Universität Regensburg

Metriken zur Bewertung und Verbesserung von Prozessmodellen

Dissertation

zur Erlangung des Grades eines
Doktors der Wirtschaftswissenschaften (Dr. rer. pol.)

eingereicht an der
Fakultät für Wirtschaftswissenschaften
der Universität Regensburg

vorgelegt von
M.Sc. Wirt.-Inf. Daniel Braunnagel

Berichterstatter:
Prof. Dr. Susanne Leist
Prof. Dr. Bernd Heinrich

Tag der Disputation:
28.08.17

Regensburg, 17. Februar 2017

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ARIS	Architektur integrierter Informationssysteme
BPM	Business Process Management
BWW	Bunge Wand Weber
CFC	Control Flow Complexity
CLT	Cognitive load theory
CM	Coupling of a Module
CoC	Conceptual Coupling
CrC	Cross Connectivity
CBO	Coupling Between Objects
DC	Direct Coupling
DS	Design Science
DSR	Design Science Research
eEPC	extended event-driven process chain
eEPK	Erweiterte ereignisgesteuerte Prozesskette
FF	Forschungsfrage
GDC	Good Decomposition Conditions
GOM	Grundsätze ordnungsmäßiger Modellierung
GPM	Geschäftsprozessmodell
GQM	Goal Question Metric
IC	Indirect Coupling
ICM	Intramodule Coupling
IS	Informationssystem / Information System
ISO	International organization for standardization
PC	Process Coupling

PPM	Process performance Management
RFC	Response for a Class
SESE	Single entry, single exit
SME	Small- and medium enterprise
TC	Total Coupling
WC	Weighted Coupling
ZHAW	Züricher Hochschule für angewandte Wissenschaften

1 Einleitung

1.1 Motivation und Problemstellung

“A model […] allows us to deal with the world in a simplified manner, avoiding the complexity, danger and irreversibility of reality” (Rothenberg 1989, S. 75)

Nach (Schütte & Rotthowe 1998, S. 59) lässt sich ein Modell als „[…] das Ergebnis einer Konstruktion eines Modellierers, der für Modellnutzer eine Repräsentation eines Originals zu einer Zeit als relevant mit Hilfe einer Sprache deklariert“ definieren. Für die hier gegenständlichen Geschäftsprozessmodelle (GPM) lässt sich weiter einschränken, dass eine Repräsentation eines Geschäftsprozesses erstellt wird. Dieses wiederum kann exemplarisch anhand von (Davenport 1993, S. 5) definiert werden, wonach ein Geschäftsprozess eine Struktur betrieblicher Aufgaben mit Ort, Zeit, Anfang und Ende, definierten benötigten Ressourcen und produzierten Leistungen sei. Die Struktur betrieblicher Aufgaben in einem (Geschäfts-) Prozessmodell wird von (Keller et al. 1992) auch als Kontrollfluss bezeichnet.

Die Nützlichkeit von Prozessmodellen ist seit einiger Zeit bekannt (vgl. Nordsieck 1962). Prozessmodelle werden erstellt und verwendet, um bspw. das Verständnis von, und das Gespräch über Geschäftsprozesse zu fördern, um Geschäftsprozesse zu planen, oder um sie zu verbessern (vgl. Kawalek & Kueng 1997; Nordsieck 1962). Die Notwendigkeit dafür ist eine Folge der aktuellen Entwicklung des Marktumfeldes vieler Unternehmen, das durch steigenden Optimierungsdruck geprägt ist. Die globale Verfügbarkeit von Dienstleistungen, Waren und Informationen erhöht das Preis- und Qualitätsbewusstsein der Kunden, so aber auch der Wunsch nach Differenzierung und Individualisierung. In der Folge müssen Unternehmen zunehmend komplexere Abläufe kosteneffizienter bewältigen und stets neue Ansprüche berücksichtigen. (Becker 2000) Um die notwendige Anpassung kontrolliert durchführen zu können, wurden Managementansätze entwickelt, die eine ingenieurmäßige Weiterentwicklung des Unternehmens (bspw. Österle 1995) oder der Prozesse (bspw. Davenport 1993; Hammer & Champy 2002; Becker 2000) verfolgen und dabei auf die Prozessmodellierung zurückgreifen. Folglich treffen Prozessmodelle in Unternehmen (Harmon & Wolf 2014) wie in der Forschung (van der Aalst 2012) auf reges Interesse.

Unternehmen verwenden Prozessmodelle u. a., um betriebliche Prozesse zu entwerfen, zu verbessern, alternative Entwürfe zu bewerten oder um auf ihrer Grundlage zu entwickelnde IT-Systeme zu spezifizieren (Kawalek & Kueng 1997). Dabei ist die Qualität der Modelle entscheidend für den Erfolg der Initiativen (Moody 2005). Wenn Prozessmodelle für den Anwender schwer verständlich sind, läuft dieser Gefahr, den modellierten Sachverhalt falsch zu verstehen, was wiederum bspw. zu Fehlern in Implementierungen von Geschäftsprozessen und Systemen oder zu einer falschen Bewertung und Entscheidung von alternativen Entwürfen führen kann (vgl. Gruhn & Laue 2006b; Vanderfeesten et al. 2007a; Mendling 2008; Mendling & Strembeck 2008; Gruhn & Laue 2007; Houy et al. 2012; Fettke et al. 2012). Daraus resultierende Schäden können, besonders wenn solche Fehler in frühen Phasen der Planung auftreten und lange unbemerkt bleiben, erheblich ausfallen, hohe Kosten verursachen, die Anpassung des Unternehmens an Änderungen im Marktumfeld erschweren und den Nutzen der Modellierungsanstrengungen erheblich einschränken (vgl. Briand & Wust 2001).

Dies macht „Verständlichkeit" zu einer wichtigen Qualitätsdimension von Geschäftsprozessmodellen (Houy et al. 2014). Eine einheitliche Definition für den Begriff der Verständlichkeit (von Prozessmodellen) hat sich in der Wirtschaftsinformatik zum aktuellen Zeitpunkt nicht etabliert. Aufbauend auf einer umfassenden Analyse der im Zusammenhang mit Prozessmodellverständlichkeit verwendeten Definitionen für den Begriff, wird Verständlichkeit von Houy et al. (2012, S. 66) definiert, dass Verständlichkeit sich „aus der Anstrengung ergäbe ein Modell korrekt zu lesen und zu interpretieren, was wiederum ein kognitiver Prozess sei, bei dem den verschiedenen Bestandteilen eines konzeptuellen Modells eine Bedeutung zugewiesen werde."[1] (vgl. Houy et al. 2012, S. 66). Das Ergebnis dieses kognitiven Prozesses zeigt sich im Wissen, genauer in den mentalen Modellen von Sachverhalten, die sich ein Modellnutzer erschafft und die es ermöglichen Probleme kreativ zu lösen (Mayer 1989). Es ist anzumerken, dass die Forschung zur Prozessmodellverständlichkeit von einem Pluralismus an Theorien geprägt ist, welche wiederum den Begriff „Verständlichkeit" unterschiedlich auslegen (Houy et al. 2014). Für diese Arbeit hervorzuheben sind zwei dieser Theorien[2], die Bunge-Wand-Weber Ontologie (Wand & Weber 1990b) und die Cognitive-Load Theory (Sweller

1 Übersetzung erfolgte durch den Autor. Im Original: „[Understandability] is related to the ease of use, respectively the effort for reading and correctly interpreting a conceptual model, which is a cognitive process of assigning meaning to the different parts of a conceptual model" Houy et al. 2012, S. 66

2 Vgl. Houy et al. 2014 für eine Übersicht.

1988). Eine detaillierte Beschreibung der Theorien findet sich auf Seite 42 (Bunge-Wand-Weber Ontologie) und auf Seite 184 (Cognitive-Load Theory).

Die Notwendigkeit, die Qualität von Prozessmodellen in der Praxis bewerten und letztlich steigern zu können, motiviert die Operationalisierung dieser Theorien sowie die Entwicklung weiterer Ansätze wie bspw. Qualitätsframeworks (vgl. Schütte 1998; Krogstie et al. 2006), Sammlungen praktischer Empfehlungen (vgl. Mendling et al. 2010; Sharp & McDermott 2009), oder aber Metriken (vgl. Vanderfeesten et al. 2007a; Overhage et al. 2012), die den Modellierer unterstützen sollen.

Qualitätsframeworks wie bspw. die Grundsätze ordnungsmäßiger Modellierung (Schütte 1998) oder das SEQUAL Framework (Krogstie et al. 2006) explizieren ihre Auffassung von Modellqualität anhand von Dimensionen wie bspw. Konstruktions- und Sprachadäquanz oder aber Knowledge Quality und Domain Quality. Diese Dimensionen beschreiben Relationen, bspw. zwischen Modellersteller und Modellnutzer, oder aber zwischen Nutzerwissen, Modell und Domäne, für die möglichst Konsens bzw. Übereinstimmung anzustreben sei, um hohe Modellqualität zu erreichen. Mit welchen konkreten Maßnahmen dieses Ziel wiederum zu erreichen sei, das wird von den Verfassern nicht vorgegeben. Diese zu finden bleibt dem Anwender überlassen. Zwar ermöglicht dies eine flexible, situative Anwendung des Frameworks. Das Erkennen passender Maßnahmen ist aber stets mit Aufwand verbunden und wird durch den Wissens- und Erfahrungshintergrund des Anwenders stark beeinflusst.

Sammlungen praktischer Empfehlungen nennen konkrete Anforderungen an qualitativ hochwertige Modelle, bspw., dass Prozessmodelle aufzuteilen seien, wenn diese mehr als 50 Elemente besäßen (Mendling et al. 2010), oder, dass stets s.g. Swimlane-Diagramme zu verwenden seien (Sharp & McDermott 2009). Diese Anforderungen kommen Maßnahmen gleich und lassen sich oft ohne weitere Interpretation anwenden. Jedoch ist die fehlende Flexibilität in der Anwendung leicht zu erkennen. So sind Swimlanes bspw. nicht angemessen, wenn kein Zuordnungsmerkmal modelliert wird. Und während 50 Elemente in einem Modell einen Anwender überfordern, geht anderen Anwendern der Zusammenhang zwischen vielen „kleinen“ Modellen verloren (vgl. „Split Attention Effekt“ Zugal 2013).

Hinsichtlich der einfachen und flexiblen Anwendung können Metriken als dritter Weg gesehen werden. Sie sind häufig automatisiert berechenbar und erlauben damit die einfache Bewertung von einer großen Anzahl an Modellen (Braunnagel & Johannsen 2013) (vgl. Malinova et al. 2013). Ob Maßnahmen notwendig sind oder nicht kann wiederum situativ vor dem Hintergrund des Wertes der Metrik und der Fähigkeiten des Anwenders entschieden werden, wobei konkrete Verbesserungsmaßnahmen von den Metriken nicht vorgegeben werden. Sie stellen das Mittel der Wahl dieser Arbeit dar. Exemplarisch für eine Metrik sei die Cross Connectivity Metrik (Vanderfeesten et al. 2008a) genannt. Diese berechnet die durchschnittliche Anzahl an Pfaden zwischen zwei Elementen im Kontrollfluss eines Prozessmodells. Damit soll eine Aussage über dessen Komplexität und ferner über die Verständlichkeit des Modells getroffen werden.

Für eine möglichst umfassende Betrachtung müssen sowohl die Syntax als auch die Semantik eines Modells berücksichtigt werden. Ein Beispiel für die Bewertung anhand der Semantik findet sich in den Arbeiten von Johannsen und Leist (vgl. Johannsen & Leist 2012b) zur Bewertung von Prozessmodelldekompositionen, einer verbreiteten Maßnahme zur Verbesserung der Modellverständlichkeit, auf Basis der Good Decomposition Conditions (GDC). Diese operationalisieren die o.g. Bunge-Wand-Weber Ontologie (Johannsen & Leist 2012b).

Die Dekomposition ist eine von vielen gängigen Modellierungssprachen unterstützte Maßnahme zur Verbesserung der Verständlichkeit der Prozessmodelle. Dabei wird der zu modellierende Inhalt derart über Ebenen mit steigendem Detailgrad verteilt, dass Elementen in einem Modell mit niedrigem Detailgrad ein Modell auf einer Ebene mit hohem Detailgrad zugeordnet wird, sodass dieses detaillierte Modell ein Element genauer beschreibt (Reijers et al. 2011) (vgl. Polyvyanyy et al. 2008). Vergleichbar mit einer Landkarte, die bei grobem Maßstab vor allem einen Überblick zur Orientierung vermittelt und bei feinem Maßstab die Details, soll ein Prozessmodell mit niedrigem Detailgrad überblicksartig die grobe Struktur des Prozesses bspw. für strategische Entscheidungen aufzeigen, während, bspw. für die genaue Analyse des Prozesses, weitere Modelle detaillierte Beschreibungen enthalten (Polyvyanyy et al. 2008). Modelle hierarchisch strukturierter Detailgrade können erstellt werden, indem Elemente eines detailarmen Prozessmodells sukzessive durch detailliertere Modelle verfeinert werden (vgl. Reijers et al. 2011), oder aber indem ausgehend von der Menge aller Elemente mit höchster Detaillierung diese Elemente nach Zusammengehörigkeit partitioniert werden und jede Partition durch ein einzelnes Element in einem Modell niedrigerer Detailstufe abstrahiert wird und dieser Vorgang iterativ für Elemente der neu gewonnenen Ebene wiederholt wird (vgl. Polyvyanyy et al. 2008).

Während einerseits die Zielsetzung der Dekomposition, eine Strukturierung von Detailstufen, in der Literatur beschrieben ist, finden sich andererseits wenige Hinweise darauf, wie eine erfolgte Dekomposition zu bewerten ist (Burton-Jones & Meso 2006) (vgl. Reijers et al. 2011). Um diese Lücke zu schließen, wenden (Johannsen & Leist 2012b) die s.g. Good Decomposition Conditions nach (Weber 1997) an, mit dem Ziel daraus Kriterien zur Bewertung einer Dekomposition zu entwickeln. Die GDC basieren auf dem Formalismus einer Dekomposition (Wand & Weber 1989a), welche Teil des ontologischen Modells für Informationssysteme (Wand & Weber 1990a) ist, das wiederum auf der Ontologie von Bunge-Wand-Weber basiert (Weber 1997). Die GDC spezifizieren fünf Anforderungen an die Dekomposition eines Informationssystemmodells, die von (Johannsen & Leist 2012b) für die eEPK formuliert werden. Bspw. sei das transferierte Losslessness-Kriterium genannt, das besagt, dass in einer Dekomposition keine Informationen verloren gehen dürfen, und, insbesondere Ereignisse der Prozessausführung bewahrt werden müssen (Johannsen & Leist 2012b).

Die GDC nach (Johannsen & Leist 2012b) sind die theoretische Grundlage eines Schwerpunktes dieser Arbeit. So ist nach der theoretischen Arbeit von (Johannsen & Leist 2012b) noch nicht ermittelt, ob die Einhaltung der bzw. der Verstoß gegen die Kriterien tatsächlich mit einem messbaren Einfluss auf die Verständlichkeit einhergeht. Weiter setzen die GDC in der Form von (Johannsen & Leist 2012b) eine manuelle Prüfung der Einhaltung voraus, was gerade in Verbindung mit einer hohen Anzahl von Modellen, bspw. in Prozessarchitekturen eine Einschränkung darstellt (vgl. Malinova et al. 2013).

Neben der Semantik nimmt auch die Syntax Einfluss auf die Verständlichkeit von Prozessmodellen. Tatsächlich finden sich in der Literatur Metriken zur Evaluation konzeptueller Modelle anhand deren Syntax, insbesondere im Software-Engineering (vgl. DeMarco & Boehm 1982; Fenton & Pfleeger 1998; Halstead 1977; Henderson-Sellers 1996; McCabe 1976). Bezogen auf Prozessmodelle ist die Entwicklung weniger fortgeschritten. So beschreibt (Vanderfeesten et al. 2007a) eine erste Klassifikation, welche wiederum aus dem Software-Engineering abgeleitet ist. Die Autoren schlagen vor, Metriken in die Klassen „Coupling“, „Cohesion“,

„Complexity“, „Modularity“ und „Size“ zu unterteilen (Vanderfeesten et al. 2007a). Die erste Klasse, „Coupling“, wird dabei definiert als ein Maß für die Stärke des Zusammenhangs, der sich aus Verbindungen zwischen Modulen ergibt, sowie aus Art und Kompliziertheit der Verbindungen (Vanderfeesten et al. 2007a). Auch diese Definition ist aus dem Software-Engineering abgeleitet. Bezogen auf Prozessmodelle lässt sie im unklaren, welche Verbindungen gemeint sind, was Module sind, oder was in diesem Kontext Kompliziertheit ist.

Für diese Klasse existieren im Vergleich zu bspw. den Klassen „Complexity“ und „Size“ verhältnismäßig wenige Metriken (González et al. 2010) (vgl. Mendling 2008), obwohl erste empirische Untersuchungen einen Zusammenhang mit der Verständlichkeit von Prozessmodellen aufzeigen (Vanderfeesten et al. 2008a). Um weitere Metriken zu finden, wird in (Braunnagel & Johannsen 2013) zunächst diskutiert Couplingmetriken aus dem Software-Engineering für die Anwendung auf Geschäftsprozessmodelle zu transferieren. Weiter werden Ideen für Metriken skizziert, die aus einem solchen Transfer entstehen können. Trotz der Skizzenhaftigkeit zeigt sich in dieser Arbeit bereits die Heterogenität der Metriken bezüglich ihrer Aussage und Anwendung, aber auch bezüglich der theoretischen Grundlagen. Manche Metriken operationalisieren die o.g. Cognitive-Load Theory, andere Metriken wiederum wenden ein Text-Mining Verfahren an. Die Aussagen der möglichen Metriken beziehen sich teils auf die Verständlichkeit der konzeptuellen Modelle, teils auf funktionale Redundanz in den Prozessen. Jedoch sind die skizzierten Metriken in Folge dieses Beitrages noch nicht anwendbar. Ihre konkrete Entwicklung ist der zweite Schwerpunkt dieser Arbeit.

