Bus Inf Syst Eng 58(1):7–17 (2016)
DOI 10.1007/s12599-015-0413-1

RESEARCH PAPER

Criteria and Heuristics for Business Process Model Decomposition

Review and Comparative Evaluation

Fredrik Milani · Marlon Dumas · Raimundas Matulevičius · Naved Ahmed · Silva Kasela

Received: 27 February 2015 / Accepted: 30 August 2015 / Published online: 8 December 2015

Abstract It is generally agreed that large process models should be decomposed into sub-processes in order to enhance understandability and maintainability. Accordingly, a number of process decomposition criteria and heuristics have been proposed in the literature. This paper presents a review of the field revealing distinct classes of criteria and heuristics. The study raises the question of how different decomposition heuristics affect process model understandability and maintainability. To address this question, an experiment is conducted where two different heuristics, one based on breakpoints and the other on data objects, were used to decompose a flat process model. The results of the experiment show that, although there are minor differences, the heuristics cause very similar results in regard to understandability and maintainability as measured by various process model metrics.

Keywords Process modeling · Decomposition · Process model metrics

Accepted after two revisions by the editors of the special issue.

F. Milani (✉) · M. Dumas · R. Matulevičius · N. Ahmed · S. Kasela
University of Tartu, Liivi 2, 504 09 Tartu, Estonia
e-mail: milani@ut.ee

M. Dumas
e-mail: marlon.dumas@ut.ee

R. Matulevičius
e-mail: rma@ut.ee

N. Ahmed
e-mail: naved@ut.ee

S. Kasela
e-mail: silva.kasela@ut.ee

1 Introduction

Business process models are used for many purposes, ranging from internal communication and knowledge management, to process improvement and information systems requirements engineering (Becker et al. 2009). Given this multifunctional character, process models need to be captured in a way that facilitates understanding by a variety of stakeholders. In this respect, it is generally accepted that large process models should be decomposed into smaller sub-processes.

While the benefits of such process decomposition are acknowledged (Johannsen and Leist 2012), there is far less consensus as to how a given process model should be decomposed (Reijers et al. 2011). Instead, several guidelines and goodness criteria for process decomposition coexist and there is a lack of evidential comparison of their relative merits. For instance, some authors propose that the goodness of a decomposition should be assessed using size (Wolter and Schaad 2007; Mendling et al. 2010), while others suggest transposing modularization criteria from information systems (Reijers and Vanderfeesten 2004; Johannsen and Leist 2012). Other studies propose to decompose processes based on data (Ivanović et al. 2010) or adopt a role-based decomposition approach (Khalaf and Leymann 2006). Despite the plethora of available approaches, it has been stated that decomposition is more an art than it is a science (Burton-Jones and Meso 2004).

In this setting, this paper addresses two research questions: (1) "How can process models be decomposed?" and (2) "How do different decomposition approaches affect a process model in terms of metrics associated with maintainability and understandability?" The first question is addressed via a literature review and classification of process model decomposition approaches. The second question is addressed via a controlled experiment. Specifically, two representative decomposition heuristics are used to decompose a real-life flat process model. Then, we compare the resulting set of decomposed process models using a range of maintainability and understandability metrics.

The rest of the paper is structured as follows. Section 2 presents a literature review on process decomposition, a categorization of proposed heuristics, and a discussion of observations made. Next, Sect. 3 introduces and discusses the results of the experiment where different heuristics are applied and compared. Finally, Sect. 4 concludes the paper and outlines future work.

2 Literature Review

In this section, we present a literature review aimed at inventorying and classifying existing decomposition approaches to address the research question of "How can process models be decomposed?" We also identify metrics used to assess process model understandability and maintainability as basis for our second research question "How do different decomposition approaches affect a process model in terms of metrics associated with maintainability and understandability?"

2.1 Search Process

The literature search process was based on the principles of Kitchenham (2004). We submitted queries to Google Scholar (which encompasses relevant databases such as ACM DL and IEEE Xplore), using three different queries (conducted October 2014). These three queries had one of the following keywords "process model", "process modeling" and "workflow" in combination with all of the following terms; "modularization", "decomposition", "sub-process", "fragment", "abstraction", "refactoring", "hierarchy". The first 400 hits of each of the three queries (1600 hits in total) were examined.

In the first round of filtering, based on title only, we eliminated duplicates and papers that were clearly off-topic. This iteration reduced the list to 250 candidate papers. We then proceeded with an inspection of the abstract and the introduction of each paper to eliminate papers that did not deal with process modeling. For instance, many papers dealt with decomposition or modularization of information systems or software code but had no significant relation to process models. We also eliminated papers that did not propose specific approaches to process decomposition, but instead dealt with another topic and referred to process decomposition as a separate issue. At the end, we obtained 72[1] relevant publications.

[1] The list of publications can be accessed at http://sep.cs.ut.ee/Main/ProcessDecomposition.

An initial analysis of these 72 publications revealed two distinct categories. On the one hand, one subset of publications (55 papers) provided prescriptive methods or guidelines for decomposing a given process model into sub-processes. The other subset of the publications (17 papers) proposed *criteria* and associated *metrics* to assess the "goodness" of a given decomposition without prescribing how a process model should be decomposed in order to achieve a suitable level of goodness. Here we use the term *decomposition heuristics* to refer to approaches of the first category and *decomposition criteria* to refer to the second category. Below we discuss each category in turn.

2.2 Decomposition Heuristics

From the 55 publications dealing with decomposition heuristics, we found, as expected, papers proposing "manual" methods or guidelines for process model decomposition. In addition, we found decomposition heuristics being employed in other related contexts such as refactoring. The heuristics found are summarized and presented below.

