Empirical Analysis of a Proposed Process Granularity Heuristic (Experimental Details)

Joachim Melcher and Detlef Seese

Institut AIFB, Universität Karlsruhe (TH)
76128 Karlsruhe, Germany
{melcher|seese}@aifb.uni-karlsruhe.de

July 14, 2009

Abstract

Choosing the adequate size of process activities (process granularity) is a problem during process design. Vanderfeesten et al. *have proposed a heuristic based on a process granularity metric and postulated a hypothesis concerning error probability about its use. The heuristic prefers process designs with high cohesion and low coupling—a principle originating in software engineering.*

In this paper, we present an experimentation system consisting of a small web-based workflow engine for empirically analyzing the error probability hypothesis. Furthermore, the results of a conducted experiment with 165 students using this experimentation system are reported. The experiment does not support the hypothesis. Instead, an alternative error probability model explaining the results is suggested.

1 Introduction

During the design phase of a workflow, one has to choose the adequate size of process activities (process granularity). Recently, Vanderfeesten *et al.* have proposed a process granularity metric inspired by software engineering [8, 10]. This metric measures the ratio between process coupling and cohesion. Based on this metric, Vanderfeesten *et al.* have suggested a heuristic for selecting between different process designs which prefers designs with high cohesion and low coupling. They have also postulated the hypothesis that those process designs are less error-prone during process execution. As they do not give an empirical validation of their heuristic and hypothesis, it is still no valid prediction system as explained in Section 2.

In this paper, we present an experimentation system for analyzing the hypothesis and report the results of a conducted experiment with 165 students using this experimentation system. Additionally, we suggest an alternative error probability model.

The remainder of this paper is organized as follows: Important basics about measurement and prediction systems are presented in Section 2. In Section 3, we give a short introduction into the process granularity heuristic proposed by Vanderfeesten *et al.* Our experimentation system for analyzing the hypothesis about error probability postulated by Vanderfeesten *et al.* is presented in Section 4. The conducted experiment and its results are shown in Section 5. The paper gives a conclusion and presents possible future work (Section 6).

2 Measurement and Prediction Systems

2.1 Definitions

The area of process measurement is inspired by the works and results of software measurement. There, many theoretical fundamentals were identified as important. Fenton and Pfleeger give a good overview in [1]. In [4], we show that these theoretical basics are also essential for process measurement—even so we had to notice that many of these findings are still ignored.

According to Fenton and Pfleeger, there are two main types of measurement:

Definition 1 (Measurement systems) *Measurement*

CC BY NC ND

This work is licensed under the Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 Germany License. To view a copy of this license, visit `http://creativecommons.org/licenses/by-nc-nd/3.0/de/`.

systems are used to assess an existing *entity by numerically characterizing one or more of its attributes [1, p. 104].*

Definition 2 (Prediction systems) *Prediction systems are used to predict some attribute of a* future *entity, involving a mathematical model with associated prediction procedures [1, p. 104].*

Besides the use for *future* entities as stated in the definition of Fenton and Pfleeger, prediction systems can also be used to predict some attribute of an *existing* entity that is measurable only in a very laborious manner.

In [4], we show how the idea of prediction systems can be transfered to process measurement (see Figure 1):

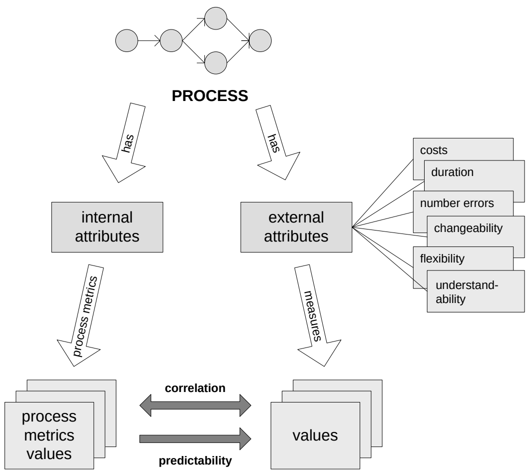


Figure 1: Prediction systems adapted to process measurement.

A process has *internal* and *external* attributes.

Internal attributes are those that can be measured purely in terms of the process separate from its behavior [1, p. 74]. Most proposed process metrics measure structural properties (internal attributes). The proposed granularity metric is an example for such a metric.

External attributes are those that can be measured only with respect to how the process relates to its environment [1, p. 74]. Examples are costs, time, understandability—and especially important for this paper—the number of errors.

2.2 Validation

Before a prediction system can be used, it has to be validated. A valid prediction system consists of two metrics both being valid measurement systems. Valid measurement systems must fulfill the following two properties:

Reliability Metric values obtained by different observers of the same process have to be consistent. Kan gives a good example [2, pp. 70–71]: If one wants to measure the height of a person, the measurements should be taken at a special time of day (e. g., in the morning) and always barefooted. Otherwise, the values of the same person could vary a lot.

Validity According to Kan [2, pp. 71–72], validity can be classified into *construct validity* and *content validity*. The first checks whether the metric really represents the theoretical concept to be measured (e. g., is church attendance a good metric for religiousness?). The second checks whether the metric covers the range of meanings included in the concept (e. g., a test of mathematical ability for elementary pupils cannot be limited to addition but should also include subtraction, multiplication, division and so forth).

The goal of a validation of a prediction system is to show a correlation between the process metric values and the corresponding external attribute in question. As Fenton and Pfleeger state, “rather than being a mathematical proof, validation involves confirming or refuting a hypothesis” [1, p. 104].

As already stated in Section 1, Vanderfeesten *et al.* do not give such a validation for their proposed heuristic. Consequently, the main goal of this paper is to examine whether the heuristic is a valid prediction system.

3 Process Granularity Heuristic

3.1 Information Element Structure

The proposed process metrics and the suggested granularity heuristic of [8, 10] are based on the methodology of product-based workflow design [6, 7, 9]. In this approach, a process is not originally represented by a process graph (e. g., using event-driven process chains, Petri nets or workflow nets). Instead, modeling starts one step earlier with a so called *information element structure* (see Figure 3).

