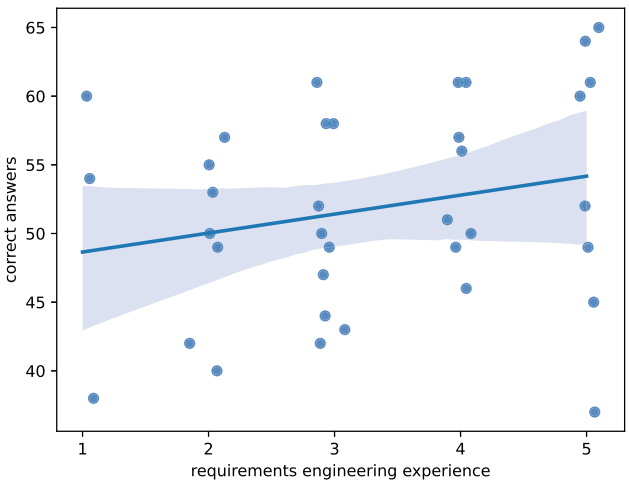


Fig. 5 The influence of experience in requirements engineering (x-axis) on the number of correct answers given in experiment one (y-axis)

There is a slight, non-significant trend of training in HANFORPL leading to more correct answers (Pearson correlation of $r(34) = 0.292$ with $p = 0.083$). The difference between both groups is statistically not significant (Mann–Whitney-U, $U = 103$ with $p = 0.348$). As both groups performed similar, we do not discern between them in the following.

In the beginning of the survey, participants had to give a self assessment of their experience in formal logics as well as requirements engineering (relating to **R2**). We assume that both disciplines give a solid foundation (be it in vocabulary or concepts) for a better understanding of requirements pattern languages.

It turned out, that training in requirements engineering does at best show a weak and statistically not significant trend (Pearson correlation of $r(34) = 0.231$ with $p = 0.175$). Astonishingly, the best and worst participants claimed to have a high understanding for requirements engineering (see Fig. 5). One could assume that this might be an artefact due to known effects on self reported ability [11], however, the distribution of correct answers in relation to reported requirements engineering skills varies strongly and may suggest that understanding requirements pattern is orthogonal to requirements engineering.

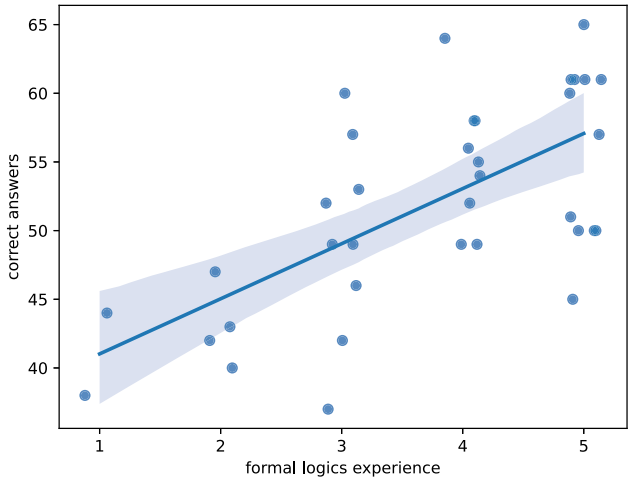


Fig. 6 The influence of experience in formal logics (x-axis) on the number of correct answers given in experiment one (y-axis)

In contrast, experience in formal logic turned out to have a strong correlation (Pearson correlation of $r(34) = 0.647$ with $p < 0.0001$) with the number of correct answers (see Fig. 6).

As the final research question (**R3**), we investigated the phrases of interest. Detailed results from the relevant questions can be seen in Table 3. For each phrase of interest, its related patterns and questions, the table shows the overall result, as well as the results of participants with prior training in formal logic (answering > 2 in the related self assessment question; $n = 30$) and with little to no training in formal logic (answering ≤ 2; $n = 6$).

Table 4 contains results of the remainder of questions with high error rates. The errors from these questions can be attributed to two kinds of formulations and underlying semantics used in the pattern language. For ease of reading, we define these ad-hoc categories analogous to the phases of interest:

(antec) *[...] if **R** holds, then **S** holds as well* : This requirement's semantic is equal to the implication $R \rightarrow S$, i. e., if ***R*** has to hold, then ***S*** has to hold as well, but not vice versa.

(atonce) *[...] if **R** holds, then **S** holds after at most **T** time units*: In this example, it is not clear if **S** is expected to be in real succession to ***R*** (as one would expect for a causal relationship), or if both happening at the same time is also valid behaviour. The latter is the case in HANFORPL.

3.6 Discussion

The overall results regarding the understanding are positive. The experiment shows that most patterns in HANFORPL can be understood even without prior training in the pattern language.

This assessment is based on the notion that, especially for engineering tasks requiring the involvement of humans, (e. g., in order to understand natural language text), judging the quality of a tool or an approach only by its ability to reach near 100 percent correctness (precision) is disregarding the complexity of the problems [12]. Pattern languages in general are thought to be applied as an interface between human readers (possibly with little to no formal background) while providing an explicit formal representation (i. e., interpretation free and machine-readable). In this context, pure natural language does not provide any additional benefit, while relying on a purely formal representation is completely incomprehensible for a large number of stakeholders [13], and may even be hard to read for a large portion of formally trained stakeholders when patterns get more involved. Thus, our measure for success is that the number of correct interpretations of participants is high enough to establish a practically sufficient understanding of the requirements. This is, a stakeholder can give mostly correct decisions for all but edge cases of the system. Remember, that the patterns provide a unique formal semantics that should be referred to when in doubt, as well as tool support enabled by the formal semantics allowing e. g., the simulation of requirements in question [14].

Results of 75–79% correct answers of participants untrained and trained in the pattern language would entail that generally more than every fifth answer to questions of whether behaviour belongs to the system are erroneous. Following the above concept on the valuation of usefulness, this shows that HANFORPL is well understandable (even without prior training). While the numbers seem low, the distribution is heavily skewed towards an unfavourable outcome, as the survey is focused on phrases of interest, i. e., on edge cases which are prone to misinterpretation. Probing these edge cases allows for the improvement of HANFORPL, but prompted errors that are seldom relevant in practical use, as explained in the detailed analysis of each phrase of interest. Requirements sets usual for industrial practice, as reported in [15], mainly contain patterns that got high success rates. This is especially the case for the *Universality* pattern and common applications of *InvarianceBoundL2* and *ResponseDelay*, i. e., excluding the answers to question A of the latter pattern, (see Table 2).

Results show that training in formal logics serves as a good predictor for the comprehension of the requirements patterns. The explanation of this effect could be twofold: First, formal logics, especially temporal logics (e. g., LTL, MTL or Duration Calculus), share similar interpretation of concepts. For example, referring to a future state requires just a non-zero interval (or single state), except it is denoted differently. Thus, the everyday understanding of these terms is, for those participants, already aligned with the formal meaning. Second, training in formal logics (in contrast to

Table 3 Correctness results for the phrases of interest in the first experiment. Each row shows the according phrase id, the pattern containing the phrase and which question in the survey prompted that exact behaviour followed by the percentage of correct answers. Column N shows participants with little to no, column L with training in formal logics (Color table online)

ID	Related pattern	Question	Average of correct answers (%)		
			N (6)	L (30)	Overall
prev	Precedence	C	50	83	78
prev	Precedence	D	67	93	89
afterw	ResponseBoundL1	D	0	63	53
afterw	EdgeResponseBoundU1	B	50	77	72
afterw*	ResponseBoundL1	C	33	57	53
afterw*	ResponseBoundL12	A	33	53	50
afterw*	EdgeResponseBoundU1	A	50	40	42
aam	UniversalityDelay	A	33	57	53
aam	UniversalityDelay	C	33	57	53
aam-cond	ResponseDelay	D	83	90	89
aam-cond	ResponseDelayBoundL1	C	83	93	92
obs	DurationBoundL	A	17	13	14
obs	DurationBoundU	B	0	17	14
obs	EdgeResponseBoundL2	C	0	20	17
obs	EdgeResponseBoundU1	D	0	13	11
obs+	DurationBoundL	C	100	90	92
obs+	DurationBoundU	D	100	97	97
obs+	EdgeResponseBoundL2	D	83	87	86
obs+	EdgeResponseDelayBoundL2	B	83	77	78
rec	ReccurrenceBoundL	B	83	83	83