Zusammenfassend besteht die Problemstellung dieser Arbeit darin, Instrumente zur Bewertung der Verständlichkeit von Prozessmodellen zu entwickeln. Die vorgestellten Ergebnisse haben zwei Ansatzpunkte: Einerseits adressiert Coupling vorwiegend die Verständlichkeit der Syntax bestehender Prozessmodelle. Demgegenüber adressiert die Dekomposition eine Maßnahme zur Verbesserung der Verständlichkeit und die Bewertung anhand der GDC die Semantik der Modelle.

1.2 Zielsetzung und Forschungsfragen

Die oben skizzierte Problemstellung leitet die beiden Zielsetzungen dieser Arbeit ein: Die Entwicklung von Metriken zur Messung der Prozessmodellverständlichkeit, einerseits auf Basis der Good Decomposition Conditions, und andererseits anhand der Qualitätsdimension „Coupling“.

Ausgangspunkt der Arbeit ist die Frage nach dem Bedarf an Forschung im Bereich BPM. Während bereits ausgeführt wurde, warum lt. herrschender Meinung in der Forschung BPM für Unternehmen in der aktuellen Marktsituation besondere Relevanz hat, soll zunächst ermittelt werden, inwieweit Unternehmen Methoden und Techniken des BPM tatsächlich einsetzen. So sollen die aktuelle Verbreitung sowie Gründe warum oder warum nicht Methoden und Techniken eingesetzt werden empirisch erhoben werden. Hieraus folgt die erste Forschungsfrage (FF):

- **Forschungsfrage 1**: Zu welchem Grad setzen Unternehmen BPM Maßnahmen ein und welche Faktoren motivieren oder behindern den Einsatz von BPM in Unternehmen?

Im Schwerpunkt Good Decomposition Conditions soll die Arbeit von (Johannsen & Leist 2012b) so fortgeführt werden, dass die Conditions zu Metriken weiterentwickelt werden. Hierfür ist es notwendig die GDC empirisch dahingehend zu untersuchen, ob die Einhaltung der GDC bzw. der Verstoß dagegen einen messbaren Einfluss auf die Verständlichkeit dekomponierter Prozessmodelle hat, zunächst unter den Idealbedingungen einer Laborsituation. Grund hierfür ist, dass ein Artefakt, dass seine Tauglichkeit unter Idealbedingungen nicht zeigen kann, im praktischen Einsatz ebenfalls keine Tauglichkeit erwarten lässt. In diesem Fall wäre es angezeigt, die theoretischen Vorarbeiten zu hinterfragen. Hieraus folgt die Forschungsfrage.

- **Forschungsfrage 2**: Haben Verstöße gegen die Good Decomposition Conditions nach (Johannsen & Leist 2012b) einen Einfluss auf die Verständlichkeit von Prozessmodellen?

Die GDC, in der von (Johannsen & Leist 2012b) vorgestellten Form kommen in ihrer Verwendung den oben diskutierten Sammlungen praktischer Empfehlungen gleich. Sie sind Kriterien für die Bewertung und gleichzeitig Empfehlungen für die Durchführung der Dekomposition. Ebenfalls bereits diskutiert wurden die Vorteile von Metriken, insbesondere, dass diese die Bewertung einer großen Anzahl an Modellen erleichtern, während die Interpretation situationsgerecht vom Anwender vorgenommen werden kann. Es soll also im Sinne der praktischen Verwendbarkeit der GDC untersucht werden, ob sich diese geeignet als Metriken formulieren lassen. Hieraus folgt die nächste Forschungsfrage:

- **Forschungsfrage 3**: Welche Metriken lassen sich aus den Good Decomposition Conditions ableiten, um Modellierer bei der Dekomposition von Prozessmodellen zu unterstützen?

Dem Forschungsschwerpunkt Coupling liegen Arbeiten aus dem Software-Engineering und die skizzenhafte Beschreibung in (Braunnagel & Johannsen 2013) zugrunde, die der Untersuchung von Coupling in der Prozessmodellierung in mehrfacher Hinsicht dienlich sind. Zunächst ist der Begriff „Coupling“ im Zusammenhang mit Prozessmodellen bisher nur vage definiert, jedoch kommt in jeder Coupling Metrik eine spezifische Auslegung zum Ausdruck. Eine genauere Untersuchung des Begriffsverständnisses in der Literatur soll zu einer präziseren Definition führen. Weiter ist es für die Entwicklung von Coupling Metriken notwendig, Anforderungen an das Artefakt zu spezifizieren. Hierfür ist es besonders wichtig zu ermitteln, welche Problemstellungen im Einsatz von BPM mit Coupling Metriken adressiert werden können, um eine Zielsetzung für das Artefakt festzulegen, und welche Anforderungen dabei bestehen. Hieraus folgt Forschungsfrage vier.

- **Forschungsfrage 4**: Wie kann Coupling im Kontext von Geschäftsprozessmodellen definiert werden und was sind mögliche Einsatzszenarien für und Anforderungen an Coupling Metriken im praktischen Einsatz?

Für die Entwicklung der Coupling Metriken für Geschäftsprozessmodelle soll auf bereits erprobte und etablierte Metriken zurückgegriffen werden, wie bspw. solche, die für konzeptuelle Modelle im Software-Engineering entwickelt wurden. Um sicherstellen zu können, dass das ursprüngliche Verständnis von Coupling möglichst erhalten bleibt bzw. weiterhin eine sinnvolle Messung stattfindet, muss die Methode für den Transfer den Qualitätsaspekt der Coupling begründet, separat ermitteln. Auf die bestehenden Metriken angewandt, können so Coupling Metriken für Geschäftsprozessmodelle entwickelt werden. Es folgt daher als fünfte Forschungsfrage:

- **Forschungsfrage 5**: Wie und welche Coupling Metriken, bzw. der jeweils gemessene Qualitätsaspekt, aus verwandten Disziplinen lassen sich für Prozessmodelle operationalisieren?

Bei der Entwicklung von Artefakten kommt der Evaluation eine besondere Wichtigkeit zu (Sonnenberg & Vom Brocke 2012a). Dem wird in dieser Arbeit Rechnung getragen, indem die Metriken auf zwei Arten evaluiert werden. Zunächst soll der Konstruktionsprozess der Metriken anhand von Kriterien in der Theorie evaluiert werden, um den Stand der Forschung adäquat zu berücksichtigen. Anschließend soll empirisch der Frage nachgegangen werden, ob die Metriken tatsächlich eine Veränderung in der Verständlichkeit von Geschäftsprozessmodellen anzeigen. Daraus folgen Forschungsfrage sechs und sieben.

- **Forschungsfrage 6**: Erfüllt die Konstruktion der Metriken bekannte Kriterien aus der Forschung?

- **Forschungsfrage 7**: Zeigen die Coupling Metriken eine verminderte oder verbesserte Verständlichkeit von Prozessmodellen an?

1.3 Aufbau der Arbeit

Um die Lösung zu o.g. Forschungsfragen zu präsentieren, ist diese Arbeit wie folgt aufgebaut. Auf die vorausgehende Vorstellung von Motivation, Problemstellung und Forschungsfragen in Kapitel 1 hin, werden in Kapitel 2 die einzelnen Forschungsfragen bearbeitet. Kapitel 3 fasst die Arbeit zusammen. Weiter werden dort auch die Ergebnisse kritisch gewürdigt und ein Ausblick auf weiteren Forschungsbedarf gegeben.

Abbildung 1 zeigt auf, wie die Beiträge in Kapitel 2 den einzelnen Forschungsfragen und den inhaltlichen Schwerpunkten „Coupling“ und „Dekomposition“ zugeordnet sind. Darüber hinaus nimmt Abbildung 1 eine methodische Einordnung der Beiträge vor. In beiden Schwerpunkten wird das Ziel verfolgt Artefakte zu entwickeln. Im Schwerpunkt „Coupling“ werden, wie vorher beschrieben, Coupling Metriken entwickelt, während im Schwerpunkt „Dekomposition“ die Conditions für die eEPK sowie die Metriken als Artefakte entwickelt werden.

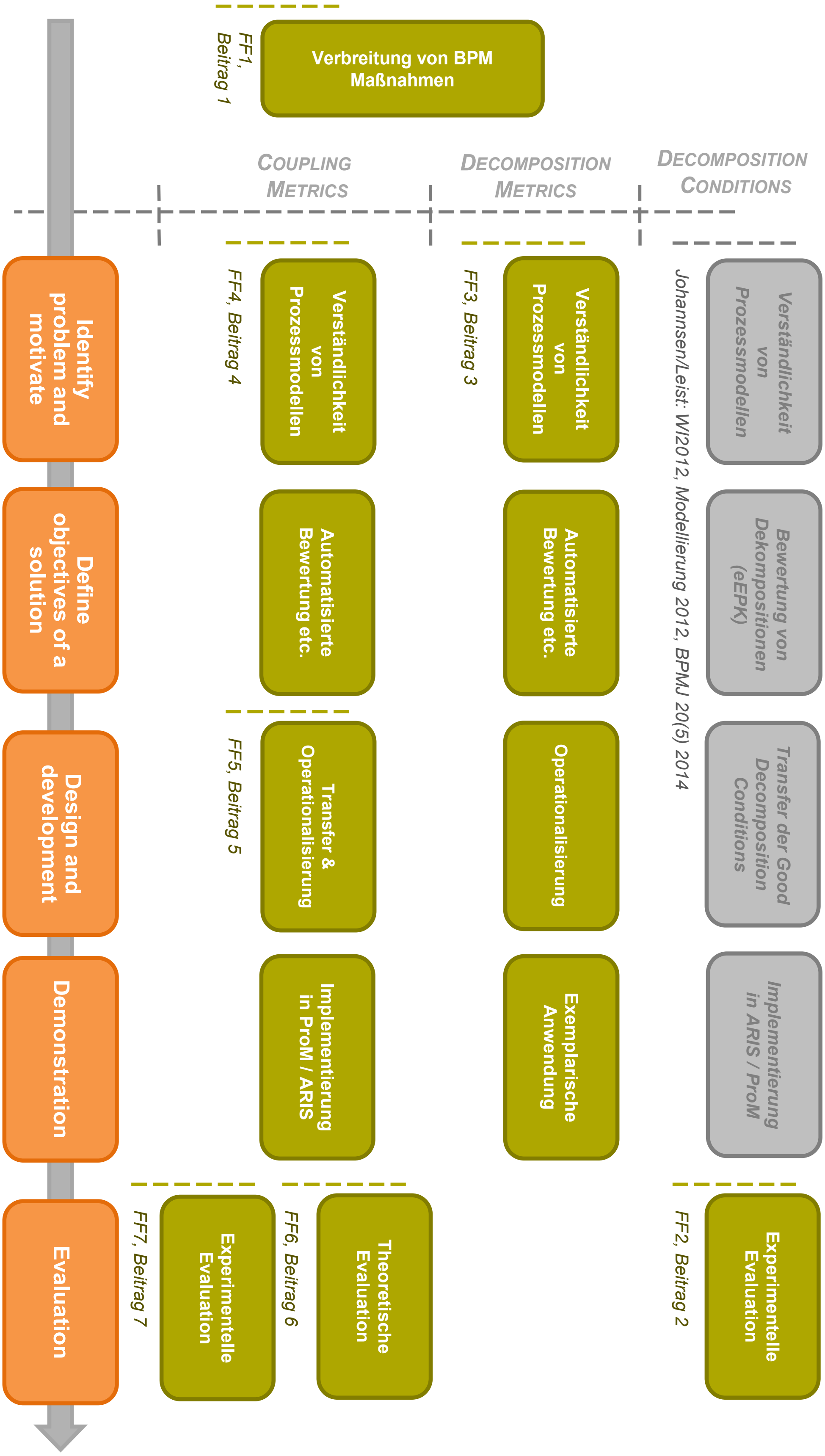


Abbildung 1: Methodische Einordnung

Aus methodischer Sicht lässt sich der Zusammenhang der Beiträge anhand der Design Science Methode nach (Peffers et al. 2006) erläutern, welche für den Forschungsprozess hinsichtlich der Artefaktentwicklung die folgenden Schritte vorschlägt:

1. Problem identifizieren und Lösung motivieren
2. Zielsetzung der Lösung definieren
3. Entwurf und Entwicklung
4. Demonstration
5. Evaluation
6. Kommunikation der Ergebnisse

Als Motivation für beide Schwerpunkte kann die empirische Erhebung des ersten Beitrages herangezogen werden. So folgt aus diesem u. a. die Erkenntnis, dass viele Unternehmen auf den Einsatz von BPM und Prozessmodellierung verzichten, weil gute Prozessmodelle nur kompliziert zu erreichen sind (vgl. Kapitel 2.1.4). Daraus wird im Rahmen dieser Arbeit die Notwendigkeit abgeleitet, zusätzliche Hilfestellung für das Modellieren von Prozessen zu bieten.

Im Schwerpunkt „Dekomposition“ liegen dem Design Science Prozess Publikationen zugrunde, die nicht Teil dieser Arbeit sind. In den Beiträgen (Leist & Johannsen 2012; Johannsen & Leist 2012a, 2012b; Johannsen et al. 2014b) wird die Entwicklung damit motiviert, dass die Verständlichkeit von Prozessmodellen für den Unternehmenserfolg entscheidend sei, sowie die Anforderungen an das Artefakt definiert, bspw., dass durch das Artefakt die Bewertung von Dekompositionen von eEPK Modellen ermöglicht werde. Weiterhin werden in diesen Arbeiten die Good Decomposition Conditions für die eEPK spezifiziert und in Form von Implementierung als Plug-Ins in den Frameworks ProM (Dongen et al. 2005) und ARIS demonstriert (Scheer 2000). Der darauffolgende Schritt, die Evaluation, wird in Forschungsfrage 2 (bzw. Beitrag 2, Kapitel 2.2) als Laborexperiment verfolgt. Die Ergebnisse dieses ersten Design Science Zyklus, die Good Decomposition Conditions, sind Grundlage für die Entwicklung der Good Decomposition Metriken.

Wie auch zuvor, wird die Entwicklung der Good Decomposition Metriken mit der Notwendigkeit der Forschung zur Verständlichkeit von Geschäftsprozessmodellen begründet (siehe Kapitel 2.3.1). Darüber hinaus erklärt Beitrag 3 (bzw. FF 3), dass Werkzeuge zur Bewertung von Prozessmodellverständlichkeit zusätzlichen Nutzen gewinnen, wenn diese eine automatisierte Bewertung erlauben, weshalb für Good Decomposition Conditions entsprechende Metriken als Ergänzung entworfen und entwickelt werden. Beitrag drei nimmt zuletzt die Demonstration des Artefakts vor, indem es die Metriken auf die Prozessmodelle aus dem Laborexperiment in Beitrag drei anwendet und die Resultate diskutiert.

Zuletzt wird der Forschungsschwerpunkt „Coupling“ behandelt. Dieser wird in Beitrag vier zunächst mit der Notwendigkeit verständlicher Prozessmodelle motiviert. Weiter wird untersucht, welchen Nutzen Coupling Metriken bei der Anwendung auf Prozessarchitekturen haben, woraus sich Anforderungen, wie bspw. die Bewertung des Kontrollflusses, insbesondere aber die Möglichkeit für eine automatisierte Bewertung ergibt. Beitrag fünf setzt den Forschungsprozess fort, indem Coupling Metriken für die eEPK auf Basis existierender Metriken in verwandten Disziplinen entwickelt werden. Das Ergebnis wird außerdem anhand einer Implementierung als Plug-Ins in ProM (Dongen et al. 2005) und ARIS (Scheer 2000) demonstriert.

Zuletzt lässt sich Forschungsfrage 6 der Evaluationsphase des Design Science Zyklus zuordnen. Diese erfolgt zunächst in Beitrag sechs, indem der Entwicklungsprozess hinterfragt wird und die Metriken anhand von Kriterien aus der Literatur geprüft werden. Beitrag sieben wiederum prüft die Metriken in einem Laborexperiment daraufhin, ob Designänderungen am Prozessmodell, die zu Änderungen im Wert der Metrik führen, auch zu einer messbar anderen Verständlichkeit des Prozessmodells führen.

Der letzte Schritt des Forschungsprozesses nach (Peffers et al. 2006), die Kommunikation der Ergebnisse, erfolgt in Kapitel 2 dieser Arbeit.

2 Wissenschaftliche Beiträge

2.1 Beitrag 1: BPM adoption in small and medium-sized companies in Bavaria

Adressierte Forschungsfrage	**Forschungsfrage 1**: Zu welchem Grad setzen Unternehmen BPM Maßnahmen ein und welche Faktoren motivieren oder behindern den Einsatz von BPM in Unternehmen?
Erscheinungsort	24th European Conference on Information Systems, ECIS 2016, Istanbul, Turkey, June 12-15, 2016 (VHB Jourqual 3: B)
Autoren	Daniel Braunnagel 30 % Thomas Falk 30 % Benjamin Wehner 30 % Prof. Susanne Leist 10 %
In diesem Beitrag wird die Verbreitung von BPM Maßnahmen in bayerischen Unternehmen ermittelt und aufgezeigt, dass der Umsetzungsgrad polarisiert: Unternehmen setzen die Maßnahmen entweder zu einem sehr hohen oder zu einem sehr niedrigen Anteil um. Ebenfalls werden mögliche Ursachen für den jeweils hohen oder niedrigen Umsetzungsgrad ermittelt und dazu genutzt, 22 Vorschläge für die Weiterentwicklung des BPM in Wissenschaft und Praxis zu entwickeln. Die Ergebnisse des Beitrages zeigen sowohl den Forschungsbedarf im Bereich BPM als auch die Tatsache auf, dass Methoden der Prozessmodellierung als wichtig wahrgenommen werden. Damit motiviert der Beitrag die allgemeine Zielsetzung dieser Arbeit und lässt sich im DS Zyklus dem ersten Schritt zuordnen. Die besondere Herausforderung dieses Beitrages liegt im Anspruch die Zielsetzung in Breite und Tiefe gleichermaßen zu bedienen. So soll der Umsetzungsgrad von BPM Maßnahmen bei einer ausreichenden Anzahl an Unternehmen erhoben werden, um das Resultat verallgemeinern zu können. Ebenso sollen Argumente für oder gegen die Umsetzung aufgezeigt werden. Gelöst wird diese Problematik durch eine eigens für diese Erhebung entwickelte Methode. Diese kombiniert den Stand aus einer Umfrage mit 114 Teilnehmern mit Ursachen, die per Fallstudien bei zehn Unternehmen erhoben wurde. Für die Erhebung wurde ein eigenes Messinstrument auf Basis der Literatur zur BPM Maturity Messung konstruiert.	

