DEVELOPMENT OF METRICS FOR EVALUATING DECOMPOSED PROCESS MODELS BASED ON WAND AND WEBER'S GOOD DECOMPOSITION MODEL

Research paper

Johannsen, Florian, University of Bremen, Bremen, Germany, fjohanns@uni-bremen.de

Leist, Susanne, University of Regensburg, Germany, Susanne.Leist@ur.de

Braunnagel, Daniel, University of Regensburg, Germany, Daniel.Braunnagel@ur.de

Abstract

Business process modelling is an important task in business transformation initiatives. Process models visualize the working procedures of a company and pinpoint the way in which business value is created. Based on process models, functional requirements on IS are derived and decisions on IS investments are made, for instance. However, in case process models become too large, employees will hardly understand them, which restricts the potential benefits associated with business process modelling. Therefore, the decomposition of process models is a means of reducing their complexity by delineating corresponding subprocess models. However, there are few commonly accepted approaches for decomposing process models only and the properties that characterize a well-done decomposition are rather unclear. We thus revert to the good decomposition model of Wand and Weber, which was established for decomposing IS, as a means to judge the quality of decomposed process models. The present study develops metrics for evaluating decomposed process models in the eEPC notation against the good decomposition model of Wand and Weber. An application of the metrics to a process model from a cooperation project shows that the metrics provide a helpful way of objectively assessing the quality of decompositions in process modelling by using the good decomposition model.

Keywords: Decomposition, Business Process Modeling, Metrics.

1 Motivation

Enterprise modelling is used to represent the *"structure, activities, processes, information, resources, people, behavior, goals"* as well as *"constraints"* of an enterprise (Fox and Gruninger, 1998, p. 109). In this respect, process modelling has increasingly gained attention in recent years (Becker et al., 2010; Harmon, 2016). Process models are not only used for process analysis and improvement efforts, they also support the design of information systems (IS) and decision-making concerning information technology (IT) investments (Becker et al., 2010). However, creating process models is a highly subjective task (Pinggera et al., 2015). Accordingly, different quality perspectives and various approaches for designing and evaluating conceptual models are proposed in literature (cf. Mendling et al., 2010; Pinggera et al., 2015; Overhage et al., 2012). In this context, "process model understandability" has been established as a widely-accepted quality criterion (Fettke et al., 2012) referring to *"the degree to which information contained in a process model can be easily understood by a reader of that model"* (Reijers and Mendling, 2011, p. 451). While there are different factors influencing the understandability of process models (e.g., modeling expertise) (cf. Mendling et al., 2012), it has been shown that the model size plays a decisive role (cf. Mendling et al., 2007). In this regard, decomposition is a means of reducing model complexity by splitting large process models into smaller subprocess models (Milani et al., 2016; Zugal et al., 2015). Though the benefits of decomposing business process models are commonly accepted, decomposition is often done in an *"ad hoc fashion"* (Reijers et al., 2011, p. 882) since generally acknowledged guidelines to do so are missing (Milani et al., 2016; Reijers and Mendling, 2011; Burton-

Jones and Meso, 2008). Thus, there is uncertainty regarding those properties that characterize a good decomposition in process modelling. Considering this, the good decomposition model of Wand and Weber (cf. Weber, 1997) is promising for guiding modellers in decomposing process models purposefully, leading to an easier understanding of the decompositions of process models (Recker et al., 2009). In previous research, we specified the conditions of the decomposition model for modelling with Event-driven Process Chains (EPCs) and derived guidelines for a good decomposition (cf. Johannsen and Leist, 2012). However, checking the conformance of a decomposed model with the decomposition conditions requires a tremendous cognitive effort, if done manually.

Against this background, a formal operationalization of Wand and Weber's decomposition conditions in the form of metrics is beneficial for the following reasons. First, metrics provide a precise explanation of decomposition with regard to the good decomposition conditions and therefore support to judge objectively as to how a decomposed process model adheres to these conditions. In consequence, they diminish the subjectivity of user assessments. Second, we develop a software tool to perform the calculation automatically. To do so, the metrics' variables are mapped to corresponding algorithms and procedures executing the evaluation of the process model. This paper describes the development of metrics that measure the coherence of a decomposed process model with Wand and Weber's decomposition conditions. The ontological expressiveness of modelling languages differs (Recker, 2011), which, in consequence, impacts the interpretation of the decomposition conditions for modelling techniques. To provide the necessary level of detail nonetheless, we restrict our research to EPCs and we pose the following research question (RQ): *Which metrics can be derived to measure the coherence of a decomposed EPC process model with Wand and Weber's decomposition conditions?*

The contribution of our work is the following: first, we operationalize the decomposition conditions as a set of formal metrics, establishing an objective base for assessing the quality of a decomposition regarding the decomposition model. The metrics resort to the conditions of Wand and Weber to unveil properties of well-performed decompositions that have not been identified for the process modelling discipline so far. Therefore, our research strongly contributes to the ongoing discussion of how to decompose properly (cf. Milani et al., 2016) and provides means to assess the quality of decompositions by using metrics. Our paper is structured as follows: the following section describes theoretical foundations on Event-driven Process Chains, gives an overview of related work, and introduces the decomposition conditions for EPCs. Then, the research procedure is described and the metrics are presented and applied to a use case before the results are discussed. The paper is rounded off with a conclusion and an outlook on future research.

2 Basics and Related Work

2.1 Event-driven Process Chains and the good decomposition model

Event-driven Process Chains (EPCs) were developed in the early 1990s and are currently one of the most frequently used techniques for business process modelling (Harmon and Wolf, 2011; Mendling, 2008). A flat EPC model comprises nodes and arcs; a node can be a function type, an event type, a connector type or a process interface (Mendling, 2008). The EPC can be enhanced by several views (e.g., organizational view, data view) providing additional information for the user (Scheer et al., 2005). In this case, we speak of enhanced Event-driven Process Chains (eEPCs).

The "understandability" of eEPC models and process models in general is a much discussed topic (e.g., Fettke et al., 2012; Mendling et al., 2010; Zugal et al., 2011). Thereby, decomposition is a means to reduce the complexity of large process models and thus to increase their understandability (Reijers and Mendling, 2011; Zugal et al., 2015). The decomposition itself can be accomplished in two ways: model abstraction and model fragmentation (cf. Zugal et al., 2015; De Lara et al., 2013). During model abstraction, a modeller aggregates information by designing an abstract process model whereas model fragmentation means to spread detailed information across several subprocess models (cf. Zugal et al., 2015; De Lara et al., 2013). Thus, a modeller may either create a high-level process model and subsequently add information via subprocess models (model fragmentation) or design a detailed "flat" model from scratch and then delineate subprocess models by abstracting from the details (model abstraction) (Davis and Brabänder, 2007). A variety of suggestions on how to do the decomposition are found in literature (e.g., Milani et al., 2016). Van der Aalst (2013) for example investigates the decomposition of Petri nets in special, whereas Ma et al. (2015) propose an algorithm for an automatic decomposition of process models. So-called "single-entry-single-exit" (SESE) components of a process model are searched for in the "block structuring" approach as these are potential candidates for subprocess models (Reijers et al., 2011). Vanhatalo et al. (2009) cluster business processes according to "fragments" that have two boundary nodes and should be objective in addition, i.e. the fragments do not overlap. Based on that, the refined process structure tree can be deduced with the fragments representing potential subprocesses (cf. Vanhatalo et al., 2009). Another approach, which analyses the connections between "nodes" of a process model, is called "graph-clustering" (cf. Reijers et al., 2011). Accordingly, nodes that are strongly connected with each other should be captured within a subprocess model (cf. Reijers et al., 2011). Milani et al. (2016) analyse various decomposition approaches in a controlled experiment, highlighting that existing heuristics (e.g., role based heuristics) do not provide sufficient criteria for decomposition or do not necessarily support the delineation of subprocess models. More, generally accepted metrics that focus on the quality of decomposed process models in special can hardly be found (cf. Reijers et al., 2011; Vanderfeesten et al., 2007). Metrics could be developed regarding specific decomposition approaches as described above (e.g., block structuring). An example would be the number of "SESE-components" (Gerth, 2013) across all subprocess models of a decomposition. Though, such metrics would not be independent of a certain decomposition approach, which decreases their general applicability. In general, as Reijers et al. (2011) summarize, there is still the need to develop metrics enabling the objective evaluation of alternative designs of a decomposed business process model.