The nodes of this graph are *information elements* which represent data that is needed during process execution. The directed edges stand for *operations* on information elements. Each operation has one or more input information elements and one output information element. The output of an operation can be the input of another one.

There are different types of operations: The simplest one has exactly one input information element (e. g., only element 18 is needed for computing element 38 in Figure 3). The second type is an AND construct which has at least two input nodes (e. g., elements 12 and 13 for the computation of element 18). The last type is an XOR construct. Here, several possible operations for the computation of an information element exist. Each operation has a boolean constraint so that only exactly one alternative is executable during each process execution. Looking at Figure 3, element 42 can be either computed by an operation using elements 39, 40 and 41 as inputs or by another operation with element 27 as input.

During the next modeling step, the information element structure is partitioned into different *activities* (consisting of a number of operations on information elements) which together form a traditional process graph. The activities A to G depicted in Figure 6, for example, are a partition of the information element structure of Figure 3 and can be combined to the process shown in Figure 5(a).

In [3], Kress *et al.* present an algorithm for directly executing the information element structure.

3.2 Process Granularity Metric

In this paper, we use the definitions of [10, pp. 426–429]—omitting the references to resource classes or roles which are able to execute the operations and activities as they are not relevant for our analysis.

Definition 3 (Operations structure) *An operations structure is a tuple (D, O) with*

- *a set D of information elements which are processed and*
- *a set $O \subseteq D \times \mathcal{P}(D)$ of operations on these information elements, such that there are no "dangling" information elements and no value of an information element depends on itself (also not indirectly).*

Based on an operations structure, the contained information elements and operations are partitioned into different activities.

Definition 4 (Activitiy) *An activity $T \subseteq O$ on an operations structure (D, O) is a set of operations.*

As a shorthand, the notation

$$\hat{T} := \bigcup_{(p,cs)\in T} (\{p\} \cup cs)$$

for the information elements processed in an activity T is introduced.

The different activities can be combined to a process which processes and computes the information elements of the operations structure in a valid sequence. For details on how to specify the control flow or how to check the correctness and soundness of the process, the reader is referred to [6]. For our purpose, the following definition is sufficient.

Definition 5 (Process) *A process on an operations structure (D, O) is a set S of activities on this operations structure.*

Based on these notations, metrics for *process cohesion* and *coupling* can be defined.

Process cohesion consists of two components. The first one, *activity relation cohesion* (see Definition 6), quantifies how much the operations of an activity are related. For that purpose, it measures the average overlap of operations. Two operations overlap if they share input or output information elements.

The second cohesion component, *activity relation cohesion* (see Definition 7), measures which fraction of information elements of an activity are used in more than one operation.

The total cohesion of an activity is simply the product of its relation and information cohesion.

Definition 8 (Activity cohesion) *For an activity T on an operations structure (D, O), the activity cohesion $c(T)$ is defined as*

$$c(T) := \lambda(T) \cdot \mu(T) \quad . \tag{3}$$

The overall cohesion of a process is computed by the average activity cohesion.

Definition 9 (Process cohesion) *For a process with set S of activities on an operations structure (D, O), the process cohesion ch is defined as*

$$ch := \frac{\sum_{T\in S} c(T)}{|S|} \quad . \tag{4}$$

Process coupling quantifies how strong the activities of a process are connected to each other. Two activities are connected if they share at least one information element. The coupling metric measures the fraction of connected activity pairs.

Definition 10 (Process coupling) *For a process with set S of activities on an operations structure (D, O), the process coupling cp is defined as*

$$cp := \begin{cases} \frac{|\{(T_1,T_2)\in S\times S | T_1\neq T_2 \wedge (\hat{T}_1\cap\hat{T}_2)\neq\varnothing\}|}{|S|\cdot(|S|-1)} & \textit{for } |S| > 1 \\ 0 & \textit{for } |S| \leq 1 \end{cases} \quad . \tag{5}$$

Finally, Vanderfeesten *et al.* define a process coupling/cohesion ratio which serves as a process granularity metric.

Definition 11 (Process coupling/cohesion ratio) *For a process with set S of activities on an operations structure (D, O), the process coupling/cohesion ratio ρ is defined as*

$$\rho := \frac{cp}{ch} \quad . \tag{6}$$

Definition 6 (Activity relation cohesion) *For an activity T on an operations structure (D, O), the activity relation cohesion $\lambda(T)$ is defined as*

$$\lambda(T) := \begin{cases} \frac{|\{((p_1,cs_1),(p_2,cs_2))\in T\times T|((\{p_1\}\cup cs_1)\cap(\{p_2\}\cup cs_2))\neq\varnothing\wedge p_1\neq p_2\}|}{|T|\cdot(|T|-1)} & \text{for } |T|>1 \\ 0 & \text{for } |T|\leq 1 \end{cases} . \quad (1)$$

Definition 7 (Activity information cohesion) *For an activity T on an operations structure (D, O), the activity relation cohesion $\lambda(T)$ is defined as*

$$\mu(T) := \begin{cases} \frac{|\{d\in D|\exists((p_1,cs_1),(p_2,cs_2))\in T\times T:(d\in((\{p_1\}\cup cs_1)\cap(\{p_2\}\cup cs_2))\wedge p_1\neq p_2\}|}{|\hat{T}|} & \text{for } |\hat{T}|>0 \\ 0 & \text{for } |\hat{T}|=0 \end{cases} . \quad (2)$$

3.3 Process Granularity Heuristic

According to Vanderfeesten and Reijers, an important issue in process design is "the proper size of the individual activities in a process (the process granularity)" [8, p. 290]. The heuristic presented in [8, 10] is thought to help designers "to select from several alternatives the process design that is strongly cohesive and weakly coupled" [10, p. 420].

Vanderfeesten *et al.* state that the proposed metrics and the heuristic are inspired by software engineering "where an old design aphorism is to strive for *strong cohesion, and loose coupling*" [10, p. 421].