A process model can be decomposed based on "milestones" in the process. For instance, when decomposing existing EPC models (Davis 2001), decomposition is performed on points in the process where there are (1) limited connection to other parts of the process, (2) connected events, (3) limited use of loops, and (4) a common distinct theme (such as order fulfillment). Another method (Sharp and McDermott 2009) adopts a similar approach where the aim is to decompose a process starting at points where significant milestones in the overall process are achieved. They are usually points of interest in terms of process measurement. In Milani et al. (2013), the authors propose aligning the decomposition with variants and milestones.

Others (Ivanović et al. 2010) propose fragmenting a workflow based on data objects by looking at which and how many data inputs an activity has and which other activities share the same data objects. In de Leoni et al. (2014) the authors utilize data objects in combination with "single entry single exit" (SESE) as a basis for decomposition heuristics. These approaches postulate that if many activities use the same data objects, they are related and thus belong in one process fragment.

Another set of approaches, as introduced in Kim et al. (2005), Khalaf and Leymann (2006), proposes that all stakeholders model their own separate processes that then are included as parts of a larger process model. The same idea permeates an approach presented in Eberle et al. (2009) who consider that process knowledge is fragmented at the local levels (building blocks for a larger process model). As such, all local experts will model their own fragments (sub-process), which are subsequently put together with other fragments. Subject-oriented BPM (S-BPM) also brings the resource of the process in focus in that decomposition captures the activities performed by a specific role (Turetken and Demirors 2011).

Yet another basis for decomposition is to consider the goal or the output of the process such as in goal modeling (Pohl 2010). In the context of business process modeling, several approaches (Antón et al. 1994; Kueng and Kawalek 1997) base the decomposition of a business process based on goals and sub-goals. Specifically, elements in the process (e.g., activities) are clustered based on the goals/sub-goals they intend to achieve, and the resulting clusters are mapped to separate sub-processes.

Another set of modeling approaches focus on the iterative characteristics of process fragments, specifically prevalent in product development processes. Product development processes concern the process of transforming a technical solution to a product that can be sold to customers (Westerberg et al. 1997) and often include several iterations of sections of the process. For instance, in Rogers (1990), Kusiak and Wang (1993), Eppinger et al. (1994), Li and Moon (2012), León et al. (2013) decomposition is based on sequential, parallel or cyclical behavior of process fragments. As such, fragments exhibiting iterative characteristics are clustered together as one sub-process.

Business Process Model Abstraction (BPMA) methods apply techniques to detailed process models for the purpose of generating generalized versions (Polyvyanyy et al. 2010). Therefore BPMA techniques collect and cluster a certain set of atomic activities or sub-processes and represent them with one aggregated sub-process by using selection criteria. The criteria used can be roles (resources), activity frequency or activity completion time (Smirnov et al. 2012), structural aspects of a process model (Polyvyanyy et al. 2009), or semantic aspects (Sadiq and Governatori 2010; Smirnov et al. 2010). Once the perspective is chosen, the process models are transformed (decomposed) accordingly.

Process model refactoring improves understandability and maintainability of process models by changing them (e.g., by decomposition) without affecting their execution semantics (Dijkman et al. 2011). Although refactoring entails many techniques, some result in decomposition of activities and sub-processes. These are, e.g., redundancies in process models (repetition of the same fragments), which are extracted and set up as a sub-process. Another example is lazy process models (sub-processes containing few activities) and techniques aiming at addressing frequently occurring deviations from the main process. These can be managed by representing them by one or more "generalizing" sub-processes (Weber et al. 2011).

In process architecture, decomposition is found either in the form of aggregation ("part-of relation" – the process is decomposed into fully contained sub-processes) or generalization ("is-a relation" – the process is decomposed into variants representing alternative ways of performing the process) (Muehlen et al. 2010). For instance, in Muehlen et al. (2010), milestone and stakeholder based decomposition heuristics is proposed. In Malinova et al. (2013), the authors found that practitioners decompose their process models based on number of elements (size), complexity, or stakeholders. Similarly in Dijkman et al. (2014), five different principles for decomposition are elicited. These are (1) goal-oriented, (2) function-based, (3) reference model-based (adapting an industry reference model), (4) object-based, and lastly, (5) based on business units.

Finally, methods to "parse" a process model into a hierarchy of (SESE) fragments (Vanhatalo et al. 2009; Huang et al. 2014) have been used to decompose process models. Changes in such fragments are locally confined and thus the fragments are independent of each other. These properties allow for fragments to be extracted as separate sub-processes, thus providing a basis for automated process model decomposition (Uba et al. 2011). For instance, label similarity based on SESE extraction techniques has been applied to determine which activities should be subsumed under one sub-process (Reijers et al. 2011). This automated decomposition method is based on the assumption that nodes with similar labels (excluding control nodes) are more likely to belong to the same sub-process than those with different labels (measured, e.g., via string-edit distance).

When examining the various approaches and their underlying foundation for decomposing process models, we found that the approaches could be categorized based on their common denominator (underlying principle for decomposing). Our analysis distinguished six classes of decomposition heuristics based on the following underlying principles; breakpoints, data objects, roles, repetition, sharing and structuredness (cf. Table 1).

The common denominator of approaches subsumed by *breakpoints* is their reliance on milestones or breakpoints of the process. In these methods, decomposition is performed at points representing natural phases of the business process in its path to fulfill an objective. For instance, heuristics based on goal-decomposition (Antón et al. 1994; Kueng and Kawalek 1997) cut the process at points where sub-goals are achieved. Similarly, other authors (Davis 2001; Sharp and McDermott 2009; Muehlen et al. 2010) propose to decompose at points where two sub-processes have distinct themes and therefore constitute different milestones or separate functions in the process. Breakpoints for decomposition are also used in the context of reference process models, for example in the MIT process handbook (Malone et al. 1993).