Table 4 Remainder of questions in experiment one with high error rates not already covered by the phrases of interest. Column N shows participants with little to no, column L with training in formal logics (Color table online)

ID	Related pattern	Question	Average of correct answers (%)		
			N (6)	L (30)	Overall
antec	Invariance	C	17	73	64
antec	InvarianceBoundL2	D	17	70	61
atonce	Precedence	B	50	53	53
atonce	ResponseDelay	A	33	50	47
atonce	ResponseDelayBoundL1	A	33	63	58
atonce	ResponseDelayBoundL1	D	50	53	53
atonce	ResponseBoundL12	A	33	53	50

requirements engineering) may allow for more detachment from the actual physical system, i. e., ignoring the question as to what might happen before or after the timing diagram.

Analysis of individual phrases allows pinpointing phrases and concepts that are not aligned with their everyday understanding (Table 3). The results show, that (rec) and (prev) are unproblematic, as questions regarding those phrases of interest were answered correctly by most participants.

For the phrase of interest in (afterw), i. e., the text ***S*** *holds afterwards*, participants leaned on the side of ***S*** only holding for a non-zero interval which matches the intended meaning (with 53% resp. 72% correct answers). For the *ResponseBoundL1 D* question, the divide between logically trained (63% correct) versus untrained (0% correct) shows that there is a different understanding of the phrases depending on training, i. e., all the latter did assume that ***S*** has to hold persistently. Again, disambiguation by including the word *persistently* in pattern where this is the case should solve this case.

Regarding (afterw*), the question whether ***S*** has to hold immediately after the trigger event (intended meaning), participants leaned to answer incorrectly (with 53%, 50%, resp. 42% correct answers). This result shows, that the behaviour has to be made explicit. The uncertainty if ***S*** has to hold immediately or at some arbitrary point (afterw*) should be addressed by including the word *immediately* as part of the patterns.

All participants performed well on the phrasing of (aam-cond) *if [...], then* ***S*** *holds after at most* ***T*** *seconds.* In contrast, for (aam) only 53% answered correctly, i. e., that the

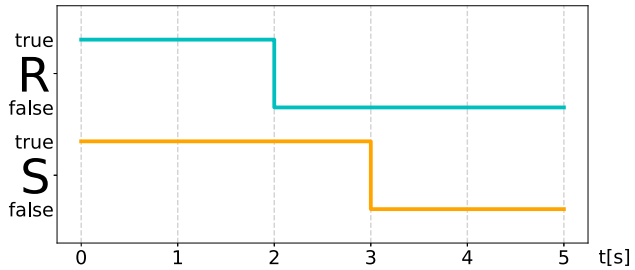


Fig. 7 Example of a *denying the antecedent* error in the survey

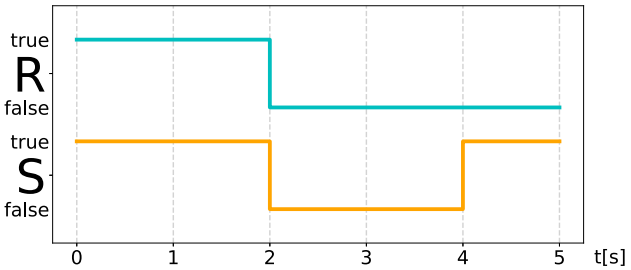


Fig. 8 Problems with immediate satisfaction of a property

observable has to hold persistently. Thus, the interpretation in (aam-cond) is in alignment with the common understanding, while the *UnversalityDelay* pattern containing the (aam) phrase should be changed to include the phrase *persistently* to be *[...] **S** holds after at most **T** seconds persistently*.

The most recent addition to the pattern language is concerned with reaction to changes of observables. Questions related to the phrase *once **R** becomes satisfied, [...]* (obs+) were consistently answered correctly, i. e., the requirement has to be evaluated after each time ***R*** becomes satisfied.

The question if an explicit rising edge is required (obs) and how especially initial behaviour is treated was highly problematic (below 17% correct answers). Answers were systematically given so, that the state of the system before the timing diagram was the missing part to satisfy the change of the observable. As we did not alter the observables, we did not include the negative case. Including the negative case would have been beneficial in analysing if participants just assumed that all observables are *false* in the beginning, or if any state was possible that suited the interpretation.

The detailed results in Table 2 show a number of questions that turned out to have a high error rate. We assigned additional ad-hoc phrases of interest: Low rates of right answers in (antec) (see Table 4) could be attributed to a common error when dealing with implications, the *denying the antecedent*. For example, the pattern *it is always the case that if **R** holds, then **S** holds as well* is satisfied by the behaviour depicted in Fig. 7. Nonetheless, ***S*** being true without ***R*** being true in time interval [2, 3] was seen as a violation by 36% of participants, especially those with little to no training in formal logic (only 17% correct in both questions). For nine participants (25%) the error was stable over both questions regarding (antec). This could point to a systematic misunderstanding of implication, or at least a difference in the understanding to the phrasing used for implication in this pattern. The existence of systematic differences of understanding conditionals has been shown by Fischbach et al. [16].

A large number of errors stem from cases in which everything relevant happens at the same point in time (atonce). An example is the requirement *if **R** holds, then **S** holds after at most **1** second* together with the behaviour depicted in Fig. 8. One can see that for time interval [0, 2] ***R*** as well as ***S*** are true, i. e., the causal relation, although it only needs to be satisfied with a delay of at most one second, is satisfied immediately. This may be again due to a notion of the requirements as more of a physical system, where the trigger results in an action with a real causal delay.

Many of the problems detected in this experiment should be fixed by small changes regarding the pattern language. As an immediate result of this experiment, several improvements for the phrases of interest were suggested, as discussed above. These modifications have to be verified carefully so, that the simplicity of the sentences is not lost. The modification could end up in overly complex sequences of adjectives trying to describe the exact behaviour of each observable.

Rather than relying entirely on the modification of the patterns, a basic understanding of formal logic, or a better understanding of formal methods in general [17, 18], should be the best mitigation for misalignment in the understanding of formal constructs. Such understanding would mitigate well known misconceptions, such as (antec), as well as support understanding for formal results gained by formal requirements tools [14, 19, 20].

Additionally, we include clarifications targeted on the misunderstandings found in this survey in our training material.

4 Pattern improvements

Table 5 summarises all patterns that we proposed improvements for.

The first modification aims to resolve issues with the persistency of assignments. While most participants interpreted the phrase *[...] **R** holds* (aam) in alignment with the formal semantic, i. e., the observable has to hold persistently, this was not the case for the phrase *if [...], then **S** holds after at most **d** time units* (aam-cond). To emphasize that the observable should hold continuously, we included the word *persistently* in both phrases.