Consequently, the statement of the heuristic is that a workflow design with a smaller value of the process granularity metric (process coupling/cohesion ratio) of Definition 11 is to be preferred over another one with a larger value. Yet, it does not describe how different alternative workflow designs can be found. [10, p. 429]

Vanderfeesten *et al.* establish the following hypothesis about the implications of their heuristic [10, pp. 425–426]:

Hypothesis 1 *The smaller the value of the process granularity metric of a workflow design, the smaller the probability of run-time mistakes.*

Instead of an empirical validation of this hypothesis, they only give some arguments as a motivation [10, p. 426]:

- "A *loose coupling* of activities will result in few information elements that need to be exchanged between activities [. . .], reducing the probability of run-time mistakes."
- "*Highly cohesive* activities [. . .] are likely to be understood and performed better by people than large chunks of unrelated work being grouped together."

4 Experimentation System

In order to analyze the hypothesis of Vanderfeesten *et al.* about their heuristic, an appropriate experimentation system had to be created. Three main requirements were identified for this system:

1. automatization of the experiments and the subsequent analysis,
2. comparability of different experiment runs with different process models (consequently, special domain knowledge must not be necessary and the actual process goal has to be abstracted from—concentrating only on process structure and granularity) and
3. cooperation of several subjects with different roles during process instance execution.

A computer-based (cf. requirement 1) experimentation system was created which is described in the remainder of this section.

We decided to use very abstract operations structures for the experimentation system (cf. requirement 2): Each information element represents a single variable of type boolean, integer or double. Operations are functions with the variables corresponding to the operation's input information elements as input parameters. According to the variable types, these functions consist of addition, subtraction, multiplication or logical AND, OR, XOR and negation. Activities consist of sets of corresponding functions which can depend on each other in a non-cyclic manner. See Figure 4 for an example of such abstract operations.

Core of the experimentation system is a small web-based workflow engine allowing several subjects to work together on process instance execution (cf. requirement 3). It is written in Java using Apache Tomcat and runs on a central server. The subjects connect to that workflow engine using a standard browser.

The workflow engine controls the execution of process instances. Each experimental subject is assigned to a resource role[1]. When an activity becomes executable, it is delegated in first-come, first-served order to the next free subject with the corresponding role. The functions of that activity together with the values of the input parameters of the basic functions[2] are displayed on the subject's screen (see Figure 2). The subject has to enter the computed values into special text fields. By clicking a button, the computed values are sent to the workflow engine for further processing. At XOR splits, the workflow engine routes automatically by evaluating the boolean constraint expressions for the different branches.

During execution, the following data is logged:

- start and end time of each activity and each process instance,
- correct or incorrect activity execution[3] and
- correct or incorrect process instance execution[4].

5 Experimental Analysis and Results

5.1 Experimental Design

In order to test Hypothesis 1 (error probability), we conducted an experiment using the experimentation system described in Section 4.

In the experiment, the independent variable is the process granularity metric value of a process design, the response variables are the rates of incorrectly executed process instances and activities.

For this experiment, we used the information element structure depicted in Figure 3, which is presented as an example in [8, 10]. The structure was used in the abstract fashion described in Section 4. So, only the structural properties—and consequently the process metric values—stayed unchanged. The used operations are shown in Figure 4.

Based on the information element structure, the three different process design alternatives (Figure 5) already proposed in [8, 10] were used. The respective partition into activities is shown in Figure 6.

[1]Consequently, one needs at least as many subjects as resource roles in the executed process instances.

[2]Basic functions are functions for which the values of its input parameters are not computed by other functions of the same activity.

[3]The correctness of an activity execution is assessed based on the values of its input parameters. So, if the values of the input parameters are incorrect—caused by an earlier activity—but the output value of the function is correctly computed based on these input values, the activity execution is assessed as correct.

[4]A process instance execution is assessed as incorrect if at least one of its activities was executed incorrectly.

Table 1: Metric values for the three process alternatives.

	cp	ch	ρ
alternative 1	0.714	0.183	3.9
alternative 2	0.611	0.105	5.8
alternative 3	0.8	0.114	7.0

The process metric values for process coupling (cp), cohesion (ch) and granularity (ρ) of the process design alternatives are listed in Table 1. So, there are three levels of the independent variable in the experiment. According to the heuristic, alternative 1 should be preferred as it has the smallest value of ρ. Following Hypothesis 1, it should also have the smallest error probability.

We created a set of ten process instances which was used for all process design alternatives. All these process instances were executable from the start of the experiment and were processed in the same order. The instances had different values for its basic information elements[5]. If they were correctly executed, the first and last instances of the set were routed directly from activity C to G at the XOR split—the others had to take the branch with all the other activities.

For each process design alternative, we used several teams which each processed that same set of process instances in order to average the team results. Each team consisted of exactly as many subjects as there are activities in its process alternative. As the subjects executing activity AE in alternative 3 have much more work than other subjects, we used two different teams for that alternative to analyze the effect of this possible bottleneck: The first got the "normal" six subjects (number of activities in alternative 3)—the second got seven subjects (two for the resource role of activity AE).

165 Business Engineering undergraduate students of the University Karlsruhe participated in our experiment. Participation was voluntary—participating students got bonus points for the accreditation to their final exam. They had no special training in the area of workflows, but the necessary mathematical knowledge for the used abstract functions (cf. Section 4). As the subjects were randomly assigned to the resource roles within the different teams for the different process alternatives, individual differences should be balanced. Finally, there were six teams for alternative 1, alternative 3 with six subjects and alternative 3 with seven subjects, respectively, as well as five teams for alternative 2.

[5]Basic information elements are information elements whose values are not computed by any operation. Instead, their values have to be given for each process instance before the execution.