In this ongoing discourse, we build on the good decomposition model of Wand and Weber (cf. Weber, 1997; Wand and Weber, 1989) to guide users in decomposing process models and to judge the understandability of decomposed models. The good decomposition model originates in the IS discipline and is part of the BWW ontology (Weber, 1997). Burton-Jones and Meso (2006) show that adhering to the decomposition conditions positively affects the understandability of conceptual models in object-oriented modelling. The potential benefits of the decomposition model to come to a manageable set of subprocess models in large modelling projects were initially proposed by Recker et al. (2009). Burton-Jones and Meso (2006) show that adhering to the decomposition conditions positively affects the understandability of conceptual models in object-oriented modelling. In general, the decomposition model is attested a wide applicability and a comprehensive scope (Recker et al., 2005). In previous works, we transferred the decomposition conditions to business process modelling and specified them for eEPC models in particular (cf. Johannsen and Leist, 2012). We also show that the perceived understandability of eEPC models strongly profits from the decomposition model (cf. Johannsen et al., 2014). Because of that, the decomposition model is promising as a step towards developing a theory for explaining the quality of decomposed process models. What is missing is a formal specification of the decomposition conditions in the form of metrics. Adequate metrics precisely capture the constructs of the decomposition conditions as variables and thus facilitate an objective judgement as to which extent a decomposed process model adheres to the decomposition conditions.

2.2 The good decomposition model of Wand and Weber for eEPC modelling

The decomposition model builds on the representational model of the Bunge-Wand-Weber (BWW) ontology, which defines fundamental constructs and constituting components of an IS (Weber, 1997). The key construct of the BWW ontology is the "thing" (e.g., human, IT-system), which has certain properties (e.g., eye color) that are expressed via attributes (Weber, 1997; Rosemann and Green, 2002). Things can be grouped into systems and subsystems (Weber, 1997). Weber (1997) gives a detailed description of the BWW ontology, for example. To determine the quality of a decomposed IS, the decomposition model proposes five conditions (Weber, 1997): (1) minimality, (2) determinism, (3) losslessness, (4) minimum coupling, and (5) strong cohesion. These conditions are described in Table 1.

However, due to its origin in IS, the terms and concepts used by the decomposition model differ from basic notions of process modelling with eEPCs. First, a system is defined as a set of “things” by Weber (1997) with no equivalent counterpart existing for eEPC process models (cf. Green and Rosemann, 2001). We therefore interpret a self-contained business process with a clearly defined starting and ending point that is visualized as a corresponding eEPC model as a “system”. Second, we use data object types representing business relevant objects (e.g., data, application types, etc.) in eEPC models (cf. Scheer et al., 2005) as representatives of “things” in the sense of Weber (1997). Generally, data object types in an eEPC model are not a one-to-one equivalent for “things” of the BWW ontology but they are the most suitable construct for representing “things” in eEPC modelling. Third, a fundamental difference regarding the term “event” exists for the BWW ontology and eEPCs. Considering the BWW model, an event describes an ordered pair that comprises the initial state and the subsequent state arising for a “thing” due to a transformation (Weber, 1997). Contrary, in process modelling with eEPCs, an event represents an instance of an event type in an eEPC process model. In that context, an event indicates the current state of a process instance during execution (e.g., the order “is delivered” in an instance of the process “management of customer orders”) (Keller et al., 1992). However, as eEPCs consider the type level, only event types are explicated in a process model. Therefore, two essentially different conceptions are allotted the identical homonym “event”, a fact that needs to be considered when interpreting the decomposition conditions. In this regard, an “event” in the BWW model can be expressed with the help of the triple “event type → function type → event type” in eEPCs, whereas the function type represents a transformation (Green and Rosemann, 2000). Consequently, in accordance with the type level of process models, we refer to event types when specifying metrics for the decomposition conditions.

Minimality condition	“A decomposition is good only if for every subsystem at every level in the level structure of the system there are no redundant state variables describing the subsystem.” (Weber, 1997, p. 153)
Determinism condition	“For a given set of external (input) events at the system level, a decomposition is good only if for every subsystem at every level in the level structure of the system an event is either (a) an external event, or (b) a well-defined internal event.” (Weber, 1997, p. 154)
Losslessness condition	“A decomposition is good only if every hereditary state variable and every emergent state variable in a system is preserved in the decomposition.” (Weber, 1997, p. 155)
Minimum coupling condition	“A decomposition has minimum coupling iff the cardinality of the totality of input for each subsystem of the decomposition is less than or equal to the cardinality of the totality of input for each equivalent subsystem in the equivalent decomposition.” (Weber, 1997, p. 161)
Strong cohesion condition	“A set of outputs is maximally cohesive if all output variables affected by input variables are contained in the same set, and the addition of any other output to the set does not extend the set of inputs on which the existing outputs depend and there is no other output which depends on any of the input set defined by the existing output set.” (Dromey, 1996, p. 42; Weber, 1997, p. 163)

Table 1. The decomposition conditions

In the following, we shortly introduce a specification of these conditions for eEPCs (see Table 2). A more comprising and embracing description of the conditions based on a representational mapping (cf. Green and Rosemann, 2000) can be found in a previous work (cf. Johannsen and Leist, 2012).

Minimality: A process model can be decomposed into subprocess models. The arrangement of the subprocess models forms a “level structure” (Reijers and Mendling, 2011). Attributes of eEPC data object types express the “state variables” (Weber, 1997). Event types of an eEPC model show the states for these attributes. Event types in an eEPC model are “not redundant” in case they express a state for a particular attribute (state variable) that changes its value during process execution (Hoffmann et al., 1993).

Determinism: Since Weber (1997) demands “internal events” to be well-defined, a modeller should avoid using modelling constructs like OR connectors or an ambiguous labelling of the modelling constructs, which lead to changes in state that are not well-defined (cf. Mendling et al., 2010; van der Aalst et al., 2002). Further, considering “external events” – as event types – is an important task for depicting well-defined process models and fulfilling the determinism condition.

Losslessness: The eEPC does not offer representation mechanisms for emergent and hereditary properties (Green and Rosemann, 2000; Recker et al., 2009). Nevertheless, Weber (1997) generalizes the condition by demanding not to lose properties at all. Properties are represented by attributes of data object types in a data model and can be related to event types in eEPCs accordingly. Thus, all “non-redundant” event types that are required for visualizing a real world situation must be captured in an eEPC model.

Minimum coupling: In terms of "coupling", the coupling concept of Vanderfeesten et al. (2008) aligns very well with the primary ideas of Wand and Weber. Hence, two subprocess models are "coupled" if the output of a function type in a subprocess model – represented as a data object type – is at the same time input to a function type in another subprocess. Therefore, the interchange of data object types between subprocess models should be minimal. More, to minimize the *"total action of all environmental things on each subsystem in the decomposition"* (Weber, 1997, p. 159), the number of event types (e.g., start event types) used for expressing external events is to be minimized.

Strong cohesion: An interpretation of cohesion for process modelling was conducted by Vanderfeesten et al. (2008) which reverts to data object types for representing "output" (in the sense of Weber (1997)). Hence, all function types transforming input to output – expressed as data object types – are to be visualized within a subprocess model. Accordingly, data object types representing "input" in a subprocess model to produce particular output cannot be found in another subprocess model on that model level.