Another modification is concerned with the phrase *if [...], then **S** holds afterwards* (afterw*). Results from the first experiment showed that the intended meaning, i. e., that the

Table 5 Patterns with modified wording. For each modified pattern, the table shows the pattern name, the original wording (top) and the modified wording (bottom); words removed from the original text are highlighted in red, words added to the new text are highlighted in green (Color table online)

Pattern Name	Pattern Text
Universality	… ***R*** holds
	… ***R*** holds *persistently*
UniversalityDelay	… ***R*** holds after at most **S** time units
	… ***R*** holds *persistently* after at most **S** time units
InvarianceBoundL2	… *if* ***R*** *holds, then* ***S*** *holds for at least* ***d*** *time units.*
	… *if* ***R*** *holds, then* ***S*** *holds immediately for at least* ***d*** *time units.*
ResponseBoundL1	… *if* ***R*** *holds for at least* ***d*** *time units, then* ***S*** *holds afterwards* .
	… *if* ***R*** *holds for at least* ***d*** *time units, then* ***S*** *holds immediately*
ResponseBoundL12	… *if* ***R*** *holds for at least* ***d1*** *time units, then* ***S*** *holds afterwards for at least* ***d2*** *time units.*
	… *if* ***R*** *holds for at least* ***d1*** *time units, then* ***S*** *holds immediately for at least* ***d2*** *time units.*
EdgeResponseBoundL2	… *once* ***R*** *becomes satisfied,* ***S*** *holds for at least* ***d*** *time units.*
	… *if* ***R*** *becomes satisfied,* ***S*** *holds immediately for at least* ***d*** *time units.*
EdgeResponseBoundU1	… *once* ***R*** *becomes satisfied and holds for at most* ***d*** *time units, then* ***S*** *holds afterwards*
	… *if* ***R*** *becomes satisfied and holds for at most* ***d*** *time units, then* ***S*** *holds immediately*
EdgeResponseDelayBoundL2	… *once* ***R*** *becomes satisfied,* ***S*** *holds after at most* ***d1*** *time units for at least* ***d2*** *time units.*
	… *if* ***R*** *becomes satisfied,* ***S*** *holds immediately after at most* ***d1*** *time units for at least* ***d2*** *time units.*
DurationBoundL	… *once* ***R*** *becomes satisfied, it holds for at least* ***d*** *time units.*
	… *if* ***R*** *becomes satisfied, it holds for at least* **d** *time units.*
DurationBoundU	… *once* ***R*** *becomes satisfied, it holds for less than* ***d*** *time units.*
	… *if* ***R*** *becomes satisfied, it holds for less than* ***d*** *time units.*

observable ***S*** should hold immediately at the occurrence of the trigger event, needs to be made explicit within the phrase. Therefore, we replaced the word *afterwards* by the word *immediately*. Moreover, we noticed three more pattern texts where the word *immediately* might be a good addition, as the formal semantics require immediate reaction to trigger events, whereas the pattern texts do not capture this explicitly.

The last modification is related to the phrase *once* ***R*** *becomes satisfied*. Currently, we do not see the possibility to address the intended interpretation of initial behaviour (obs) directly within the phrasing without overcomplicating it. We rather see this as inevitable knowledge that needs to be established when working with HANFORPL. The second issue within the given phrase is related to whether every or only the first rising edge should be considered (obs+). The former being the intended meaning. This showed to be a minor problem in the first experiment and might result from a misunderstanding of non-native English speakers, misinterpreting the word *once* to mean *only one time*. We try to address this issue by exchanging the word *once* by *if*, which should clearly indicate that all occurrences are referred to.

5 Experiment 2: effect of pattern modifications on the intuitive understanding

The experiment described in Sect. 3 was a short, industry friendly foray into the comprehensibility of HANFORPL. In this section, we describe a second, complementary experiment which serves a twofold purpose: The main objective is to investigate the effect of pattern modifications on the intuitive understanding of HANFORPL. The second purpose is to replicate the first study with a higher number of participants in order to gain confidence in the observed errors.

5.1 Goal and hypotheses

In the first experiment, we suggested a number of changes to improve the understandability of HANFORPL. The main goal of the second experiment is to evaluate these improvements, leading to Hypothesis **H1**. We assume that the improved patterns are easier to understand, thus leading to a higher number of correct answers in comparison to the unmodified patterns.

As the language improvements have to be seen within the context of HANFORPL, this experiment uses the same selection of patterns as the first experiment (i. e., patterns that are of practical relevance or cover phrases of interest). This enables

us to also use the control group of this second experiment as a replication of the first experiment (Hypothesis **H3**).

Also, in the first experiment, we argued that most *common cases*, meaning interpretations decidable without regarding edge cases, could be disregarded in order to keep the questionnaire as short as possible. We substantiate this claim by investigating these non-edge cases (Hypothesis **H2**).

As HANFORPL is a restricted English language developed by native and non-native speakers, we investigate whether there is an impact of participants' English language skills as well as their native language on the understandability of HANFORPL (Hypothesis **H4**).

In summary, the second experiment is designed around the following hypotheses:

- **H1** Each improved pattern in HANFORPL is better understood as the original pattern.
- **H2** For *common cases*, HANFORPL is very well understood.
- **H3** Results from the first experiment are reproduced.
  - (a) HANFORPL is generally understandable.
  - (b) The understandability of the defined phrases of interest is similar.
  - (c) Training in formal logics is the main predictor of performance in HANFORPL.
- **H4** Does the native language of the participants or their English skills have an impact on the understandability of HANFORPL?

5.2 Subject selection

Due to the limited availability of requirements engineers from industry projects, participants in the second experiment were mainly computer science students or students in related fields. Students in several university courses supervised by the authors and second-degree university contacts were asked to participate in the experiment. These courses encompassed both Bachelor's and Master's programs, resulting in a student cohort with diverse yet comparably more uniform levels of experience compared to the engineers involved in the first experiment. Student participants received bonus points (which counted towards passing the course) for completing the survey. The number of bonus points that could be achieved was at the discretion of the respective supervisors. However, participation in the experiment was not compulsory. We decided that the actual number of achieved points should be independent of the answers given in the survey. When investigating the intuitive understanding, there are no correct or incorrect answers, but only answers in alignment or misalignment with, in our case, the predefined formal semantics. While the decision to give points independent of performance may cause participants to not seriously work on the questionnaire, the alternative would have incentivised trying to meet our expectations (e. g., by search for a language documentation). We see the latter issue as a more serious threat to the validity of our results, especially as the lowest of effort attempts can be filtered out by control questions. Conducting the survey in a more controlled environment (e. g. in presence) to avoid either of the issues was not viable.

Although students and requirements engineers occupy different roles, there should be a sufficient degree of similarity between them, considering that the students are enrolled in STEM subjects. Nonetheless, we assume the group of students to be much more homogenous than the group of participants in the first experiment, thus we expect effects to be less pronounced. As students were not addressed in the course of the first experiment, the groups of participants in both experiments are assumed to be disjoint.

5.3 Object selection

The control group of this experiment should serve as a replication for the first experiment (Hypothesis **H3**). Therefore, we used the same selection of patterns and example behaviours as in the original survey. To investigate whether *common cases* of usage (i. e., behaviour without complex corner cases) of the patterns in HANFORPL are intuitively understandable (Hypothesis **H2**), we included two additional questions, respectively example behaviours, for each pattern. The additional example behaviours were chosen manually, taking into account that no timing with possibly ambiguous acceptance was included, the acceptance or violation of a behaviour was not triggered at the beginning of the timing diagram (as initial behaviour showed to be a source of misalignment in the first experiment), and a similar behaviour was not yet included in the remaining example behaviours. The questions from the first survey are thus a subset of the questions of the second survey.

In the first experiment, certain patterns exhibited a significant misalignment between intuitive understanding and formal semantics. Based on these results and obtained feedback, some modification to the phrasing was suggested. Although there are multiple possible modifications per misleading phrase, we decided to test only one modification against the original phrasing. Testing more than one modification for the same phrase against each other would result in some sort of ranking effect and carries the risk of confusing participants. We think that the different wording that participants are already exposed to within the survey might have an impact on their intuition in later questions – although or because these words might not be used in upcoming questions. That is, participants may think that the different wordings coexist in HANFORPL and thus carry different meaning. For example, if a participant is confronted with the word *immediately* as a modification to a pattern in one question, and is then confronted with the word *directly* as

a modification in another question, their answer to the second question might be impacted by the absence of the word *immediately*.

The second survey contains the modifications described in Table 5 for ten of the original patterns, as they are, in our opinion, the best candidates for improving the understandability (Hypothesis **H1**).

