

Modularity in Process Models: Review and Effects

H.A. Reijers¹ and J. Mendling²

¹ Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
h.a.reijers@tue.nl

² Queensland University of Technology
Level 5, 126 Margaret Street, Brisbane QLD 4000, Australia
j.mendling@qut.edu.au

Abstract. The use of subprocesses in large process models is an important step in modeling practice to handle complexity. While there are several advantages attributed to such a modular design, including ease of reuse, scalability, and enhanced understanding, the lack of precise guidelines turns out to be a major impediment for applying modularity in a systematic way. In this paper we approach this area of research from a critical perspective. Our first contribution is a review of existing approaches to process model modularity. This review shows that aside from some limited insights, a systematic and grounded approach to finding the optimal modularization of a process model is missing. Therefore, we turned to modular process models from practice to study their merits. In particular, we set up an experiment involving professional process modelers and tested the effect of modularization on understanding. Our second contribution, stemming from this experiment, is that modularity appears to pay off. We discuss some of the limitations of our research and implications for future design-oriented approaches.

1 Introduction

Modularity is the design principle of having a complex system composed from smaller subsystems that can be managed independently yet function together as a whole [19]. Such subsystems – or *modules* – can be decomposed in a similar vein. In many domains, modularity is a key principle to deal with the design and production of increasingly complex technology. For example, it has been argued that the computer industry has dramatically increased its rate of innovation by adopting modular design [5]. Modules can also be found in business process models, where they are commonly referred to as *subprocesses*. Most popular process modeling techniques support this concept, e.g. flowcharts, IDEF0 diagrams, UML Activity Diagrams, EPCs, BPMN, and YAWL.

Various advantages are attributed to the use of subprocesses in process models. At build-time, subprocesses support a *modeling style* of stepwise task refinement, stimulate *reuse* of process models, and potentially *speed up* the (concurrent) development of the overall process model [2, 23]. At run-time, when a

process model is enacted, subprocesses allow for *scaling* advantages: Each subprocess, for example, may be executed on a different workflow server [23]. Finally, when a process model is used to facilitate the understanding of complex business processes among various stakeholders, subprocesses are supposed to ease the *understanding* of the model [15, 36].

However, the way that modularity is currently utilized in modeling practice raises some questions about the actual benefits. First of all, there are no objective criteria for applying granularity. Accordingly, there is no absolute guideline if a particular subprocess should be on level X or $X + 1$ in the model hierarchy [13]. Neither is there a unique way to modularize a process model [13]. As a consequence, modularity is often introduced in an ad-hoc fashion. Furthermore, there are clearly drawbacks when the process logic is fragmented across models. In particular, it “becomes confusing, less visible, and tracking its paths is tiring” [12] if a subprocess is decomposed in further subprocesses. The fact that the semantic check in **ARIS Toolset** mainly addresses consistency issues between events in the subprocess and around the refined function illustrates the seriousness of this problem. Finally, even if modularization is useful for maintenance purposes, it is questionable whether advantages materialize in practice as many organizations fail to keep their models up to date.

The greater research challenge we see here is to provide explicit guidance for using modularization in process models. But, this would be a dubious undertaking at the present state of the art: We simply do not have the evidence whether modules in process models pay off. Therefore, this paper is concerned with establishing an empirical foundation as a necessary preparation for a design-oriented approach to the subject. We start this investigation from a critical review of existing approaches to introduce modularity in process models.

Our null hypothesis is that *modularization does not increase process model understanding*, and we introduce an experimental design to challenge it. In this approach, we worked together with a group of professional process modelers to evaluate a set of professional process models. The controlled variable in the design is whether subprocesses are used or not; the response variable is the degree of understanding that the subjects display with respect to the models. Note that we focus on the *understanding* of a process model as the major point of evaluation. Our motivation is that in most business applications, the primary purpose of a process model is to act as a means of communication [25, 31]. As Van der Aalst and Van Hee put it when discussing the introduction of subprocesses in a process model “[.] the most critical consideration is that the process be understood by the people carrying out the work. If this is not the case, the result can be a difficult-to-manage process.”

Against this background, the structure of this paper is as follows. In the next section, we will give a broader background for the concept of modularity, in particular with respect to process modeling. In Section 3, we will present our research method, after which the results of the experiment we carried out are given in Section 4. Before we come to a conclusion in Section 6, we discuss our findings and their limitations in Section 5.

2 A Review of Modularity and Process Modeling

2.1 Concepts and terms

A first issue that should be considered here is that the terms *modularity*, *decomposability*, and *hierarchy* are sometimes used interchangeably. However, according to [19], a modular system is not automatically *decomposable*, since one can break a system into modules whose workings remain highly interdependent with the internal workings of other modules. Furthermore, as Parnas points out in his seminal paper on “information hiding”, a modular system is not necessarily *hierarchical* [32]. That would be the case if the “uses” relation between modules gives a partial ordering, which is not always so. One can easily imagine, for example, a software program where software modules mutually call each other. These subtleties also hold in the context of process models. In most practical cases, however, a modular process model will probably be hierarchical too although perhaps not decomposable, i.e. its subprocesses may still be highly interdependent. In this paper we consider the more general phenomenon of “modularity” as the main point of interest.