Tabelle 1: Bibliographische Angaben zu Beitrag 1

Small and medium sized (SMEs) companies are a pillar of the Bavarian economy. With business process management (BPM) providing an important competitive advantage in the globalized economy, the adaption of BPM by SMEs has societal relevance. However, the reasons why, or why not, SMEs implement BPM measures are still not fully understood. Previous research addressed this topic either breadthwise as surveys or in depth as case studies, and thus only has a limited perspective. Therefore, in our work, we carry out a mixed method analysis.

We conduct 10 case studies to analyse the current state of adoption as well as the reasons for or against implementing further BPM measures. The insights gained guide the design of the subsequent survey. 114 results allow us to evaluate how widespread a particular reason may be. Lastly, the combined discussion of the results of both the case studies and surveys allow us to identify reasons that hinder or foster BPM adoption in SMEs, which are in-depth as well as generalizable.

The study results are analyzed to derive propositions to research and practitioners alike that support SMEs to introduce further measures of BPM and improve their global competitiveness. For example, we could identify that BPM is in some cases enforced by customers, that stricter certifications are necessary, and that BPM trainings aligned to the needs of SMEs are desirable.

2.1.1 Introduction

Business Process Management (BPM) is one of the key concepts in information systems and represents a comprehensive approach for managing an organization's business operations (Hammer 2010). The BPM concept has continually evolved over the last few decades by integrating methods, techniques and tools from various fields (Harmon 2010). Today BPM is recognized as a holistic management approach encompassing a wide range of aspects (e.g., strategic alignment, governance, methods, information technology, people, and culture (Rosemann & Vom Brocke 2015). Over the years, BPM has been adopted by organizations in various industries all over the world. One main motivation is found in the development of the globalized markets. Increasing demands on organizations for e.g., delivery speed, quality and flexibility together with a growing information transparency force companies to continuously optimize their processes in order to survive in competition.

BPM, if applied in an appropriate manner, facilitates process optimization and is thus considered a competitive advantage (Trkman 2010). A significant number of studies confirm positive effects of BPM on organizational performance (e.g., Kohlbacher 2010; Komus 2011). Further, several studies investigate the adoption of BPM by organizations, both on a worldwide scale (e.g., Harmon & Wolf 2014) and for specific regions (e.g., Brucker-Kley et al. 2014; Minonne et al. 2011). Studies dealing with BPM adoption commonly show (cf. Roeser & Kern 2015 for an overview), despite the generally acknowledged importance of BPM, considerable differences in the BPM adoption between countries, industries and company sizes. In addition, many organizations do not fully exploit the potential of BPM (cf. Minonne & Turner 2012). Whereas large organizations have been making effective use of BPM for quite some time, especially for small and medium sized enterprises (SMEs), the successful adoption of BPM seems to be a particular challenge. Possible reasons identified by (Kolář 2014) include, among other things, the lack of internal manpower dedicated to BPM and the different levels of process rigidity in SMEs compared to large organizations. As smaller companies tend to have a higher portion of flexible or ad-hoc processes, it is even more difficult for them to apply existing BPM measures in a SME context. However, a broader evaluation of BPM adoption in SMEs is missing.

Our research, for this reason, focuses on BPM in small and medium-sized enterprises. Especially in the German economy, SMEs play an important role. Currently, they represent 47% of the gross value added and 39% of the aggregated turnover (Söllner 2014). Further, SMEs currently employ 94% of the employees in the private sector; even more, e.g., in the German state Bavaria, they employ 99.6% of the employees in the private sector (DESTATIS 2015). In summary, because of their high societal relevance and the role of BPM to sustain their competitiveness, the adoption of BPM by SMEs is a highly relevant topic for research. However, what is needed are reliable insights into this domain to derive the pivotal aspects of how to extend or enrich the future BPM research agenda towards SMEs.

The aim of this paper is to capture the status quo of BPM adoption in Bavarian SMEs. This aim is operationalized with three research questions. (1) To which extent are BPM measures realized in those companies? (2) Which factors influence the adoption of BPM? (3) Which of these factors are distinctive to foster or hinder SMEs in adopting BPM? To address these questions, we use a mixed method approach combining qualitative (i.e. case study) and quantitative (i.e. survey) research.

The remainder of this paper is organized as follows. Section 2.1.2 describes the conceptual basics including related work as well as our research method, which is a mixed method approach that integrates a survey and case studies. The results of the survey and the findings of the case studies are presented in section 2.1.3. There we elaborate on the findings related to specific BPM topics and carry out an overall evaluation. In section 2.1.4, the results are summarized and discussed in the light of SME-specific characteristics. Section 2.1.5 concludes the paper.

2.1.2 Conceptual Basics

2.1.2.1 Related Work

Previously, the adoption of BPM was empirically adressed by surveys. For example, the biannual survey “The State of Business Process Management”, a survey on the adoption of BPM, has been focusing operational BPM measures and tools from companies of all sizes and locations since 2006 (cf. Harmon & Wolf 2014). Another example is the ZHAW study (Brucker-Kley et al. 2014; Minonne et al. 2011), which has a particular focus on the strategic aspects of BPM adoption (see section 2.1.3.4). A literature review by Roeser and Kern (2015) examines the status quo and the use of surveys published in the BPM domain. They classify the surveys based on the research goals into six classes. Class IV shows surveys on the status quo of BPM in practice. However, none of these surveys answer our research questions because they focus different objectives or subjects. More empirical research in this area has been conducted by means of case studies, which also follow a slightly different focus in their research. E.g., Dallas and Wynn (2014) carried out a BPM initiative in a middle-sized Australian accounting firm analysing whether BPM can be successfully applied in this particular SME, and Chong (2007) conducted a BPM initiative in an Australian wine company analysing factors that drive or hinder BPM adoption.

2.1.2.2 Methodology

To address the research questions, we follow a mixed method approach combining qualitative (i.e. case study) and quantitative research (i.e. survey). Since these two methods complement each other well, they have been advocated for the study of organizations in IS in particular. Whereas a case study allows an in-depth investigation into the fuzzy and complex nature of an organization, its findings can be tested for generalization with a survey. (cf. Goes 2013; Huysmans & Bruyn 2013; Venkatesh et al. 2013)

Gable (1994) formalized such an approach. He argues that a preceding case study may inform the survey design, e.g., by pilot testing the survey instruments or construct validation. Also, in his case, notes from the case study were used to interpret survey findings (cf. Gable 1994). Our work instantiates the mixed method approach as presented by Gable (1994). The methods' consistency is ensured in two ways. First, in both methods, the targets were selected from the same list of companies. Second, both methods' instrument is built on the same theoretical groundwork.

Regarding the aforementioned statements, our case studies involved repeated visits at the companies' sites over a longer period of time to conduct the studies in person. To make the case studies logistically possible and ensure their consistency, we decided to focus the research context on SMEs in Bavaria. As for that, Bavaria is suitable as it is an economically strong state in Germany having the majority of private sector employments in SMEs. Thus participants of both the case study and the survey were recruited from a list of Bavarian companies that have previously declared their interest in research cooperation. This list—provided by the Bavarian State Ministry of Economy—contained 10,864 companies fitting the criteria of SMEs.

Our measurement is based on the literature on BPM maturity models. BPM maturity models are meant to measure an organization's capabilities of implementing business processes which achieve their business goals (van Looy et al. 2011). Characteristically, the said models provide, among others, lifecycle levels and capability areas for improvement. The levels reflect the progress of implementing measures towards a mature BPM (van Looy et al. 2011). The adaption for our study was done as follows.

First are the assessment items resulting from a systematic search for BPM maturity literature. From these sources, we assembled the means for survey-based maturity assessment. Van Looy et al. (2011) found a common structure among maturity models' capabilities according to which we grouped our assessment items (see Table 3). For example, the category #1 items "Is the process documentation of your organization maintained permanently?" or "Does your organization have a process map?" stem from the maturity model by Schmelzer and Sesselmann (2008).

The second aspect of our instrument is the level classification. While the maturity models do not share a common calculation scheme, they are designed for an assessment of a finer scope than used in our study. However, each assessment item can be mapped to a category of BPM measures (see Table 3), and the item categories have a natural order. E.g., it is clearly mandatory to properly identify and document processes before it makes sense to introduce process performance management (PPM). In fact, this is the systematics of process maturity models and, as Paulk et al. (1993) observe, companies should follow this order. Also, we do realize that companies do not implement a full set of BPM measures for each and every auxiliary process. Because of that, we map the current progress of such companies to the highest category whose items are rated with at least 50% fulfilment on average. To further avoid terminological confusion, we will use the term 'category' for rating progress instead of 'maturity level'. The common list and the common measurement instrument were used in both the case study and the survey to ensure consistency.

2.1.2.3 Case Studies

To find participants for the case studies, we randomly selected companies from the list of Bavarian SMEs (see section 2.1.2.2) until ten companies agreed to participate in the study (see Table 2). In total, we contacted 137 companies. After 10 companies had agreed, the case studies, which consisted of two phases, were performed between March and October 2014.

First, we conducted a semi-structured interview with a representative from each of the participating companies, with these interviews being based on an interview protocol asking about the current situation of the company and its market, current and previous initiatives of BPM, the existence of process documentation, and the measure regarding process performance management. Due to the explorative nature of the interviews, we refrained from more specific interview items. The interviews, which usually took about one day, were conducted by two researchers, protocolled and consolidated afterwards.

Second, we launched a basic BPM initiative with the company, e.g., documenting or revising the existing documentation of a process, which served to provoke a very intensive discussion about BPM. Regarding the documentation, we conducted separate interviews with all employees involved in the process. We used these interviews to also ask about their knowledge of, experience with and attitude towards BPM. Finally, we presented the results of the initiative to the leading board in a workshop. This workshop also served to initiate a discussion of both potential uses and benefits of BPM initiatives and potential drivers or hindrances of implementing further BPM measures. In summary, this second phase produced a rich background of information about why or why not BPM is installed at Bavarian SMEs.

Integrating the case study, we could realize some of the benefits of the mixed method approach by Gable (1994). We piloted the survey at the companies, testing whether the participants were able to understand and answer the survey correctly. Moreover, while not generalizable, the case study notes provide in-depth information for the interpretation of the quantitative results. Even though these information are valid only for the company where the study was conducted, they can serve for triangulation with the survey results. Section 2.1.3 combines findings from the survey with our notes from the case study.

2.1.2.4 Survey

Our survey is built from measures for maturity assessment in literature and informed by the case study. The questionnaire contains the following items. After starting with five demographic questions (e.g., industry, number of employees, etc.), four questions about BPM as a strategic asset are asked. Subsequently we addressed process documentation (five questions), definition of process goals (five questions), process controlling and reporting (six questions), and process improvement (two questions).

Wherever possible, the answers are formulated as a 5-point Likert scale (e.g., degree of agreement, degree of fulfilment). In other cases, the items ask for a yes/no answer or for an open text. The survey was originally in German and items were translated into English for this publication.

The survey was conducted anonymously and the questionnaire was implemented as an interactive PDF file that could be sent by pressing a button at the end. All terminology was explained by mouse-overs, to reduce subjective interpretations. The PDF was distributed via e-mail to 10,864 Bavarian SMEs in total 128 of which responded.

Responses were filtered for relevance, completeness and internal consistency. First, we checked whether the demographics actually fitted with the definition of a SME. Regarding completeness, we eliminated incompletely answered responses, e.g., when the survey was blank from some point on, as this would have distorted our analysis. The consistency check refers to the natural order of categories mentioned before. If a responder claimed to have established all measures of performance measurement without even defining processes at all, we removed the survey for lack of plausibility.

After filtering the responses for completeness and internal consistency, 114 responses remained for evaluation and built the basis for the interpretation and discussion in the next section. If logical dependencies among survey items reduced the number of relevant responses (e.g., “Is your companies process documentation organized in a process map: Yes or No” and “Does your process map show dependencies”), the size of the subset is noted.

#	Industry	Employees
A	Cereal R&D	20
B	Automatization machines	240
C	Bottling machines	250
D	Car accessory	10
E	Electronics	230
F	Measurement instruments	50
G	Electric components	200
H	Fittings and Couplings	120
I	Steel construction	25
J	Steel processing	150

Table 2: Interviewed Companies

#	Category
0	Initial category
1	Processes are defined and documented
2	Roles and resources are defined and documented
3	Process goals are continually revised and communicated
4	Process performance is continually measured and evaluated
5	Processes are continually optimized

Table 3: Categories for the classification of maturity progress

The demographic distribution of responses is as follows. As to the number of employees, 65.2 % of the companies report to have less than 50, while the remaining 44.8% have more employees. The most represented industries are electrical & mechanical engineering (20.9%) as well as the service sector (20.0%). The persons who answered the survey are usually head or

CEO of the company (77.2%) and BPM is part of their daily work (71.1%). The results and their interpretation are subjects of the following sections.

2.1.3 Results

2.1.3.1 BPM and Strategy

The first part of the questionnaire aims at discovering the value of BPM for the companies' strategy and accordingly for their top management. 31.3% of the companies evaluate BPM as very important, as contrasted with 37.4%, which rate BPM as not important for their strategic planning. This corresponds to the objectives that SMEs try to achieve with BPM. The objectives mentioned most frequently are standardization (91.5%), increasing productivity (89.6%) and quality management (87.7%) all of which put the emphasis on operational activities. On the contrary, the impact of BPM for the companies' strategy is not well developed: only a few companies use BPM to support in-/outsourcing decisions (31.1%) or for the application of new technologies (e.g., support of mobile processes; 33.0%). The commitment of top management for BPM is respectable, 53.9% rate a strong commitment, which reflects the widespread knowledge and use of BPM in the Bavarian SMEs. However, it seems that, foremost, BPM means standardization and cost reduction to top management, while they do not see the potential of the knowledge achieved by conducting BPM measures to support strategic planning.

From our preceding case study research, we found two examples, which provide possible reasons for the most frequently named objectives (quality management, standardization) for the use of BPM in SMEs. A certification according to ISO (International Organization for Standardization) standards was mandatory for some companies to prove a certain level of quality to their partners, e.g., suppliers and customers. In those cases, meeting the certification requirements was the main motivation for e.g., documenting a company's processes. The introduction of ERP systems was another reason for a detailed process analysis with the aim of selecting an appropriate ERP system or replacing the existing one.

2.1.3.2 Process Purpose, Documentation, Quality and Capabilities

After the strategic perspective on BPM, the further items address the operational dimension. Items asked whether the processes achieve their goals and whether the goal is achieved reliably, a differentiation pointed at by company C. If the process runs through, it produces the expected results. However, lacking in overview, errors, e.g. delays, remain undiscovered until the customer reacts. Here, the process is not reliable. Figure 2 shows the respective results of the survey. The most frequent answer (36.0%) of the questions combined is that the processes generally achieve their goal and are mostly reliable as well. However, the number of companies considering their processes as generally failing (goal: not at all and occasionally: 7.0%) or mostly unreliable (10.5%) may, in total, be low. Nonetheless, it is still surprisingly high, considering that failing and unreliable processes most presumably have a strong negative impact on the company's performance.

In fact, in none of our case studies, we uncovered processes that predominantly failed or were predominantly unreliable. At company C, purpose achievement was rated high whereas the reliability was subject to improvement. The case study notes uncovered possible reasons: a high number of coordinative tasks ran over a very long period. It was prone to delays and other deviations, which went undetected over long periods of time. Also, the process was new and not fully established yet. Since the majority of tasks was performed by a small number of people, the company board did not consider any form of documentation necessary. Only when the project sizes and numbers increased, the need for change was perceived. In the BPM initiative, our process models made the process transparent, and the board realized that they had completely underestimated the complexity of the coordinative tasks in general. Further, the initiative uncovered many issues in detail that had never been communicated by the employees before, e.g., the lack of consistent data or diverging assumptions about the process in general. The board assumed that tackling these issues would improve the reliability greatly.

Figure 3 shows the results regarding process documentation and management capabilities. Less than a third (22.7%) of the participants declared to "not at all" or only "sporadically" update their documentation. We doubt that all of these companies fully realize the benefit and potential of BPM initiatives. It becomes evident that more advanced measures, e.g., the process map or role descriptions, are less frequently installed. The case study notes revealed possible reasons. Some of the companies had installed a process documentation because important customers had urged them to do so. E.g., company J is a supplier for the automotive industry and is thus required to have an ISO certification and basic BPM measures installed. Still, BPM was essentially considered a costly nuisance. Hence the company had not trained any of their personel to perform BPM initiatives, was not willing to invest in BPM initiatives and maintained the least possible amount of documentation to sustain the certification. There were no attempts to manifest BPM as a means of improvement in any way. Other companies, e.g., company E, installed BPM staff out of their own motivation to improve processes. Thus, the persons involved implemented measures such as a process map being, in fact, well trained to do so.