Minimality regarding eEPCs	The decomposed eEPC process model should not hold any event types that are "redundant" and thus indicate "states" that never occur during process execution. Event types used for representing states of state variables (attributes) on a type level that are not needed for the continuation of a process on an instance level are to be avoided. Function types and modelling constructs from other ARIS views (e.g., of the organizational view, data view, output view) related to these must be reflected as to their necessity regarding this fact as well.
Determinism regarding eEPCs	To fulfil the determinism condition, the decomposed business process model has no OR connectors, while subprocesses are built around external events. Rules for decision nodes have to be established and the event types have to be labelled appropriately.
Losslessness regarding eEPCs	No information must get lost during decomposition. Event types related to attributes describing properties are of central importance and should be preserved.
Minimum coupling regarding eEPCs	Each subprocess of a process must have less input data object types and external event types than in any other comparable decomposition of the same process.
Strong cohesion regarding eEPCs	All function types transforming a set of input to output (data object types) are captured within a subprocess. Each input within this subprocess cannot be found in any other subprocess at the same model level and produce other output.

Table 2. The decomposition conditions specified for eEPCs (Johannsen and Leist, 2012)

3 Metrics for the Decomposition Conditions

3.1 Procedure of the research

Our research follows the "Goal Question Metric (GQM) approach" for the systematic development of metrics by Basili et al. (1994). The GQM approach draws upon the idea that measurements in an entrepreneurial context require a thoroughly defined goal, which is then operationalized by relevant enterprise data, which are then interpreted regarding the goal (Basili et al., 1994). Especially for our research, the GQM approach is well qualified since it assures a very systematic research procedure in which reproducible results are achieved. We state our goal as the development of means to judge the quality of a decomposed process model. In that context, we refer to Wand and Weber's decomposition model as an approach for obtaining decompositions that are of high quality. In our study, the five decomposition conditions provide the questions of the GQM approach. We thus ask: To what degree does a decomposed process model adhere to the (I) minimality, (II) determinism, (III) losslessness, (IV) minimum coupling and (V) strong cohesion condition? To answer these questions, we derive metrics that help to evaluate the coherence of a decomposed eEPC process model in view of the decomposition conditions. To specify metrics, we follow a three-step approach that builds on the interpretation of the decomposition conditions for eEPC modelling (see Table 2). In a first step **(step 1)**, we identify the central constructs of the decomposition conditions, which have been specified for eEPCs. Afterwards **(step 2)**, we derive the corresponding variables for a metric. These variables are then arranged in the form of metrics that capture the initial idea of the decomposition conditions and allow to objectively judge to which extent a decomposed model adheres to these conditions **(step 3)**.

In the remainder of this chapter, we show the derivation of metrics for the "minimum coupling condition" for demonstration purposes. Then, considering the page restrictions, we briefly summarize the metrics for the other conditions, which, however, have been deduced in the same manner. To shorten the notation of the metrics, S_i refers to a subprocess model and |S| to the set of subprocess models. Likewise, M_j refers to a level of the decomposition and |M| to the set of levels in a decomposition.

3.2 Metrics to assess minimum coupling

Step 1: The specification of minimum coupling for eEPCs (see Table 2) puts a major emphasis on external event types and input data object types to determine the coupling degree. Because of that, a corresponding metric has to capture both these concepts. Coupling is given if a data object type is shared by a function type 1 in S_1 and a function type 2 in S_2 that are allocated to the same model level. The construct "(function type 1; function type 2)" is called a "coupled pair of function types" in the following. Thus, a "coupled pair of function types" consists of the function type producing the data object type as output and the function type receiving this data object type as input. Further, external event types are an important aspect to be considered for determining the environmental impact on a subprocess.

Step 2: The variables derived from these central constructs to develop corresponding metrics are "coupled pair of function types" and "external event type". For normalization purposes, the number of potential "pairs of coupled function types" is part of a corresponding metric, too. It expresses all possible constellations of coupling between function types. Further, the subprocess models (S_i) and model levels (M_j) of a level structure are required to focus either single model levels or the holistic decomposition. Take Figure 1 as an example, which shows the two subprocess models "complaint receipt – S_1" and "complaint handling – S_2". We assume that both these models are assigned to the same model level (M_j) of a decomposition. S_1 has two output data object types, namely the "confirmation of receipt" and the "complaint". The complaint is also an input data object type in S_2 for the function type "analyse complaint reason". Further, the output data object type "customer letter" is given. Accordingly, one "coupled pair of function types", namely *"(transmit complaint to employee responsible; analyse complaint reason)"*, exists. The variable "coupled pair of function types" takes into account these pairs of function types. Further, each subprocess model has an external event type, represented as start event types.

Step 3: In total, we propose four metrics for measuring the degree as to which a decomposition coheres to the minimum coupling condition. To reduce complexity, both concepts that determine the environmental impact on an eEPC subprocess model – namely the "coupling" between function types and the number of external event types – are dealt with by separate metrics. In this respect, it needs to be acknowledged that certain users may consider subprocess models on a specific level exclusively (e.g., operating staff focusing on M_0, developers focusing on M_1) to find the information sought after (cf. Bobrik et al., 2007). For instance, for deriving functional requirements on an IS, which are documented in the rough concept, a general perspective on a business process and its activities is taken, whereas in the fine concept a more detailed perspective on the tasks constituting an activity is aspired. In this respect, it is important that the requirements of the decomposition conditions are not only diligently followed regarding the holistic decomposition but also regarding selected model levels, a circumstance to be taken into account during the metric definition. The first metric (metric 1 – Table 3) addresses coupling between function types and focuses on a certain model level of a level structure only. To measure the degree of coupling, the "coupled pairs of function types" between subprocess models are to be counted for that model level. The result is divided by the total number of "potential pairs of coupled function types". Since modelling is a subjective task, modellers might not explicate all data object types in their models and, hence, all function types are used for calculating the potential pairs of coupled function types. To calculate the "potential pairs of coupled function types" consider that the "potential coupling" between function types can be directed in both directions. Reverting to Figure 1, we acknowledged one pair of coupled function types. This number is divided by the total number of potential coupled pairs of function types. Therefore, for each function type in a subprocess model, one builds a couple with each of the function types originating from different subprocess models on the model level under consideration. In Figure 1, S_1 and S_2 comprise three function types each. Thus, 2*(3*3) "potential pairs of coupled function types" regarding S_1 and S_2 can be built, amounting to a total of 18 "potential pairs of coupled function types". The value for metric 1 regarding Figure 1 is thus "0.056 (=1/18)".

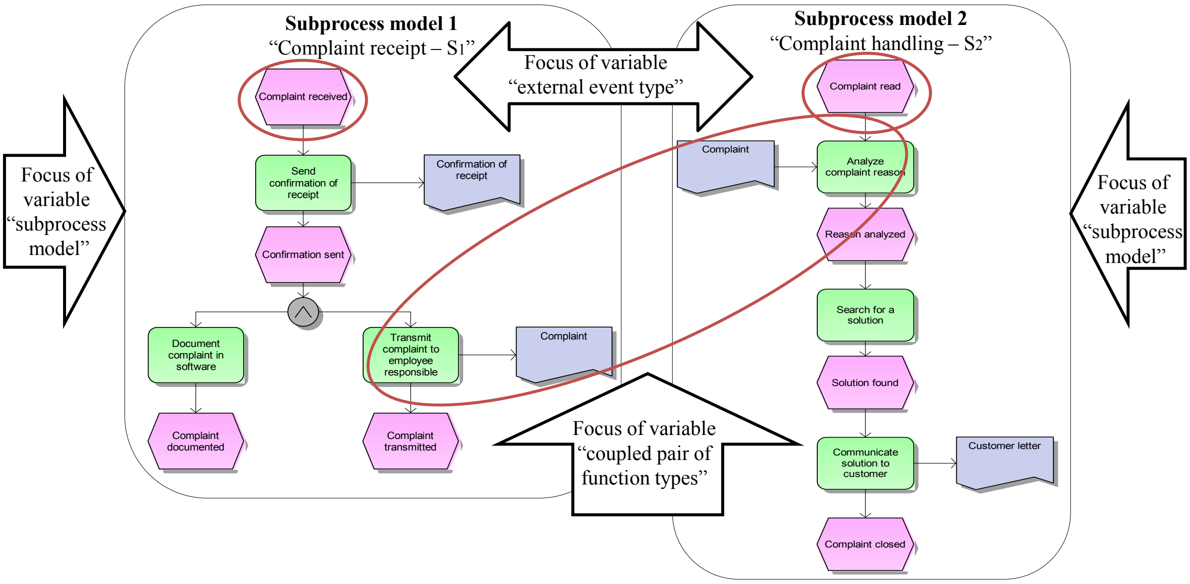
Subprocess model 1
"Complaint receipt – S1"
Complaint received
Send confirmation of receipt
Confirmation of receipt
Confirmation sent
Document complaint in software
Transmit complaint to employee responsible
Complaint
Complaint documented
Complaint transmitted
Focus of variable "subprocess model"
Focus of variable "external event type"
Subprocess model 2
"Complaint handling – S2"
Complaint read
Complaint
Analyze complaint reason
Reason analyzed
Search for a solution
Solution found
Communicate solution to customer
Customer letter
Complaint closed
Focus of variable "subprocess model"
Focus of variable "coupled pair of function types"