5.4 Survey design

Overall, the second experiment is a two-group design: The first group serves as a control group for this experiment as well as a replication of the first experiment (cf. Sect. 3). The second group is the treatment group to investigate the impact of the modified pattern texts. To enable the replication of experiment one, the overall survey design remained unchanged. We hence only describe changes or additions made to the original design.

At the beginning of the survey, we included additional demographic questions asking for the participants' profession and their native language (both as free text questions), as well as a self-assessment of their English skills (Likert scale from *poor* to *very good*). The later two questions being the basis for the evaluation of Hypothesis **H4**.

The main part of the survey still only contains one type of question, asking whether a given instantiated requirement in HANFORPL is fulfilled by an example system behaviour given as a timing diagram. Since we included the two *common cases* for each pattern, we asked this question for six consecutive timing diagrams. The order of patterns presented to the participants is static to prevent participants from being overwhelmed by the early appearance of complex patterns. However, we decided to randomize the order in which the six timing diagrams are shown within a pattern group to minimize the risk of learning to answer the trivial cases first, thus preserving the best possible comparability to the former experiment.

In order to filter participants who do not answer the survey conscientiously, two attention check questions are inserted into the survey. Given that the survey only contains one type of question, it is crucial to align these questions to the common format. Otherwise, there is a high risk that the attention check questions standout by their distinct presentation, failing to achieve their intended impact. We decided to directly instruct the participants on how to answer these questions, hidden in the style of a pattern text accompanied by six example behaviours. Figure 9 shows the design of the attention check questions. The survey thus contains 19 question groups (17 patterns under query and two control patterns). The control patterns were inserted at the eighth and 17th position, respectively. Another control mechanism to detect participants not seriously interested in the survey is the time spent

Question 8-A:
Is the requirement "*It is always the case that this is a control question, please answer yes to all.*" fulfilled in the following example?

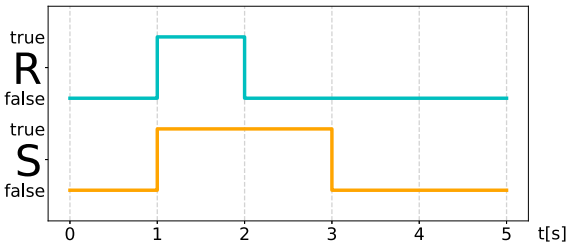


Fig. 9 Design of the attention check questions used in experiment two

on answering the survey, which is recorded for each question group.

The group of participants is randomly split into a *control* and a *treatment group*. Participants in the control group are exposed to the questions using the original phrasing of pattern texts, i. e., questions from the previous experiment enriched by the additional cases. Participants assigned to the treatment group are exposed to the same questions, however this time using the modified pattern texts.

5.5 Study results

The survey was completed by 180 participants from which 164 answered the attention check questions correctly. For these participants, the average time to complete the survey was 25:49 minutes (median 17:48 minutes). Eight participants completed the survey in less than ten minutes (one in 4:07 min, the other seven in a time span of 7:53 min to 9:54 min). During a plausibility check of these answer sets, we found that the fastest participant answered all questions (except for the attention check) in a fixed pattern and was therefore excluded. For the other seven participants, we found no conspicuous answering scheme, and since their achieved number of correct answers range from 66% to 85%, we assume that these are valid responses that should be considered. Thus, from all participants, 16 were excluded because of the attention check questions, and one was excluded due to their low time and conspicuous answers. In the following, we refer to this subset of 163 participants as the participants of the second experiment.

The participants have an average experience in requirements engineering of 2.0, in HANFORPL of 1.1, and in formal logics of 3.0 on a self assessment Likert scale of 1 (not experienced at all) to 5 (very experienced). The median age group is 21 to 30. The native language of most participants is German (122). Of the participants, 14 answered to be native English speakers, and the remaining spread around 20 different Euro-

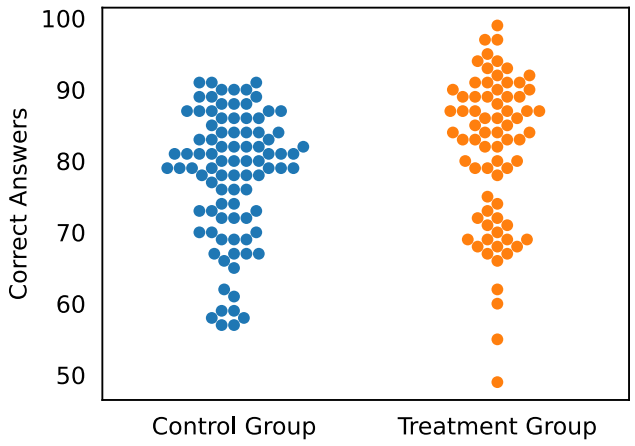


Fig. 10 Number of correct answers given for all patterns (questions A to F) in experiment two

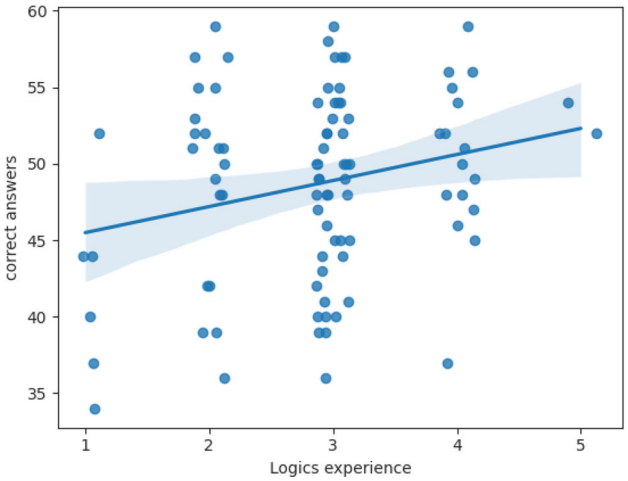


Fig. 11 The influence of experience in formal logics (x-axis) on the number of correct answers given in experiment two (y-axis), based on correct answers of questions A-D in the control group

pean and non-European languages. The participants have an average English language skill of 4.1 on a self assessment Likert scale of 1 (poor) to 5 (very good).

At the beginning of the survey, each participant was randomly assigned to either the control or the treatment group. From the 163 participants, 90 participants were assigned to the control group, and 73 to the treatment group.

Overall results. Table 6 shows the detailed performance of all participants in the control group and the treatment group, over all patterns and questions. In total, 76% (72% for questions A to D, 86% for questions E and F regarding the *common case*) of all questions were answered correctly by participants in the control group. Participants in the treatment group answered correctly with an accuracy of 80% (77% for questions A to D, 86% for questions E and F regarding the *common case*). Figure 10 shows the distribution of correct answers within the two groups.

Former training. To compare the influence of former training in different fields to the results of the first experiment, we only consider questions A to D (68 questions in total). Table 7 summarises the results regarding the influence of former training on the number of correct answers for the control and treatment group, respectively, including their Pearson correlation and significance values. There is no trend of training in HANFORPL leading to more correct answers. Only eight participants in the control group said to have experience with HANFORPL (answering > 1 in the related self assessment question). These participants answered slightly worse with an accuracy of 69% (on average 46.9 of 68 questions answered correctly), while the remaining 82 participants with no experience (answering 1 in the related self assessment question) reached an accuracy of 72% (on average 48.9 of 68 questions answered correctly). In the treatment group, the four participants with experience in HANFORPL answered correctly with an accuracy of 70% (47.3 out of 68 answered correctly), while the 69 participants with no experience reached an average accuracy of 77% (52.4 out of 68 answered correctly).

For experience in requirements engineering, there is no trend of training in requirements engineering leading to more correct answers. For the treatment group, there is at most a weak inverse, non-significant trend (Pearson correlation of $r(71) = -0.204$ with $p = 0.083$).