Object-based heuristics assume that activities sharing common objects belong together and thus should be

Table 1 Decomposition heuristics

Decomposition heuristics	References
Breakpoints	Antón et al. (1994), Kueng and Kawalek (1997), Smith and Morrow (1999), Davis (2001), Sharp and McDermott (2009), Muehlen et al. (2010), Milani et al. (2013), Dijkman et al. (2014)
Data objects	Ivanović et al. (2010), Conforti et al. (2014), de Leoni et al. (2014), Dijkman et al. (2014)
Role	Pimmler and Eppinger (1994), Kim et al. (2005), Khalaf and Leymann (2006), Eberle et al. (2009), Muehlen et al. (2010), Turetken and Demirors (2011), Smirnov et al. (2012), Malinova et al. (2013), Dijkman et al. (2014)
Shared processes	Rogers (1990), Kusiak and Wang (1993), Eppinger et al. (1994), Weber et al. (2011)
Repetition	Dijkman et al. (2011), Uba et al. (2011), Weber et al. (2011), Li and Moon (2012), León et al. (2013)
Structuredness	Sadiq and Governatori (2010), Smirnov et al. (2010), Reijers et al. (2011), Huang et al. (2014)

located in one sub-process. These approaches consider the objects as primary driver for decomposition decisions.

Role based heuristics ground their decomposition decisions on “who” is performing the activities (Pimmler and Eppinger 1994; Muehlen et al. 2010; Smirnov et al. 2012). These approaches are applied to collaborative process modeling where different organizations or business units contribute with their own fragments as proposed by Kim et al. (2005), Eberle et al. (2009), or when modeling for outsourcing purposes (Khalaf and Leymann 2006).

Shared processes subsume approaches, such as (Dijkman et al. 2011; Uba et al. 2011; Weber et al. 2011), that seek to reduce redundancy by modeling process fragments that are called upon multiple times in different parts of a process, into one sub-process.

Repetition-based heuristics, on the other hand, consider the frequency of activities as their basis for decomposition. For instance, some (Rogers 1990; Kusiak and Wang 1993; Eppinger et al. 1994) propose separating sets of activities that are repeated more often (cyclical) from those that are sequential or parallel. Another example is the generalization of frequently occurring instance and variant changes into a separate sub-process (Weber et al. 2011).

The final class of heuristics for decomposition is based on the *structuredness* of the process models, i.e., using SESE fragments as a basis for identifying candidate sub-processes (Sadiq and Governatori 2010; Smirnov et al. 2010; Uba et al. 2011; Reijers et al. 2011).

2.3 Decomposition Criteria

In this section we review metrics associated with process model decomposition rather than specific decomposition heuristics.

It has been shown that larger process models tend to hamper understandability (Reijers and Mendling 2011) and increase the probability of making errors (Mendling et al. 2007). On this basis, “good” sub-processes are neither too small nor overly large (Weber et al. 2011). For instance, the IDEF0 method proposes 4–6 activities (Muehlen et al. 2010) while others state 5–15 (Kock and McQueen 1996), or 5–7 (Sharp and McDermott 2009) activities per process model. Others still propose limiting the number of elements to less than 50 (Mendling et al. 2007) or 31 nodes (Rosa et al. 2012). Another study (Cardoso and Mendling 2006) proposes considering other size related metrics, including the number of specific elements such as activities and control-flow elements, or the number of activities, joins, and splits in a process model.

Complexity (as various ratios of process model elements) is reversely correlated to understandability and increases the probability for errors (Cardoso and Mendling 2006). For measuring complexity, the CFC (aggregated number of branches from all split constructs of a process model) metrics (Cardoso 2005) and HPC (measuring length, volume and difficulty of a process model) metrics (Cardoso and Mendling 2006) have been proposed. Other complexity metrics are CNC (number of arcs divided by nodes) (Latva-Koivisto 2001) and CI (number of node reductions required to reduce a process model to one node). Density (relation between nodes and arcs) of a process model is also an approximation of its complexity. A high density value indicates a more complex process model and is negatively related to understandability (Vanderfeesten et al. 2008a; Dumas et al. 2012).

Coupling metrics, as inspired by software engineering, have been transposed to process models such as “density metrics” (Mendling 2006), “cross-connectivity metric” (Vanderfeesten et al. 2008a), “connectedness” (Reijers et al. 2011), “weighted coupling metric” (Vanderfeesten et al. 2007), “process coupling” (Reijers and Vanderfeesten 2004; Vanderfeesten et al. 2008b), and adaptation of coupling metrics for eEPC models (Braunnagel et al. 2014). A common feature of these metrics is that they look at the connectedness of control-flow elements in a sub-process. Therefore, when a collection of nodes are seen as connected to each other, they are more likely to be related and should belong to the same sub-process.

Cohesion is closely related to the notion of coupling. Cohesion refers to how closely the sub-elements of a given

Table 2 Metrics for decomposing

Process model metrics	References
Size	Kock and McQueen (1996), Cardoso and Mendling (2006), Sharp and McDermott (2009), Muehlen et al. (2010)
Coupling and cohesion	Daneva et al. (1996), Reijers (2003), Reijers and Vanderfeesten (2004), Mendling (2006), Vanderfeesten et al. (2007, 2008a, b), Reijers et al. (2011), Braunnagel et al. (2014)
Complexity	Latva-Koivisto (2001), Cardoso (2005), Cardoso and Mendling (2006)

sub-process are internally connected. Cohesion measures such as "functional", "event" and "logical connectors" are proposed in Daneva et al. (1996). A cohesion metrics has also been developed based on the "steps" composing an activity and their associated data objects (Reijers 2003; Reijers and Vanderfeesten 2004; Vanderfeesten et al. 2008b).