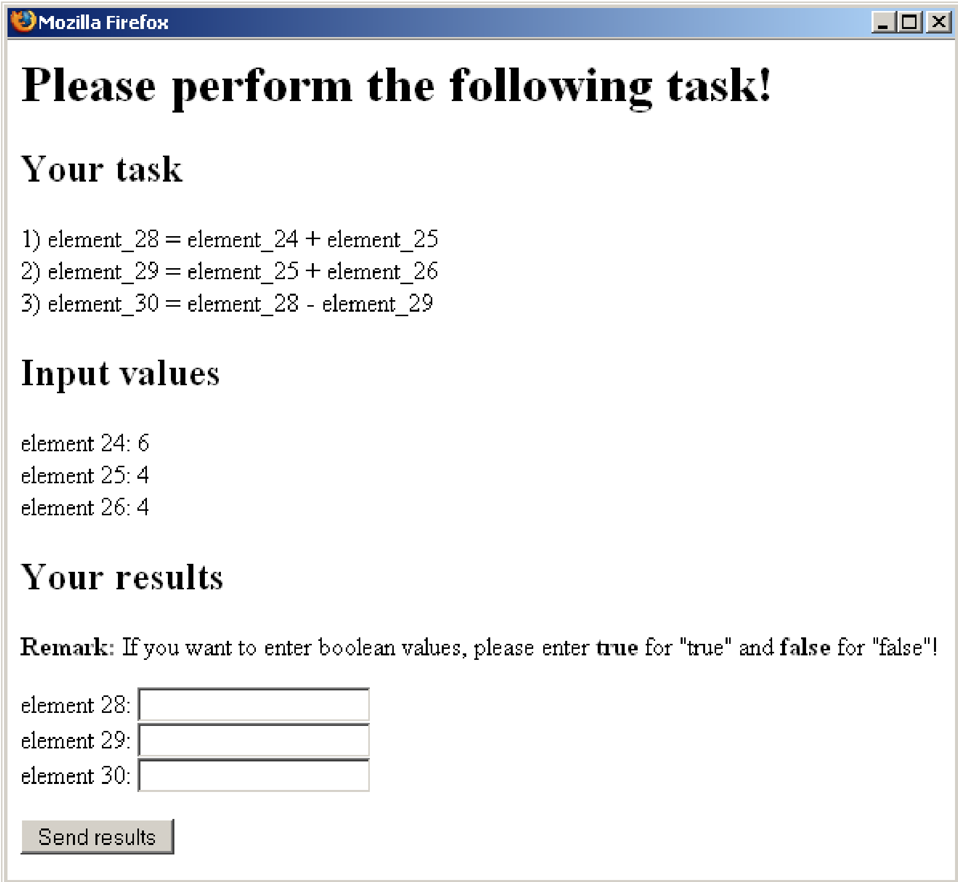


Figure 2: Screenshot of subject's web browser.

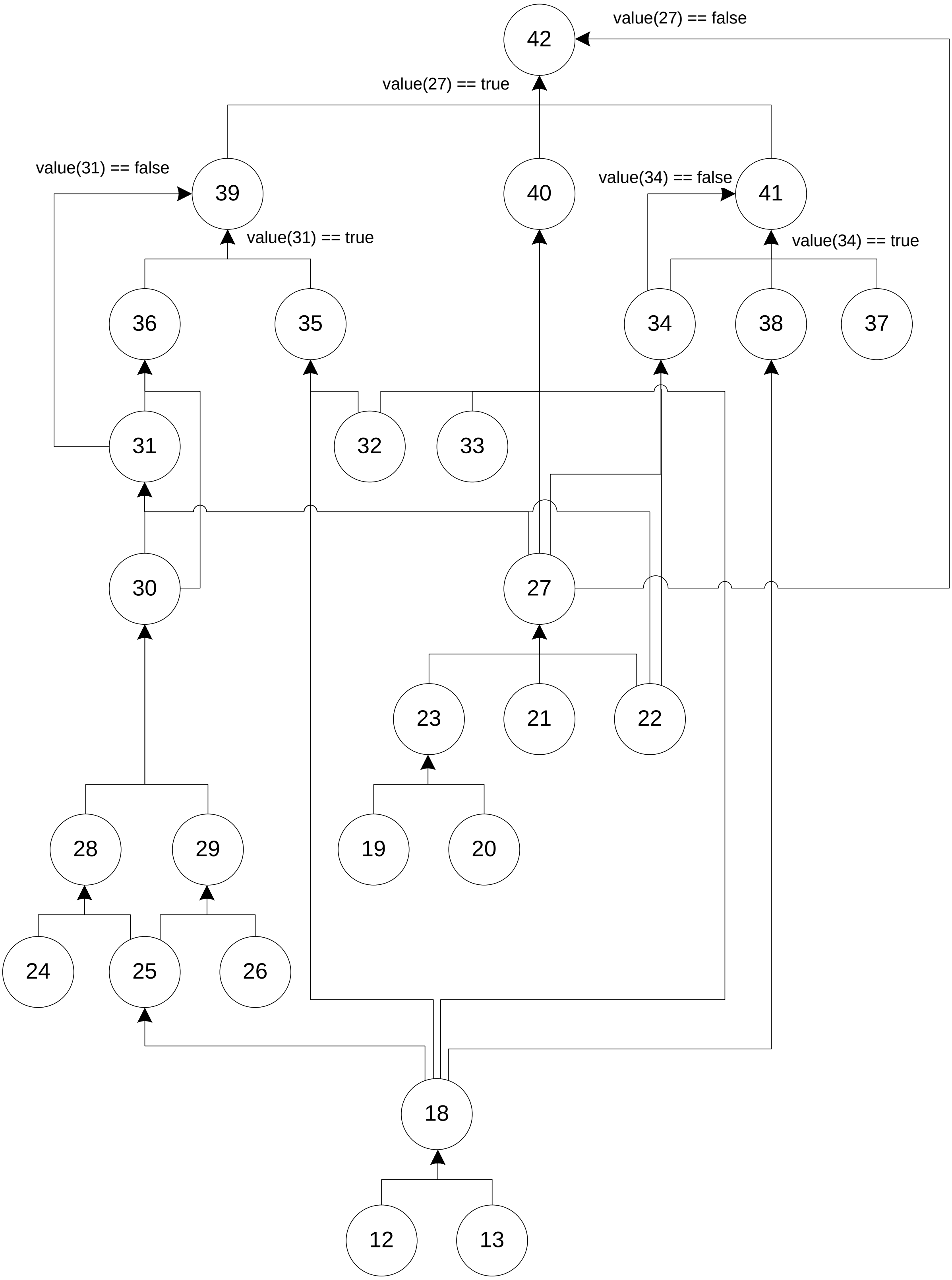
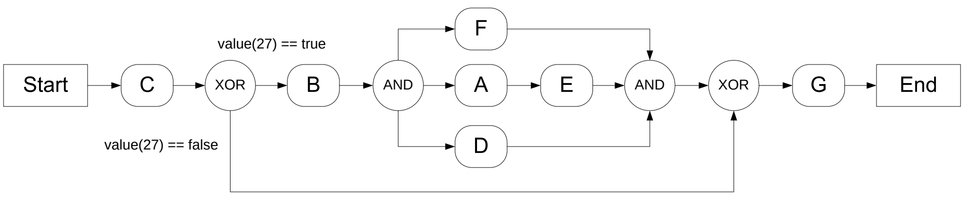