There is a weak, statistically significant trend of experience in formal logic leading to more correct answers (Pearson correlation of $r(88) = 0.240$ with $p = 0.023$ and $r(71) = 0.259$ with $p = 0.027$, in the control and treatment group, respectively), as can be seen in Fig. 11. In the control group, the 64 participants with knowledge in formal logics (answering > 2 in the related self assessment question) answered with 72% accuracy (on average 49.1 of 68 questions answered correctly). The 26 participants with little or no knowledge in formal logics (answering ≤ 2) answered slightly worse with 70% accuracy (on average 47.6 of 68 questions answered correctly). The result for the treatment group is slightly better: The 55 participants with knowledge in formal logics answered with 78% accuracy (on average 53.1 out of 68 questions). The 18 participants with little or no knowledge reached 72% accuracy (48.8 out of 68 questions).

Participants were asked to rate their English language skills at the beginning of the survey. We separate the participants of both control and treatment group based on their English language skills: Participants with poor to medium English skills (answering ≤ 3 in the related self assessment question) and those with better skills (answering > 3). The 71 participants in the control group with high English proficiency answered questions A to D correctly with an accuracy of 72% (49.2 out of 68 questions), while the 19 participants with lower language skills answered correctly with an accuracy of 69% (46.6 out of 68 questions). In the treatment group,

Table 6 Survey results of the second experiment, per pattern (listed in the order they occur in the survey) and question (columns A to F) of the *Control group* and *treatment group*, respectively. Questions A to D correspond to the questions in the first survey, questions E and F represent the additional trivial cases introduced for the second survey (Color table online)

	Pattern name	Average of correct answers (%)								
		A	B	C	D	E	F	$\text{Total}_{A\text{-}D}$	$\text{Total}_{E\text{-}F}$	Total
Control Group	Universality	92	100	98	98	98	97	97	98	97
	Invariance	63	96	77	87	97	98	81	98	86
	Initialization	96	96	97	92	96	92	95	94	95
	Persistence	72	100	73	95	92	93	85	92	88
	Precedence	90	36	87	83	96	86	74	91	80
	DurationBoundL	10	98	86	60	97	100	64	98	75
	DurationBoundU	93	8	98	94	94	98	73	96	81
	RecurrenceBoundL	95	81	80	85	92	93	85	92	88
	UniversalityDelay	48	88	36	90	94	22	66	58	63
	InvarianceBoundL2	23	30	96	62	94	92	53	93	66
	ResponseDelay	54	71	77	77	84	81	70	82	74
	ResponseDelayBoundL1	55	56	82	27	65	58	55	62	57
	ResponseBoundL1	17	93	37	72	63	74	55	68	59
	ResponseBoundL12	16	100	94	85	95	86	74	90	79
	EdgeResponseBoundL2	96	41	6	88	97	95	58	96	70
	EdgeResponseBoundU1	28	78	90	13	83	45	52	64	56
	EdgeResponseDelayBoundL2	95	76	74	47	75	95	73	85	77
Treatment Group	Universality	69	98	95	97	95	98	90	96	92
	Invariance	58	89	68	90	100	98	76	99	84
	Initialization	91	97	97	93	97	95	94	96	95
	Persistence	69	98	60	97	94	100	81	97	86
	Precedence	87	30	87	86	94	86	72	90	78
	DurationBoundL	10	100	83	61	95	98	64	96	74
	DurationBoundU	91	8	94	98	94	90	73	92	79
	RecurrenceBoundL	98	82	76	83	80	91	85	86	85
	UniversalityDelay	87	98	87	93	93	87	91	90	91
	InvarianceBoundL2	53	82	87	61	78	90	71	84	75
	ResponseDelay	50	84	80	75	89	76	72	82	76
	ResponseDelayBoundL1	57	58	78	39	68	57	58	62	60
	ResponseBoundL1	79	80	94	80	64	78	83	71	79
	ResponseBoundL12	83	98	97	68	82	61	86	72	82
	EdgeResponseBoundL2	95	83	15	90	93	93	71	93	78
	EdgeResponseBoundU1	84	57	56	17	73	52	54	62	56
	EdgeResponseDelayBoundL2	91	76	71	47	80	87	71	84	75

the results for both subgroups are a bit higher: The 56 participants with high English proficiency answered questions A to D correctly with an accuracy of 78% (52.9 out of 68 questions), while the 17 participants with lower proficiency answered correctly with an accuracy of 72% (49.3 out of 68 questions). There is at most a very weak, non-significant trend of English language skills on the number of correct answers for questions A to D, as shown in Fig. 12.

In the control group, the 67 participants with German native language answered with an accuracy of 74% (50.3 out of 68 questions). On average, their logic experience is 2.9. The three native English participants reached 63% (43.0 out of 68 questions, logic experience of 3.3), the three participants being native in both English and German reached 66% (45.0 out of 68 questions, logic experience of 3.7), and the 17 participants with any other native language reached 65% (43.9 out of 68 questions, logic experience of 2.6). In the treatment group, results are similar: The 51 participants with German native language reached 79% (54.0 out of 68 questions, logic experience of 3.1), the seven English natives

Table 7 Influence of former training in different fields on the correct answers in experiment two (based on correct answers of questions A-D)

Skill	Group	Level	n	$\text{Correct}_{\text{A-D}}$ (%)	Pearson Correlation
HANFORPL	Contr.	> 1	8	69	$r(88) = -0.058$ with $p = 0.588$
		$= 1$	82	72	
	Treat.	> 1	4	70	$r(71) = -0.117$ with $p = 0.323$
		$= 1$	69	77	
Requirements Eng.	Contr.	> 2	23	76	$r(88) = -0.033$ with $p = 0.756$
		≤ 2	67	76	
	Treat.	> 2	21	75	$r(71) = -0.204$ with $p = 0.083$
		≤ 2	52	81	
Formal Logic	Contr.	> 2	64	72	$r(88) = 0.240$ with $p = 0.023$
		≤ 2	26	70	
	Treat.	> 2	55	78	$r(71) = 0.259$ with $p = 0.027$
		≤ 2	18	72	
English Proficiency	Contr.	> 3	71	72	$r(88) = 0.129$ with $p = 0.226$
		≤ 3	19	69	
	Treat.	> 3	56	78	$r(71) = 0.068$ with $p = 0.565$
		≤ 3	17	72	

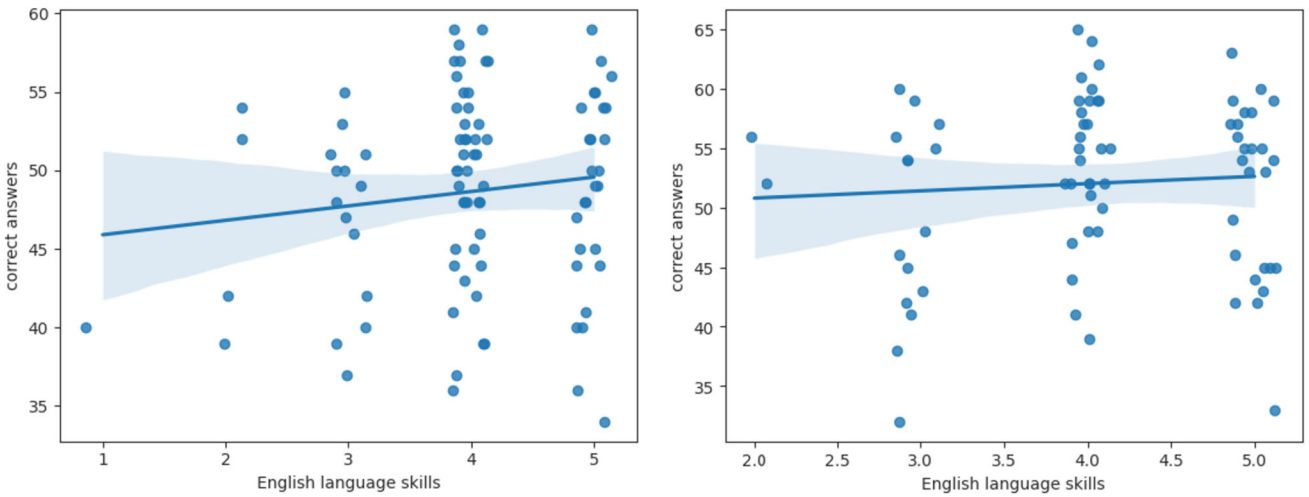


Fig. 12 The influence of English language proficiency (x-axis) on the number of correct answers given in experiment two (y-axis), based on correct answers of questions A-D in the control group (left) and treatment group (right)

62% (42.0 out of 68 questions, logic experience of 2.9), the one participant being native in both English and German 81% (55.0 out of 68 questions, logic experience of 3.0), and the 14 participants with any other native language reached 73% (49.9 out of 68 questions, logic experience of 3.1).