2.2 Modularization in systems

In many settings, “the real issue is normally not to be modular but *how* to be modular” [19]. But at the same time, modular systems are much more difficult to design than comparable interconnected systems [5]. Beyond that, problems with incomplete or imperfect modularization tend to appear only when the modules come together and work poorly as an integrated whole. It has been argued that many of the most attractive and durable systems are developed through an “unselfconscious” design process [4]. In this mode, the design rules that are used are not explicit; inconsistencies and interdependencies are revealed by trial and error only. However, it is by no means obvious that unselfconscious design must always, or even usually, result in modularity [19].

Quality criteria to consciously decompose a system into modules have been discussed by Wand and Weber on a general level [41, 43]. The authors identify five criteria. The first three are absolute criteria that are either met or not and focus on the content of the modular model, not its structure. *Minimality* requires that there is no redundant state information in the modular model. In data models this basically matches normalization requirements. *Determinism* requires that a state change is clearly identified to be triggered by an internal or an external event. If that is not the case the behavior of a module can only be understood by knowing the state of another subsystem. *Losslessness* demands that emergent properties are not lost in a modularization. Furthermore, the two criteria coupling and cohesion should be optimized, cf. [45]. *Coupling* should be minimal such that the sum of inputs of each subsystem is less or equal to the sum of inputs in any other modularization. *Cohesion* should be maximal such that all output affected by input variables are contained in the same set, and

adding another output does not extend the set of input variables on which they depend.

Wand and Weber’s criteria had a strong influence on the object-oriented design metrics proposed by Chidamber and Kemerer [10]. The usefulness of the five criteria has been demonstrated for UML class diagrams and state charts in an experimental setting [8]. Yet, an application in the area of process modeling, either by designing good decomposition operations or by testing their suitability, is missing.

2.3 Modularization in process models

The area of related research in the context of process models is huge, covering works on *process modularization*, e.g. [3, 7, 42], *process inheritance*, e.g. [6, 26], and *reduction rules*, e.g. [14, 34, 44]. Since the latter two categories are mainly utilized in process model analysis, we will focus on the first category. Furthermore, we do not consider modular design of process-aware information systems such as in [16, 24]. In the context of process model modularization, three aspects can be distinguished: modularization operations, modularization prerequisites, and modularization selection.

Modularization Operations: The idea that *basic operators* should facilitate modularization was already proposed in the 1980s for data flow diagrams [3]. Refinement operations have also been defined for Workflow Nets [1]. Also, some modeling approaches impose the use of block structures of nested control primitives, which favor the creation of decomposable modules, as in e.g. BPEL. Recently, the ability to extract a subprocess from a process model has been described as a change pattern for process-aware information systems [42]. This pattern must be implemented reflecting the syntactic requirements of the modeling language. In ARIS there are two ways to extract a subprocess: by modularization (refining function with subprocess) and by segmentation (cutting a model in different parts) [13]. Both these options are tailored to yield syntactically correct EPCs.

Modularization Prerequisites: There are some recommendations regarding *when* a process model should be considered for modularization. Some of the practitioners books state that modularization should be introduced in a model with more than 5–15 [18] or 5–7 activities [36], yet without giving any support for this rule. Recently, it has been recommended based on empirical findings that process models with more than 50 elements should be decomposed [28]. Depending on the process modeling language the amount of activities can vary for 50 elements, e.g. EPCs use connectors for routing and events to separate functions while YAWL essentially only uses tasks. Still, up to now no objective criteria has been proposed for identifying which subprocess should be on which level in the model hierarchy [13].

Modularization Selection: There are some guidelines on how to *select* parts of process models for modularization. Good candidates for subprocesses are fragments of a model that are components with a single input and a single output control flow arc [22, 7, 39]. Furthermore, long and thin process models

should be preferred to square models [13, p.278]. This argument points to the potential of metrics to guide the modularization. The idea here would be to use quality metrics like the ones proposed in [28, 29] to assess which modularization should be preferred. An application of metrics to compare design alternatives is reported in [38]. Yet, there is no dedicated approach to guide modularization based on metrics.

Overall, the main focus of research on process modularization is of a conceptual nature. Clearly, there are no objective and explicit guidelines that modelers in practice can rely on. The aim of our research as reported in the following sections is to contribute to a better understanding of the effects of modularization as a stepping stone towards such guidelines.