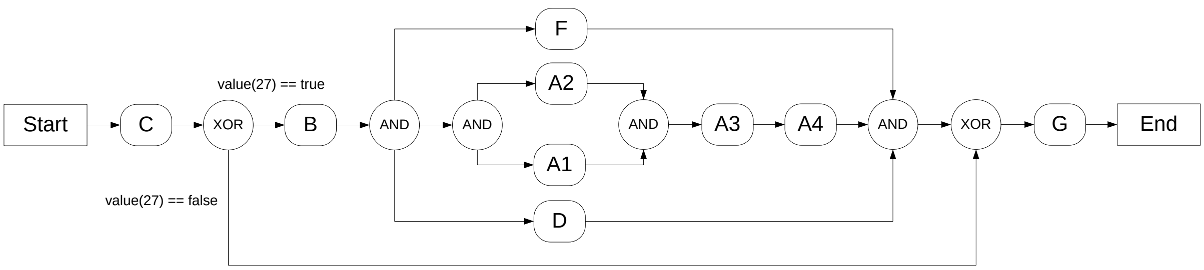
Figure 3: Information element structure used in experiment.

$$
\begin{aligned}
v_i(18) &:= v_i(12) + v_i(13) \\
v_i(23) &:= v_i(19) + v_i(20) \\
v_i(25) &:= v_i(18) - 1 \\
v_b(27) &:= v_i(23) > [v_i(21) + v_i(22)] \\
v_i(28) &:= v_i(24) + v_i(25) \\
v_i(29) &:= v_i(25) + v_i(26) \\
v_i(30) &:= v_i(28) - v_i(29) \\
v_b(31) &:= v_b(27) \wedge [v_i(22) \neq v_i(30)] \\
v_b(34) &:= v_b(27) \wedge [v_i(22) \neq 0] \\
v_i(35) &:= v_i(18) + v_i(32) \\
v_b(36) &:= v_b(31) \wedge [v_i(30) > 20] \\
v_i(38) &:= v_i(18) + 1 \\
v_b(39) &:= \begin{cases} v_b(31) & \text{for } v_b(31) = \text{FALSE} \\ v_b(36) \vee [v_i(35) \neq 0] & \text{for } v_b(31) = \text{TRUE} \end{cases} \\
v_b(40) &:= n_b(27) \wedge \{[v_i(18) + v_i(32)] > v_i(33)\} \\
v_b(41) &:= \begin{cases} v_b(34) & \text{for } v_b(34) = \text{FALSE} \\ v_b(34) \wedge [v_i(37) > v_i(38)] & \text{for } v_b(34) = \text{TRUE} \end{cases} \\
v_b(42) &:= \begin{cases} \neg v_b(27) & \text{for } v_b(27) = \text{FALSE} \\ [v_b(39) \wedge v_b(40)] \vee v_b(41) & \text{for } v_b(27) = \text{TRUE} \end{cases}
\end{aligned}
$$

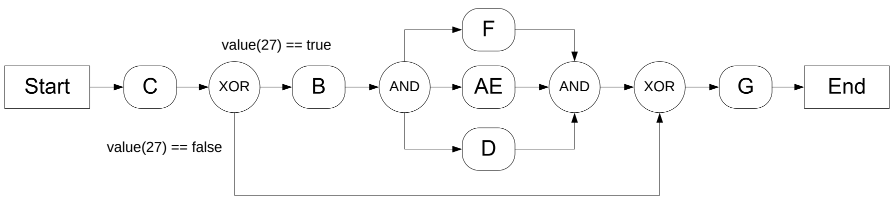
Figure 4: Operations used in experiment. $v_i(18)$ stands for the integer variable representing information element 18, $v_b(27)$ for the boolean variable representing information element 27.



(a) Process alternative 1.



(b) Process alternative 2.



(c) Process alternative 3.

Figure 5: Three different process alternatives used in experiment.