Figure 1. Example for the minimum coupling condition

All values calculated for the model levels of a decomposition via metric 1 can then be aggregated across all model levels to come to an overall value for the decomposed model as shown by metric 2 in Table 3. The other aspect of minimum coupling refers to event types indicating external events. According to Green and Rosemann (2000), start event types are typically used for representing external events in an eEPC model. Thus, first, the number of start event types should be minimal to reduce the total impact of the environment on a subprocess (cf. Weber, 1997).

Ratio of "coupled pairs of function types" across subprocess models on a model level – metric 1 *(minimum coupling)*
$Metric\ 1\ (M_j) = \frac{number\ of\ coupled\ pairs\ of\ function\ types\ across\ all\ subprocess\ models\ on\ a\ model\ level\ M_j}{number\ of\ potential\ pairs\ of\ coupled\ function\ types\ on\ a\ model\ level\ M_j}$
Calculation & Interpretation – metric 1
This metric counts the number of "coupled pairs of function types" across all subprocess models of a model level $\mathbf{M_j}$ and divides this number by the total number of "potential pairs of coupled function types" on that model level. The metric value represents the ratio of coupled pairs of function types in regards to all possible constellations of coupling between function types for a certain model level (M_j).
Average ratio of "coupled pairs of function types" for a model level – metric 2 *(minimum coupling)*
$Metric\ 2 = \frac{\sum_{j=0}^{\lvert M \rvert} \frac{numer\ of\ coupled\ pairs\ of\ function\ types\ across\ all\ subprocess\ models\ on\ a\ model\ level\ M_j}{number\ of\ potential\ pairs\ of\ coupled\ function\ types}}{\lvert M \rvert}$
Calculation & Interpretation – metric 2
Metric 2 aggregates the values received by applying metric 1 across all model levels $\mathbf{M_j}$ and divides the resulting sum by the total number of model levels **|M|**. The metric value shows the average ratio of coupled pairs of function types in regards to all possible constellations of coupling between function types for model levels across all model levels of a decomposition.
Average number of external event types for subprocess models on a model level – metric 3 *(minimum coupling)*
$Metric\ 3\ (M_j) = \frac{\sum_{i=1}^{\lvert S \rvert} number\ of\ external\ event\ types\ of\ subprocess\ model\ S_i}{\lvert S \rvert}$
Calculation & Interpretation – metric 3
This metric counts the number of external event types across all subprocesses $\mathbf{S_i}$ of a model level $\mathbf{M_j}$ and divides this number by the total number of subprocesses **|S|** on that model level. The metric answers the question of how many external event types subprocess models on a model level M_j have in average.
Average number of external event types for subprocess models of the decomposition – metric 4 *(minimum coupling)*
$Metric\ 4 = \frac{\sum_{j=0}^{\lvert M \rvert} \frac{\sum_{i=1}^{\lvert S \rvert} number\ of\ external\ event\ types\ of\ subprocess\ model\ S_i}{\lvert S \rvert}}{\lvert M \rvert}$
Calculation & Interpretation – metric 4
Metric 4 aggregates the values received by applying metric 3 across all model levels and divides the resulting sum by the total number of model levels **|M|**. The resulting value indicates the average number of external event types of subprocess models – in regards to a model level – across all model levels of the decomposition.

Table 3. Proposed metrics to assess minimum coupling

Further, the user needs to consider whether further external events impact a process and are considered as intermediate event types in the model (cf. Scheer et al., 2005). The total number of external event types is counted and divided by the number of subprocess models of that model level, which is done by metric 3 (Table 3). For example, there are two start event types in Figure 1 and no further external event types. This number is divided by the amount of subprocess models, which results in a metric value of "1". Therefore, in Figure 1, the external input is minimal. To obtain a value for the holistic decomposition, the values for metric 3 are aggregated for all model levels and then divided by the total number of all model levels existing, a procedure captured by metric 4 (Table 3). Both the metrics 3 and 4 result in values of "1" or higher, while "1" represents a perfect decomposition in that context. The normalization across subprocess models on a model level is performed for assuring the results to be interpretable and comparable (Heinrich et al., 2007), which has been posed as a central requirement on metrics in the IS domain (cf. Hinrichs, 2002). Hence, since particular users may focus on certain subprocess models only to retrieve the information sought after – which is a principle idea behind process model decomposition (e.g., Zugal et al., 2015) – an average value for subprocess models is strived for (see metric 3 or denominator of metric 4). As an example, when specifying the fine concept in the course of IS development, certain employees may focus on those subprocess models exclusively that visualize the working procedures performed by themselves.

In summary, minimum coupling is determined based on "coupled pairs of function types" and "external event types". For both, corresponding metrics were introduced. Since the aspect of coupled function types focuses on structural aspects of a decomposition, no domain knowledge is required for calculating metrics 1 and 2. This enables an automatized assessment of the coherence of a model to the minimum coupling definition. However, domain knowledge is required for determining external event types. To reduce the coupling degree – and thus the number of coupled pairs of function types as well as external event types –, a modeller can merge subprocess models. However, completely reducing the interaction between subprocess models may not be appropriate under all circumstances as, e.g., the modeller might come up with one, merged, process model only, which counteracts the intention of decomposition.

3.3 Mimimality, determinism, losslessness and strong cohesion

The following Tables 4 and 5 summarize the metrics proposed for the remaining conditions. The metrics have been derived following the procedure shown in section 3.1. In summary, we propose two metrics (5 and 6) to determine "minimality". Both metrics direct the attention to redundant event types that need to be identified in a model. A redundant event type refers to an attribute that never changes its value during process execution and, hence, such an event type is never reached (cf. Scheer et al., 2005; Weber, 1997). The ideal value for both metrics is "0". That way, the initial idea of Weber (1997) to avoid unnecessary state variables is perfectly preserved, even though the challenge of applying the metrics lies in finding redundant event types of an eEPC model. The first two metrics developed to assess "determinism" (metrics 7 and 8) focus on the requirement that internal events of a decomposition need to be well-defined (cf. Weber, 1997). Thus, OR split operations in the subprocess models are in the centre of attention. These can be identified quite simply without requiring domain knowledge of the real world situation modelled. The consideration of external events can be challenging (cf. Weber, 1997), which requires the user to have profound knowledge of a process and its environment to apply metric 9. An external event type represents a state that occurs due to actions of environmental components, e.g., customers or partners (Weber, 1997). All metrics have a value of "0" in an ideal decomposition.