3 Research Design

In the previous sections we discussed that the ad-hoc way in which modularity is currently introduced in modeling practice raises doubts about its benefits. In this section, we will explain our design to test the following null hypothesis:

H0: Use of modularization does not improve understanding of a process model.

There are several challenges in testing the presumed absence of a relation between modularity and understanding, in particular in pursuing results that have a potential to be generalizable on the one hand while applying methodological rigor on the other. In particular, it would be unsatisfactory to rigorously test the effects of modularity in small, toy-like process models, as any effect would possibly be hard to spot anyway. To achieve a realistic setting for our research, we set up a collaboration with Pallas Athena Solutions¹ in the Netherlands, a specialized provider of BPM services. This company provided us with real-life models as study objects. Furthermore, their process modelers participated in our investigation. As will be explained in this section, we applied an *experimental* design to achieve sufficient control over the dependent variable (modularization) and to allow a meaningful evaluation of the response variable (understanding) from our hypothesis.

In lack of specific literature on empirical research with respect to modular process modeling, we build on approaches and classifications used in the field of software experimentation [17, 33]. In particular, we use an experimental design that is comparable to what was applied in a recent study to evaluate various types of BPM technology [30]. To test the hypothesis we carried out a so-called *single factor experiment*. In general, this design is suitable to investigate the effects of one *factor* on a common *response variable*. This design also allows to analyze variations of a factor: The *factor levels*. The *response variable* is determined when the participants of the experiment – the *subjects* – apply the factor or factor levels to a particular *object*. The overall approach in our experiment is visualized in Figure 1. We will address the most important elements in our design in more detail now.

¹ See <http://www.pallas-athena.com>.

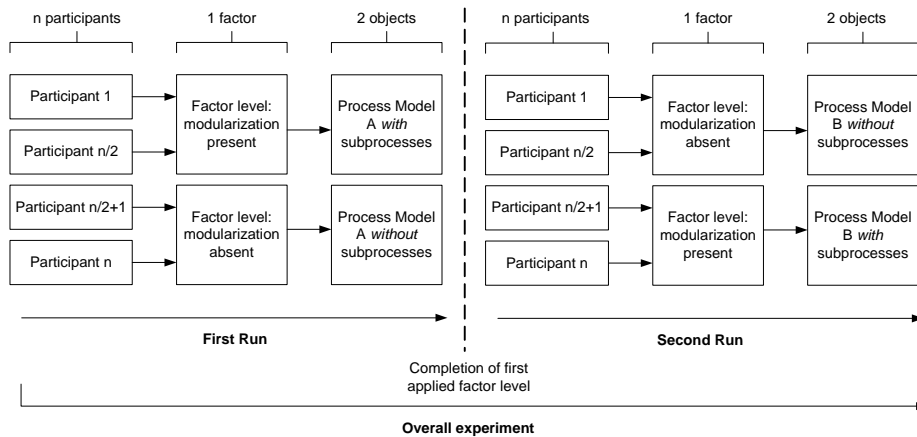


Fig. 1. Experiment design

Objects. The basic *objects* that were evaluated by the participants, were two process models from practice. The models were used in the experiment both in their original form – displaying modularity – and in their flattened version where modularity is completely removed: All dependencies between model elements are then on the same level of abstraction. Note that for any particular process model the absence or presence of modularity does not affect the business logic.

Both process models were selected from a little over 80 process models that were created and delivered by the consultancy company for its clients. We focused our search for suitable objects using three criteria: (1) presence of modularity, (2) size, and (3) access to the original creators of the models. The process models we looked for needed to display modularity, as consciously applied by the modeler to deal with the complexity of a large model. We only considered models of more than 100 tasks, which can be considered as *very large* using the process size classification provided in [9]. Our line of reasoning here is that if modularity does not help to understand very large models, it will not help to distinctively understand smaller models either. Finally, we needed access to the modelers of the model to validate questions on the content of the model.

From our search, four candidate models emerged. One of these models was specifically developed for automated enactment. It was not further considered because understanding is generally not a prime issue with this modeling purpose. Of the remaining three, which were all developed for the support of stakeholders in a process improvement project, the two process models were selected that were most similar to each other in terms of process size, number of subprocesses, and modularity depth. Both models had been modeled with the *Protos* tool [40]. The flattened versions of the process models can be seen² in Figure 2, so that the reader can get an impression about their structure and size. Note that we are not allowed to disclose the content of the models.

² For larger images, see <http://www.reijers.com/models.pdf>.