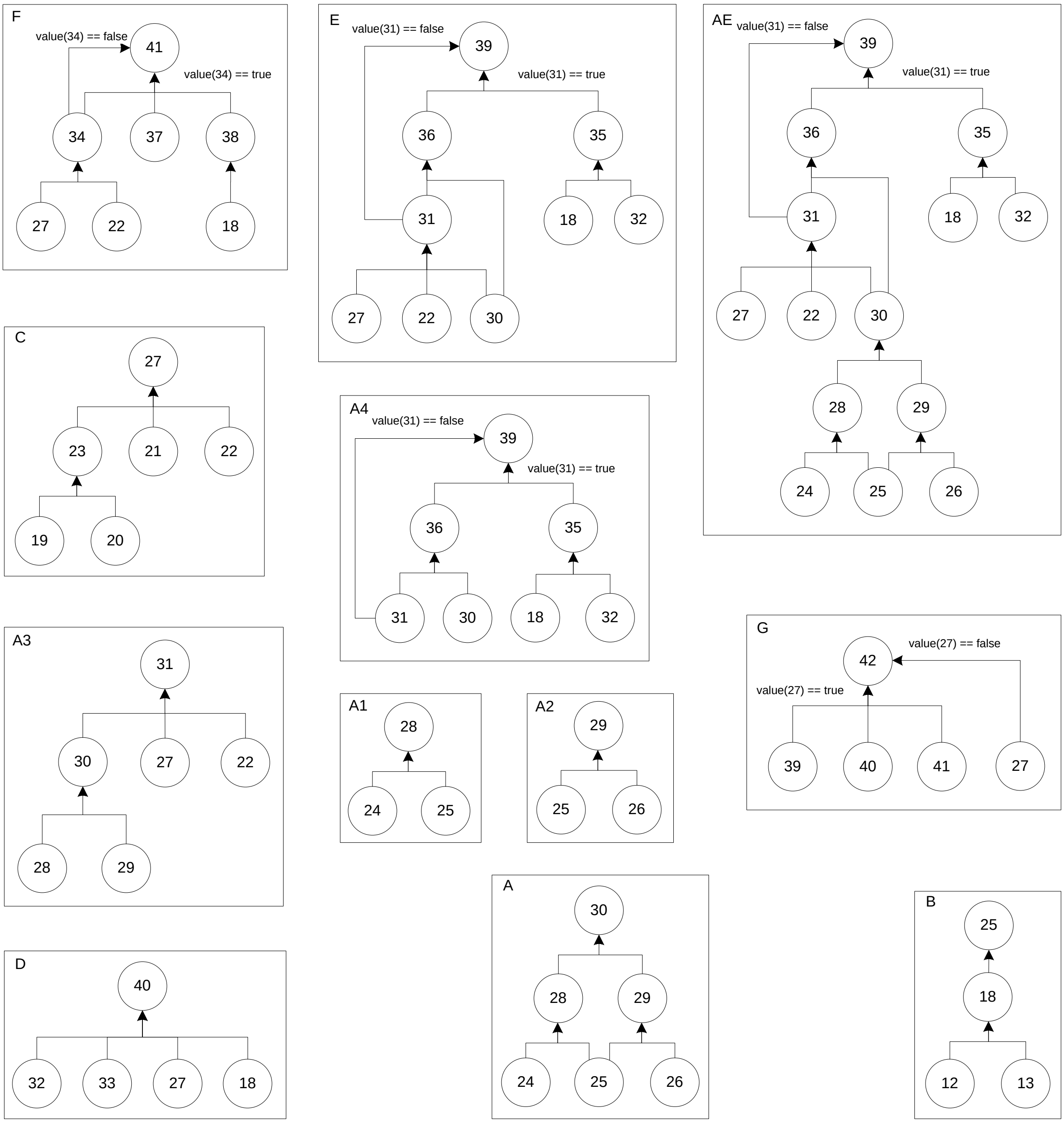


Figure 6: Partitioning of the information element structure in smaller activities.

5.2 Results

The number of incorrect process instances and activities (over all teams) for the different process alternatives are shown in Table 2.

First, we checked whether there is a significant difference between alternative 3 with six and seven subjects. For that purpose, we used Pearson's chi-square test [5, pp. 643–648]. The null-hypothesis that the numbers belong to the same distribution could not be rejected on the $\alpha = 0.05$ level. Consequently, both cases were mixed together for the further analysis (row "sum alt. 3" in Table 2).

Afterwards, we look at the actual hypothesis. As one can see in Table 2, alternative 1, which should be the best process design according to Hypothesis 1, has the highest ratio of incorrect process instances closely followed by alternative 3, which should be the worst design. Again, we did a chi-square test to test the alternatives for significant differences. Only for the pair alternative 1 and 2, the null-hypothesis (no difference) could be rejected ($p \approx 0.030$). So, the results of our experiment do not support Hypothesis 1.

Next, we did an analysis on activity level. The results of pairwise chi-square tests are shown in Table 3. Looking at Table 2, one sees that the error probabilities of activities A–AE have exactly the opposite order than predicted by Hypothesis 1—even though, there is only a significant difference between alternatives 1 and 3. That was motivation to us to search for alternative factors of influence.

In a next step, we analyzed the possible influence of the number of information elements and operations (see Table 4) on the error probability of activities (see last row of Table 2) and depicted the values in Figure 7. For the analysis, we computed both Spearman's rank correlation coefficient (arbitrary monotonic function) [5, pp. 42–45] and Pearson's correlation coefficient (linear correlation) [5, pp. 38–40]. For the number of information elements, we got 0.95 (Spearman) and 0.78 (Pearson) respectively—as well as 0.97 (Spearman) and 0.85 (Pearson) respectively for the number of operations. So, roughly speaking, larger activities are more error-prone.

We interpret these results as follows: Hypothesis 1 that process granularity (a global process property) influences the error probability during process execution might not be true. Instead, activity size seems to have a big influence on the error probability of an activity. From the point of view of a subject, the remaining process is some kind of "black box". It only sees its own activity with the contained operations. This fact motivates the following alternative error probability model.

5.2.1 Error Probability Model

If the probabilities p_i that activity i is executed erroneously for a process instance are stochastically independent, then the probability P_{err} that the process instance is executed erroneously is

$$P_{err} = 1 - \prod_i (1 - p_i) \quad . \tag{7}$$

If one further assumes for simplicity that all error probabilities p_i of the n activities of a process are equal with value p, one gets

$$P_{err} = 1 - (1 - p)^n \quad . \tag{8}$$

Comparing the error probabilities P_{err_A} and P_{err_B} of two alternative process designs, one gets the following theorem.

Theorem 1 *Given two alternative process designs A and B with n_A and n_B activities, respectively, and activity error probability p_A and p_B, respectively.*

Then, process design A is more error-prone than process design B ($P_{err_A} > P_{err_B}$) if

1. $p_A > 1 - (1 - p_B)^{\frac{n_B}{n_A}}$ *or*
2. $n_A > n_B \cdot \frac{\ln(1-p_B)}{\ln(1-p_A)}$.

The proof of Theorem 1 can be found at the top of page 13.

If one applies Theorem 1 on the more special case that one process design alternative has larger and more error-prone but less activities than the other one, one gets the following corollary.