For assessing "losslessness" via metric 10, domain knowledge is required for identifying "missing non-redundant event types". Accordingly, the calculation of the metric cannot be automatized as the semantics of the process model must be reflected against the real world. However, the manual calculation of the metric's value provides valuable insights as to what degree a process model coheres to Wand and Weber's idea of losslessness. Again, ideally, the application of the metric results in a value of "0".

Finally, we propose two metrics for determining "strong cohesion" (metrics 11 and 12 – Table 5). Both metrics focus on structural aspects of a decomposition and analyse the delineation of subprocess models more closely. The semantics of the process model is not investigated. Thus, the calculation of the metric can be automatized, as no domain knowledge is required. Similarly to minimum coupling, the calculation requires to count the data object types across the subprocess models. The metric values are “0” in the case of a perfect decomposition in regards to “strong cohesion”.

Average ratio of “redundant” event types for subprocess models on a specific model level – metric 5 *(minimality)*
$Metric\ 5\ (M_j) = \frac{\sum_{i=1}^{\lvert S\rvert} \frac{number\ of\ redundant\ event\ types\ in\ subprocess\ model\ S_i}{number\ of\ event\ types\ in\ subprocess\ model\ S_i}}{number\ of\ subprocess\ models\ \lvert S\rvert\ on\ model\ level\ M_j\ considered}$
Calculation & Interpretation – metric 5
This metric counts the “redundant event types” of a subprocess model **S**$\mathbf{S_i}$ on a model level $\mathbf{M_j}$. This number is divided by the total number of event types of that subprocess model $\mathbf{S_i}$. This is done for all subprocess models of the model level $\mathbf{M_j}$, and the partial results are aggregated. The resulting number is divided by the total number of subprocess models **|S|** on that model level to obtain an average ratio for the subprocess models and thus to increase comparability. Hence, two count variables **i** and **j** are used with **i** addressing the subprocess models (for example S_1, S_2, etc.) and **j** addressing the model levels (for example M_0, M_1, etc.). The final result represents the average ratio of redundant event types for subprocess models on a model level M_j.
Average ratio of “redundant” event types for subprocess models across all model levels of a decomposition – metric 6 *(minimality)*
$Metric\ 6 = \frac{\sum_{j=0}^{\lvert M\rvert} \frac{\sum_{i=1}^{\lvert S\rvert} \frac{number\ of\ redundant\ event\ types\ in\ subprocess\ model\ S_i}{number\ of\ event\ types\ in\ subprocess\ model\ S_i}}{number\ of\ subprocess\ models\ \lvert S\rvert\ on\ model\ level\ M_j}}{\lvert M\rvert}$
Calculation & Interpretation – metric 6
In summary, the value for metric 6 is received by applying metric 5 to all model levels of a decomposition (e.g. M_0, M_1, etc.). The resulting values are summed up and the result is divided by the total number of model levels **|M|** of a decomposed process model. The value shows the average ratio of redundant event types (regarding all event types modelled) for subprocess models of a model level across a decomposition.
Average ratio of OR splits of subprocess models on a specific model level – metric 7 *(determinism)*
$Metric\ 7\ (M_j) = \frac{\sum_{i=1}^{\lvert S\rvert} \frac{number\ of\ OR\ splits\ in\ subprocess\ model\ S_i}{number\ of\ split\ operations\ in\ subprocess\ model\ S_i}}{number\ of\ subprocess\ models\ \lvert S\rvert\ on\ the\ model\ level\ M_j\ considered}$
Calculation & Interpretation – metric 7
This metric counts the number of OR splits of a subprocess model $\mathbf{S_i}$ on a model level $\mathbf{M_j}$. This number is divided by the total number of split operations (XOR, OR, AND) of that subprocess model. This is done for all subprocess models of that model level and the partial results are aggregated. The result is divided by the number of subprocess models on that level to achieve an average ratio. The value represents the average ratio of OR splits in regards to all split operations for subprocess models on a model level M_j.
Average ratio of OR splits of subprocess models across all model levels of a decomposition – metric 8 *(determinism)*
$Metric\ 8 = \frac{\sum_{j=0}^{\lvert M\rvert} \frac{\sum_{i=1}^{\lvert S\rvert} \frac{number\ of\ OR\ splits\ in\ subprocess\ model\ S_i}{number\ of\ split\ operations\ in\ subprocess\ model\ S_i}}{number\ of\ subprocess\ models\ \lvert S\rvert\ on\ the\ model\ level\ M_j\ considered}}{\lvert M\rvert}$
Calculation & Interpretation – metric 8
The average ratio of OR splits of the subprocess models on a specific model level is calculated for all model levels and the values are summed up. The result is divided by the total number of model levels of a decomposed process model. The metric value represents the average ratio of OR split operations in regards to all split operations for subprocess models of a model level across all model levels of a decomposition.
Ratio of missing “external event types” of a decomposition – metric 9 *(determinism)*
$Metric\ 9 = \frac{\sum_{j=0}^{\lvert M\rvert} \sum_{i=1}^{\lvert S\rvert} \frac{number\ of\ missing\ external\ event\ types\ in\ subprocess\ model\ S_i}{number\ of\ external\ event\ types\ modeled\ +number\ of\ missing\ external\ event\ types\ in\ subprocess\ model\ S_i}}{total\ number\ of\ subprocess\ models\ of\ a\ decomposition\ across\ all\ model\ levels}$
Calculation & Interpretation – metric 9
Metric 9 counts the number of “missing external event types” for a subprocess model and divides this number by the sum of “missing external event types” and explicitly modelled “external event types”. Missing event types are those that are “not redundant”, however, have not been considered by the modeller. The values are summed up for all subprocess models across the decomposition and the result is divided by the total number of subprocess models that can be found in the decomposition. The metric value represents the average number of missing external event types in regards to all external events that should have been captured by a decomposed process model. Note that the average value considers all subprocess models across all model levels without differentiating between model levels.

Table 4. Proposed metrics to assess minimality and determinism

Following the GQM approach, twelve metrics were defined for measuring the coherence of a decomposed business process model with the decomposition conditions. Most decomposition conditions allow to consider a decomposed model as a whole (metrics 2, 4, 6, 8, 9, 10, 12), facilitating the comparison of alternative decompositions, but also enabling the analysis of model levels separately (metrics 1, 3, 5, 7, 11). Further examples for the metrics’ application can be found at: *https://tinyurl.com/ybyjp2ky.*

Ratio of missing “non-redundant event types” of a decomposition – metric 10 *(losslessness)*
$Metric\ 10 = \frac{\sum_{j=0}^{\lvert M\rvert} \sum_{i=1}^{\lvert S\rvert} \frac{number\ of\ missing\ non\ redundant\ event\ types\ in\ subprocess\ model\ S_i}{number\ of\ event\ types\ in\ subprocess\ model\ S_i + number\ of\ missing\ non\ redundant\ event\ types\ in\ subprocess\ model\ S_i}}{total\ number\ of\ subprocess\ models\ of\ a\ decomposition\ across\ all\ model\ levels}$