Our observation distinguished three main classes of metrics for decomposition based on the following underlying principles: size, coupling and cohesion, and complexity (cf. Table 2).

2.4 Discussion

We note that some decomposition heuristics can be applied surgically, i.e., on sections of process models that exhibit specific patterns. The "repetition", "shared processes" and "role based" heuristics can be implemented on process fragments if and only if certain conditions are fulfilled. For instance, "repetition" and "shared processes" heuristics are only applicable to process fragments that exhibit such patterns. Likewise, "role based" heuristics offer guidelines for cases with several stakeholders but cannot be applied when for instance, the process of one stakeholder is large and is in need of further decomposing. As such, these heuristics do not provide enough support to be generally applied on a set of process models but rather function as complementary to other heuristics. The "breakpoint", "data object", and "structuredness" heuristics on the other hand can be applied generally on process models, as they are not dependent upon certain conditions being fulfilled.

Furthermore, we note that the heuristics do not provide sufficient criteria for determining which fragment of the process model to include as a separate sub-process. A heuristics approach might offer necessary criteria (statements on how to decompose) but not sufficient criteria (statements determining which process fragments to include in a sub-process). For instance, "structuredness" provides necessary criteria (SESE blocks) but not sufficient criteria for determining which SESE fragment to use (usually there are many such fragments in a sub-process). It is possible to set threshold values that function as a "guide" when determining which activities to include in a sub-process, but such strategies are not sufficiently refined yet (Reijers et al. 2011). In a similar manner, "breakpoint" and "data object" provide necessary but not sufficient criteria. For instance, the breakpoint heuristics do not give sufficient guidance for identifying "milestone steps" in a process. Similarly, data object heuristics lack proper guidance for determining when a set of activities shares the same set of data objects. As such, there are currently no heuristics that provide both necessary and sufficient criteria for process model decomposition. However, in contrast to "structuredness", breakpoint and data object heuristics, while not offering sufficient criteria, can be applied by relying on the knowledge of the domain experts and should, intuitively, produce process models that better reflect the actual business processes. It should be noted that the data object heuristics not only require that the data objects be modeled but that they are captured in a consistent manner across the process models. Otherwise it would be very difficult, if not impossible to apply these heuristics.

We also note that the heuristics for decomposition are highly inspired by conceptual modeling while metrics for assessing decompositions are direct transposition of metrics from programming and software design to process models (Cardoso and Mendling 2006; Muketha and Ghani 2010). For instance, coupling and cohesion metrics are inspired by Wand and Weber (Johannsen and Leist 2012), size by lines of code (LOC) (Cardoso and Mendling 2006), modularity by information flow by Henry and Kafura (Cardoso and Mendling 2006), and complexity by McCabe's cyclomatic complexity metrics (Muketha and Ghani 2010). As such, decomposition heuristics and metrics have emerged quite independently of each other as separate streams of research. In the literature on process model metrics, there are no substantiated claims that a certain metric is more suited for use together with certain decomposition heuristics.

3 Controlled Experiment

In this section we analyze the second research question on "How do different decomposition heuristics affect a process model in terms of metrics associated with maintainability and understandability?” To achieve an answer we performed an experiment.

Subjects The experiment was conducted in April 2015 with second year master students of a BPM course. The population consisted of 36 (voluntary) students who were in their third month of a business process management course. As such, they had gained the required background and familiarity required for reading, understanding, and working with process models. Each student was randomly assigned to one of two groups (decomposition based on data object or breakpoint heuristics).

Objects The flat process model used as input originates from the operations of a mid-sized European bank managing fixed income products. The process model used had been modeled for documentation purposes by a team of consultants. This model is flat, i.e., no decomposition has been made. It begins with a start event and continues until the end of the process (including data objects).[2]

Factor and factor levels The main factor in this experiment is the heuristics used to decompose the process model. As such, the same process model is given to two groups of students to be decomposed according to predetermined heuristics.

Response variable The response variable used was the process model metrics of the decomposed process models (size, cohesion, complexity, and density).

Hypothesis formulation The goal of the experiment is to investigate if the different decomposition heuristics (data object and breakpoint) cause any significant differences as to size, cohesion, complexity and density metrics. As two different heuristics are applied, we expect the resulting models to exhibit differences in process model metrics. Accordingly, the following hypotheses are formulated.

- *Hypothesis*: Different decomposition heuristics (data object and breakpoint) cause significant differences in size, cohesion, complexity and density metrics.
- *Alternative hypothesis*: Different decomposition heuristics (data object and breakpoint) do not cause significant differences in size, cohesion, complexity and density metrics.

Experimental design Both groups received (1) an introduction to the experiment, (2) an explanation of the concepts of process model decomposition, (3) an overview of the flat process model, and (4) practical instructions such as submission format. The only difference in the given introductions was the decomposition heuristics. Each group was only introduced to the heuristics they were assigned to. The students were randomly assigned to one of the two groups. The students were given 6 days to decompose the flat process model. In addition to the decomposition task, they also answered questions regarding the extent of their previous experience with process models. The students were also asked to rate the difficulty of applying the heuristics they had been assigned.