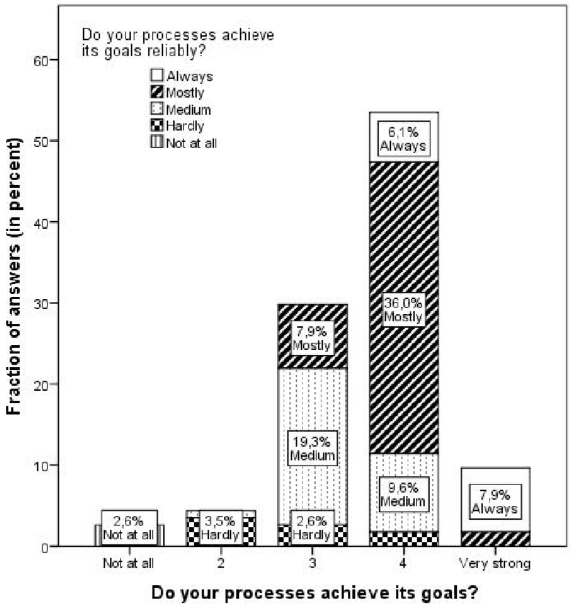


Figure 2: Goal achievement of processes

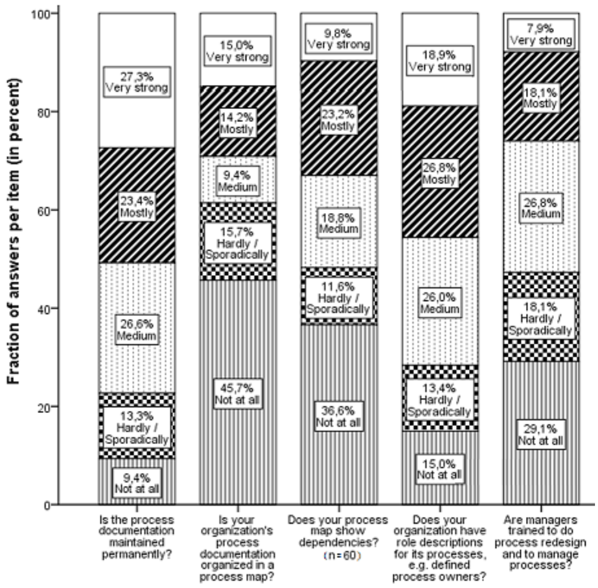


Figure 3: Documentation measures and capabilities

2.1.3.3 Process Controlling and Improvement

Further, the questionnaire focuses on process controlling and process improvement asking whether goals for processes are defined and aligned to the business strategy. 42.5% of the respondents state that process goals have been defined for the majority of their processes, in contrast to 30.1% declaring that they do not use process goals at all, or only rarely. Companies that widely use process goals mostly also link them to the business strategy. Still, the majority of companies with only few goals in place do not derive them from strategy although there is a broader distribution to be observed. In no case, company-wide use of process goals goes without anchoring them to strategy (see Figure 4).

To evaluate if process goals are reviewed and adjusted where necessary, we considered a subset of companies (n=98) that had defined goals in the first place. About 58% of these companies claimed to be doing this at least once a year (Figure 5). As goals serve as a benchmark for process performance, they should be communicated to and understood by everyone involved in the process. However, in 51.3% of the SMEs, goals are not or only partially known to the employees involved (Figure 5).

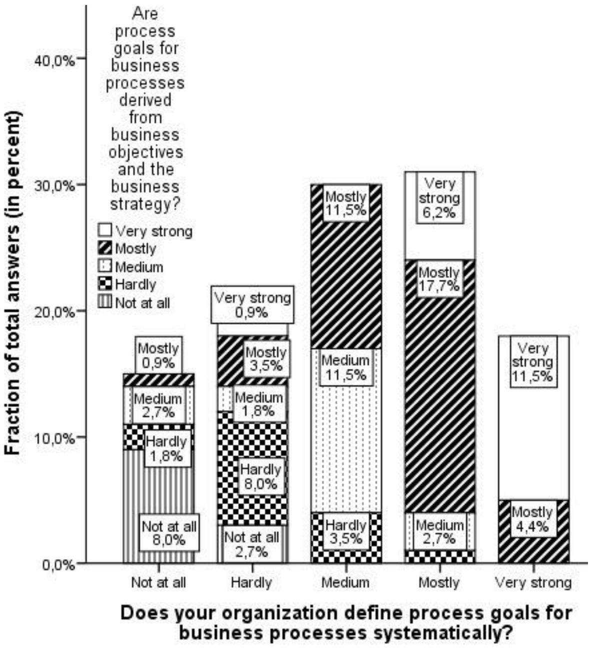


Figure 4: Definition of process goals

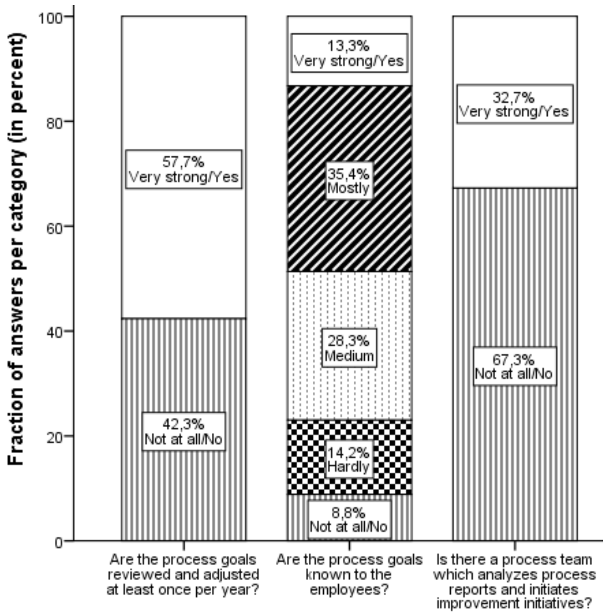


Figure 5: Process goals and organizational integration (n=98)

These results correspond to the findings in our previous case studies. Here, too, the majority of the companies defined process goals that were derived from strategy. However, checking the documents in company H, the last time they actually were updated had been 3 to 6 years ago. A regular review and an adaption to current requirements were missing. Also, it became apparent that employees involved in the process were not aware of the process goals, because these goals had not been communicated to them.

An important aspect relating to process monitoring and controlling is the operationalization of goals in terms of measurable performance indicators. As expected, a positive correlation between the determination of process goals and the use of process performance indicators was found. However, a considerable number of SMEs (21.9%) do not take advantage of process performance indicators at all. More than 42% have defined process goals for the majority of their processes but less than half of them have specified performance indicators to operationalize process goals. Most interestingly, even when the companies stated to have specified goals, for each business process, about 17.6% of them do not use any performance indicators whatsoever for measurement.

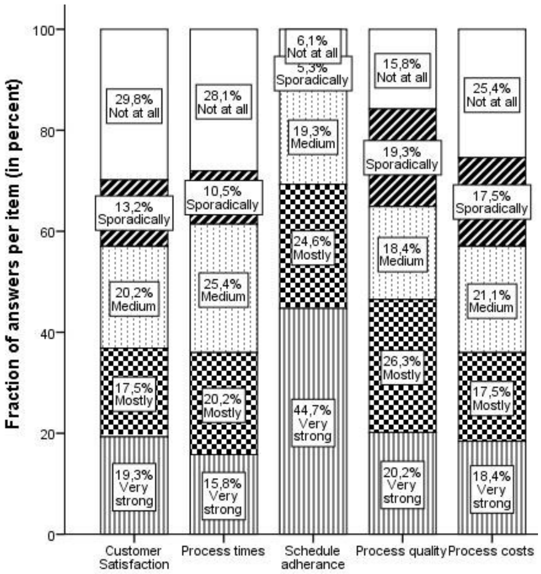


Figure 6: PPM indicators used by SMEs

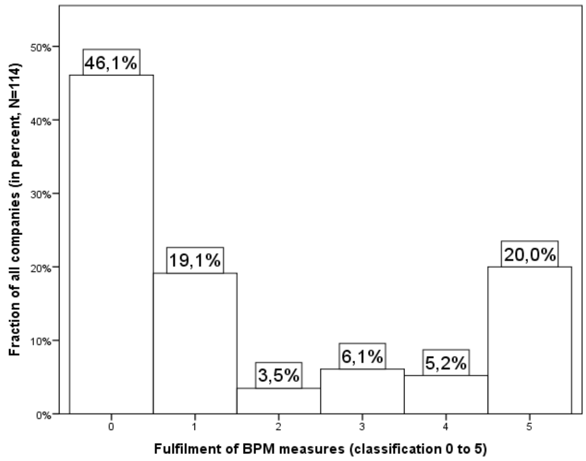


Figure 7: Overall classification of SMEs

In addition, we asked which indicators SMEs usually employ for process performance measurement (Figure 6). By far the most important is “Adherence to schedule”, which is used either regularly or often by 69.3% of the companies. Most notably is the rare usage of cost and time indicators, which show, with 18.4% and 15.8%, the lowest values of regular usage and are never used by about 25% to 28% of the companies. This stands, to some extent, in contrast to the answers given regarding the organizations’ strategic objectives with productivity and cost transparency ranking relatively high. An example can be found in company G. After we introduced high quality process models as part of the BPM initiative, in the following workshop, also due to the rather small process size, a systematic measurement was easily derived.

To exploit the maximum possible benefit from BPM, it is crucial not only to collect measurements but also to establish a reporting system and analyse the data for active process controlling. For that purpose, it is necessary to regularly assess deviations from planned performance targets, identify and analyse their causes, and initiate corrective actions. 31.9% of the SMEs stated to be performing these tasks continually or very often whereas a similarly sized group made up of 38.9% of the companies never or only rarely do so. Altogether, 27.2% of the respondents have available a full or almost complete process reporting system whereas for a bigger share of companies (43.9%) reporting is either non-existent or only exists in a rudimentary form. A positive example in this regard was company E. They installed a completely automatized IT system for performance measurement providing regular performance reports to the management. This allowed a strict monitoring and quick reaction to occurring problems. Driving factors for this initiative were the available BPM capabilities and the management interest in BPM.

Regarding the domain of process improvement, we asked the SMEs if they regularly assessed the potential for improvement and actively search for measures to enhance existing processes. As the results show, most of them fall into the two categories of companies that do so occasionally (34.8%) or at frequent intervals (27.7%). At the top end, we found 10.7% of the SMEs having implemented a constant identification and assessment of process improvement possibilities. On the other hand, there is a considerable group of companies for which process improvement is not part of their BPM approach (15.2%) or only plays a minor role (11.6%). Comparing these results to other items shows a considerably lower number of companies that can rely on a previously conducted analysis for process improvement. For example, of those companies engaged in process improvement quite regularly, less than half of them (48.8%) have systematic data (e.g., target deviations, root cause analysis, etc.) available as a basis for taking decisions. Regarding the organizational integration, only 32.7% have put a designated process team in place, which is responsible for analysing process reports and initiating improvement activities (Figure 5). This clearly indicates that the majority of SMEs neglect a clear allocation of tasks as well as the corresponding competences, which are necessary to effectively carry out process improvement initiatives.

2.1.3.4 Overall evaluation

After interpreting the items in detail, we cover the overall results. We use the order of the categories to classify the companies (see Table 3). A company's class is the highest class the items of which have been fulfilled for at least 50%, due to the nature of the survey targeting the overall BPM. Probably, a company will not implement the whole set of measures even for the least important auxiliary business process. The classification shows the fulfilment of BPM measures while considering these conditions.

Figure 7 shows how many companies reached the respective category. Surprisingly, the majority of answers tend towards the borders. First, 46.1% of the companies reside in category zero. Hence, they reportedly document very little, if at all. Our case study notes show that company J is an example of this observation. Mostly, their employees are involved in the manufacturing process, which is determined by the manufacturing necessities. However, due to changing markets, the time and flexibility of their design and tender process (designing the customized component and calculating a tender for the production) came into focus. The company has about ten employees covering this process among others. By their own account, the existing process documentation written to acquire a certification is not related to the actual process. In fact, the involved personnel know their own tasks very well but only have an abstract notion of the overall process. Nonetheless, our case study notes show that, a process model covering the entire process reveals several possibilities to increase the flexibility and performance of the process. As a consequence, the head of the company considered modelling very useful. However, since the employees have neither the time nor the capabilities, they do not plan to conduct further BPM initiatives.

The second most frequented class, class 5 with 20.0% of the companies, includes companies that continuously optimize their processes and have implemented the measures from the previous classes as well. An example for class 5 is company E. While we were revising the process model for the product design, we realized that the company is highly process driven. They perform continuous monitoring and the feedback is used for optimization. Asked for the reasons of the high degree of BPM involvement, the head of quality referred to the complexity of their processes. The product design combines mechanical engineering, optical engineering (i.e. optical lenses and sensors) and software development as well as a large amount of external regulatory demands and engineering tools and processes for mass production. Since the company does not outsource any steps, they need BPM to deal with the immense complexity. While the company had already implemented most of the known BPM measures, they were still interested in further methods and techniques to increase the efficiency of their processes.

Third, about 19.1% of the companies reports' are classified as class 1, having a process documentation. Even though the example from class 0 showed that realistic process models are no mandatory requirement for a certification, the certification was the initial motivation for company G to document their processes. The models were further used to define the responsibilities of the employees and monitor their delivery. However, responsibilities and roles from a process perspective (e.g., process ownership etc.) were not defined, since that concept was not known to the company. The existing documentation was created with MS Visio flowcharts and e.g., routing constructs of modelling languages (e.g., OR, XOR, AND connectors) were omitted. We revised the models introducing a proper control flow and important milestones. Now using the model, the company showed great interest in introducing further BPM measures, especially a process performance measurement.

Last, only few companies fall into classes 2 to 4. Reflecting our previous experience and our notes, we found a possible explanation for this distribution. While class 2 can already be achieved with few measures (i.e., process documentation), achieving class 3 to 5 requires continuous efforts. BPM initiatives are either done with as little effort as possible, for which they seldom produce a proper documentation and remain at the initial level. This is sometimes the case if initiatives are conducted for certification purposes only, for example. However, when the initiatives are actually considered beneficial, e.g., due to complexity issues, a proper BPM with the necessary resources is installed. In this case, due to the small structures of the SMEs, the effort to implement the additional measures from class 4 and 5 is manageable. A closer look at class 5 reveals that the majority of companies score 50% to 65% of the items in class 5 (39.1% of the companies in class 5). A higher effort is assumingly not warranted for.

Our results differ strongly from the prior studies by (Harmon & Wolf 2014) and (Brucker-Kley et al. 2014; Minonne et al. 2011). There, companies were predominantly categorized to either level 2 (Harmon & Wolf 2014) or level 2-3 (Brucker-Kley et al. 2014; Minonne et al. 2011). We argue that this is due to vastly different methods and subjects, which make a comparison of the results very difficult. For example, Brucker-Kley et al. (2014) and Minonne et al. (2011) ask one single question only to identify the BPM maturity level of their subjects. We argue that this very abstract question (almost) encourages uncertain answers regressing to the mean. Harmon and Wolf (2014) conclude that level 2 is the dominant maturity level since, overall, the answer "occasionally" was the most frequent answer. In our study, each subject is classified individually based upon a large set of questions (see section 2.1.2.2.). Last, our study focuses SMEs in Bavaria, whereas the other studies have no such focus. In summary, the results of the two previous studies cannot be compared with ours in a meaningful way.

2.1.4 Discussion

The integration of the results of our survey and multiple case studies puts us in a position to evaluate the status quo of BPM adoption in Bavarian SMEs in general, and to identify motivations and reasons that help to explain the results observed. In this section, we discuss the main findings in a broader perspective. First of all, the overall results provide a divergent picture regarding the adoption of BPM in Bavarian SMEs. As highlighted in section 2.1.3.4, there is a notable cluster of companies that are clearly dedicated to BPM and have implemented most of the relevant BPM measures. This shows that it is indeed possible for SMEs to comprehensively adopt BPM. However, the vast majority reaches only lower levels of BPM adoption. It is an interesting question whether those companies do not see any benefit from adopting BPM given their concrete situation or if they are actually willing to adopt BPM but struggle with the realization for various reasons.

Thus, we discuss the issues regarding the adoption of BPM on the basis of our previous findings in more detail. We link these findings to possible reasons, and compare them to the requirements of the maturity levels. Hence, we are able to derive propositions, which enables practitioners to define next steps to possibly arrive at a higher maturity level or solve individual problems, and define requirements for BPM research focusing on SMEs. Table 4 gives and overview of the propositions. The columns in Table 4 show the observations, starting points and the derived propositions, and the rows are grouped to the domains that we identified among the observations: scientific foundation, strategy, implementation of BPM measures, and organizational embedding. In the following, each of these domains is explained.

	Observation	Reason/Starting Point	Propositions
Scientific Foundation	Established BPM approaches not implemented	Complexity of single BPM approaches; lack of BPM knowledge and manpower	Development of BPM approaches and trainings adapted for SMEs; usage of best practices
	Well-known modelling notations not used; missing quality requirements for documentation; self-designed graphical representation for processes visualization	Lack of modelling skills; lack of manpower; expensive BPM tools; missing awareness for the benefits	Development of and participation in BPM trainings; affordable BPM tools
Strategy	Mismatch of used performance indicators, process goals and strategic objectives	Lack of communication between management and employees; avoidance to measure performance	Consistent delineation of performance indicators from the strategy; establishment of a comprehensive measurement and reporting system; raise awareness for strategic benefits and of long-term planning
	Poor quality of process documentation due to external requirements (e.g., certifications)	Missing awareness for the benefits of process documentation and qualitative benefits of certification	Raise awareness for the qualitative benefits of certification; participation in modelling trainings; raise awareness for the benefits of process documentation; rigorous certification audits
	Development of business strategy on basis of BPM not performed;	BPM only performed on an operational level; strategic planning as a separate task; poor operationalization of process goals to performance indicators	Raise awareness for the contribution of BPM to strategic planning; participation in BPM trainings on methodological knowledge; usage of BPM for forecasting to review and adapt the strategy
Implementation of BPM measures	Selective or isolated implementation of BPM Measures	BPM is used to cover current needs; fulfillment of certain requirements; established BPM approaches not used; lack of manpower; no definition of responsibilities; poor communication/ information sharing	Participation in method trainings; awareness for benefits of a consistent, integrated BPM approach; creation of a holistic view on diverse BPM measures
	Focus on single business processes	Short-term and problem oriented focus; no process map showing interdependencies	Participation in method trainings; awareness for benefits of holistic BPM approaches; definition of responsibilities for a comprehensive BPM
Organizational embedding	Lack of anchoring BPM in the organization: -limited use of BPM measures for decision making -limited process improvement or process redesign possibilities -limited reporting of performance achievements	No clear role definitions; no employees with main topic BPM; not sufficient resources; lack of communication; lack of employee skills regarding BPM	Establishment of roles and provision of sufficient resources; organizational embedding of measurement and reporting system; communication of goals and achievements

Table 4: Derivation of propositions

A question worth paying attention to is whether SMEs draw on the broad **scientific foundation** that exists in the BPM domain. For example, do they utilize established concepts and approaches that have already proven their usefulness for the intended purposes? To our surprise, when we initially asked the case study participants which one of the manifold BPM approaches they use, all of them replied that they do not adhere to a special one. Rather, they developed a company-specific ad-hoc procedure that worked for their individual purposes. This, in turn, leads to problems such as an incomplete or inconsistent BPM implementation, which are reflected in the survey results, too. For example, we found companies extensively measuring the performance of their processes but never using the gathered data for process controlling activities (see section 2.1.3.3). The same is true for the documentation of business processes where well-known modelling languages (e.g., EPC, BPMN) are not used but, instead, self-designed graphical representations. Together with a missing awareness of quality requirements, this leads to a process documentation that is not appropriate for many BPM related topics. In summary, we found that, for SMEs in Bavaria, the orientation on existing BPM approaches and instruments is rather low. This may involve the danger that some extra effort is necessary for the implementation and that the resulting BPM is less effective in the end as common best practices are not exploited (see Table 4).