Calculation & Interpretation – metric 10
This metric counts the number of "missing non-redundant event types" for a subprocess model and divides this number by the sum of missing "non-redundant event types" and explicitly modelled event types. This is done for all subprocess models of a decomposition. The values are summed up and the result is divided by the total number of subprocess models that can be found in the decomposition (across all model levels). The value shows the average ratio of "missing non-redundant event types" to the sum of "missing non-redundant event types" plus all event types explicitly modelled for subprocess models of a decomposition. Note that the average value considers all subprocess models across all model levels without differentiating between model levels.
Ratio of "duos of function types" sharing a common input data object type across subprocess models on a model level – metric 11 *(str. coh.)*
$Metric\ 11\ (M_j) = \frac{number\ of\ duos\ of\ function\ types\ sharing\ a\ common\ input\ data\ object\ type\ but\ have\ different\ output\ data\ object\ types\ across\ all\ subprocess\ models\ on\ a\ model\ level\ M_j}{number\ of\ potential\ duos\ of\ function\ types\ on\ a\ model\ level\ M_j}$
Calculation & Interpretation – metric 11
This metric counts the number of duos of function types that share a common data object type as input (so called "duos of function types") but have different output object types across all subprocess models **S_i** of a model level **M_j** and divides this number by the total number of "potential" duos of function types on that model level. The metric shows the ratio of function types that have different output object types but share a common data input type in regards to all possible constellations of function types on a specific model level M_j.
Average ratio of "duos of function types" sharing a common input data object type for a model level – metric 12 *(strong cohesion)*
$Metric\ 12 = \frac{\sum_{j=0}^{\lvert M\rvert} \frac{number\ of\ duos\ of\ function\ types\ sharing\ a\ common\ input\ data\ object\ type\ but\ have\ different\ output\ data\ object\ types\ across\ all\ subprocess\ models\ on\ a\ model\ level\ M_j}{number\ of\ potential\ duos\ of\ function\ types\ on\ a\ model\ level\ M_j}}{\lvert M\rvert}$
Calculation & Interpretation – metric 12
The ratio of the "duos of function types" across subprocess models on a model level is calculated for all model levels, the partial results are aggregated and the result is divided by the total number of model levels. The value stands for the average ratio of function types that have different output data object types but share a common data input type in regards to all possible constellations of function types (taking the model level as a calculation base) across all model levels.

Table 5. Proposed metrics to assess losslessness and strong cohesion

4 Application of the Metrics & Prototypical Implementation

Prior to specifying the metrics, we evaluated the decomposition conditions for eEPCs in an experimental setting to judge their impact on process model understandability (cf. Johannsen et al., 2014). The results showed that decomposed process models are more understandable if the decomposition abides the conditions. Further, abiding the conditions strongly increased the perceived ease of understanding. An overview of the material and the decomposed process model with detailed explanations is available at: *https://tinyurl.com/ybpj3hv5*. In that context, three alternative decompositions "A", "B", and "C" of a process model depicting the "student enrolment process" at a German university, which violated the conditions to varying degrees, were created. Each decomposition comprised four model levels (M_0 to M_3). Alternative "A" complied with the conditions as far as possible. Alternative "B" violated the minimality and losslessness conditions, which focus the semantics of a process and less its structure. Alternative C violated all the decomposition conditions to the same degree. In a previously conducted experiment, we found that models complying with the decomposition conditions are perceived as significantly easier to understand by users (cf. Johannsen et al., 2014). For this work, we reuse the process models and apply the newly developed metrics to see whether they indicate a difference as well (see Table 6). Almost all of the metrics applied to alternative "A" result in ideal values for these particular measures. However, some results emerge from applying the metrics that seem counterintuitive at first sight. They are highlighted by the colorations in Table 6 and further explained in the following. First, the results for metrics 3 and 4 need explaining. Both metrics focus external event types of eEPC subprocess models. To minimize the external influence on a system as demanded by Weber (1997), each subprocess model should – ideally – only have one start event type initially. However, the subprocess models in alternative "A" have more than one start event type on average. That particular circumstance is also reflected by the values for the metrics 3 and 4, which are quite similar for the model alternatives "A", "B" and "C". Essentially, it needs to be acknowledged that student enrolment is a complex process, thus requiring the modelling of numerous start event types for the subprocess models. Second, the results for metrics 7 and 8 (determinism) need clarifying. Regarding the application of metric 7, variant "B" performs worse than "C" for model levels M_1 and M_3, although alternative "B" is perceived as easier to understand by model users. Thereby, both alternatives, "B" and "C", have only one OR split operation on M_1. However, the delineation of subprocess models is different and, thus, alternative "B" comprises two subprocess models on M_1 whereas there are four subprocess models in alternative "C". Accordingly, the model user focusing on particular subprocess models exclusively to find certain information will less likely come across the subprocess model with the OR connector in alternative “C” compared to alternative “B”. This circumstance also affects metric 8, aggregating the values for metric 7 across all model levels.

Despite these peculiarities, the results of the calculation match with the assumption that decompositions with fewer violations of the conditions are not only easier to understand but also perform better regarding the metrics’ values than equivalent decompositions violating the conditions. Hence, the metrics provide a good indicator as to what degree a decomposition adheres to the decomposition conditions as defined allowing to systematically assess its perceived quality. However, it is rather unlikely to perfectly adhere to each condition in practice, e.g., due to the complexity of the real world.

Condition	Minimum Coupling				Minimality		Determinism			Losslessness	Strong cohesion	
Metric	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Alternative A												
Level 0 (M_0)	0	0	3	2.312	0	0	0	0	0	0	0	0
Level 1 (M_1)	0		2		0		0				0	
Level 2 (M_2)	0		1.25		0		0				0	
Level 3 (M_3)	0		3		0		0				0	
Alternative B												
Level 0 (M_0)	0	0	3	2.063	0	0.049	0	0.148	0	0	0	0
Level 1 (M_1)	0		2		0.118		0.083				0	
Level 2 (M_2)	0		1.25		0.079		0.177				0	
Level 3 (M_3)	0		2		0		0.333				0	
Alternative C												
Level 0 (M_0)	0	0.037	3	2.188	0	0.063	0	0.075	0	0	0	0
Level 1 (M_1)	0.012		2.25		0.143		0.063				0	
Level 2 (M_2)	0.011		1		0.107		0.236				0	
Level 3 (M_3)	0.125		2.5		0		0.25				0	

Table 6. Calculation results

Many of the metrics presented here require a laborious calculation, which is time-consuming considering the various model levels and subprocess models that may be created. In most practical settings, however, process models are available electronically in tools such as ARIS or MS Visio. Thus, we used the formalization of the metrics to implement an automatic calculation to be used by modellers evaluating a large number of models. The prototype also serves as a proof-of-concept for the metrics (cf. Hevner et al., 2004). For the metrics 1, 2, 7, 8, 11 and 12, the implementation was straightforward. First, we mapped the variables from the metrics to eEPC modelling elements. Then, we implemented the calculation procedure as algorithms. For the implementation, we used the ProM process mining and analysis framework *(http://bit.ly/2pQYNwc)*. This framework is well known for its analytical capabilities and easy extensibility. The implementation expects the models to be available in the EPML-notation (cf. Mendling and Nüttgens, 2006), which is an open standard supported by many frequently used modelling tools. The source code of our implementation is available at *http://bit.ly/2pQNVi3*. Currently, there is no implementation for a fully automatized calculation of the metrics 3, 4, 5, 6, 9 and 10 as process knowledge is required for that purpose. As of now, redundant event types, missing external event types, etc. have to be identified by the user via their process knowledge or the advice of experts.

5 Discussion

5.1 Reflection

In this research, the decomposition conditions of Wand and Weber have been transformed into metrics to evaluate the quality of decomposed eEPC models. The metrics for the minimality condition focus on the event types of eEPC models, which are used as representations for state variables. This interpretation is based on the fact that the BWW ontology explicitly focuses the instance level, whereas the eEPC works on the type level. Thus, for unambiguously determining redundant event types, a modeller needs to reflect on the instances of an eEPC model. Depending on the size of the process model, this may require considerable cognitive efforts. However, the instances of a process model clearly show which event types are redundant and can thus be deleted on the type level. The metrics for the determinism condition suggest to avoid “OR splits” in a process model to ensure that internal events are well-defined (cf. Weber, 1997) on the one hand. Taking a structural perspective on process models helps to avoid uncertainty when instantiating an eEPC model. However, in future research, further semantics-based metrics are to be developed that consider the labelling of the nodes in a process model as well, because inadequate labels might lead to additional ambiguities counteracting the ideas of Weber (1997). On the other hand, metric 9 deals with external events as mentioned. As previously said, event types in an eEPC model cannot be mapped to events of the BWW ontology in a one-to-one manner. Thus, process knowledge is required from the user side to determine whether an eEPC model actually considers all relevant event types that point to external events (cf. Weber, 1997).