Phrases of interest. For the control group, we consider the results regarding the phrases of interest (Research Question **R3**) and ad-hoc phrases (cf. Sect. 3.5). Detailed results from the relevant questions can be found in Table 8. As in the evaluation of the first survey, we divided the participants into the group of participants with prior training in formal logics (answering > 2 in the related self assessment question; n = 64) and those with little or no former training (answering ≤ 2; n = 26).

Modified patterns. For the treatment group, we investigated whether modifications (cf. Table 5) suggested as a result of the first experiment are beneficial to the understandability of our patterns. Table 9 shows the ten modified patterns, the applied modifications, and the average difference of correct answers in the treatment group to the respective value in the control group.

Results are mixed. The highest overall improvements are achieved for the *UniversalityDelay* and the *ResponseBoundL1* pattern, with an increase of correct answers of 28% and 20%, respectively. The highest increase in correct answers for individual questions are given for questions *ResponseBoundL12-A* and *UniversalityDelay-F* with 67% and 65%, respectively. The highest deterioration occurred

Table 8 Correctness results of the *control group* in experiment two for the phrases of interest (as defined in the first experiment). Each row shows the according phrase id, the pattern containing the phrase and which question in the survey prompted that exact behaviour followed by the percentage of correct answers. Column N shows participants with little to no, column L with training in formal logics (Color table online)

ID	Related Pattern	Question	Average of correct answers (%)		
			N (26)	L (64)	Overall
prev	Precedence	C	84	89	87
prev	Precedence	D	73	87	83
afterw	ResponseBoundL1	D	65	75	72
afterw	EdgeResponseBoundU1	B	80	78	78
afterw*	ResponseBoundL1	C	30	40	37
afterw*	ResponseBoundL12	A	19	15	16
afterw*	EdgeResponseBoundU1	C	92	89	90
aam	UniversalityDelay	A	42	51	48
aam	UniversalityDelay	C	38	35	36
aam-cond	ResponseDelay	D	76	78	77
aam-cond	ResponseDelayBoundL1	C	76	84	82
obs	DurationBoundL	A	7	10	10
obs	DurationBoundU	B	0	12	8
obs	EdgeResponseBoundL2	C	3	7	6
obs	EdgeResponseBoundU1	D	15	12	13
obs+	DurationBoundL	C	76	90	86
obs+	DurationBoundU	D	92	95	94
obs+	EdgeResponseBoundL2	D	100	84	88
obs+	EdgeResponseDelayBoundL2	B	65	81	76
rec	ReccurrenceBoundL	B	88	78	81
antec	Invariance	C	73	79	77
antec	InvarianceBoundL2	D	88	87	87
atonce	Precedence	B	26	40	36
atonce	ResponseDelay	A	50	56	54
atonce	ResponseDelayBoundL1	A	50	57	55
atonce	ResponseDelayBoundL1	D	15	32	27
atonce	ResponseBoundL12	A	19	15	16

for the pattern *Universality*, *DurationBoundU*, and *EdgeResponseDelayBoundL2*, with −5% and −2% of correct answers, respectively. The highest deterioration for individual questions occur for questions *EdgeResponseBoundU1-B* and *ResponseBoundL12-F* with −34% and −25% of correct answers, respectively. Over all ten modified patterns, the number of correct answers in the treatment group was on average 5.8% higher than in the control group.

5.6 Discussion

Regarding Hypothesis **H1**, the improvements of HANFORPL increased the rate of correct answers slightly. Participants in the treatment group reached slightly more correct answers (79%) than participants in the control group (76%), while the accuracy for unmodified patterns showed a similar pattern in both groups. As argued in the discussion of our first experiment, the introduction of modifications may introduce new ambiguities. We will discuss the impact of each modification on the accuracy of participants (cf. Table 9): The addition of the word *persistently* lead to a strong decline in correct answers for question *Universality-A* (Fig. 13). This may be explained as, at first glance, the new phrasing may be read with an emphasis on the persistence part, i. e., it is fulfilled as long as any valuation of ***R*** holds persistently. The change nonetheless increased the understandability for all examples that obviously violated persistence (e. g., *UnversalityDelay-C* and *UnversalityDelay-F*) and thus not require any further evaluation of interval lengths or interaction of observables.

While changing the phrase *afterwards* to *immediatly* disambiguated some instances (e. g., *ResponseBoundL1-A* and *ResponseBoundL12-A*), it seemingly introduced a new ambiguity (e. g., *ResponseBoundL12-D, E* and *F*). This ambiguity is concerned with the question whether the effect ***S*** has to hold as soon as the condition is fulfilled or immediately at the beginning of the interval that will fulfil the condition in the future. For example, in the modified pattern *if* ***R*** *holds for at least* ***1*** *time unit, then* ***S*** *holds immediately*

Table 9 Difference Δ_i (%) of correct answers for modified patterns in the *treatment group* to corresponding patterns in the *control group* in experiment two. Modifications adding words are highlighted in green, removals in red (Color table online)

Pattern Name	Modification	Δ_A	Δ_B	Δ_C	Δ_D	Δ_E	Δ_F	Δ_{avg}
Universality	*persistently*	-23	-2	-3	-1	-3	+1	-5
DurationBoundL	*if* *once*	+0	+2	-3	+1	-2	-2	-1
DurationBoundU	*if* *once*	-2	+0	-4	+4	+0	-8	-2
UniversalityDelay	*persistently*	+39	+10	+51	+3	-1	+65	+28
InvarianceBoundL2	*immediately*	+30	+52	-9	-1	-16	-2	+9
ResponseBoundL1	*immediately* *afterwards*	+62	-13	+57	+8	+1	+4	+20
ResponseBoundL12	*immediately* *afterwards*	+67	-2	+3	-17	-13	-25	+3
EdgeResponseBoundL2	*if* *immediately* *once*	-1	+42	+9	+2	-4	-2	+8
EdgeResponseBoundU1	*if* *immediately* *once* *afterwards*	+56	-21	-34	+4	-10	+7	+0
EdgeResponseDelayBoundL2	*if* *immediately* *once*	-4	+0	-3	+0	+5	-8	-2

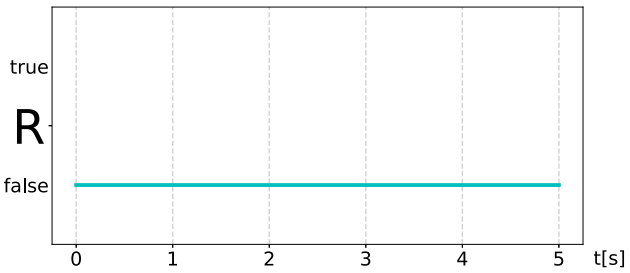


Fig. 13 Satisfying example behaviour used for question A of the *Universality* pattern: ***R*** *holds persistently*

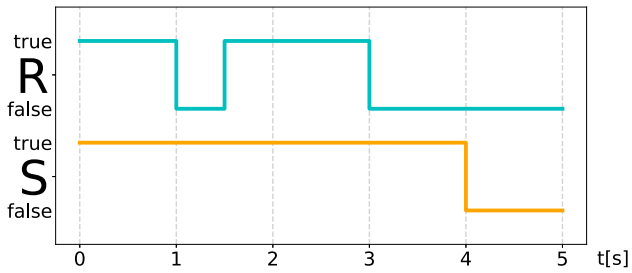


Fig. 14 Violating example behaviour used for question D of the *ResponseBoundL12* pattern: *if* ***R*** *holds for at least* ***1*** *time unit, then* ***S*** *holds immediately for at least* ***2*** *time units*

for at least ***2*** *time units.* (*ResponseBoundL12*), does ***S*** need to hold for 2 time units starting when ***R*** already holds for at least 1 time unit or at the point in time where ***R*** starts holding (and will continue to hold for at least 1 time unit). In the latter interpretation, the example behaviour shown in Fig. 14 is satisfying the requirement. In the intended interpretation, this example is a violation of the requirement since at time $t = 1.5$, ***R*** holds for 1 time unit, but starting at that point, ***S*** keeps holding for less than 2 time units.