The interplay between BPM and business **strategy** is an interesting topic. This more prominent relation covers the contribution of business strategy for the definition of business and process goals. Our investigations reveal a mostly consistent derivation of process goals. However, the operationalization of those goals by means of performances indicators is not done consistently in many SMEs. As a result, the defined performance indicators are aligned to the process goals to a limited extent only and do not fully reflect the business strategy (see section 2.1.3.1 and 2.1.3.3). A possible reason disclosed in the case studies is a lack of communication, which is why employees are simply not aware of the strategic goals. Further, SMEs avoid monitoring the performance of their employees. In either case, the results indicate that SMEs do not use the potential of BPM to pursue long-term goals. They rather monitor their production to prevent deviations from schedule or quality problems, which may be subject to a contractual penalty. Interestingly, certifications (e.g., according to DIN EN ISO 9001) are often not considered as a chance to adopt BPM but are rather regarded as a duty, which has to be fulfilled in some way or other. We have observed that some companies hold the certificate, even though their process documentation was mostly not up-to-date and its quality on a low level. The potential of BPM for strategic purposes, such as processes for mobile business or in-/outsourcing decisions, is mostly overlooked by SMEs. They rather perform BPM on an operational level and regard strategic planning as a separate task. The missing linkage is reflected in e.g., the use of performance indicators being inconsistent to the business strategy (see section 2.1.3.3). Thus, the achievement of strategic objectives cannot be measured by means of BPM. Hence, we found that SMEs often lack an appropriate instrument to review and adapt their strategic focus. Another point is that the use of performance indicators does not only enable to measure the current performance of business processes, but also allows to rate the possible process performance in the near future by using techniques of mathematical forecasting and simulation (see Table 4).

Another issue relates to the **implementation of BPM measures** where we differentiate three aspects that we could observe in the course of our study: (I) the degree of fulfilment of BPM measures, (II) the consistent implementation across different categories, such as strategy, documentation, PPM, etc., and (III) the pervasiveness in the company with regard to complete process coverage. In general, we found that only a small group of SMEs adopted BPM measures to the full extent (see section 2.1.3.4). The majority selectively implements measures to fulfil a current demand. In this context, process documentation takes a special position as it is often introduced only to fulfil certain requirements for e.g., ISO certifications. We also found that BPM measures are inconsistent with each other since they were introduced in isolation without following a systematic approach. Hence, e.g., process targets do not match the strategic goals and extensively gathered data is never used for process controlling (see section 2.1.3.3). The main reason for that is the absence of an employee who is solely responsible for BPM tasks. Further, poor communication and information sharing encourages the emergence of isolated measures. With regard to pervasiveness and process coverage, our results show that SMEs in Bavaria mostly focus on single processes when implementing BPM measures (see section 2.1.3.2 and 2.1.3.3). Though it might be a sensible approach to focus the efforts on important key processes, an over-excessive concentration may cause problems. Since we found that most companies do not describe the interdependencies among their processes (e.g., by depicting them in process maps), mismanagement and high coordination efforts are the consequences. As a general view on business processes is not available for managers, overall management control and alignment to strategy becomes difficult (see Table 4).

A further issue that we found important in the SME domain is the missing organizational embedding of BPM. Many companies do not provide sufficient resources, first and foremost staff, for BPM activities, and clear definitions of roles (e.g., process owner etc.) are also scarce (see section 2.1.3.2). Whereas in large companies there usually are positions dedicated to BPM topics, smaller-sized companies of focus on operational daily business. During all of our on-site visits, we never met any employees whose main task was BPM. Mostly, it was the quality manager who had been assigned the additional responsibility for this topic, and only a small number of SMEs have process teams to discuss problems and develop measures for improvement. In case BPM standards or requirements are defined, they are poorly communicated in most SMEs (see section 2.1.3.3). As a consequence, e.g., process goals defined by the management are unknown to those employees working in the respective processes. On the other hand, process reporting, which is supposed to provide decision makers with relevant data (e.g., process performance measures), is poorly implemented, too. This may cause wrong decisions both at the operational and the strategic levels. Another problem SMEs struggle with is the lack of BPM knowledge and qualified personnel. In particular, we found the quality of the process documentation at a rather low level. Other companies having successfully implemented a PPM do not succeed in drawing the right conclusions from it as they were not trained in redesigning business processes (see Table 4).

The deficits are also a great challenge for scientists since all the itemized problems can be supported by methods, techniques and tools that have already been available for a long time. Further, there is a tremendous amount of scientific literature in which, mostly based on the design science research method, the development and the evaluation of these BPM methods, techniques and tools are described. The fact that many practitioners do not use them, although they are available and have proven to be useful according to the relevant literature, indicates either a knowledge gap or a lack of willingness. The latter contradicts the contribution described in scientific BPM literature and can be interpreted as a request to scientists to develop new or update existing methods, techniques and tools that are more suitable for everyday routines of SMEs. We see our research as a starting point to investigate the usefulness of the available BPM methods, techniques and tools and to bring them more in line with the needs of SMEs. The former, namely the knowledge gap, motivates to develop further possibilities to train managers in BPM to close the said gap. However, it has to be considered that managers nowadays are confronted with a tremendous amount of BPM courses with different emphases and in different learning settings. Therefore, the reasons of this knowledge gap should be closely investigated, and new training possibilities need to be developed that are explicitly in line with the needs of the employees in SMEs.

2.1.5 Conclusion

In this paper, we assessed the state of BPM adoption in Bavarian SMEs. A mixed method approach combines the results of 10 in-depth case studies with 114 responses of a survey. Considered together, they uncover the state of adoption regarding measures about BPM and strategy, purpose achievement and reliability, documentation, capabilities, performance management and redesign. This assessment shows potentials for improvement left untapped, together with possible reasons.

On the base of the identified reasons, we developed our main contribution, next to the results of the survey and the case studies (research question 1 and 2), namely propositions for practitioners and propositions for researchers (research questions 3). These propositions expectedly improve the BPM adoption and thus support the competitiveness of SMEs. In this regard, we answered the three research questions defined in the introduction (a summary of which is shown in Table 4).

Still, our work is not without limitations. First, the interpretation of the case studies, though conducted by two researchers and discussed in a group of four, leaves room for subjectivity. They have to be cross-verified by the results of the survey. Another limitation originates from the selection of participants. As they were contacted at random, it is possible that only those companies responded that had a particular interest in implementing BPM measures. In the light of the previous discussion, companies without any interest in BPM at all might contribute to the size of category 0 (see section 2.1.3.4).

Results and limitations lead to further research. First, the study was conducted in Bavaria and needs to be extended to structurally different regions. In that context, further case studies may sustain or contradict the present results. Last, in future work, we will implement and evaluate the propositions in order to verify their relevance and extend their number.

2.2 Beitrag 2: Testing the Impact of Wand and Weber‘s Decomposition Model on Process Model Understandability

Adressierte Forschungsfrage	**Forschungsfrage 2**: Haben Verstöße gegen die Good Decomposition Conditions nach (Johannsen & Leist 2012b) einen Einfluss auf die Verständlichkeit von Prozessmodellen?
Erscheinungsort	35th International Conference on Information Systems, ICIS 2014, Auckland, New Zealand, December 14-17, 2014 (VHB Jourqual 3: A)
Autoren	Dr. Florian Johannsen 70 % Daniel Braunnagel 15 % Prof. Susanne Leist 15 %
Dieser Beitrag behandelt ein Laborexperiment, in dem die Entwicklung der GDC mit einer besseren Verständlichkeit von Prozessmodellen einhergeht. Die Nullhypothesen für den Einfluss von persönlichen und modellbezogenen Faktoren können erfolgreich abgelehnt werden. Weiter zeigen Verstöße gegen die Conditions einen statistisch signifikanten Einfluss auf die wahrgenommene Verständlichkeit der und die benötigte Zeit für die Problemlösung und einen teilweise signifikanten Einfluss auf die Güte der Antworten. Der Beitrag lässt sich im DS Entwicklungszyklus der GDC dem letzten Schritt, der Evaluation zuordnen. Er ist damit auch Voraussetzung für die Entwicklung der GDC Metriken. Eine besondere Herausforderung dieses Beitrages ist, dass sich die GDC auf die Semantik von Konsensmodellen beziehen. Für das Experiment muss also ein realitätsnaher Sachverhalt konstruiert und gleichzeitig ausgeschlossen werden, dass Hintergrundwissen der Subjekte das Experiment beeinflusst. In der Folge kommt den validitätssichernden Maßnahmen besondere Wichtigkeit zu. Bspw. werden bezüglich der Messinstrumente Wert auf die Multidimensionalität der Instrumente (vgl. (Houy et al. 2012) gelegt und, soweit möglich, auf bewährte Instrumente zurückgegriffen (vgl. (Burton-Jones & Meso 2006), in der Literatur auffindbare Einflussfaktoren berücksichtigt und die statistische Validität hinterfragt, denn gerade die Konstruktvalidität ist in der Forschung zur Prozessmodellverständlichkeit häufig kritikwürdig (Houy et al. 2012).	

Tabelle 5: Bibliographische Angaben zu Beitrag 2

Process modeling is becoming increasingly important for business process management initiatives. However, for being able to exploit the benefits associated with business process modeling the process models need to be understandable for its users. It has been shown in literature that the growing size of a process model has a negative effect on its understandability. Decomposition is a means for splitting large process models into smaller subprocess models to increase the understandability. However, it is still unclear what properties actually characterize a good decomposition, while generally accepted guidelines on decomposing process models are missing. We analyze Wand and Weber's good decomposition model as an approach for creating decomposed process models that are easy to understand. The paper at hand investigates in how far the decomposition conditions influence the understandability of a process model. Using an experiment we show that the decomposition conditions actually do have a positive influence on model understandability.

2.2.1 Introduction

Business process modeling is a crucial task in business reorganization projects (Becker et al. 2010; Mendling et al. 2010). It captures employees' process knowledge to be used for entrepreneurial initiatives (Becker et al. 2010). Business process models enable decisions on IT investments, reorganizations or the implementation of information systems (IS) for example (Becker et al. 2010) and thus support the alignment of a company's IT infrastructure with the business strategy (Branco et al. 2014). The proper documentation of processes as models is seen as a central success factor for the development of information systems (Aguilar-Savén 2004; Mendling et al. 2012a). However, process modeling is a highly subjective task (Schütte & Rotthowe 1998), and process model quality remains a fuzzy topic (Fettke et al. 2012). Several approaches were taken to assess process model quality, such as the development of quality metrics (cf. Gruhn & Laue 2007; Mendling 2008; Vanderfeesten et al. 2007a) or quality frameworks for example (cf. Krogstie et al. 2006; Overhage et al. 2012). In the light of this discussion, model understandability has been established as a key quality criterion (Houy et al. 2012). Model understandability refers to "the degree to which information contained in a process model can be easily understood by a reader of that model" (Reijers & Mendling 2011, S. 451). It is widely acknowledged, that the understandability of process models is of high importance to exploit the benefits associated with process modeling (e.g., process analysis)

(Houy et al. 2012; Mendling et al. 2010). In addition, the larger a process model gets in size the harder it is to understand (Mendling et al. 2010; Zugal et al. 2011).

Literature introduces “decomposition” or “modularization” of process models as a way to improve the understandability of a model (cf. Reijers & Mendling 2011; Zugal et al. 2013). Thereby, a business process is decomposed into several subprocesses which are depicted as corresponding subprocess models. They are assigned to certain levels of a hierarchy differing in detail (Davis & Brabänder 2007; Reijers et al. 2011). However, the lack of guidelines supporting a modeler in delineating subprocesses and assigning models to hierarchy levels is problematic (Davis & Brabänder 2007; Reijers et al. 2011). As a result, decomposition or modularization is often reached in an “ad hoc fashion” (Reijers et al. 2011). It is still unclear which properties characterize a “good” decomposition in business process modeling. Metrics that have gained general acceptance and focus on the quality of a decomposed business process model are still missing (Reijers et al. 2011).

We intend to provide a better understanding for the quality of decomposed process models by using Wand and Weber’s decomposition model (cf. Wand & Weber 1989b; Weber 1997). It is our aim to find out whether the adherence of a decomposed process model to the decomposition conditions positively influences its understandability or not. In this paper we focus on Event-driven Process Chains (EPCs) (cf. Scheer et al. 2005), which is one of the most widespread process modeling languages (Reijers & Mendling 2011), to provide an adequate level of detail. We pose the following research question: *Is a decomposed process model (using the EPC notation) adhering to Wand and Weber’s decomposition conditions easier to understand than an equivalent decomposed process model violating the decomposition conditions?*

To answer this question we conducted an experiment, which is acknowledged as a promising approach to get epistemological insights in business process management (BPM) research (Houy et al. 2010). In case of a positive effect, guidelines for decomposing process models can be derived in future work to support business analysts in designing manageable process models.

The paper is structured as follows: First, basics on model understandability and decomposition are described. Afterwards, we substantiate the relevance of the decomposition model for the research at hand and present our interpretation of the decomposition conditions for the domain of business process modeling. Subsequently, the design of our investigation is introduced. In the main part of this paper, we describe the results of our experiment and interpret them in a successive step. We close with limitations and an outlook on further research.

2.2.2 Basics and Theoretical Foundations

2.2.2.1 Event-driven Process Chains

Event-driven Process Chains (EPCs) have gained significant attention in the field of business process modeling with user acceptance and tool support underpinning this development (Mendling 2008). It is currently one of the most frequently used modeling languages (Adam et al. 2013; Reijers & Mendling 2011). EPCs can be understood as a set of event types, function types and connector types (Mendling et al. 2010; Scheer et al. 2005). They are part of the ARIS (Architecture of Integrated Information Systems) framework which provides additional perspectives on a process, e.g., the organizational view, data view or output view (Scheer et al. 2005). In that context, the term “enhanced” Event-driven Process Chain (eEPC) is used. In this paper we solely focus on the eEPC notation in a first step since the interpretation of the decomposition conditions (cf. Wand & Weber 1989b; Weber 1997) varies for different modeling languages due to their heterogeneous ontological expressiveness (cf. Recker et al. 2009).

2.2.2.2 Understandability and Decomposition in Business Process Modeling

Process model understandability is a topic of lively discussion. But still, the topic is not fully understood yet (Houy et al. 2014; Zugal 2013). According to Vanderfeesten et al. (2007a), “understandability” is closely related to “complexity”. Decreasing the complexity of a process model thus improves its understandability. In literature, e.g., the visual layout of a model, its size, the modeling notation used, users’ knowledge concerning process modeling or the problem domain considered are mentioned as factors that influence the perceived understandability of a model (cf. Mendling et al. 2007; Reijers & Mendling 2011). Further, quality frameworks and guidelines for process modeling provide criteria (e.g., completeness) or rules (e.g., avoidance of OR-connectors) which support the design of easy-to-understand models (cf. Krogstie et al. 2006; Mendling et al. 2010; Overhage et al. 2012). For example, the seven process modeling guidelines (7 PMG) represent recommendations on how to design a process model properly, focusing on structural aspects (e.g., use of one start and one end event only) as well as an appropriate labeling of model elements (e.g., use of verb-object activity labels) (cf. Mendling et al. 2010). Additionally, more pragmatic suggestions for model design can be found (cf. Pieterse 2005; Silver 2008), which were developed for practitioners in special (Mendling et al. 2010). Beyond that, Weber et al. (2011) introduce generally valid refactoring opportunities to improve the understandability of a process model. Moreover, the process of model creation is examined more closely in that context (cf. Pinggera et al. 2013). Furthermore, the modeling purpose needs to be considered when dealing with understandability of process models (Fettke et al. 2012; Reijers & Mendling 2011). Typical modeling purposes are, e.g., the creation of transparency concerning business processes, process documentation (e.g., for ISO 9000 certification), process analysis, or the design of information systems (Jeanneret et al. 2012; Koch 2011). Depending on the purpose, not all elements of a model may be necessary (cf. Bobrik et al. 2007). This is determined by the so called “model usage” (Jeanneret et al. 2011). For example, software engineers are usually interested in details of a business process (e.g., information flow between activities), whereas a call center employee may only be interested in a very abstract view on the customer retention process.

Decomposition helps to provide different levels of abstraction for a process model for heterogeneous model users (e.g., software engineer, call center employee). Thus, they may focus on specific model levels in a level hierarchy to find the information sought after (Bobrik et al. 2007; Davis 2008). Certain information can be aggregated (model abstraction) or placed in corresponding subprocess models (model fragmentation) (Lara et al. 2013; Zugal et al. 2013).