Corollary 1 *Given two alternative process designs A and B. Alternative B has larger and more error-prone ($p_A < p_B$) but less activities ($n_A > n_B$) than alternative A.*

Then, process design A is more error-prone than process design B ($P_{err_A} > P_{err_B}$) if

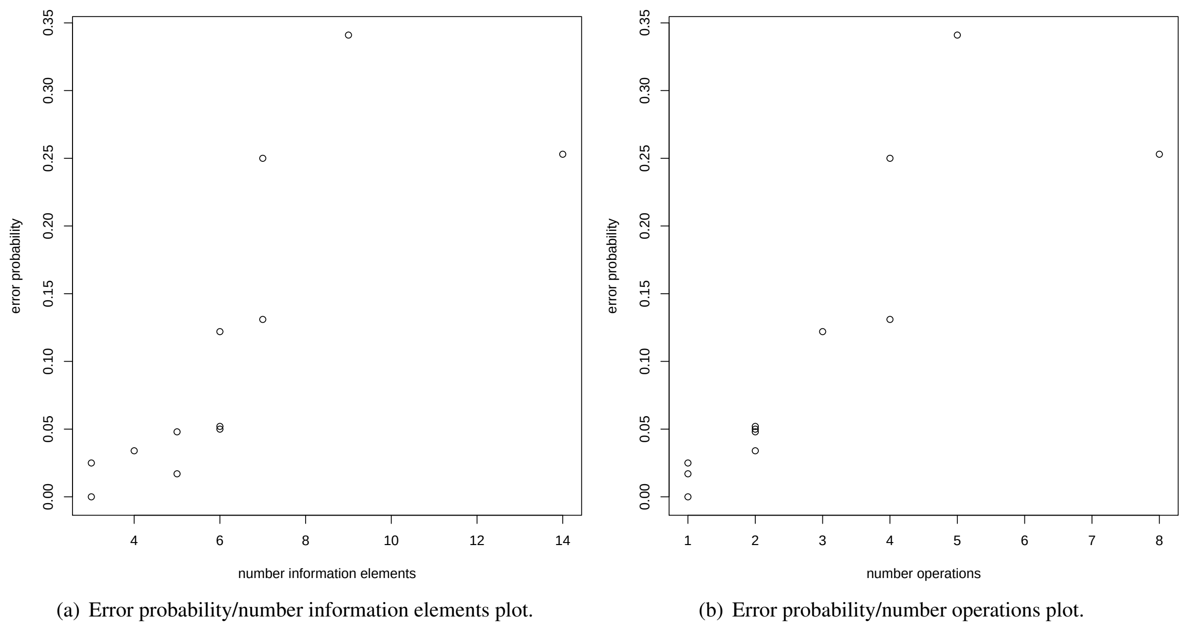
1. $1 - (1 - p_B)^{\frac{n_B}{n_A}} < p_A < p_B$ *or*
2. $n_A > n_B \cdot \frac{\ln(1-p_B)}{\ln(1-p_A)}$.

Proof. Let $p_A < p_B$ and $n_A > n_B$ with $p_A, p_B \in (0, 1)$ and $n_A, n_B \in \mathbb{N} \backslash \{0\}$.

Regarding 1.) The proposition follows from case 1 of Theorem 1 together with the precondition $p_A < p_B$.

Table 2: Error statistics for the different process alternatives (alternative 3 with six and seven subjects, respectively).

	# incorrect process instances	# incorrect activities C	# incorrect activities B	# incorrect activities F	# incorrect activities D	# incorrect activities G	# incorrect activities A	# incorrect activities E	# incorrect activities A1	# incorrect activities A2	# incorrect activities A3	# incorrect activities A4	# incorrect activities AE	# process instances with at least one of A–AE incorrect
alt. 1	29/60 48.3%	9/60 15.0%	1/41 2.4%	3/41 7.3%	1/41 2.4%	0/60 0.0%	5/41 12.2%	14/41 34.1%	-	-	-	-	-	18/41 43.9%
alt. 2	14/50 28.0%	0/50 0.0%	0/40 0.0%	2/40 5.0%	0/40 0.0%	0/50 0.0%	-	-	0/40 0.0%	1/40 2.5%	2/40 5.0%	10/40 25.0%	-	13/40 32.5%
alt. 3, 6 s.	26/60 43.3%	3/60 5.0%	3/47 6.4%	9/47 19.1%	1/47 2.1%	7/60 11.7%	-	-	-	-	-	-	9/47 19.1%	9/47 19.1%
alt. 3, 7 s.	24/60 40.0%	0/60 0.0%	2/48 4.2%	9/48 18.8%	1/48 2.1%	4/60 6.7%	-	-	-	-	-	-	15/48 31.3%	15/48 31.3%
sum alt. 3	50/120 41.7%	3/120 2.5%	5/95 5.3%	18/95 18.9%	2/95 2.1%	11/120 9.2%	-	-	-	-	-	-	24/95 25.3%	24/95 25.3%
sum		12/230 5.2%	6/176 3.4%	23/176 13.1%	3/176 1.7%	11/230 4.8%	5/41 12.2%	14/41 34.1%	0/40 0.0%	1/40 2.5%	2/40 5.0%	10/40 25.0%	24/95 25.3%	



(a) Error probability/number information elements plot.

(b) Error probability/number operations plot.

Figure 7: Possible influences on error probability.

Regarding 2.) According to case 2 of Theorem 1,

$$n_A > n_B \cdot \frac{\ln(1-p_B)}{\ln(1-p_A)} \tag{9}$$

holds.
As

$$\begin{aligned} & p_A, p_B \in (0,1) \text{ and } p_A < p_B \\ \Rightarrow\ & 0 < 1-p_B < 1-p_A < 1 \\ \Rightarrow\ & \ln(1-p_B) < \ln(1-p_A) < 0 \\ \Rightarrow\ & \tfrac{\ln(1-p_B)}{\ln(1-p_A)} > 1 \quad , \end{aligned}$$

the precondition $n_A > n_B$ is already contained in (9). ■

Let us now look at the following example for Corollary 1: Alternative B has larger and more error-prone but less activities than alternative A. So, while alternative B has $n_B = n$ activities with error probability $p_B = 0.075$, alternative A has $n_A = 2n$ activities with error probability $p_A = 0.05$. As one can easily check using case 2 of Corollary 1, alternative B is less error-prone for all values of n.