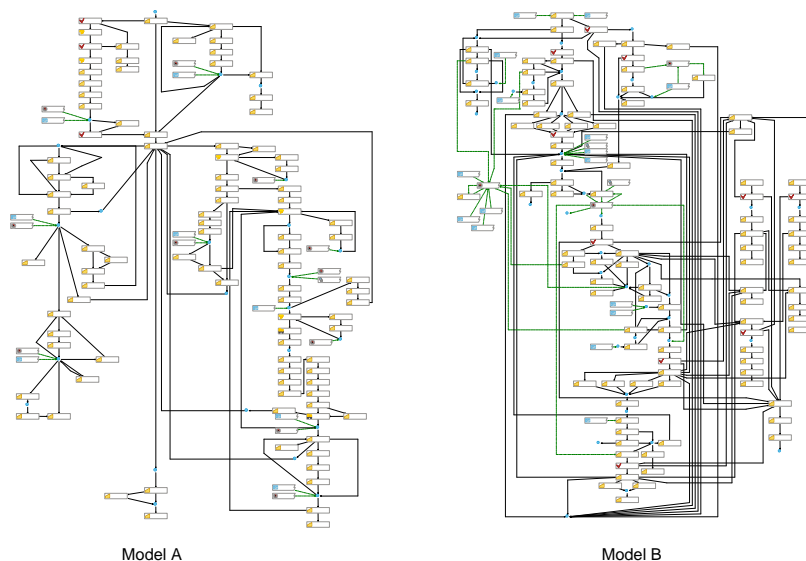


Fig. 2. Flattened versions of the used process models

Model A describes the procedure in use by an organization that is responsible for handing out driver’s licences. The process in question deals with clients that cannot directly obtain their driver’s license because of physical or psychological disabilities that can influence their driving. Model B captures how a certain category of unemployed citizens is coached and receives advice in finding a job. Note that labels in Figure 2 have been removed to protect the confidentiality of the involved organizations; the subjects in our experiment saw the entire model in full (including the labels).

Factor and factor levels. In our experiment, the *use of modularity* is the considered *factor*, with factor levels “present” and “absent”. Note that we deliberately collected real process models from practice already exhibiting modularity and derived flattened versions from it, instead of doing it the other way around. In this way, we could build on a real-life application of modularity.

Response variable. The *response variable* in our experiment is the level of understanding that the respondents display with respect to the process models, both in their modularized and flattened form. To measure the response variable, a specific set of questions was developed for each of the two models to be answered by the subjects. We used the percentage of correctly answered questions given by a subject as measure for his or her level of understanding of the particular model. This approach is similar to the one we applied in a previous study into model understandability [29]. An example question for model A is: “If an AA-investigation is required, then a number of alternative settlements is possible. How many of these settlements exist?”. For model B an example question is: “If a client does not appear on an appointment, is it always so that a new

appointment is scheduled?”. Note that the question sets are different for each of the models because both their content and structure differs. The questions were formulated in Dutch, the same language used by the creator of the modeler to name model elements, and also being the native language for all subjects. The model-specific questions were preceded by a general introduction to the experiment, some specific background information for each of the models, and a number of general questions with respect to the subject’s background. As will be explained later, we used the latter information for comparison purposes (see Section 4.2).

Subjects. The participants in this experiment were 28 experienced consultants from Pallas Athena Solutions. They were randomly assigned to the two groups used in our set-up (*block design*). Each group was presented two models: One model that displayed modularity and the other model in the flattened version. This way each participant received two *different processes* – models A and B – and two *different styles* – modular and flattened. Participation in the experiment was voluntary; the single reward offered for participation was access to the research results.

Instrumentation. The experiment was then carried out in the following way. The groups of subjects were provided with the process models *on paper*, together with the questions; an alternative would have been to show the models on a computer display, e.g. using the software that was used to create the models. The involved consultancy company indicated that paper is a common form to interact with their clients. Recall that the original versions of the models were divided into subprocesses by their respective authors. These models could therefore be presented to the respondents as a set of A4-sized papers, one for the main process and one for each subprocess. The alternative, flattened model versions were presented on A3 paper format, making task labels clearly legible.

Prior to the actual experimentation, all questions and correct answers were discussed with the creators of the models. They approved of these and validated that the question sets were a proper way to test understanding of the models. Then, five graduate students from Eindhoven University of Technology were involved in a pre-test. This led to the reformulation of 10 questions to remove ambiguities and the removal of 3 questions. The latter was explicitly required to keep the experiment within a reasonable time frame. For each model, 12 questions were included in the final version of the experiment.

Data collection and analysis. During the experiment, the subjects were asked to spend at most 25 minutes per model for answering its related questions. This limit was imposed to keep the time spent on the entire questionnaire under one hour and to prevent an imbalance in time spent on each model. Both at the start and at the end of answering a set of questions for each model, subjects were asked to write down the current time to allow for exact comparisons.

For our data analysis, well-established statistical methods and standard metrics are applied, as provided by the software package STATGRAPHICS XV. II. From the description the elements in this section, it follows that the experiment is *balanced*, which means that all factor levels are used by all participants of

the experiment. In general, such an approach enables repeated measurements and the collection of more precise data as every subject generates data for every treated factor level. As can be seen in Figure 1, we went through two runs, so this experiment displays a *repeated measurement*. But in contrast to the approach in [30], two objects instead of one were used (process models A and B) to repeat the experiment in a second run. This setup prevents confronting the same group of subjects to the same model more than once. In this way, we could avoid learning effects to take place while still varying the factor levels.