Several process modeling languages as well as software tools (e.g., ARIS Business Architect) support model abstraction and fragmentation. This is achieved by introducing modeling constructs such as “collapsed subprocesses” for Business Process Diagrams (BPDs) (cf. OMG 2011), or “hierarchical functions” for EPCs (cf. Davis & Brabänder 2007).

In general however, different forms of decomposition exist while generally accepted guidelines for doing so are missing. For example, the business analyst can design a high-level process model and subsequently use subprocess models to provide more detailed information (Davis & Brabänder 2007). Alternatively, the business analyst might create a detailed “flat” model from scratch. In the following steps, more abstract views can be generated by delineating corresponding subprocess models (Davis & Brabänder 2007). In both cases, the subprocess models are assigned to model levels of a hierarchy[3]. Nevertheless, decomposing in an ill-considered manner may affect model understandability in a negative way (cf. Zugal et al. 2011).

Literature introduces some approaches supporting a modeler in decomposing process models but does not give a recommendation which one should be used regarding a specific project situation. For example, during “block-structuring” blocks are built around the nodes of a “flat” process model (Reijers et al. 2011). The blocks denote subprocesses. The blocks are supposed to have one incoming control-flow arc and one control-flow arc leaving the block (“single-entry-single-exit (SESE) component”) (Gerth 2013). Furthermore, strongly connected nodes can be clustered to form a subprocess (“graph clustering”) (Reijers et al. 2011). When business process models are “disaggregated” (hierarchically decomposed), tasks are described by using subprocess models assigned to certain levels of a hierarchy (Davis & Brabänder 2007; La Rosa et al. 2011; Malone et al. 1999). During specialization (horizontal segmentation/modularization) smaller subprocess models (having the same abstraction level) are derived from an original model (Davis & Brabänder 2007; La Rosa et al. 2011; Malone et al. 1999). Process model abstraction produces more abstract views on a model by aggregating activities of a lower level (Smirnov et al. 2011). De Lara et al. (2013) focus on the metamodels of modeling languages to derive generally usable abstraction operations for example. Further, Sharp and McDermott (2009) present a heuristic enforcing a particular structure from a practitioner’s perspective. Ma et al. (2015) develop an algorithm for an automatic decomposition of models which are transformed into corresponding graphs. An approach for decomposing Petri Net models by partitioning its edges is provided by van der Aalst (2013).

3 An illustrative example for a decomposed process model is available at: http://wi3.uni-regensburg.de/fileadmin/user_upload/Questionnaire.pdf
Ein Abdruck des verlinkten PDF findet sich in Anhang 1

Despite these suggestions which support a business analyst in decomposing process models, the quality of a decomposition remains a fuzzy topic (Reijers et al. 2011; Zugal et al. 2013). It is still an open question, which properties characterize a “good” decomposition and how the quality of a decomposed model can be adequately measured (e.g., by means of metrics). Using existing process model quality metrics for that purpose is problematic, since these have been developed for “flat” models (cf. Braunnagel & Johannsen 2013; Cardoso 2006; Gruhn & Laue 2007; Mendling 2008; Mendling et al. 2010). For example, Cardoso (2006) developed a metric for the complexity of control-flows by analyzing different types of split-operations (AND, OR, XOR). Additionally, graph-based complexity metrics exist (Latva-Koivisto 2001). Mendling et al. (2012a) specified existing process model metrics with corresponding thresholds to reduce a model’s error probability and increase its understandability that way. Jung et al. (2011) developed an entropy-based measure to analyze the uncertainty of performing certain tasks within a business process during runtime.

Nevertheless, such metrics can only be used on single subprocess models and the results need to be aggregated to consider the “holistic” model including all model levels. It is strongly doubted that this procedure reflects the quality of a decomposition (Reijers et al. 2011), since this approach contradicts the original idea of those metrics. Quality metrics, that focus decomposed process models in special, still need to be developed (Reijers et al. 2011; Vanderfeesten et al. 2007a). It seems feasible to derive criteria or metrics for a good decomposition from the decomposition approaches described above. An example would be the number of “SESE components” that could be derived for a process model in the “block-structuring” approach. Nevertheless, these metrics or criteria would be closely related to a specific decomposition approach, as introduced. A general means for judging the quality of a decomposition, independent from a certain decomposition approach, has not been developed yet.

In summary, decomposition of process models is seen as an effective way of reducing complexity and enhancing understandability, since it leads to smaller, more manageable subprocess models (Mendling et al. 2010; Reijers et al. 2011; Zugal et al. 2013). However, generally accepted guidelines for decomposing process models and means for measuring the quality of the decomposition are missing (Reijers et al. 2011). Therefore, different approaches are used by practitioners to delineate and organize process models (Malinova et al. 2013), often based on “ad hoc” decisions (Reijers et al. 2011). Consequently, the “split-attention” effect may outweigh the benefits of decomposition in a modeling project (Zugal et al. 2011). A better understanding of what constitutes a “good” decomposition is thus required to avoid such drawbacks and to truly profit from the benefits (e.g., reduced complexity) of decomposing large process models.

2.2.3 The Decomposition Model of Wand and Weber

2.2.3.1 Relevance of the Decomposition Model

A generally accepted theory on “good” decompositions for business process modeling does not exist yet (cf. Houy et al. 2014). We thus analyze whether the decomposition model of Wand and Weber, originally developed for information systems, can be used for judging the quality of decomposed process models and for deriving guidelines on how to decompose properly in a subsequent step. The reasons for choosing the decomposition model are the following: the decomposition model is based on the BWW (Bunge-Wand-Weber) representation model which serves as a theoretical foundation for the ontological analysis of modeling languages (cf. Green & Rosemann 2000; Recker et al. 2005; Recker et al. 2009; Recker et al. 2011). Thus, the BWW ontology is an established concept for evaluating process modeling languages. In addition, the BWW ontology has a wide applicability and a comprehensive scope (Recker et al. 2005) and can therefore be a useful foundation for the evaluation of decomposed process models. Further, the decomposition model has already been successfully used for judging the quality of models in object-oriented modeling (cf. Burton-Jones & Meso 2008). Recker et al. (2009) consider the decomposition model to be promising for supporting a business analyst in large-scale business process modeling initiatives as well. Smaller, more manageable subprocesses can be derived with the help of the decomposition conditions (Recker et al. 2009). However, an analysis of the decomposition conditions regarding the understandability of process models has not been done yet. Because of the reasons mentioned, we build on the BWW ontology, a well-known theory in information systems research, to provide a theoretical foundation for decomposing process models.

2.2.3.2 The Decomposition Model and its Specification for eEPCs

The decomposition model is part of the BWW ontological models, comprising the representation model, the state-tracking model and the decomposition model (Rosemann & Green 2002; Weber 1997). The decomposition model itself builds on constructs introduced in the representation model. The representation model presents ontological constructs needed for specifying the behavior of an information system (Weber 1997). The key construct is a "thing" (e.g., human) (Rosemann & Green 2002; Weber 1997). The "thing" is characterized by properties (e.g., hair color) represented by attributes (Weber 1997). "Things" are grouped to form a system which can be broken-down into several subsystems. A "thing" experiences several changes throughout its lifetime documented in its history (Weber 1997). Further details can be found in Weber (1997). The decomposition model presents five conditions which characterize a "good" decomposition (cf. Weber 1997): (1) minimality, (2) determinism, (3) losslessness, (4) minimum coupling and (5) strong cohesion.

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, S. 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, S. 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, S. 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, S. 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, S. 42; Weber 1997, S. 163)

Table 6: The decomposition conditions

For using the decomposition conditions in the context of business process modeling they need to be interpreted for that area of application. As mentioned, we focus on eEPCs in our research. For deriving an interpretation of the conditions we build on the results of an ontological analysis by Green and Rosemann (2000). This ontological analysis of eEPCs helps to decide in how far specific constructs of the BWW ontology can be expressed by corresponding modeling constructs (cf. Green & Rosemann 2000). In the following, we shortly introduce a specification of these conditions for eEPCs (see Table 7) whereas an embracing description and derivation based on a representational mapping can be found in a previous work of ours (cf. Johannsen & Leist 2012b).

Minimality: A business process model can be decomposed into subprocess models which are assigned to certain model levels of a “level structure” (Davis & Brabänder 2007; Reijers et al. 2011). “State variables” (Weber 1997) are represented as attributes of data objects – involved in a business process – and are generally not redundant, when there is a change of value for those attributes. A “state” (Weber 1997) for these attributes is described by event types in an eEPC (Hoffmann et al. 1993; Keller et al. 1992). In addition, event types that do not refer to data objects but specify the behavior of an information system (e.g., system locked) can be required for an eEPC model (Hoffmann et al. 1993).

Determinism: “Internal events” that occur within a system are to be well-defined (cf. Weber 1997). Regarding process modeling with eEPCs, the business analyst is thus expected to avoid OR-connectors and use an unambiguous labeling of the modeling constructs (cf. Mendling et al. 2010; van der Aalst et al. 2002). “External events” are modeled as start event types within an eEPC (Green & Rosemann 2000). That way, subprocess models can be delineated that are triggered by its environment (e.g., arrival of an order).

Losslessness: Hereditary as well as emergent properties are supposed to be preserved during decomposition (cf. Weber 1997). However, the eEPC lacks corresponding representation mechanisms (Green & Rosemann 2000; Recker et al. 2009). Weber (1997) mitigates this circumstance by demanding “not to lose properties” in general. Thus, event types that are related to non-redundant state variables (see minimality) need to be preserved in the decomposition.

In addition, constructs of other ARIS views that are associated to these event types must be found in the decomposed process model as well.

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 modeling constructs from other ARIS views (e.g., of the organizational view, data view, output view) related to these must be reflected upon their necessity regarding this fact as well.
Determinism regarding eEPCs	To fulfill 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 labeled appropriately.
Losslessness regarding eEPCs	No information must get lost during decomposition. Event types being related to attributes describing properties are of central importance and should be preserved. Function types associated to these together with modeling constructs from other ARIS views must be considered as well, whether they can be found in corresponding subprocess models of a decomposition or not.
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.

Table 7: The decomposition conditions specified for eEPCs (Johannsen & Leist 2012b)

Minimum coupling: “Input” expresses those states that arise within a system due to actions of the environment (Weber 1997). “Coupling” indicates that “things” are associated with each other (Weber 1997). In the context of business process modeling, coupling exists within each subprocess model but also between subprocess models (Braunnagel & Johannsen 2013). In the field of decomposition, the latter form of coupling is relevant, whereas only the coupling concept as introduced by Vanderfeesten et al. (2008c) for the workflow domain meets the primary ideas of Wand and Weber. Therefore, the interaction between subprocesses (cf. Vanderfeesten et al. 2008c) of a decomposition should be minimized. Two subprocess models are “coupled” in case the output of a function type in a subprocess model is input to a function type in another subprocess at the same time. Further, the number of start event types must be as minimal as possible to minimize the “total action of all environmental things on each subsystem in the decomposition” (Weber 1997, S. 159).

Strong cohesion: There are neither clear instructions on how to model “cohesion” nor on how to express “output” in eEPCs (cf. Green & Rosemann 2000). Not much work can be found dealing with cohesion in business process modeling. An exception is Vanderfeesten et al. (2008c) who focus on data objects. Therefore, we state that strong cohesion is given for a subprocess, if all function types that transform a set of input to output are captured within a subprocess. The input as well as output is represented by data object types of an eEPC model. The subprocesses are to be delineated in a way that the data object types representing input within a subprocess cannot be found in any other subprocess on that model level as input.

Table 7 summarizes the interpretation of the decomposition conditions for eEPCs once again (cf. Johannsen & Leist 2012b). Further on, we evaluate whether the coherence with these conditions has a positive effect on the understandability of a decomposed eEPC process model or not.

2.2.4 Experimental Design for Testing the Decomposition Conditions

Laboratory experiments are an established means for testing theories in IS research (Wohlin et al. 2012) and they are increasingly used in the field of BPM as well (Houy et al. 2010). They enable the systematic search for relations between variables and dependent measures within a controlled environment (Burton-Jones & Meso 2002; Sarshar & Loos 2005; Wilde & Hess 2007). In the following, a laboratory experiment is described that analyzes differences in model understandability for alternative decompositions of a process model. The experiment was conducted in accordance with the process as shown by Wohlin et al. (2012). The objective of the experiment was to test, in how far the adherence of a decomposed eEPC model to the decomposition conditions influences model understandability. Focusing on the decomposition model as a potential means for creating “good” decompositions that are easy to understand, we propose that a decomposed process model adhering to the conditions is considered to be more understandable than a decomposed model violating the decomposition conditions.

2.2.4.1 Scope, Context, Research Model and Hypotheses

In accordance with Wohlin et al. (2012), we briefly introduce the scope of the experiment, its context, the research model and the hypotheses.

Scope: The object of the study were participants of a “business process modeling and process reengineering” course and their problem solving abilities respectively their judgment on model understandability regarding decomposed process models. We evaluated whether violating the decomposition conditions significantly influences model understandability of a decomposed process model or not. We took the perspective of researchers, investigating the benefits of the decomposition conditions for designing decomposed process models that are easy to understand. The main effect studied was the model understandability, whereas the “problem solving abilities” and the “perceived ease of understanding” were emphasized in special. The experiment comprised 53 students of the aforementioned course at a German university. The students could be assumed being novices in process modeling (cf. Recker et al. 2012) at that stage. In their perception, model understandability was a decisive aspect of the process modeling discipline which made them eligible candidates for the experiment. The materials of the experiment (decomposed process models) were randomly assigned to the participants.

Context: The experiment was run off-line (and thus not in an industrial modeling project) (Wohlin et al. 2012), conducted with bachelor degree students (in their third year at university). The experiment dealt with a real problem often encountered in large modeling projects, namely the decomposition of process models. The experiment focused on eEPCs as a modeling language only.

Research model and hypotheses: The focus of this work is on the decomposition conditions and their influence on model understandability. Currently, there is no dominating or commonly accepted theory in the field of process model understandability research (Houy et al. 2014). Consequently, an established theory serving as a base for the definition of a research model to test the decomposition conditions cannot be referred to. Therefore, we draw upon insights from experimental studies and meta-analyses to identify factors influencing model understandability (e.g., Houy et al. 2012; Reijers et al. 2011). Based on these, a research model is derived for this investigation.

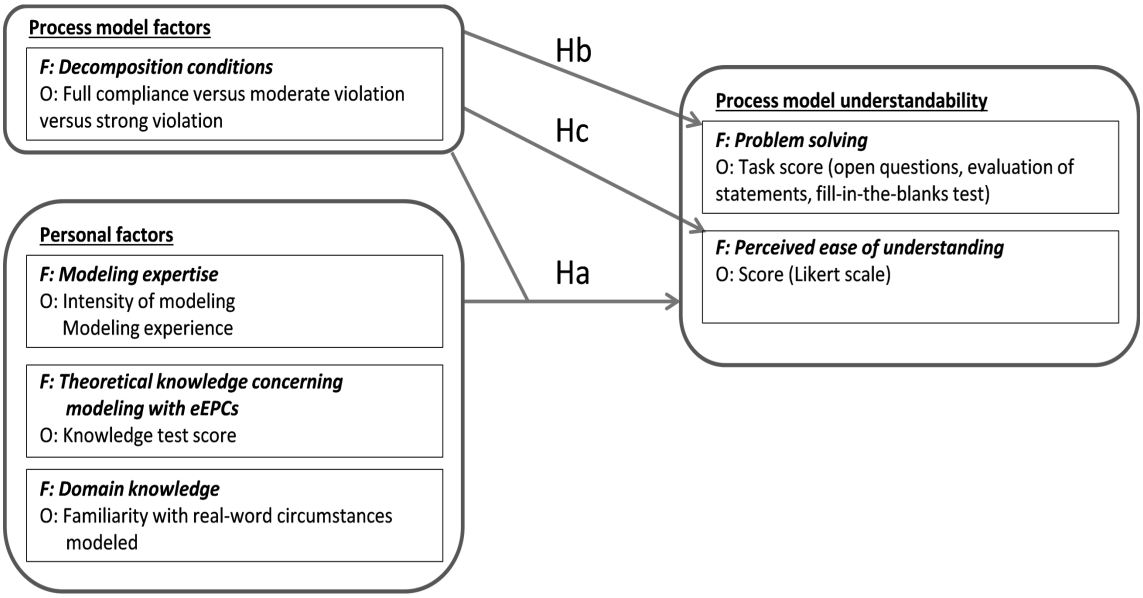


Figure 8: Research Model

According to recent research on model understandability, one has to differentiate between "process model factors" and "personal factors" affecting a modeler's perception (cf. Mendling et al. 2012b). Whereas process model factors refer to the process model and its characteristics (e.g., size), personal factors relate to the reader of the model and may comprise theoretical knowledge concerning process modeling concepts or user's domain knowledge for example (Khatri et al. 2006; Mendling et al. 2012b; Reijers & Mendling 2011). We build on this differentiation between process model factors as well as personal factors and derive the research model as depicted in Figure 8. The operationalization of these factors is described in the section "Materials of the Experiment".

As mentioned, the process model factors focus on the process model respectively its design (Mendling et al. 2012b). The decomposition conditions are the process model factors to be investigated. It is our purpose to find out in how far the adherence to the decomposition conditions positively influences the understandability of a process model. Therefore, the decomposition conditions are the key factor in our experiment. However, considering recent studies, further process model factors, such as the layout, the representation of modeling concepts via graphical elements or the labeling of model elements exist (cf. La Rosa et al. 2011; Reijers & Mendling 2011). Those factors are directed at the "concrete syntax" of a process model (La Rosa et al. 2011) while their influence is not investigated in this experiment. Therefore, their impact has to be minimized in our experimental setting. The same holds true for the aspects “modeling notation”, “model purpose” and “size” (Mendling et al. 2007; Reijers & Mendling 2011) potentially threatening the validity of the experiment. We address these aspects by the design of the materials for the experiment as described in the section “Materials of the Experiment”.