Replacing *once* by *if* to indicate that the signal change may be reoccurring did overall slightly increase the understandability, but also introduced instances which suggest a similar shift to the scope of the requirement to the time at which the condition is evaluated (cf. Fig. 15).

Most of the modifications lead to mixed results, but overall supporting a slightly better understanding of the patterns (cf. Table 9). For a modification of the pattern language, we only consider changes that had a predominately positive impact

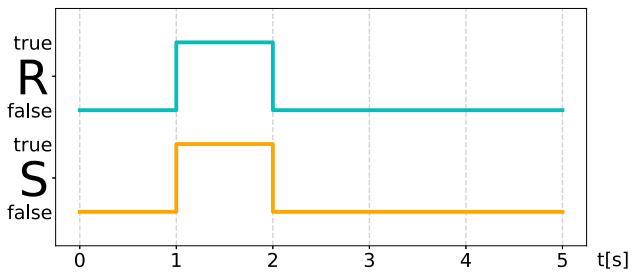


Fig. 15 Violating example behaviour used for question C of the *EdgeResponseBoundU1* pattern: *if* ***R*** *becomes satisfied and holds for at most* ***1*** *time unit, then* ***S*** *holds afterwards*

with emphasis on the most used patterns. For example, adding the *persistently* modification worsens the understanding of the *Universality* pattern slightly (−5% on average), but vastly increases the understandability for the *Universality-Delay* pattern (+28% on average), which is used far more frequent.

Regarding Hypothesis **H2**, the second experiment supports the assumptions that we made for the first experiment. On average, the *common cases* of requirements usage were answered correctly in 86% in the control group (+15% in relation to the non-common cases presented in questions A to D of experiment one), and 85% in the treatment group. Exceptions (e. g., *UniversalityDelay-F*), can be explained by the ambiguity removed through the inclusion of *persistently* as evident through the treatment group not making this mistake. Again, the number of correct answers decreases with more complex patterns, as there are far more possibilities for misunderstandings.

Regarding Hypothesis **H3**, the replication of results of the first experiment was mostly successful. The number of questions answered correctly (A to D) were 71% in this experiment (control group). This is slightly worse than in the original experiment, which may be an effect of the classroom study, as students were awarded course points and thus did not have the same interest in the study than voluntary requirements engineers from industry projects. The similarity extends to the phrases of interest (Hypothesis (b)), except for (antec). Regarding (antec), students performed much better (+19.5% on average) than their industrial counterparts. Here, the separation using logical training did not impact the results in any way (Table 8). Similarly, the effect of previous training in formal logics as the main predictor (Hypothesis (c)) for performance in understanding the requirements pattern decreased in strength (cf. Fig. 11). The decrease in effect size as well as vanishing of the (antec) errors can be explained by the selection of participants for this study. The selection of students in a number of computer science courses allowed us to gather a large number of participants, but narrowed the distribution of participants' background and experience in comparison to professional requirements engineers. Students in the sourced courses should have already received a thorough introduction into formal logics, thus being warned of the pitfalls of implication (antec). Students should also not have had too much exposure to temporal logics. This effect may further be supported by students lacking the background to give a good self-assessment of their logics or requirements engineering experience apart from the comparably small differences provided by their performance in maths and software engineering lectures, respectively. Remarkably, the best students in this experiment are only slightly worse than the best requirements engineers in experiment one.

Participants reported an average experience in HANFORPL of 1.1 on the self assessment Likert scale. Those claiming to have some experience in HANFORPL (value > 1 on Likert scale) performed worse, both in the control and treatment group, than those claiming to have no experience (cf. Table 7). Although not statistically significant, this inverse correlation is opposed to the results in the first experiment, where a slight but also non-significant trend of experience in HANFORPL leading to more correct answers was found. The limited group size does not allow for a meaningful conclusion to be drawn. Only twelve participants (eight in the control group and four in the treatment group) said to have some experience in HANFORPL. Combined with the only slight improvement attributed to training in the first experiment, this again may just hint the complexity of the underlying problem, emphasising the need for tools and documentation to verify and refresh one's mental model while working with requirements patterns, as well.

Regarding Hypothesis **H4**, proficiency in the English language, at least over some introductory level, does not have a strong effect on the understandability of HANFORPL. As one should have expected, there were few participants with little to no knowledge of English. The data in Fig. 12 shows, that at the language level of a student, English proficiency has little impact on understanding the pattern. This result is especially positive with regard to international teams and the low barrier of participation using HANFORPL. Figure 12 shows, that modifications in the treatment group had a positive effect on the understandability for participants with lower English skill.

We decided to include the results on native languages spoken by participants to encourage further investigation in this direction. Best results were achieved by German speakers, which might be an effect of the pattern language mainly being edited by non-native English speakers (e. g., [6–8, 14]).

6 Threats to validity

6.1 Internal validity

The threat of *Repeated Testing* is concerned with participants learning over the run of an experiment. As participants were not informed if their answers were correct, they should not have been able to gain information on the correct interpretation of the pattern. An acclimatization to the pattern language was intended though in order to prevent participants from being overwhelmed with the more complex pattern. Questions to query trivial cases added in the second experiment were intermixed randomly within each pattern group. Thereby, we lower the risk that participants of the second experiment learn from these cases before answering queries on the edge cases. As the surveys were performed either in an industrial or academical context, *Maturation*, i. e., changes over the duration of the survey can influence the results. We tried to keep the survey as short as possible in order to prevent tiring and impatience (or loss of participants) due to more pressing concerns. Nevertheless, we are aware that there is the risk of a fatigue effect. However, we cannot analyse the sole impact of fatigue, as, by the design of our study, the complexity of patterns is increasing throughout the survey.

The threat of *Instrumentation* is concerned with the influence of the experimental material itself on the results. We tried to make the examples of system behaviour as accessible as possible for the use by not formally trained participants [21]. Nonetheless, problems with phrases of interest starting in the beginning of the timing diagram may have suffered from a notion of the system too commonly associated with a real system, i. e., where there is always a previous state even if switched off. To guarantee that the questions themselves do not contain errors, all timing diagrams were automatically verified by the pattern simulator being part of HANFOR [6].

6.2 Construct validity

The threat of *Interaction of setting and treatment* is concerned with non-aligning circumstances of experiment and reality.

In fact, the experiments are presented in a form focusing on the patterns itself, not on realistic requirements. In a real setting, expressions over observables in pattern instantiations can add another layer of complexity, that is abstracted away, to get data on the pattern themselves. In reality, the correctness numbers can be much lower as requirements get considerably more complex. Nonetheless, the expression language is not likely to have interactions with the phrasing of the surrounding pattern.

As discussed in Sect. 5.2, students received bonus points for their participation in the second experiment. The motivation of the participants might hence be to complete the survey instead of giving considered answers. Therefore, the overall results may be lower than for a typical industrial setting.

6.3 External validity

The threat of *Interaction of selection and treatment* is concerned with the selection of non-representative participants. For the first survey, our participants were selected by contacting cooperation partners from different engineering divisions, and the chair mailing list. This way, we tried to spread the risk of convenience sampling over different businesses and person groups likely to be in a position or likely to be in the near future of using a requirements pattern language.