4 Results

In this section, we will first present our main analysis results, after which we will explore some alternative explanations for these to decide on our hypothesis.

4.1 Main results

Our main analysis for each model focuses on the comparison between the group performance in terms of correctly answered questions for its modularized and flattened version. In other words, does it matter whether someone sees a modularized or a flattened version of a process model? As explained, we calculated for each of the subjects the percentage of correct answers given for each model to make this comparison. Recall that each subject saw a modular model for one process and a flattened model for the other. The values are shown in Table 1.

Table 1. Average percentages of correct answers for the model variants.

	Flattened	Modular
MODEL A	38.54%	42.36%
MODEL B	37.50%	58.33%

As can be seen from this table, for both models the modular version generates a *higher* average percentage of correct answers, which suggests a better understandability. To determine whether the differences are statistically significant, it is important to select the proper statistical test. Therefore, we first explored for each of the models the distribution of correct answers for each of its variants, i.e. the modular and flattened version. Because the standardized skewness and standardized kurtosis are within the range of -2 to +2, for each model the correctly answered questions can be assumed to be normally distributed. Additionally, F-tests indicated that with a 95% confidence the standard deviations of the samples for each of the models are also the same. These two conditions justify the application of Student’s t-test [37].

Application of the t-test results in a P-value for each comparison; a P-value lower than 0.05 signals a significant difference when assuming a 95% confidence level. The results are then as follows:

- For model A, there is *no difference* between the modular and the flattened version in terms of the average percentage of correctly answered questions ($P = 0.562$).
- For model B, there is a *significant difference* between the modular and the flattened version in terms of the average percentage of correctly answered questions ($P = 0.001$).

The difference for model B seems to support rejection of H_0 . However, we must first explore whether alternative explanations exist to properly decide on the acceptance or rejection of this hypothesis.

4.2 Supporting results

The main *alternative* explanation for the difference for model B is that the group that produced better results for the modular version is simply different from the group that looked at the flattened version. Recall from Section 3 that our experiment is characterized by a *block design*, i.e. subjects are randomly assigned to the two experimental groups. If the groups are different with respect to a characteristic that may influence their ability to understand process models, then this would not allow us to reject H_0 – despite the noted statistical difference. A second, *alternative* explanation would be that one group of respondents simply spent more time than the other on answering the corresponding questions.

To assess these alternative explanations, we analyzed the characteristics as shown in Table 2. Each entry in the table lists an investigated factor, the considered factor levels, and the P-value resulting from a statistical test. Note that we applied a standard t-test to determine a statistical difference between the groups with respect to each factor, unless its basic requirements were not met with respect to the assumed normal distribution and variance equality. In the latter case, we used the non-parametric Mann-Whitney W test to compare the medians across both groups [37].

All P-values in this table are far greater than 0.05, so none of the investigated factors signals anything close to a statistical difference between the groups at a 95% confidence level. Therefore, in lack of knowledge on other plausible influences, we must *reject* hypothesis H_0 . We conclude that *modularity appears to have a positive connection with process understanding*.

5 Discussion

We single out two questions that emerge from considering the results from the previous section:

1. Why does modularity matter for understanding model B, but not for A?
2. What is the explanation for modularity influencing the understanding of model B?

In this section, we will first address these questions and then discuss some limitations of our experiment.

Table 2. Group comparison.

Factor	Factor levels	P-value
DOMAIN KNOWLEDGE	Knowledgeable with the process context or not	0.386
COMPANY EXPERIENCE	Actual number of years within company	0.411
FIELD EXPERIENCE	Actual number of years working as process consultant	0.726
EDUCATION	University degree or not	0.453
JOB TYPE	Business consultant or technical consultant	1.000
MODELING AMOUNT	Estimated number of process models created	0.504
MODELING SIZE	Estimated average size of process models created (nodes)	0.764
TIME OVERALL	Actual time spent on entire experiment	0.948
TIME A	Actual time spent on model A in the experiment	0.641
TIME B	Actual time spent on model B in the experiment	0.417

5.1 Model differences

We recall that we selected models A and B from a wide range of models, keen on satisfying a number of requirements (see Section 4). From the four models that met these, models A and B were *most* similar, notably with respect to the number of tasks they contain and their depth. To determine why modularity plays a bigger role in understanding model B, we carried out a further analysis of both models by using the metrics shown in Table 3. At the top of the table, some basic metrics are given, followed by metrics that have been proposed as indicators for process model complexity in general, and at the bottom some metrics that are explicitly proposed for assessing modular process models.