The losslessness condition focuses on hereditary and emergent state variables in special, a differentiation which does not become obvious in an eEPC model due to ontological deficiencies of the modelling notation (cf. Recker et al., 2009). Nevertheless, this circumstance is negligible because our metric builds on the requirement to preserve all types of properties during decomposition (cf. Weber, 1997). Hence, the primary idea of the decomposition condition was enhanced for eEPC modelling. To operationalize the input to or the environmental influence on a subsystem, concerning the minimum coupling condition, our metrics focus data object types and external event types of subprocess models. Considering the ontological expressiveness of eEPCs (cf. Green and Rosemann, 2000), these modelling constructs are most appropriate to define coupling for subprocess models from a structural perspective. In terms of the metrics for the strong cohesion condition, data object types of a decomposed eEPC model characterize the dependence of a set of output state variables on the corresponding input state variables. This approach builds on a particular operationalization for cohesion in process modelling as introduced by Vanderfeesten et al. (2008) and was adapted for this research. Actually, the interpretation of cohesion on the base of data object types is most appropriate to capture the condition's initial purpose.

5.2 Benefits and Restrictions

Judging the quality of a decomposition by referring to the metrics as introduced brings about some restrictions: first, because the values of the metrics are not standardized, and considering the complexity of entrepreneurial working procedures documented as process models in practice, the results can currently only be thoroughly interpreted in case equivalent decompositions are compared to each other. Additionally, a general proposition as to whether some conditions are more important than others cannot be done. Generally, the aggregation of all metrics to an overall value across all decomposition conditions remains an open issue. More, the application of some of the metrics requires, to a certain degree, users' process knowledge. This is because the metrics do consider more than only the structural aspects of a model (e.g., metrics 3, 4 or 10). In addition, it is hard to determine how much effort (time or resources) would be necessary to improve a metric's value of "0.148" for metric 8 to a value of "0.005" for example (see Table 6). No practical experiences exist on that yet. Furthermore, by merging subprocess models, the values for the metrics 1 and 2 regarding "minimum coupling" can be optimized. In an extreme case, this might result in a single process model only, which, however, is counterproductive to the idea of decomposition. Considering this, a decomposed model should be assessed on base of all conditions and metrics accordingly.

Besides these restrictions, the application of the metrics gives valuable insights into the quality of a decomposition, and is beneficial for the following reasons: **first**, the metrics capture properties of a well-performed decomposition by reverting to Wand and Weber's decomposition conditions and represent a manageable approach for evaluating decomposed eEPC models. In this regard, the metrics for the minimum coupling condition, the strong cohesion condition, and the determinism condition (for dealing with internal events) provide advice on how to assess a decomposition based on its structure and the design of the subprocess models without requiring process knowledge or a deeper analysis of the underlying semantics. However, contrary to approaches that focus the structure of a process model, the decomposition model also takes into account the semantics, e.g., by the minimality or losslessness condition. Further, we propose ideal values for the metrics. The "distance" of the values received from applying the metrics to the "ideal values" as proposed enable the initial assessment of the coherence of a decomposed process model with the decomposition conditions. **Second**, the calculation can partly be automatized by the implementation of the metrics as a tool, which considerably speeds up the quality assessment procedure, reduces cognitive efforts and the likelihood of errors of a manual calculation. Corresponding means to assess the quality of decompositions are missing yet. So, the user is supported in choosing an alternative from equivalent decompositions. **Third**, we introduce metrics for both the model level and the holistic decomposition, which is decisive because excerpts from a large process model may be relevant for particular users only. For instance, employees may focus on those subprocess models for specifying functional requirements during IS development that visualize the working procedures they are involved in. Therefore, particular users will search for information on certain model levels only, which, in turn, should meet the quality requirements as stipulated by the decomposition conditions just as the decomposition as a whole should do. Because of that, the metrics do not only allow to judge the decomposition as a whole but also particular model levels, which substantiates their practical applicability. Summing up, despite the mentioned restrictions, the metrics are beneficial means to assess the quality of a decomposition objectively, supporting the delineation of subprocess models.

6 Conclusion and Outlook

Process modelling is a decisive task for today’s companies considering business transformation initiatives in terms of digitalization. Decomposition is seen as an effective means for raising the understandability of a process model (Reijers et al., 2011; Mendling et al., 2010). Still, the properties that characterize a “good” decomposition remain an open question. In our work, we develop metrics to judge the coherence of a decomposed eEPC process model with the decomposition conditions of Wand and Weber.

Our work is beneficial for research and practice alike: our set of metrics, based on the decomposition conditions of Wand and Weber, contributes to the academic discussion on how to decompose process models purposefully and on properties that characterize a well-performed decomposition (cf. Milani et al., 2016). They provide a clear definition of the concept of decomposition in the light of Wand and Weber’s decomposition conditions and emphasize the data view (external events, data object types) for the delineation of subprocess models. In sum, our research constitutes elementary groundwork for the discussion of decomposition in process models as well as for the fuzzy concept of process model quality and understandability in general. Corresponding means to evaluate decomposed process models are not found in literature yet. Thus, we provide practitioners with measures to judge the quality of decomposed models. For this, we implemented a prototype, which partly automatizes the calculation. This reduces the effort required to assess the quality of a decomposition drastically. In addition, this eliminates the likelihood of calculation errors. The clear advice the metrics provide regarding determinism, minimum coupling and strong cohesion can support a modeller in properly delineating subprocesses and choosing one of several equivalent alternatives of a decomposed model.

A limitation of our research is that the consolidation of the single results of each metric to form a holistic view on a decomposition – across all conditions – still remains an open issue. The main reason for this are the ontological differences between Wand and Weber’s decomposition conditions and the eEPC that, in consequence, restrain a one-to-one mapping of the decomposition conditions for eEPCs. As a further consequence, it is not possible for the metrics to capture completely the conditions’ initial purpose of Wand and Weber. Further, a fully automatized calculation is not possible, as some metrics need domain knowledge. To receive a manageable set of metrics, not all the ideas of the original decomposition conditions can be mapped to modelling with eEPCs. Moreover, the practical experience of applying the metrics is limited to the mentioned modelling project with the university administration so far. Thus, we will realize further applications in future.

In a future work, we will develop metrics for other modelling languages, e.g., BPMN, as well. In addition, our metrics will be applied in further case studies for evaluation purposes. More, we will conduct empirical studies with modellers to obtain feedback on the metrics’ usability.

References

Basili, V., Caldiera, G. and Rombach, H. D. (1994). *The Goal Question Metric Approach,* In Encyclopedia of Software Engineering, John Wiley & Sons, pp. 528–532.

Becker, J., Thome, I., Weiß, B. and Winkelmann, A. (2010). "Constructing a Semantic Business Process Modelling Language for the Banking Sector." *Enterprise Modelling and Information Systems Architectures* 5 (1), 4–25.

Bobrik, R., Reichert, M. and Bauer, T. (2007). "View-Based Process Visualization." *Business Process Management* Vol. 4714, 88–95.

Burton-Jones, A. and Meso, P. (2006). "Conceptualizing Systems for Understanding: An Empirical Test of Decomposition Principles in Object-Oriented Analysis." *Information Systems Research* 17 (1), 38–60.

Burton-Jones, A. and Meso, P. N. (2008). "The Effects of Decomposition Quality and Multiple Forms of Information on Novices' Understanding of a Domain from a Conceptual Model." *Journal of the Association for Information Systems* 9 (12), 748–802.

Davis, R. and Brabänder, E. (2007). *ARIS Design Platform: Getting Started with BPM*, London: Springer.

De Lara, J., Guerra, E. and Sánchez Cuadrado, J. (2013). "Reusable abstractions for modeling languages." *Information Systems* 38 (8), 1128–1149.

Dromey, G. (1996). "Cornering the chimera." *IEEE Software* January 1996, 33–43.