The submitted sets of sub-processes were analyzed to ensure that they were complete. Submissions requiring interpretation (such as to which sub-process an activity should belong) were discarded. At the end of the filtering process, 25 valid submissions remained. Of these, 14 had applied data object heuristics and 11 had applied breakpoint heuristics. For each valid submission, the number of sub-processes, size (number of activities and nodes per sub-process), cohesion, CNC, and density of all sub-processes were calculated. The average of these values was calculated for the whole set of sub-processes of each submission and used as value for the entire set of decomposed process models. For instance, if a student had decomposed the flat process model into 10 different sub-processes, the values of each of the ten sub-processes were first calculated. Then the average value of each metric for all the 10 sub-processes was calculated and used in the data analysis presented below.[3]

The two groups were comparable in terms of prior experience with process models and familiarity with BPMN. For instance, both groups had created or edited about 7.5 (7.64 for breakpoint and 7.43 for data object) process models during the past year. These process models had an average size of 18 activities (18.64 for breakpoint and 18.79 for data object). In response to questions regarding familiarity and confidence in understanding and using BPMN, both groups stated on average that they ‘somewhat agree’ (on a 7-step scale of ‘strongly agree’, ‘agree’, ‘somewhat agree’, ‘neutral’, somewhat disagree’, ‘disagree’, and ‘strongly disagree’). Tests conducted to verify their statistical significance (t test and Mann–Whitney test) showed that the averages of both groups were similar with 95 % confidence.[4]

3.1 Findings

The decomposed process models were analyzed in terms of number of sub-processes, size (as measured by number of activities and nodes), cohesion, CNC, and density. For each set of decomposed process models, the average of the above metrics were calculated. The average values of these metrics are shown in Table 3 below.

[2] The flat process model can be accessed at http://sep.cs.ut.ee/Main/ProcessDecomposition.

[3] The values for the experiment can be accessed at http://sep.cs.ut.ee/Main/ProcessDecomposition.

[4] The statistical analysis can be accessed at http://sep.cs.ut.ee/Main/ProcessDecomposition.

Table 3 Average value of metrics

Heuristics	No of sub-processes	Size (activities)	Size (nodes)	Cohesion	CNC	Density
BreakPoint	16.73	6.48	12.46	0.26	1.01	0.15
DataObject	15.14	5.34	10.08	0.38	0.96	0.15

Table 4 Shapiro–Wilk test determining if the data is normally distributed

Heuristics	No of sub-processes	Size (activities)	Size (nodes)	Cohesion	CNC	Density
BreakPoint	0.990	0.000	0.000	0.953	0.124	0.859
DataObject	0.454	0.001	0.008	0.323	0.004	0.006

Table 5 p value of two-sample t tests and Mann–Whitney tests

	Test	p value
Average no of sub-processes	Two-sample t test	0.502
Average size (activities)	Mann–Whitney test	0.809
Average size (nodes)	Mann–Whitney test	0.687
Average cohesion	Two-sample t test	0.004
Average CNC	Mann–Whitney test	0.309
Average density	Mann–Whitney test	0.689

In order to determine if the average values of both groups are similar (with statistical confidence), the two-sample t test (for normal distributed samples) or Mann–Whitney test (for non-normal distributed samples) was conducted. In order to determine which of these tests should be applied, the Shapiro–Wilk test (results shown in Table 4) of normality was conducted.

The Shapiro–Wilk test shows that the two-sample t test can be applied on 'number of sub-processes' and 'cohesion' as the p value for both sets is above 0.05. For the other metrics, which are not normally distributed, the Mann–Whitney test is conducted to determine if there are significant differences in the averages of the two sets.

The results show that there is not enough evidence to support the hypothesis that the values of the two sets, in regards to 'average no of sub-processes', 'average size (activities)', 'average size (nodes)', 'average CNC', and 'average density' are significantly different as the p value is above 0.05 (cf. Table 5). We infer therefore that the average values of the two sets (BreakPoint and DataObject) are similar. However, as the p value for 'average cohesion' is below 0.05 (cf. Table 5), it can be stated with 95 % confidence that their averages differ. This is expected as data object heuristics base decomposition on the cohesion of the activities. Any other results would have been surprising.

The violin plot and box plot of the metrics, as visually shown in Fig. 1, express the same results. An interesting observation is that the standard deviations of all metrics in breakpoint heuristics are noticeably higher than for data object heuristics.

For instance, as can be seen from Table 6, the standard deviation of the breakpoint set is often about twice as high as for data object heuristics (with the exception of cohesion). Furthermore, the figures of the box plots in Fig. 1 (encompassing 50 % of the data) for breakpoint heuristics are higher than those of the data object heuristics. This indicates that following data object heuristics results in more consistency with regard to the number of sub-processes, size, complexity, and density. In other words, breakpoint heuristics allow more interpretation as to where the breakpoints are in the flat process model during decomposition.

The participants also answered a questionnaire that included 4 questions regarding the degree of difficulty in (Q1) understanding the flat process model, (Q2) difficulty in identifying logical breakpoints or common data objects, (Q3) difficulty in determining which activities should belong to one sub-process, and (Q4) difficulty in applying the heuristics when decomposing the flat process model. The responses were given on a scale between 1 and 5 ('very simple', 'simple', 'neutral', 'rather difficult' and 'very difficult'). In order to detect any dissimilarity between the responses given by those who applied breakpoint as compared to data object heuristics, the Fisher's Exact test was conducted (cf. Table 7).

A closer look at the results for Q3 and Q4 (cf. Table 8) show that there is a slight overweight of 4 (rather difficult) from those who applied data object heuristics for both questions.

The Fisher's Exact test indicates that for Q1 and Q2, the responses given by the two groups are similar (high p value). However, for Q3 and Q4 (in particular for Q3) there seems to be an indication that it was more difficult to determine which activities should belong to one sub-process (Q3) and to apply data object heuristics when decomposing a flat process model (Q4).