In addition, personal factors are to be considered (see Figure 8). They refer to the model user (Mendling et al. 2012b). We differentiate between “modeling expertise”, “theoretical knowledge concerning modeling with eEPCs” and “domain knowledge” (cf. Khatri et al. 2006; Mendling et al. 2012b). The modeling expertise refers to the frequency a user deals with process models in everyday life (e.g., every day, once a week) and the point in time a user has encountered process models for the first time (cf. Mendling et al. 2012b). In addition, users may have a different theoretical knowledge concerning the process modeling discipline (e.g., the correct modeling of cycles in eEPC models) which affects their perception of a model (cf. Mendling et al. 2012b). Further, the domain knowledge of a user must be considered. A user with deep knowledge regarding the real-world situation may have less problems understanding the corresponding model than a user unfamiliar with the domain (cf. Khatri et al. 2006).

Process model understandability addresses the degree, to which a model reader is able to easily understand the information captured in the process model (Reijers & Mendling 2011). For measuring model understandability, a reference framework has been introduced by Houy et al. (2012), differentiating between objective and subjective dimensions. Regarding an objective measurement, Burton-Jones and Meso (2002) emphasize that understanding is best to be judged by a “deep processing” approach like “problem solving” (cf. Gemino & Wand 2003; Mayer 1989). This enables to check the “deeper-level understanding” of participants (Bodart et al. 2001). In our context that means, participants should be able to solve tasks requiring a deeper analysis of the process model respectively subprocess models. This can be tested by using problem solving questions (cf. Bodart et al. 2001; Burton-Jones & Meso 2006; Houy et al. 2012). In our research model, the objective measures for assessing model understandability are coded into the dependent variable “problem solving”.

Process model perception is also a matter of subjectivity (Houy et al. 2012). This is to be considered by the "perceived ease of understanding" (Burton-Jones & Meso 2002; Davis 1989; Houy et al. 2012) which is a dependent variable of our experiment as well. Both variables "problem solving" and "perceived ease of understanding" are used to determine "process model understandability".

Based on this research model and considering the objective of the experiment as described, we pursue the analysis of the following three relationships:

- **a**: The influence of the process model factors and the personal factors on the process model understandability.
- **b**: The influence of the process model factors on the problem solving capabilities.
- **c**: The influence of the process model factors on the perceived ease of understanding.

To support the statistical validity of our conclusions (cf. Shadish 2002) we operationalize the relationships "a-c" by the following hypotheses:

- **H_0a**: The process model factors and the personal factors have no influence on the process model understandability.
- **H_1a**: The process model factors and the personal factors have influence on the process model understandability.

- **H_0b:** The process model factors have no influence on the problem solving.
- **H_1b**: The process model factors have influence on the problem solving.

- **H_0c**: The process model factors have no influence on the perceived ease of understanding.
- **H_1c**: The process model factors have influence on the perceived ease of understanding.

Hypothesis "a" tests our overall model as preliminary before we focus on the topic of our work, the influence of the decomposition conditions on the problem solving capability and the perceived ease of understanding. In our experiment, the decomposition conditions were the variable to be modified to test their influence on process model understandability (see following section).

2.2.4.2 Materials of the Experiment

In the following, we describe the material for operationalizing the variables (see Figure 8).[4] In order to analyze the influence of the decomposition conditions (process model factors) on process model understandability, three alternative decompositions of a business process model (alternative A, B and C), describing one real-world situation, were created. The model considered was the "student enrollment process" at a German university, stemming from a modeling project to document the processes of the administration department. Therefore, the process model used has been created for documentation purposes using the ARIS Business Architect and the eEPC. In comparison to other modeling languages, such as Petri Nets, the eEPC is said to be easy to understand (Reijers & Mendling 2011; Sarshar & Loos 2005). In the course "business process modeling and process reengineering" – where the participants came from— a strong emphasis was given to modeling with eEPCs and the use of the ARIS Business Architect as a software tool. The influence of the modeling language on our experiment can thus be neglected.

Alternative A fulfilled all the decomposition conditions (see Table 7). Alternative B showed moderate violations by violating the "minimality", "determinism" and "losslessness" condition (cf. Burton-Jones & Meso 2002). We chose these three conditions because they are less focused on the structure of the decomposed model (delineation of subprocesses) than the "minimum coupling" and "strong cohesion" condition. Thus it was interesting to see in how far differences on the understandability regarding alternative A and alternative B existed. In a third alternative C, all five conditions (see Table 7) were violated. The manipulation of the model was done by two researchers.

All alternatives (A, B and C) comprised four model levels (level 0—level 3) similarly. The number of subprocess models did not vary greatly between the alternatives as well. In total,

4 The questionnaire as well as the decomposed model of alternative A can be downloaded at: http://wi3.uni-regensburg.de/fileadmin/user_upload/Questionnaire.pdf
Ein Abdruck des verlinkten PDF findet sich in Anhang 1

alternatives A and B each comprised eight subprocess models whereas alternative C had ten subprocess models. This difference resulted from the application of the decomposition conditions, since some subprocess models needed to be merged to stick to the minimum coupling condition (alternatives A and B). Careful attention was given to the number of modeling elements for each subprocess model to eliminate the influence of “size” on model understandability (cf. Mendling et al. 2007). All of the subprocess models of alternative A had less than 50 modeling elements as proposed by Mendling et al. (2010). In alternative B, seven out of eight subprocess models, and nine out of ten subprocess models in alternative C had less than 50 modeling elements. The largest subprocess model in alternative A held 45 modeling elements, 59 elements in alternative B and 55 elements in alternative C. The smallest subprocess models in alternatives A and B comprised nine elements each. In alternative C it consisted of five elements. Regarding these figures, “size” played a negligible factor in our experiment (cf. Zugal 2013).

Figure 9 shows the subprocess model “create student file” from alternative C. To minimize the impact of factors that refer to the “concrete syntax” of a process model (La Rosa et al. 2011), e.g., the labeling of model elements, we used a strict top-down modeling of control-flows for all subprocesses (except the most abstract model level) and set up naming conventions. Thus, the labels of function types always start with a verb while the labels of event types start with a noun. Further, the students were familiar with the graphical presentation of eEPCs modeled with the ARIS Business Architect, so that the graphical representation was no disturbing factor in our experiment as well.

A questionnaire was designed to gain insights on the personal factors and test the process model understandability in the experiment. For reasons of construct validity, we built on instruments (e.g., “fill-in-the-blanks” test) that were used successfully in previous experimental investigations in the IS domain (e.g., Bodart et al. 2001; Burton-Jones & Meso 2006).

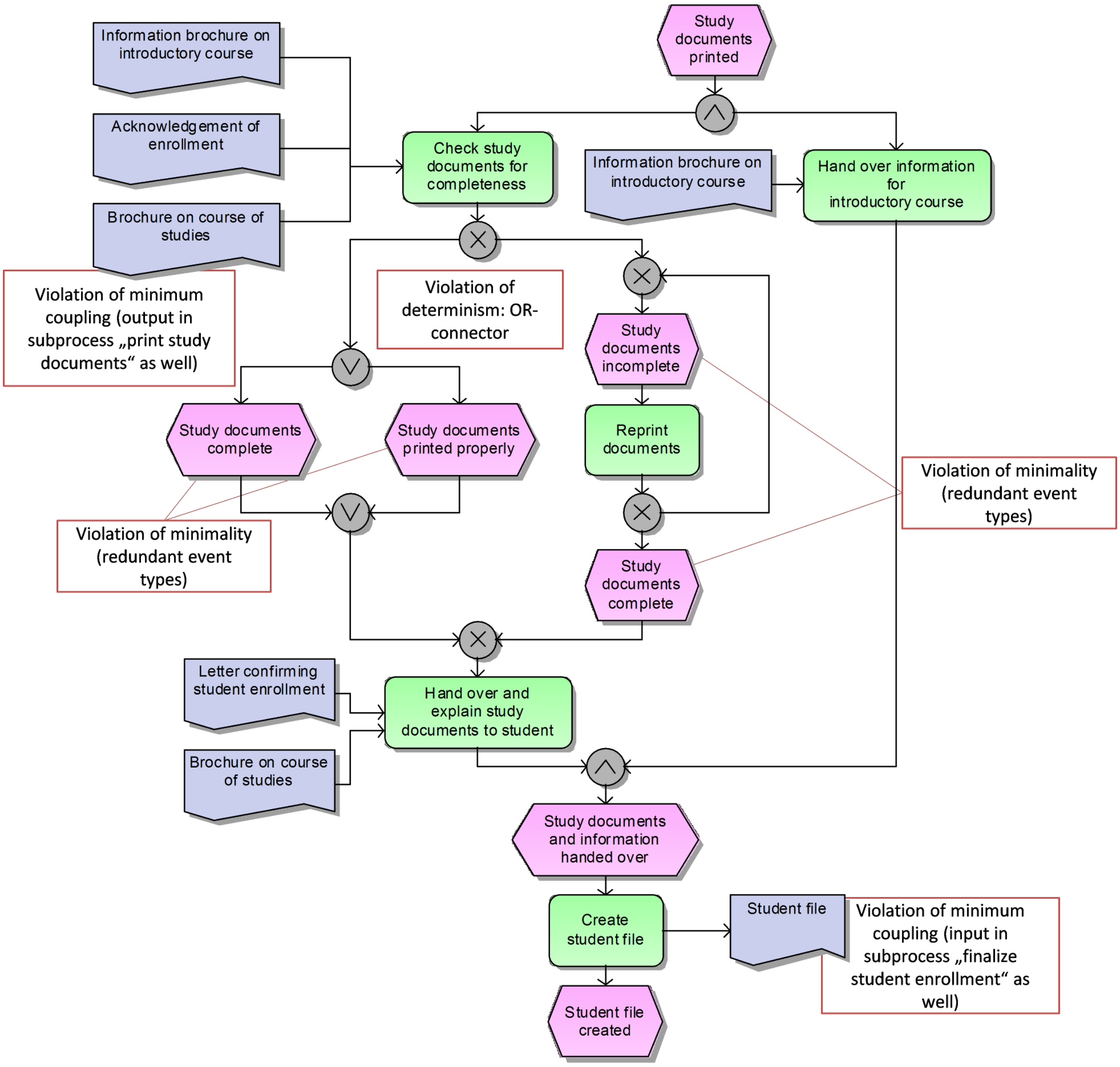


Figure 9: Subprocess "create student file" - model alternative C (strong violation)

The personal factor “modeling expertise” (see Figure 8), is used to judge how often users actually deal with process models in everyday life (intensity of modeling) and to find out when users were confronted with process models for the first time (modeling experience) (cf. Mendling et al. 2012b). A suggestion for requesting that sort of information can be found in Mendling et al. (2012b) for example, which was adapted and modified for the study at hand. This led to three questions (e.g., “how often do you deal with process models in your studies?”). Each question had a four item scale (see questionnaire: block 1). They were summed up to the variable “modeling expertise” ranging from zero to nine so that a higher value indicated more experience.

In addition, the theoretical knowledge of the participants concerning business process modeling constructs was considered by a series of “yes/no questions” (cf. Mendling et al. 2012b). Again, the answers were coded and summed up, forming the variable “theoretical knowledge concerning modeling with eEPCs” ranging from zero to six (see questionnaire: block 2). A higher value indicated more knowledge. Furthermore, the users’ domain knowledge (cf. Khatri et al. 2006) was assessed (see Figure 8). Therefore the participants’ familiarity with the modeled real-world situation (student enrollment) was analyzed using four questions (e.g., “did you gather information which documents are needed by the registry office before registering at your university?” – see questionnaire: block 1). They were condensed into the variable “domain knowledge” ranging from zero to six.

Problem solving questions can be designed as open questions or as evaluations (“true”/ “false”) of statements on the domain (Burton-Jones & Meso 2006). Thus, to measure the problem solving abilities (see Figure 8) both types of questions were used. On the one hand, open questions were asked, such as “which documents does a student have to provide if he wants to matriculate?”. They were answered using the process model as a source (see questionnaire: block 3). On the other hand, statements such as “the matriculation process can be continued even if the student does not bring a bachelor’s certificate” were given. The participants were supposed to agree or disagree on these statements and give an explanation for their decision (see questionnaire: block 4). Demanding participants to explain their answers requires them to profoundly examine the process model (cf. Bodart et al. 2001; Burton-Jones & Meso 2006). That way, their deeper-level understanding can be tested (cf. Bodart et al. 2001; Burton-Jones & Meso 2006). Additional insight into participants’ surface-level understanding can be gained by a recall of model content (Bodart et al. 2001; Burton-Jones & Meso 2006; Houy et al. 2012). For that purpose, a “fill-in-the-blanks” test (cf. Burton-Jones & Meso 2006) was used and users were supposed to fill the gaps with information retrieved from the models (see questionnaire: block 5). The results were coded into the variable “problem solving” ranging from zero to 41 with a higher value indicating a better performance.

The perceived ease of understanding (see Figure 7) was tested on a 7-point likert scale comprising four items (e.g., “understanding the process model requires enormous mental effort”) with a higher value indicating a higher degree of agreement (see questionnaire: block 6). Such an approach is suggested by Burton-Jones and Meso (2002) or Gemino and Wand (2005) for example, which was taken up and adapted for the investigation at hand. The answers were added up and labeled “perceived ease of understanding” in a range from zero to 28.

In summary, the questionnaire covered subjective as well as objective dimensions for measuring model understandability. Regarding the objectively measurable dimensions of “effectiveness” (Houy et al. 2012) a correct answering of questions, a solving of problems as well as a recall of model content was considered (questionnaire: blocks 3-5). The dimension “verification of model content” (Houy et al. 2012) was the only dimension not considered in our questionnaire, since this would require each participant to have profound knowledge of the real-world student enrollment process. However, this was no premise in our experiment. The dimension “efficiency” was considered as students were asked to note the time needed for answering the questions. Further, the order of questionnaires given back was documented by the experimenters. The subjective dimension was covered by the variable “perceived ease of understanding” (questionnaire: block 6). Our experiment thus adequately considered the dimensions for measuring model understandability according to Houy et al. (2012).

2.2.4.3 Procedure, Setting and Participants of the Experiment

53 bachelor degree students of a course in “business process modeling and process reengineering” participated in the experiment. The participants could earn additional credit points for their studies, an incentive for taking the experiment seriously (cf. Wohlin et al. 2012). Since each student had at least once experienced “enrollment” at university, a basic understanding for the domain could be assumed for all students. After all participants had arrived, the experimenters shortly introduced the decomposition of eEPC models without mentioning the decomposition conditions. Afterwards, the questionnaire and the process models were handed out. Participants were randomly assigned to one of the groups A, B or C, whereas group A received model alternative A, group B model B and group C model C. In the end, group A (no violations) comprised 14 students, group B (moderate violations) embraced 18 students and group C (strong violations) 21 students. The questionnaire was identical for all groups. The participants did not receive a textual description of the process. In our experiment all participants were given the process model and the questionnaire on paper. This is because the medium of the model (paper or computer screen) is another factor potentially influencing

model understandability (Reijers et al. 2010). To mitigate this influence, the same conditions must be given for all participants in the experimental setting (Wohlin et al. 2012).

As soon as every participant had received the questionnaire and the process model, they were asked to start filling out the questionnaire. The experimenters stayed in the room for the whole experiment assuring there was no collaboration among the students. Students could leave the room as soon as they were finished and every student finished within 1,5h. The fastest student needed 15 minutes to fill out the questionnaire completely, whereas the majority required about 20-25 minutes. Afterwards, the answered questionnaires were corrected by two researchers to mitigate subjectivity regarding the correction.

2.2.5 Results of the Experiment

2.2.5.1 Results Received

Asked about their experience with process modeling, the majority of students reported to deal with process models as part of their studies nearly every day (67.9% – 36 students) or several times a month (30.2% – 16 students). Asked about their modeling experience as part of their work or internships, 13.2% stated to deal with process modeling nearly each day, 18.9% several times a month and 50.9% less than once a month. In total, the majority deals with process modeling in their studies nearly every day while several students are confronted with process models in internships or extramural work additionally. Thus, we consider the participants to have experience with process modeling. On their domain knowledge, 88.7% of the students reported to have matriculated once, or twice (9.4%), or more (one student). Further, 56% reported to not know how the matriculation is processed administratively, 41% know so partially and one student reported to know it. Finally, students were asked if they dealt with the said process in other courses. 7.5% reported to have done so, 92.5% have not. In summary, the students were in contact with the process before, however their knowledge is vague. Finally, we tried to assess the students’ knowledge on eEPC modeling. Asked six detailed questions on eEPC modeling, 52.8% of the students answered all questions correctly and therefore scored six points, 35.8% five points, 7.5% four points and one student had three correct answers. One response was missing. In summary, the majority of participants can be considered to know eEPC modeling.

The proposed hypotheses were tested using a regression analysis. The regression analysis is applicable for this experiment: First, our dependent variable is a rating scale. Second, independent variables are either ordinally scaled, i.e., the number of decomposition conditions violated (group C: five conditions violated, group B: three conditions violated, group A: zero conditions violated) or rating scales, i.e., the personal factors. Third, the regression allows us to interpret the results in their direction. We did refrain from an ANOVA analysis since this method does not allow inferences about the tendency.

Our sample has 51 valid cases if the personal factors were involved and 53 cases otherwise. The questionnaire contained 28 questions covering the six categories as shown in Figure 8. Since many of the questions turned out to have hardly any explanatory power on their own, they were aggregated into the variable “personal factors”. Checking the correlations among the independent variables “modeling expertise”, “theoretical knowledge concerning modeling with eEPCs” and “domain knowledge“, the corresponding variance influence factors ranged between 1.018 and 1.096, dismissing significant interferences. Therefore the regression analysis was applicable.