Fettke, P., Houy, C., Vella, A.-L. and Loos, P. (2012). "Towards the Reconstruction and Evaluation of Conceptual Model Quality Discourses – Methodical Framework and Application in the Context of Model Understandability." In: *Enterprise, Business-Process and Information Systems Modeling.* Ed. by Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P. and Wrycza, Berlin Heidelberg: Springer, pp. 406–421.

Fox, M. S. and Gruninger, M. (1998). "Enterprise modeling." *AI magazine* 19 (3), 109–121.

Gerth, C. (2013) *Business Process Models. Change Management.* Berlin/Heidelberg: Springer.

Green, P. and Rosemann, M. (2000). "Integrated process modeling: An ontological evaluation." *Information Systems* 25 (2), 73–87.

Green, P. and Rosemann, M. (2001). "Ontological Analysis of Integrated Process Models: testing hypotheses." *Australasian Journal of Information Systems* 9 (1), 30–38.

Harmon, P. (2016). *The State of Business Process Management – 2016*. URL: https://www.bptrends.com/bpt/wp-content/uploads/2015-BPT-Survey-Report.pdf (visited on 11/21/2017).

Harmon, P. and Wolf, C. (2011). *Business Process Modeling Survey.* URL: http://www.bptrends.com/reports/Process_Modeling_Survey-Dec_11_FINAL.pdf (visited on 11/21/2017).

Heinrich, B., Kaiser, M. and Klier, M. (2007). "Metrics for measuring data quality-foundations for an economic oriented management of data quality." In: *2nd International Conference on Software and Data Technologies (ICSOFT).* Barcelona.

Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004). "Design Science in Information Systems Research." *MIS Quarterly* 28 (1), 75–105.

Hinrichs, H. (2002). "Datenqualitätsmanagement in Data Warehouse-Systemen." Universität Oldenburg.

Hoffmann, W., Wein, R. and Scheer, A.-W. (1993). *Konzeption eines Steuerungsmodells für Informationssysteme – Basis für die Real-Time Erweiterung der EPK (rEPK).* Report No. 106. Universität Saarbrücken.

Johannsen, F. and Leist, S. (2012). "Wand and Weber's Decomposition Model in the Context of Business Process Modeling." *Business & Information Systems Engineering* 54 (5), 263–280.

Johannsen, F., Leist, S. and Braunnagel, D. (2014). *Testing the Impact of Wand and Weber's Decomposition Model on Process Model Understandability*. 35th International Conference on Information Systems (ICIS), Auckland (NZ).

Keller, G., Nüttgens, M. and Scheer, A.-W. (1992). *Semantische Prozeßmodellierung auf der Grundlage „Ereignisgesteuerter Prozeßketten (EPK)“*. Heft 89. Institut für Wirtschaftsinformatik. Universität Saarbrücken.

Ma, Q., Kelsen, P. and Glodt, C. (2015). “A generic model decomposition technique and its application to the Eclipse modeling framework.” *Software & Systems Modeling* 14(2), 921–952.

Mendling, J. (2008). *Metrics for Process Models – Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness.* Berlin et al.: Springer.

Mendling, J. and Nüttgens, M. (2006). “EPC markup language (EPML): an XML-based interchange format for event-driven process chains (EPC).” *Information Systems and E-Business Management* 4 (3), 245–263.

Mendling, J., Reijers, H. and van der Aalst, W. (2010). “Seven process modeling guidelines.” *Information and Software Technology* 52 (2), 127–136.

Mendling, J., Reijers, H. A. and Cardoso, J. (2007). “What Makes Process Models Understandable?” *Lecture Notes in Computer Science* Vol. 4714/2007, 48–63.

Mendling, J., Strembeck, M. and Recker, J. (2012). “Factors of process model comprehension - Findings from a series of experiments.” *Decision Support Systems* 53 (1), 195–206.

Milani, F., Dumas, M., Matulevičius, R., Ahmed, N. and Kasela, S. (2016). “Criteria and Heuristics for Business Process Model Decomposition.” *Business & Information Systems Engineering* 58 (1), 7–17.

Overhage, S., Birkmeier, D. Q. and Schlauderer, S. (2012). “Quality Marks, Metrics, and Measurement Procedures for Business Process Models: The 3QM-Framework.” *Business & Information Systems Engineering* 4 (5), 229–246.

Pinggera, J., Soffer, P., Fahland, D., Weidlich, M., Zugal, S., Weber, B., Reijers, H. A. and Mendling, J. (2015). “Styles in business process modeling: an exploration and a model.” *Software & Systems Modeling* 14 (3), 1055–1080.

Recker, J. (2011). *Evaluations of Process Modeling Grammars – Ontological, Qualitative and Quantitative Analyses Using the Example of BPMN,* Berlin et al.: Springer.

Recker, J., Indulska, M., Rosemann, M. and Green, P. (2005) *Do Process Modelling Techniques Get Better? A Comparative Ontological Analysis of BPMN*, 16th Australasian Conference on Information Systems, Sydney.

Recker, J. C., Rosemann, M., Indulska, M. and Green, P. (2009). “Business process modeling: a comparative analysis.” *Journal of the Association for Information Systems* 10 (4), 333–363.

Reijers, H. A. and Mendling, J. (2011). “A Study Into the Factors That Influence the Understandability of Business Process Models.” *Man, and Cybernetics – Part A* 41 (3), 449–462.

Reijers, H. A., Mendling, J. and Dijkman, R. M. (2011). “Human and automatic modularizations of process models to enhance their comprehension.” *Information Systems* 36 (5), 881–897.

Rosemann, M. and Green, P. (2002). “Developing a meta model for the Bunge–Wand–Weber ontological constructs.” *Information Systems* 27 (2), 75–91.

Scheer, A.-W., Thomas, O. and Adam, O. (2005). “Process Modeling Using Event-Driven Process Chains.” In: *Process-aware information systems*. Ed. by Dumas, M., van der Aalst, W. and Hofstede, A. T. John Wiley and Sons, pp. 119–146.

van der Aalst, W. M. (2013). “Decomposing Petri nets for process mining: A generic approach.” *Distributed and Parallel Databases* 31 (4), 471–507.

van der Aalst, W. M. P., Desel, J. and Kindler, E. (2002). “On the semantics of EPCs: A vicious circle.” In: *EPK 2002: Business Process Management using EPCs*. Ed. by Nüttgens, M. and Rump, F. J., Trier, pp. 71–80.

Vanderfeesten, I. T. P., Cardoso, J., Mendling, J., Reijers, H. A. and van der Aalst, W. M. P. (2007) *Quality Metrics for Business Process Models,* In: *BPM and Workflow Handbook 2007*. Ed. by Fischer, L. Future Strategies, pp. 179–190.

Vanderfeesten, I., Reijers, H. A. and van der Aalst, W. M. P. (2008). “Evaluating workflow process designs using cohesion and coupling metrics.” *Computers in Industry* 59 (5), 420–437.

Vanhatalo, J., Völzer, H. and Koehler, J. (2009) “The refined process structure tree.” *Data & Knowledge Engineering* 68 (9), 793–818.

Wand, Y. and Weber, R. (1989). "A Model of Systems Decomposition." In: *10th International Conference on Information Systems*. Ed. by DeGross, J. I., Henderson, J. C. and Konsynski, B. R., Boston, pp. 42–51.

Weber, R. (1997). *Ontological Foundations of Information Systems.* Queensland: Coopers & Lybrand.

Zugal, S., Pinggera, J. and Weber, B. (2011). "Assessing Process Models with Cognitive Psychology." In: *EMISA 2011*, pp. 177–182.

Zugal, S., Soffer, P., Haisjackl, C., Pinggera, J., Reichert, M. and Weber, B. (2015). "Investigating expressiveness and understandability of hierarchy in declarative business process models." *Software & Systems Modeling* 14 (3), 1081–1103.