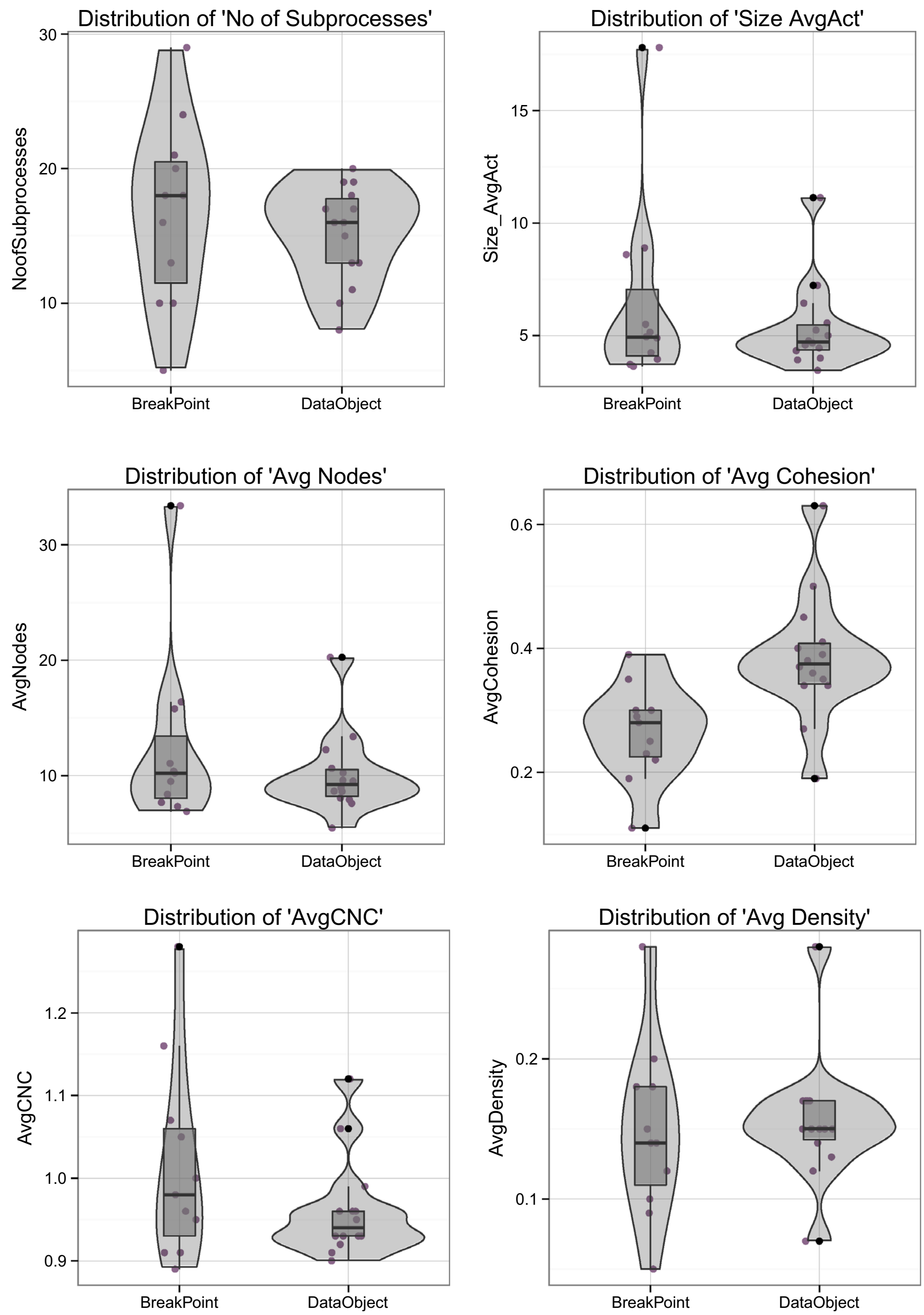

Fig. 1 *Violin plot* and *boxplot* of the metrics

Table 6 Standard deviation

Heuristics	No of sub-processes	Size (activities)	Size (nodes)	Cohesion	CNC	Density
BreakPoint	6.92	4.16	7.63	0.08	0.12	0.06
DataObject	3.65	1.94	3.53	0.10	0.06	0.04

Table 7 Fisher's Exact test

	Q1	Q2	Q3	Q4
Fisher's exact test (p value)	0.5844	0.6006	0.1882	0.3661

3.2 Threats to Validity

The sample size and one flat process model to be decomposed are not sufficient to draw conclusions about the effects

Table 8 Distribution of responses for Q3 and Q4

Scale	1	2	3	4	5	Total
Q3 – How easy or hard was it to determine which activities should be in one sub-process model?						
BreakPoint	0.00	0.27	0.55	0.18	0.00	1.00
DataObject	0.00	0.14	0.29	0.57	0.00	1.00
Q4 – How easy or hard was it to apply the breakpoint/data object approach when decomposing the large flat process model?						
BreakPoint	0.00	0.27	0.46	0.27	0.00	1.00
DataObject	0.00	0.29	0.21	0.50	0.00	1.00

of different heuristics on understandability and maintainability. Due to this, our discussion in the previous section is indicative and based on observations rather than statements or conclusions. Further experiments are required to make the findings more conclusive. Another threat to validity results from the use of size and complexity metrics as proxy for understandability and maintainability. Understandability of process models entails factors beyond metrics, such as cognitive factors, perceived understandability and layout. This limitation is shared with several other studies (Dumas et al. 2012). Finally, it should be noted that the experiment was conducted with students who areas such novice modelers. However, their level of expertise was uniform, i.e., no expert modelers participated.