Two metrics display values that differ more than a factor 2 between the models under consideration, i.e. **Subprocesses** and **FanIn-Out**. According to [20], the relatively high value of the latter metric for model B (33.42) would suggest a poorer structuring of model B compared to model A, which would make it more difficult to use. However, an additional test to determine whether a difference exists in model understandability between the modular version of model A and the modular version of model B does not show a higher average percentage of correct answers for the former. In lack of other empirical support for the use of this metric, the relatively high number of subprocesses (20) in model B seems more relevant: It suggests that the difference between the modular and flattened version of this model is more distinct than for model A.

For the remaining factors, models A and B display quite similar characteristics, even though model B is the slightly larger one. There is no general trend that suggests that one model is considerably more complex than the other and none of the metrics display substantial and meaningful differences other than the number of subprocesses. So, the most reasonable answer to the question why modularity has an impact on understanding model B but not on model A is

Table 3. Complexity metrics.

Metric	Description	Source	Model A	Model B
TASKS	Total number of tasks	–	105	120
NODES	Total number of nodes	–	130	175
ARCS	Total number of arcs	–	171	248
SUBPROC	Total number of subprocess in original model	–	9	20
TO	Average number of outgoing arcs from transitions (tasks)	[21]	0.81	1.03
PO	Average number of outgoing arcs from places (milestones)	[21]	3.42	2.24
CYCN	McCabe’s cyclomatic number (adjusted for Petri nets)	[21]	43	75
CONNECT	Number of arcs divided by the number of nodes	[27]	1.32	1.42
DENSITY	Number of arcs divided by the maximal number of arcs	[27]	0.020	0.016
CONDEG	Average number of input and output arcs per routing element	[27]	1.10	1.21
FAN-IN	Average number of modules calling a module	[20]	1.25	2.26
FAN-OUT	Average number of modules called by a module	[20]	1.5	2.26
FANIN-OUT	Average $((\text{Fan-In}) * (\text{Fan-Out}))^2$ per module	[20]	3.63	33.42
DEPTH	Degree of nesting within the process model	[27]	3	3

that B’s original version displayed *a much higher degree of modularization* than model A, which eased its understanding.

5.2 The influence of modularity

In search for an explanation of *how* modularity increases model understanding, we re-examined the questions we used in our experiment. Recall that these questions were validated by the original creators of the model (see Section 3): The questions were considered to be to the point, reasonable, and a good way to test someone’s understanding of the model.

In the *ex post* analysis of our results, we pursued the idea that by using a modular model perhaps one type of question would be answered better than another. In particular, we categorized our questions as being of a *local* or *global* type. The answer for a local question can be found within the confinements of a single subprocess in the modular version, where the examination of more subprocesses is required to answer a global question. As it turned out, model B contained 2 global questions and 10 local questions. In a comparison between the group that used the modular model and the group that used the flattened model, the following results emerged:

- Too few global questions were used to determine whether there is a difference in terms of the average percentage of correctly answered questions between using the modular or the flattened version of model B .
- For local questions, there is a *significant difference* in terms of the average percentage of correctly answered questions between the modular and the flattened version of model B (P=0.002).

From this analysis, we cautiously infer that modularity may be helpful for understanding a process model because it *shields the reader from unnecessary information*. Where the reader of flattened model always sees the entire context, the reader of the modular version is confronted with precisely the right set of information when the proper subprocess is selected. In this sense, it resembles the positive effect of Parnas’ “information hiding” concept [32]: Programmers are most effective if shielded from, rather than exposed to the details of construction of system parts other than their own.

Whether there is also an opposite effect, i.e. the correct answer for a global question would be easier to find with a flattened model, could not be established for model B. However, it does not seem too likely; an analysis of the results for model A did not show such an effect.

5.3 Limitations

Only a small number of 28 subjects were involved in this experiment and only 2 process models were considered. Both aspects are threats to the *internal validity* of this experiment, i.e. whether our claims about the measurements are correct. But these small numbers result from our choices to (1) involve experienced process modelers and (2) process models from industrial practice. Most experienced modelers from the company already participated and the confidential models could not be shown outside the company. Also, to keep the experiment’s duration under one hour – a pragmatic upper bound to avoid non-response – it was not feasible to use, for example, more models. The choice for professional modelers and real models clearly positively affects the *external validity* of our study, i.e. the potential to generalize our findings. Therefore, our experiment shows how “internal and external validity can be negatively related” [11].

Another aspect is the choice for displaying the process models *on paper*. It is by no means certain that similar findings would result from an experiment where models are shown on a *computer display*. In the latter mode, “information hiding” is achievable in other ways than by applying modularity. For example, the **Protos** tool that was used to create the models allows to zoom in on part of a process model, which is another form of shielding away irrelevant data.