Generally, one finds many parameters for the above model so that the process design with the larger and more error-prone but less activities has a smaller error probability than the alternative design.

These findings about the error probability model are consistent with our interpretation of the results of our experiment. Hypothesis 1 could be wrong. Instead of process granularity, the size (and consequently error probability) and the number of activities in a process could be the main reasons for different error probabilities of alternative process designs.

Table 3: Results of chi-square tests for error statistics on activity level ($\alpha = 0.05$). For cells marked with "+", the null-hypothesis (no difference) was rejected.

	activity C	activity B	activity F	activity D	activity G	activities A–AE
# alt. 1 vs. 2	+	-	-	-	-	-
# alt. 1 vs. 3	+	-	-	-	+	+
# alt. 2 vs. 3	-	-	+	-	+	-

6 Conclusion and Future Work

In this paper, we gave a short introduction into the process granularity heuristic of Vanderfeesten *et al.* We presented a web-based experimentation system for analyzing the hypothesis of Vanderfeesten *et al.* that process designs with smaller process granularity metric values are less error-prone. Furthermore, we reported the results of an experiment involving 165 students.

The experiment does not support the hypothesis. Instead, we presented an alternative error probability model which is able to explain the results.

For future work, we suggest to conduct further and even larger experiments to re-check our results about the heuristic of Vanderfeesten *et al.* as well as our proposed alternative error probability model.

Acknowledgment

The authors want to thank Roland Küstermann for his technical assistance as well as the participating students.

References

[1] Norman E. Fenton and Shari Lawrence Pfleeger. *Software Metrics: A Rigorous and Practical Approach*. International Thomson Computer Press, 2nd edition, 1996.

[2] Stephen H. Kan. *Metrics and Models in Software Quality Engineering*. Addison-Wesley, 2nd edition, 2002.

Table 4: Number of information elements and operations per activity.

	activity C	activity B	activity F	activity D	activity G	activity A	activity E	activity A1	activity A2	activity A3	activity A4	activity AE
# information elements	6	4	7	5	5	6	9	3	3	6	7	14
# operations	2	2	4	1	2	3	5	1	1	2	4	8

Proof. (Theorem 1) Let $p_A, p_B \in (0,1)$ and $n_A, n_B \in \mathbb{N}\backslash\{0\}$.

Regarding 1.)

$$\begin{array}{rrcll}
 & P_{err_A} & > & P_{err_B} & \\
\Leftrightarrow & 1-(1-p_A)^{n_A} & > & 1-(1-p_B)^{n_B} & |-1 \\
\Leftrightarrow & -(1-p_A)^{n_A} & > & -(1-p_B)^{n_B} & |\cdot(-1) \\
\Leftrightarrow & \underbrace{(\underbrace{1-p_A}))^{n_A}}_{0<\cdot<1} & < & \underbrace{(\underbrace{1-p_B})^{n_B}}_{0<\cdot<1} & |(\cdot)^{\frac{1}{n_A}} \\
\Leftrightarrow & 1-p_A & < & (1-p_B)^{\frac{n_B}{n_A}} & |-1 \\
\Leftrightarrow & -p_A & < & -1+(1-p_B)^{\frac{n_B}{n_A}} & |\cdot(-1) \\
\Leftrightarrow & p_A & > & 1-(1-p_B)^{\frac{n_B}{n_A}} &
\end{array}$$

Regarding 2.)

$$\begin{array}{rrcll}
 & P_{err_A} & > & P_{err_B} & \\
\Leftrightarrow & 1-(1-p_A)^{n_A} & > & 1-(1-p_B)^{n_B} & |-1 \\
\Leftrightarrow & -(1-p_A)^{n_A} & > & -(1-p_B)^{n_B} & |\cdot(-1) \\
\Leftrightarrow & \underbrace{(1-p_A)^{n_A}}_{>0} & < & \underbrace{(1-p_B)^{n_B}}_{>0} & |\ln(\cdot) \\
\Leftrightarrow & n_A\cdot\ln(1-p_A) & < & n_B\cdot\ln(1-p_B) & |\div\underbrace{\ln(\underbrace{1-p_A}_{0<\cdot<1})}_{<0} \\
\Leftrightarrow & n_A & > & n_B\cdot\frac{\ln(1-p_B)}{\ln(1-p_A)} &
\end{array}$$

■

---

[3] Markus Kress, Joachim Melcher, and Detlef Seese. Introducing executable product models for the service industry. In Ralph H. Sprague, editor, *Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS 2007)*, page 46, 2007.

[4] Joachim Melcher and Detlef Seese. Process measurement: Insights from software measurement on measuring process complexity, quality and performance. Research report, Universität Karlsruhe (TH), Institut AIFB, 2008. `http://digbib.ubka.uni-karlsruhe.de/volltexte/1000009225`.

[5] Michael J. Panik. *Advanced Statistics from an Elementary Point of View*. Elsevier Academic Press, 2005.

[6] Hajo A. Reijers. *Design and Control of Workflow Processes : Business Process Management for the Service Industry*, volume 2617 of *LNCS*. Springer, 2003.

[7] Hajo A. Reijers, Selma Limam, and Wil M. P. van der Aalst. Product-based workflow design. *Journal of Management Information Systems*, 20(1):229–262, 2003.

[8] Hajo A. Reijers and Irene T. P. Vanderfeesten. Cohesion and coupling metrics for workflow process design. In Jörg Desel, Barbara Pernici, and Mathias Weske, editors, *Business Process Management: Proceedings of the Second International Conference BPM 2004*, volume 3080 of *LNCS*, pages 290–305, 2004.

[9] W. M. P. van der Aalst, H. A. Reijers, and S. Limam. Product-driven workflow design. In Weiming Shen, Zongkai Lin, Jean-Paul Barthès, and Mohamed Kamel, editors, *Proceedings of the 6th International Conference on Computer Supported Cooperative Work in Design*, pages 397–402, 2001.

[10] Irene Vanderfeesten, Hajo A. Reijers, and Wil M. P. van der Aalst. Evaluating workflow process designs using cohesion and coupling metrics. *Computers in Industry*, 59(5):420–437, 2008.