4 Conclusion

The benefits of process model decomposition for increased comprehensibility and maintainability are widely recognized, however, there is no comparable consent regarding the question of how to decompose a process model. In fact, some argue that this is more an art than science. Nevertheless a variety of approaches have been proposed. In light of the proposed approaches, we examined how a process model can be decomposed and how different decomposition heuristics affect the process models.

Our survey showed (as discussed in Sect. 2.3) that three heuristics (repetition, shared processes, and role based processes) do not require certain conditions to be fulfilled. Applying these heuristics would not produce enough candidates for sub-processes, as sections of process models that do not fulfill the required conditions would not be fragmented to sub-processes. These heuristics are therefore useful as complementary to other heuristics. On the other hand, three heuristics (breakpoint, data object and structuredness) provide necessary but not sufficient criteria for decomposition. These heuristics can be generally applied for process model decomposition but fall short of determining which process fragments to model as a sub-process.

We examined how the different heuristics (breakpoint and data object-based) affect understandability and maintainability of process models by conducting an experiment. Using quantitative metrics as approximation of understandability, the experiment showed that the two heuristics are similar in terms of understandability and complexity. Therefore choosing heuristics could be influenced by factors such as the type of stakeholders involved or the level of details captured in the models.

However, understandability of process models is not restricted to its metrics only. Therefore, exploring the effect of decomposition on cognitive understandability is a relevant and important direction for future work. We also observed that the research on decomposition of process models has been conducted independently of goodness criteria and associated metrics. As such, little, if any, research has been conducted on the correlation between the type of processes and metrics. Hence, it is important to conduct further empirical studies on the relation between decomposition heuristics and understandability and maintainability metrics.

Acknowledgments This research was supported by the European Social Fund via the Doctoral Studies and Internationalisation Programme – DoRa.

References

Antón A, McCracken W, Potts C (1994) Goal decomposition and scenario analysis in business process reengineering. Adv Inform Syst LNCS 811:94–104

Becker J, Becker J, Winkelmann A (2009) Developing a business process modeling language for the banking sector – a design science approach. In: Proceedings AMCIS 2009. http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1718&context=amcis2009. Accessed 18 Oct 2015

Braunnagel D, Johannsen F, Leist S (2014) Coupling and process modeling – an analysis at hand of the eEPC. In: Proceedings Modellierung 2014. Wien, pp 121–136. http://epub.uni-regensburg.de/29719/. Accessed 18 Oct 2015

Burton-Jones A, Meso PN (2004) Conceptualizing systems for understanding: an empirical test of decomposition principles in object oriented analysis. Inf Syst Res 17(1):38–60

Cardoso J (2005) How to measure the control-flow complexity of web process and workflows. http://www.academia.edu/3156014/How_to_Measure_the_Control-flow_Complexity_of_Web_Processes_and_Workflows. Accessed 18 Oct 2015

Cardoso J, Mendling J (2006) A discourse on complexity of process models. Bus Process Manag Workshops LNCS 4103:117–128

Conforti R, Dumas M, García-Bañuelos L, La Rosa M (2014) Beyond tasks and gateways: discovering BPMN models with subprocesses, boundary events and activity markers. Bus Process Manag LNCS 8659:101–117

Daneva M, Heib R, Scheer A (1996) Benchmarking business process models. Technical Report, Saarland University

Davis R (2001) Business process modelling with ARIS: a practical guide. Springer, New York

De Leoni M, Munoz-Gama J, Carmona J, Van der Aalst WMP (2014) Decomposing alignment-based conformance checking of data-aware process models. In: On the move to meaningful internet systems: OTM 2014 Conferences. Springer, Heidelberg, pp 3–20

Dijkman R, Gfeller B, Küster J, Völzer H (2011) Identifying refactoring opportunities in process model repositories. Inf Softw Technol 53(9):937–948

Dijkman R, Vanderfeesten I, Reijers HA (2014) Business process architectures: overview, comparison and framework. Enterp Inf Syst. doi:10.1080/17517575.2014.928951

Dumas M, La Rosa M, Mendling J, Raul M (2012) Understanding business process models: the costs and benefits of structuredness. In: CAiSE'12 Proc 24th Int Conf Adv Inf Syst Eng, vol 7328, pp 31–46

Eberle H, Unger T, Leymann F (2009) Process fragments. In: On the move to meaningful internet systems. LNCS, vol 5870, pp 398–405

Eppinger SD, Whintey DE, Smith RP, Gebala DA (1994) A model-based method for organizing task in product development. Res Eng Des 6(1):1–13

Huang Y, He K, Feng Z, Huang Y (2014) Business process consolidation based on E-RPSTs. In: Serv. (SERVICES), 2014 IEEE World Congr. IEEE, pp 354–361

Ivanović D, Carro M, Hermenegildo M (2010) Automatic fragment identification in workflows based on sharing analysis. LNCS 6470:350–364

Johannsen F, Leist S (2012) Wand and Weber's decomposition model in the context of business process modeling. Bus Inf Syst Eng 4(5):275–286

Khalaf R, Leymann F (2006) E role-based decomposition of business processes using BPEL. In: Proceeding ICWS'06 Proc IEEE Int Conf Web Serv, pp 770–780

Kim K, Won J, Kim C (2005) A fragment-driven process modeling methodology. In: Computational science and its applications – ICCSA 2005. LNCS, vol 3482, pp 817–826

Kitchenham B (2004) Procedures for performing systematic reviews. Technical Report, Keele Univ, vol 33, p 28

Kock NFJ, McQueen RJ (1996) Product flow, breadth and complexity of business processes: an empirical study of 15 business processes in three organizations. Bus Process Re-eng Manag J 2(2):8–22