Finally, the lay-out of a process model may be a factor that influences understandability, as we hypothesized before in [29]. As a limited understanding of this effect exists at this point, we are restrained in properly controlling this variable. We used the same modeling elements, the same top-down modeling direction, and roughly a similar breadth and width for both models on paper to limit this effect – if any (see Figure 2).

6 Conclusion

On the basis of the controlled experiment we described in this paper, the main conclusion of this paper must be that modularity in a process model (through use of subprocesses) appears to have a positive connection with its understandability. However, this effect manifests itself in *large* models if modularity is applied to a sufficiently *high* extent and particularly seems to support comprehension that requires insight into *local* parts of the model.

These results should be considered within the limitations of the experiment we described, but in principle favor further efforts into the development of more explicit design guidance towards modularizing process models. As we noted, this is a major gap in our knowledge on process modeling. From the review of process modularization approaches that we presented in the paper, we identified several attractive ingredients for such an approach. In particular, Wand and Weber's quality criteria have already been successfully applied for other types of models and the use of metrics to guide process modularization seems a fruitful direction. Our future work is aimed at the development of such guidance and metrics.

Aside from this research agenda, we hope that publications like [8, 30, 35], and this paper as well, may serve as an inspiration for further integrating methodologies from behavioral science in the design-science approaches common to the BPM field. This could be particularly helpful to provide explicit support for both the necessity and the utility of the models, algorithms, systems, and other artifacts that BPM scholars are concerned with.

Acknowledgements

We would like to thank Bela Mutschler, Barbara Weber, Maarten Boote, and all participants from Pallas Athena Solutions for their contribution to this research.

References

1. W.M.P. van der Aalst. Verification of Workflow Nets. In P. Azéma and G. Balbo, editors, *Application and Theory of Petri Nets 1997*, volume 1248 of *Lecture Notes in Computer Science*, pages 407–426. Springer Verlag, 1997.
2. W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT press, Cambridge, MA, 2002.
3. M. Adler. An algebra for data flow diagram process decomposition. *IEEE Transactions on Software Engineering*, 14(2):169–183, 1988.
4. C. Alexander. *Notes on the Synthesis of Form*. Harvard University Press, 1970.
5. C.Y. Baldwin and K.B. Clark. Managing Modularity. *Harvard Business Review*, 75(5):84–93, 1997.
6. T. Basten and W.M.P. van der Aalst. Inheritance of Behavior. *Journal of Logic and Algebraic Programming*, 47(2):47–145, 2001.
7. A. Basu and R.W. Blanning. Synthesis and Decomposition of Processes in Organizations. *Information Systems Research*, 14(4):337–355, 2003.

8. A. Burton-Jones and P. Meso. How good are these UML diagrams? An empirical test of the Wand and Weber good decomposition model. In L. Applegate, R. Galliers, and J.I. DeGross, editors, *Proceedings of ICIS*, pages 101–114, 2002.
9. J. Cardoso. Poseidon: A Framework to Assist Web Process Design Based on Business Cases. *Int. Journal of Cooperative Information Systems*, 15(1):23–55, 2006.
10. S.R. Chidamber and C.F. Kemerer. A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, 20(6):476–493, 1994.
11. T.D. Cook, W.R. Shadish, and D.T. Campbell. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Houghton Mifflin, 2002.
12. N. Damij. Business Process Modelling Using Diagrammatic and Tabular Techniques. *Business Process Management Journal*, 13(1):70–90, 2007.
13. R. Davis. *Business Process Modelling With Aris: A Practical Guide*. 2001.
14. J. Desel and J. Esparza. *Free Choice Petri Nets*, volume 40 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1995.
15. M. Dong and F.F. Chen. Petri Net-Based Workflow Modelling and Analysis of the Integrated Manufacturing Business Processes. *The International Journal of Advanced Manufacturing Technology*, 26(9):1163–1172, 2005.
16. S. Jablonski. MOBILE: A Modular Workflow Model and Architecture. In *Proceedings of the International Working Conference on Dynamic Modelling and Information Systems*, 1994.
17. N. Juristo and A.M. Moreno. *Basics of Software Engineering Experimentation*. Kluwer Academic Publishers, 2001.
18. N.F. Kock Jr. Product Flow, Breadth and Complexity of Business Processes: An Empirical Study of 15 Business Processes in Three Organizations. *Business Process Re-engineering & Management Journal*, 2(2):8–22, 1996.
19. R.N. Langlois. Modularity in Technology and Organization. *Journal of Economic Behavior and Organization*, 49(1):19–37, 2002.
20. R. Laue and V. Gruhn. Complexity metrics for business process models. In W. Abramowicz and H.C. Mayr, editors, *Proceedings of BIS 2006*, volume 85 of *Lecture Notes in Informatics*, pages 1–12, 2006.
21. G.S. Lee and J.M. Yoon. An Empirical Study on Complexity Metrics of Petri Nets. *Microelectronics and reliability*, 32(9):1215–1221, 1992.
22. F. Leymann. Workflows Make Objects Really Useful. *EMISA Forum*, 6(1):90–99, 1996. http://sunsite.informatik.rwth-aachen.de/Societies/GI-EMISA/forum/content_96_1/Emisa.1_96_S90-99.pdf.
23. F. Leymann and D. Roller. Workflow-based Applications. *IBM Systems Journal*, 36(1):102–123, 1997.
24. F. Leymann and D. Roller. *Production Workflow - Concepts and Techniques*. Prentice Hall, 2000.
25. A. Lindsay, D. Downs, and K. Lunn. Business Processes: Attempts to Find a Definition. *Information and Software Technology*, 45(15):1015–1019, 2003.
26. T.W. Malone, K. Crowston, J. Lee, and B. Pentland. Tools for Inventing Organizations: Toward a Handbook for Organizational Processes. *Management Science*, 45(3):425–443, 1999.
27. J. Mendling. *Detection and Prediction of Errors in EPC Business Process Models*. PhD thesis, Vienna University of Economics and Business Administration, 2007.
28. J. Mendling, G. Neumann, and W.M.P. van der Aalst. Understanding the occurrence of errors in process models based on metrics. In R. Meersman and Z. Tari, editors, *OTM Conference 2007, Proceedings, Part I*, volume 4803 of *Lecture Notes in Computer Science*, pages 113–130. Springer, 2007.