The participants in the second survey were mostly students. They were reached through first-degree contacts of the authors in the field of university teaching. No individual students were addressed, but participants in basic lectures both at the Bachelor's and Master's level, mainly in the field of computer science. Hereby, we tried to spread the risk of convenience sampling over different groups of students at different stages of their studies. Nonetheless, educational background and distribution over factors such as training in formal logics and requirements engineering experience are far more homogenous than this would be the case for industrial participants. This impacts the generalisability of results regarding those factors. The main focus of the second experiment, evaluation of the improvements, in a two-group design should only suffer little impact from this.

By selecting the participants for both studies from very different contexts, we lower the risk of a participant taking part in more than one of the surveys.

7 Related work

Winter et al. [3] conduct a survey on the understandability of quantifiers and their negation (such as *all*, *more than* or *at least*) in natural language requirements. Results show, that there are significant effects on reading speed and error rate between the different quantifiers and their negated forms. Based on the results, advice for writing requirements is given. This recent work shows the relevance of investigations into the understanding of requirements in general. Phrasings are chosen once and reproduced in each instantiation, i. e., any problem introduced to a pattern is multiplied over a requirements specification. Therefore, ensuring understanding by a broad audience is even more relevant.

Giannakopoulou et al. [22] address the problem of pattern understanding by presenting several representations of the instantiated requirement, both graphical and as formal logic, e. g., LTL. This is a necessary support for error recovery by comparison to the intended result, while the pattern language should itself prevent errors in the first place by being aligned with the intuitive understanding of the patterns.

A different approach is taken by Moitra et al. [23], designing the requirements language in the style of a programming language. This surely aligns the intuitive understanding with stakeholders from a computer science background, but may exclude other stakeholders entirely, because of the condensed syntax.

8 Conclusion

Over the two experiments presented in this paper, we demonstrated how an inquiry on the alignment of the formal semantics of HANFORPL and the intuitive understanding of requirements engineers can help to understand and improve the pattern language. We followed an exploratory approach to locate problematic phrases in the pattern language and validated the suggested changes as well as assumptions made for the first experiment, in a second experiment.

Almost half of the patterns considered in the survey are contained in the SPS [8]. Parts of the results can therefore be generalized to SPS-like languages.

The analysis results are positive, and the pattern language performed very well in hiding the formal complexity behind

intuitively understandable sentences for both, requirements engineers within the industry, and university students in computer science. Nonetheless, the language contains several phrases that lead to near random decisions, and misconceptions of logic can lead to misinterpretations that cannot be mitigated entirely by phrasing. For suggested phrasings, the understandability of the pattern language increased slightly. Several large increases in the understandability partly came at the cost of new, but less impactful, ambiguities. The unexpected side effects show that suggested changes, although seeming highly beneficial to understanding, have to be verified. Thus, further consideration will be necessary to apply improvements to HANFORPL, and to decide what errors are best prevented by training and documentation. Due to the complexity of requirements and the underlying problem of formulating the behaviour of a whole system, intuitive understanding of a pattern language should only be one building block amongst many, also including tools supporting quick verification of one's mental model, good documentation and processes that mitigate the random misunderstanding.

Further investigations into HANFORPL should include patterns and their scopes in order to evaluate if scopes are clearly understood and to see if scopes alter the interpretation of phrases within patterns.

Acknowledgements We thank all participants, for taking part in the study.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability Availability of data and materials: The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

1. Yang H, Roeck AND, Gervasi V, Willis A, Nuseibeh B (2011) Analysing anaphoric ambiguity in natural language requirements. Requir Eng 16(3):163–189
2. Berry DM, Kamsties E (2005) The syntactically dangerous all and plural in specifications. IEEE Softw 22(1):55–57
3. Winter K, Femmer H, Vogelsang A (2020) How do quantifiers affect the quality of requirements? REFSQ. Lecture notes in computer science, vol 12045. Springer, Berlin, pp 3–18
4. Dietsch D, Langenfeld V, Westphal B (2020) Formal requirements in an informal world. In: 2020 IEEE workshop on formal requirements (FORMREQ), pp 14–20. IEEE
5. Konrad S, Cheng BHC (2005) Real-time specification patterns. In: ICSE, pp 372–381. ACM
6. Becker S, Dietsch D, Hauff N, Henkel E, Langenfeld V, Podelski A, Westphal B (2021) Hanfor: semantic requirements review at scale. In: REFSQ Workshops. CEUR Workshop Proceedings, vol. 2857. CEUR-WS.org
7. Henkel E, Hauff N, Eber L, Langenfeld V, Podelski A (2023) An empirical study of the intuitive understanding of a formal pattern language. REFSQ. Lecture notes in computer science, vol 13975. Springer, Berlin, pp 21–38
8. Post AC (2012) Effective correctness criteria for real-time requirements. PhD thesis, University of Freiburg
9. Post A, Hoenicke J (2012) Formalization and analysis of real-time requirements: a feasibility study at BOSCH. VSTTE. Lecture notes in computer science, vol 7152. Springer, Berlin, pp 225–240
10. Langenfeld V, Dietsch D, Westphal B, Hoenicke J, Post A (2019) Scalable analysis of real-time requirements. In: RE, pp 234–244. IEEE
11. Dunning D (2011) The dunning-Kruger effect: on being ignorant of one's own ignorance. Advances in experimental social psychology, vol 44. Elsevier, Amsterdam, pp 247–296
12. Berry DM (2017) Evaluation of tools for hairy requirements and software engineering tasks. In: IEEE 25th international requirements engineering conference workshops, RE 2017 workshops, Lisbon, Portugal, September 4-8, 2017, 284–291. IEEE Computer Society. https://doi.org/10.1109/REW.2017.25
13. Gervasi V, Zowghi D (2005) Reasoning about inconsistencies in natural language requirements. ACM Trans Softw Eng Methodol 14(3):277–330
14. Langenfeld V (2023) Formalisation and analysis of system requirements. PhD thesis, University of Freiburg. https://doi.org/10.6094/UNIFR/240644
15. Post A, Menzel I, Hoenicke J, Podelski A (2012) Automotive behavioral requirements expressed in a specification pattern system: a case study at BOSCH. Requir Eng 17(1):19–33
16. Fischbach J, Frattini J, Mendez D, Unterkalmsteiner M, Femmer H, Vogelsang A (2021) How do practitioners interpret conditionals in requirements? PROFES. Lecture notes in computer science, vol 13126. Springer, Berlin, pp 85–102
17. Bjørner D, Havelund K (2014) 40 years of formal methods–some obstacles and some possibilities? FM. Lecture notes in computer science, vol 8442. Springer, Berlin, pp 42–61
18. Westphal B (2021) On education and training in formal methods for industrial critical systems. FMICS. Lecture notes in computer science, vol 12863. Springer, Berlin, pp 85–103
19. Easterbrook SM, Chechik M (2002) Guest editorial: special issue on model checking in requirements engineering. Requir Eng 7(4):221–224
20. Katis A, Mavridou A, Dimitra Pressburger T, Schumann J (2022) Capture, analyze, diagnose: realizability checking of requirements in FRET. CAV 2022 LNCS, vol 13372. Springer, Berlin, pp 490–504. https://doi.org/10.1007/978-3-031-13188-2_24
21. Dietsch D, Feo-Arenis S, Westphal B, Podelski A (2011) Disambiguation of industrial standards through formalization and graphical languages. In: RE, pp 265–270. IEEE Computer Society
22. Giannakopoulou D, Pressburger T, Mavridou A, Schumann J (2020) Generation of formal requirements from structured natural language. REFSQ lecture notes in computer science, vol 12045. Springer, Berlin, pp 19–35
23. Moitra A, Siu K, Crapo AW (2018) Towards development of complete and conflict-free requirements. In: RE, pubIEEE, pp 286–296