Kueng P, Kawalek P (1997) Goal-based business process models: creation and evaluation. Bus Process Manag J 3:17–38

Kusiak A, Wang J (1993) Efficient organizing of design activities. Int J Prod Res 31(4):753–769

Latva-Koivisto A (2001) Finding a complexity measure for business process models. Tech. Rep. Helsinki Univ. Technol. Syst. Anal., pp 1–26

León HCM, Farris JA, Letens G, Hernandez A (2013) An analytical management framework for new product development processes featuring uncertain iterations. J Eng Technol Manag 30(1):45–71

Li W, Moon YB (2012) Modeling and managing engineering changes in a complex product development process. Int J Adv Manuf Technol 63(9):863–874

Malinova M, Leopold H, Mendling J (2013) An empirical investigation on the design of process architectures. In: Proceedings of the 11th international conference on Wirtschaftsinformatik 2013, Leipzig

Malone TW, Crowston K, Lee JLJ, Pentland B (1993) Tools for inventing organizations: toward a handbook of organizational processes. Proc Second Work Enabling Technol Collab Enterp 45:425–443

Mendling J (2006) Testing density as a complexity metric for EPCs. Tech. Rep. Vienna Univ. Econ. Bus. Adm.

Mendling J, Neumann G, van der Aalst W (2007) Understanding the occurrence of errors in process models based on metrics. Lect Notes Comput Sci 4803:113–130

Mendling J, Reijers HA, van der Aalst WMP (2010) Seven process modeling guidelines (7PMG). Inf Softw Technol 52(2):127–136

Milani F, Dumas M, Matulevičius R (2013) Decomposition driven consolidation of process models. Adv Inform Syst Eng LNCS 7908:193–207

Muehlen MZ, Wisnosky D, Kindrick J (2010) Primitives: design guidelines and architecture for BPMN models. In: Australas. Conf. Inf. Syst

Muketha G, Ghani A (2010) A survey of business processes complexity metrics. Inf Technol J 9(7):1336–1344

Pimmler TU, Eppinger SD (1994) Integration analysis of product decompositions. Alfred P. Sloan School of Management, MIT, Cambridge

Pohl K (2010) Requirements engineering: fundamentals, principles, and techniques. Springer, New York

Polyvyanyy A, Smirnov S, Weske M (2009) The triconnected abstraction of process models. In: Dayal U, Eder J, Koehler J, Reijers H (eds) Business process management, vol 5701. LNCS, pp 229–244

Polyvyanyy A, Smirnov S, Weske M (2010) Business process model abstraction. In: Handb. Bus. Process Manag. 1. Springer, Heidelberg, pp 149–166

Reijers HA (2003) A cohesion metric for the definition of activities in a workflow process. In: Proc. EMMSAD. pp 116–125

Reijers HA, Mendling J (2011) A study into the factors that influence the understandability of business process models. IEEE Trans Syst Man Cybern 41(3):449–462

Reijers HA, Vanderfeesten I (2004) Cohesion and coupling metrics for workflow process design. In: Proc Bus Process Manag – Second Int Conf BPM 2004, Potsdam, pp 290–305

Reijers HA, Mendling J, Dijkman RM (2011) Human and automatic modularizations of process models to enhance their comprehension. Inf Syst 36(5):881–897

Rogers JL (1990) Knowledge-based tool for decomposing complex design problems. J Comput Civ Eng 4(4):298–312

Rosa L, Mendling J, La Rosa M (2012) Thresholds for error probability measures of business process models. J Syst Softw 85(5):1188–1197

Sadiq S, Governatori G (2010) Managing regulatory compliance in business processes. In: vom Brocke J, Rosemann M (eds) Handb. Bus. Process Manag. 2. Springer, Heidelberg, pp 159–175

Sharp A, McDermott P (2009) Workflow modeling: tools for process improvement and applications development. Artech House

Smirnov S, Dijkman R, Mendling J, Weske M (2010) Meronymy-based aggregation of activities in business process models. Concept Model LNCS 6412:1–14

Smirnov S, Reijers HA, Weske M, Nugteren T (2012) Business process model abstraction: a definition, catalog, and survey. Distrib Parallel Databases 30(1):63–99

Smith RP, Morrow J (1999) Product development process modeling. Des Stud 20(3):237–261

Turetken O, Demirors O (2011) Plural: a decentralized business process modeling method. Inf Manag 48(6):235–247

Uba R, Dumas M, García-Bañuelos L, La Rosa M (2011) Clone detection in repositories of business process models. Bus Process Manag LNCS 6896:248–264

Vanderfeesten I, Cardoso J, Reijers HA (2007) A weighted coupling metric for business process models. CEUR Workshop Proc 247:41–44

Vanderfeesten I, Reijers HA, Mendling J, van der Aalst WMP, Cardoso J (2008a) On a quest for good process models: the cross-connectivity metric. LNCS 5074:480–494

Vanderfeesten I, Reijers HA, van der Aalst WMP (2008b) Evaluating workflow process designs using cohesion and coupling metrics. Comput Ind 59(5):420–437

Vanhatalo J, Völzer H, Koehler J (2009) The refined process structure tree. Data Knowl Eng 68(9):793–818. doi:10.1016/j.datak.2009.02.015

Weber B, Reichert M, Mendling J, Reijers HA (2011) Refactoring large process model repositories. Comput Ind 62(5):467–486

Westerberg AW, Subrahmainan E, Reich Y, Konda S (1997) Designing the process design process. Comput Chem Eng 21(Supplement):S1–S9

Wolter C, Schaad A (2007) Modeling of task-based authorization constraints in BPMN. In: Alonso G, Dadam P, Rosemann M (eds) Business process management, vol 4714. LNCS, pp 64–79