29. J. Mendling, H.A. Reijers, and J. Cardoso. What makes process models understandable? In G. Alonso, P. Dadam, and M. Rosemann, editors, *International Conference on Business Process Management (BPM 2007)*, volume 4714 of *Lecture Notes in Computer Science*, pages 48–63. Springer-Verlag, Berlin, 2007.
30. B. Mutschler, B. Weber, and M.U. Reichert. Workflow management versus case handling: Results from a controlled software experiment. In L.M. Liebrock, editor, *Proceedings of the ACM Symposium on Applied Computing, Volume I*, pages 82–89. 2008.
31. M.A. Ould. *Business Processes: Modelling and Analysis for Re-engineering and Improvement*. Wiley, 1995.
32. D. Parnas. On the Criteria for Decomposing Systems into Modules. *Communications of the ACM*, 15(12):1053–1058, 1972.
33. L. Prechelt. *Kontrollierte Experimente in der Softwaretechnik: Potenzial und Methodik*. Springer, 2001.
34. W. Sadiq and M.E. Orlowska. Analyzing Process Models using Graph Reduction Techniques. *Information Systems*, 25(2):117–134, 2000.
35. K. Sarshar and P. Loos. Comparing the control-flow of EPC and Petri nets from the end-user perspective. In W.M.P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors, *Proceedings of BPM 2005, Lecture Notes in Computer Science, Volume 3649*, pages 434–439. Springer-Verlag, Berlin, 2005.
36. A. Sharp and P. McDermott. *Workflow Modeling: Tools for Process Improvement and Application Development*. Artech House, 2001.
37. D.J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. CRC Press, 2004.
38. I. Vanderfeesten, H.A. Reijers, and W.M.P. van der Aalst. Evaluating workflow process designs using cohesion and coupling metrics. *Computers in Industry*, 2008.
39. J. Vanhatalo, H. Völzer, and F. Leymann. Faster and more focused control-flow analysis for business process models through SESE decomposition. In B.J. Krämer, K.J. Lin, and P. Narasimhan, editors, *ICSOC*, volume 4749 of *Lecture Notes in Computer Science*, pages 43–55. Springer, 2007.
40. H.M.W. Verbeek, M. van Hattem, H.A. Reijers, and W. de Munk. Protos 7.0: Simulation made accessible. In G. Ciardo and P. Darondeau, editors, *Proceedings of the 24th International Conference on Application and Theory of Petri Nets*, pages 465–474. Springer, 2005.
41. Y. Wand and R. Weber. On the Deep Structure of Information Systems. *Information Systems Journal*, 5:203–223, 1995.
42. B. Weber, S. Rinderle, and M.U. Reichert. Change patterns and change support features in process-aware information systems. In J. Krogstie, A.L. Opdahl, and G. Sindre, editors, *Proceedings of CAiSE*, volume 4495 of *Lecture Notes in Computer Science*, pages 574–588. Springer, 2007.
43. R. Weber. *Ontological Foundations of Information Systems*. Coopers & Lybrand and the Accounting Association of Australia and New Zealand, Melbourne, 1997.
44. M.T. Wynn, H.M.W. Verbeek, W.M.P. van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Reduction rules for YAWL workflow nets with cancellation regions and or-joins. BPMCenter Report BPM-06-24, BPMcenter.org, 2006.
45. E. Yourdon and L.L. Constantine. *Structured Design*. Prentice Hall, 1979.