Table 8 shows the regression estimation for hypothesis “a”. The three violation groups were dummy coded with group C as the base level. The coding followed the suggestions of Walter et al. (1987). Regarding process model understandability, both groups A (violating no decomposition condition) and B (violating three decomposition conditions) perform significantly better than group C whose model violates all five conditions. In both groups, the direction of effect remains positive over the confidence interval ([2.920; 9.957], [2.305; 8.754]) and the effect is significant (0.001, 0.001). The std. error is 1.749 and 1.603. Finally, for the personal factors the estimation shows a weak negative influence with changing sign. Nonetheless, the influence of the process model factors is significant. In consequence, H_0a has to be rejected and we feel supported in our initial assumption that abiding the decomposition conditions (see Table 7) positively influences the understandability of process models. On a side note, for above model R^2 has a value of 0.297. If the model is modified using group B as base, group C performs in a 95% range of [-8.754; -2.305] and still has a clear effect relative to B. The difference to group A, however, becomes insignificant on a range of [-2.784; 4.602]. Therefore, the difference between “no violations” and “moderate violation” groups is too low to be significant, but still the difference between “no violations” and “strong violations” is significant, supporting our initial assumption.

H_0a: The process model factors and the personal factors have no influence on the process model understandability. **H_1a:** The process model factors and the personal factors have influence on the process model understandability.

Model	Unstandardized Coeff.		Standard. Coeff.	t	Sig.	95,0% Confidence Interval for B	
	B	Std. Error	Beta			Lower bound	Upper Bound
(Constant)	51.174	5.667		9.031	.000	39.774	62.574
Group A	6.438	1.749	.503	3.681	.001	2.920	9.957
Group B	5.529	1.603	.474	3.450	.001	2.305	8.754
Personal factors	-.520	.406	-.162	-1.281	.206	-1.337	.297
a. Dependent Variable: Process model understandability; Overall F-Value: 7.972							

Table 8: Coefficients for H_0a/H_1a

H_0b: The process model factors have no influence on the problem solving. **H_1b**: The process model factors have influence on the problem solving.

Model	Unstandardized Coeff.		Standardized Coeff.	Sig.	t	95,0% Confidence Interval for B	
	B	Std. Error	Beta			Lower Bound	Upper Bound
(Constant)	35.048	.803		.000	43.666	33.436	36.660
Group A	1.741	1.269	.211	.176	1.372	-.808	4.290
Group B	.536	1.181	.070	.652	0.453	-1.837	2.909
a. Dependent Variable: Problem solving, Overall F-Value: 0.952							

Table 9: Coefficients for H_0b/H_1b

Table 9 shows the regression estimation used for hypothesis “b”. As can be seen, the influence on group A, models with no violations, is significant to a level of 0.824, which appears sufficiently high in our context. However, the significance for group B with moderate violations is too low to reject the possibility, that the measured influence is coincidental. The R^2 for above model is 0.037, which is to be expected in the multifaceted context of process model quality measurement. In summary, we chose to reject H_0b only partly. We are certain to have shown a significant difference regarding the problem solving capabilities between models with no violations and strong violations. However, we cannot conclude with certainty a difference between no violations and moderate violations. The std. error is 1.269 and 1.181.

H_0c: The process model factors have no influence on the perceived ease of understanding. **H_1c**: The process model factors have influence on the perceived ease of understanding.

Model	Unstandardized Coeff.		Standardized Coeff.	Sig.	t	95,0% Confidence Interval for B	
	B	Std. Error	Beta			Lower Bound	Upper Bound
(Constant)	9.095	.778		.000	11.695	7.533	10.657
Group A	4.548	1.230	.487	.001	3.698	2.078	7.017
Group B	4.571	1.145	.526	.000	9.994	2.272	6.871
a. Dependent Variable: Perceived ease of understanding, Overall F-Value: 10.384							

Table 10: Coefficients for H_0c/H_1c

Table 10 shows the regression estimation for relationship “c”. The direction of the effect of group A and group B is positive. This means, the students rated the models A and B better than model C. The direction of effect does not change over the confidence interval and the effect itself is highly significant. We can therefore clearly refute the hypothesis that the model factors have no influence on the perceived ease of understanding. This result supports our initial assumption that violating the decomposition conditions decreases the ease of understanding of process models.

Finally, the results are further visualized in Figure 10. The boxplots reveal a visible difference of group C towards groups A and B. Though the range of group B is wider, the difference in average between A and B is small. Further, all groups perform in the higher half of the possible results. This may be attributed to the selection of students with a fond background in process modeling.

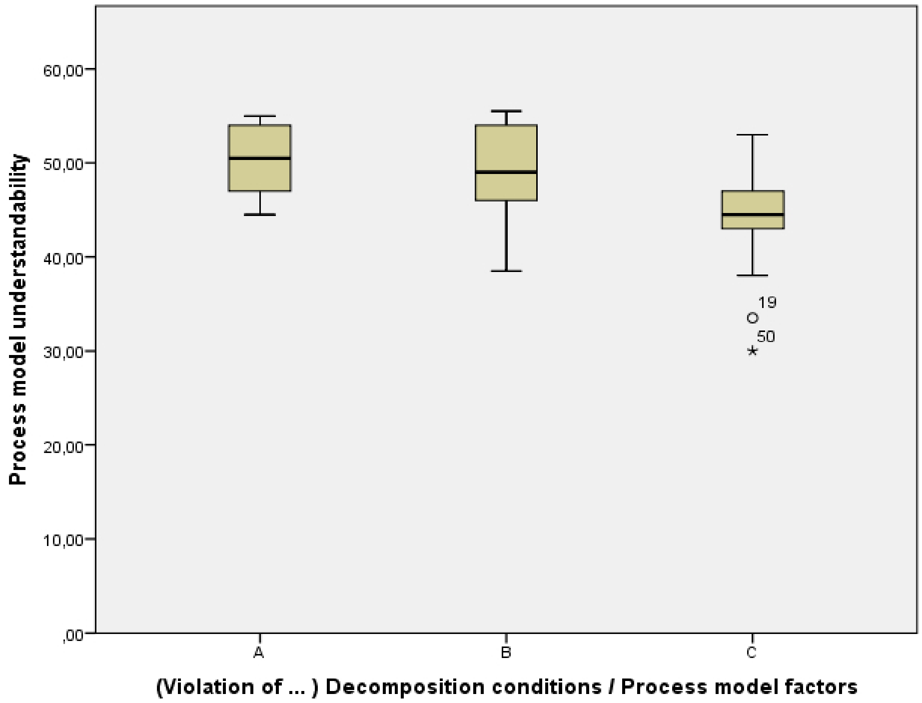


Figure 10: Boxplots showing the difference between groups A, B and C

In summary, the results show a significant difference between "no violations" (group A) and "strong violations" (group C) for all our hypotheses. On the other hand, the difference between "no violations" (group A) and "moderate violations" (group B), as well as "moderate violations" and "strong violations" (group C) is weaker. This was to be expected due to the differences of groups A, B and C regarding the decomposition conditions (see section "Materials of the Experiment").

2.2.5.2 Validity

Validity in experimentation describes the assessed truth of an inference (Shadish 2002). Validity discusses in how far the results of our experiment support our claim about the influence of the decomposition conditions on the concept of model understandability. For discussing common threats to validity, Shadish et al. (2002) present the following typology:

The first category of threats affects the statistical conclusion validity (Shadish 2002). Since our experimental design did not include repetitions or different locations, threats referring to variance in implementation or selection of participants can be excluded. Therefore, it remains to discuss the statistical assumptions of our tests. We used a regression analysis with the coding scheme of Walter et al. (1987). For hypothesis "a", our regression analysis uses the process model understandability as dependent variable, the process model factors and personal factors as independent variables. A visual examination of the scatterplot supported a linear regression approach. Further, referring to prior experiments and their measures (cf. Khatri et al. 2006; Mendling et al. 2012b; Reijers & Mendling 2011) we expect to have used the statistically relevant independent variables. We also tested for multi-collinearity. Variance influence factors between 1.0 and 1.1 allow us to refute an influence. Finally, our interpretation is, as proposed by Shadish et al. (2002), based on the confidence intervals.

Internal validity refers to the question whether the difference in reported understandability is actually a consequence of the manipulation of the decomposition conditions. In this regard, the experiment was conducted once in one moment and place. Consequently, we could mitigate learning, maturation, attrition or instrumentation effects. The participants' background was checked in the questionnaire.

Construct validity refers to the question, in how far our instruments reflect the constructs of process model understandability, personal factors and violation of the decomposition conditions correctly. Considering the artificial nature of the concepts and their vague definition, the construct validity can hardly be assessed objectively. Nonetheless, to mitigate the most obvious pitfalls, our instrumentation referred to multiple methods of assessing constructs (e.g., questionnaire, "fill-in-the-blanks" test) that were pretested and used by other authors before (cf. Burton-Jones & Meso 2006; Mendling et al. 2012b).

External validity describes in how far the causal relation discussed in this experiment can be generalized over different settings. As discussed before, a wide range of factors (e.g., media, modeling purpose, modeling language) can possibly influence process model understandability. However, they are not part of our experimental setup. We intentionally choose the form of a laboratory experiment to analyze the effect of the decomposition conditions in isolation. The isolated analysis allows an interpretation whether our results are caused by the treatment or result from an incidental combination of other factors (see internal validity). However, this isolated analysis limits the generalizability of our results. Obviously, it takes further research in the field to analyze how strong the influence of adhering to the decomposition conditions is in more general settings.

Indeed, a limitation of our experiment is the use of students in a class setting. However, students can be considered as novices in process modeling (cf. Recker et al. 2012), a target group for whom model understandability is of central importance. Hence, the students are eligible candidates for our experiment.

2.2.5.3 Interpretation of the Results

We now interpret the results of the quantitative analysis. The results indicate that adherence to the decomposition conditions, or their violation, has an influence on model understandability, especially on the perceived ease of understanding. Delineating subprocesses of a decomposition, considering the decomposition conditions, obviously leads to a design which is perceived as more understandable than an alternative decomposition violating the conditions. It needs to be mentioned that there were no large differences in the quality of answers given to the problem solving questions between the groups B and C. A possible explanation for this is that the split-attention effect (Zugal et al. 2011) was compensated by participants through continuously scanning the process model during answering the questionnaire over and over again, which of course is time consuming. However, a difference became obvious for the problem solving abilities regarding groups A and C. Therefore, violating all decomposition conditions negatively affected problem solving abilities in comparison to an equivalent decomposition fulfilling all decomposition conditions.

The large difference between groups A and C as well as between B and C regarding perceived ease of understanding is an indicator that the minimum coupling as well as the strong cohesion condition strongly influence the perceived ease of understanding. An explanation is that minimum coupling demands to reduce input-output-relations between subprocess models (e.g., by merging them) and decreases the fragmentation of process models that way (Zugal et al. 2013). Further, strong cohesion promotes model abstraction by holding related objects together in a subprocess model (Zugal et al. 2013). Both, model abstraction as well as fragmentation are recognized to support, respectively hamper, model understandability (Zugal et al. 2013), which is in accordance with our results. Whereas the influence of process fragmentation on problem solving can obviously be mitigated by thoroughly scanning a process model, this appeasing effect is not given for the subjective perception of model understandability. The small difference between group A and B indicate that the minimality, determinism and losslessness condition have a small influence. The weak explanatory power of the personal effects may be caused by the selection of students. The fact that all have a common background in process modeling may lead to the little influence on the overall prediction, leaving high residuals. Experiments with participants from a wider background, which will be done in future research, may clarify this.

Model	Unstandardized Coeff.		Standardized Coeff.	t	Sig.	95,0% Confidence Interval for B	
	B	Std. Error	Beta			Lower Bound	Upper Bound
(Constant)	33.429	3.201		10.442	.000	26.998	39.859
Group A	-13.500	5.062	-.389	-2.667	.010	-23.667	-3.333
Group B	-8.429	4.712	-.261	-1.789	.080	-17.894	1.037
a. Dependent Variable: Order of returned questionnaires							

Table 11: Coefficients for order of returned questionnaires

Further, since problem solving was the most time consuming task, we analyzed the influence of the model factors towards the order the tests were given back (see Table 11). As can be seen, the models from group A were given back before those of group B which were finished before group C. Therefore, problem solving for alternatives violating the decomposition conditions proved to be more time consuming than for alternative A showing no violations. This supports our assumption regarding the effect of process fragmentation on problem solving as mentioned above. In consequence, even though we could not measure a significant difference in problem solving capabilities among B and C, we could measure a difference in the order of returns. Finally, it remains to interpret the results from hypotheses “a” and “c”. Both tests support our initial assumption, that the violation of decomposition conditions decreases the understandability of process models. Especially hypothesis “c” shows a clear direction of the effect with a high significance. Therefore, the topic cannot be dismissed since we have found a strong influence the violation of the decomposition conditions has on the perceived ease of understanding.

2.2.6 Summary, Limitations and Outlook

In the paper at hand, we tested the impact of the decomposition conditions on the understandability of decomposed process models. After introducing the conditions, we experimentally tested the assumption that the adherence to the decomposition conditions positively influences the understandability of a decomposed process model. 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. This result does not only support our initial assumption but also underpins the importance of the research. It justifies continuing our work in developing guidelines for decomposing process models based on the decomposition conditions. The results show that the decomposition model is actually a promising approach for assessing the still fuzzy topic of process model decomposition. Further, the large difference between the “moderate” and “strong violation” groups regarding the “perceived ease of understanding” is remarkable. In addition, we have a small difference between the groups “no violation” and “moderate”. This is an indicator that the conditions (minimum coupling and strong cohesion) focusing structural aspects (delineation of subprocesses) have a major influence on understandability.

There are some limitations. At first, we restricted ourselves to eEPCs. We have done this because modeling languages differ in their ontological expressiveness (Recker 2011; Recker et al. 2009). The ontological characteristics of modeling languages influence the interpretation of the decomposition conditions (cf. Johannsen et al. 2014b). Specifying the decomposition conditions to be valid for different modeling languages would neglect the individual strength and peculiarities of a language and thus result in a very abstract specification not feasible for the purpose of process model decomposition. Thus, to receive detailed and beneficial insights, a focus on one modeling language is essential. We focus on the eEPC, which is currently one of the most frequently used process modeling languages (Reijers & Mendling 2011). Second, the results of the experiment might be different if practitioners were considered instead of students. Whereas the students in our experiment represent quite a homogeneous group, there might be more variability concerning the personal factors “modeling expertise”, “theoretical knowledge on process modeling” and “domain knowledge” among practitioners. Reproducing our results with practitioners is a topic for future research. Third, the effort required for judging a decomposed process model regarding the decomposition model varies for the conditions (cf. Johannsen et al. 2014b). Some of them (e.g., minimality and losslessness) require a certain degree of domain knowledge and go beyond a mere analysis of the process model structure. It further needs to be mentioned, that our interpretation of the conditions for eEPCs is based on a representational mapping (cf. Johannsen & Leist 2012b) and does not generally exclude other perspectives on the decomposition model.

To our knowledge, our study is the first one to analyze the benefits of the decomposition model in the context of business process modeling. Based on these findings our upcoming work will address the following: First, additional empirical studies will be done to verify our results with practitioners. Further, we will specify guidelines for decomposing process models based on the decomposition conditions. Additionally, a more detailed analysis of interdependencies between the decomposition conditions will be done. Whereas minimum coupling tends to merge subprocesses, strong cohesion tends to delineate subprocesses for example. The balance of the conditions regarding model understandability will be tested more precisely. Is it better to violate the minimum coupling condition completely but stick to the strong cohesion condition than violating both conditions at the same time to a lesser degree? In that context, the impact of each single condition on model understandability will be tested more detailed. Furthermore, the effect of combining certain violations against the conditions on model understandability is to be explored more profoundly. Moreover, it is worth investigating whether thresholds for the number of violations that differentiate a well-understandable from an incomprehensible decomposition can be derived or not. Finally, the interpretation of the decomposition conditions will be extended to other modeling languages.

2.3 Beitrag 3: Metrics for evaluating decomposed process models based on Wand and Weber’s good decomposition model

<table>
<tr><td>Adressierte Forschungsfrage</td><td>Forschungsfrage 3: Welche Metriken lassen sich aus den Good Decomposition Conditions ableiten, um Modellierer bei der Dekomposition von Prozessmodellen zu unterstützen?</td></tr>
<tr><td>Erscheinungsort</td><td>Zur Begutachtung eingereicht, Journal of Information Technology (VHB Jourqual 3: A)</td></tr>
<tr><td>Autoren</td><td>Dr. Florian Johannsen 70 %
Daniel Braunnagel 15 %
Prof. Susanne Leist 15 %</td></tr>
<tr><td colspan="2">In diesem Beitrag werden die GDC in Form von 12 Metriken operationalisiert. Zunächst wird für jede Condition ermittelt, welche Informationen für ihre Berechnung benötigt werden. Anschließend werden Elemente der eEPK Notation identifiziert, in denen sich diese Informationen finden. Im dritten Schritt wird berücksichtigt, wie eEPK Modelle dekomponiert werden und welche Metriken vor diesem Hintergrund die Conditions sinnvoll operationalisieren können. Die entwickelten Metriken werden auf Material eines Laborexperimentes angewandt, um zu demonstrieren, inwieweit die Metriken die empirisch ermittelten Unterschiede in der Verständlichkeit der Modelle anzeigen. Weiter wird vorgestellt, inwieweit die Metriken implementiert werden konnten. Zuletzt werden die Metriken den Conditions gegenübergestellt.

Damit behandelt der Beitrag den DS Entwicklungszyklus für die Entwicklung der Metriken mit Ausnahme der Evaluation vollständig.

Für dieses Vorhaben verdienen folgende Herausforderungen besondere Erwähnung. Zunächst müssen die formal spezifizierten Metriken berücksichtigen, wie Dekompositionen für eEPK Modelle im Alltag durchgeführt werden. Weiter muss diskutiert werden, inwieweit die für die Berechnung notwendigen Informationen alleine aus dem Prozessmodell ermittelt werden können. Zuletzt ist, wie schon bei der Formulierung der Conditions, zu berücksichtigen, dass die BWW Ontologie zur Beschreibung real existierender Dinge, also Instanzen, entwickelt wurde. Für die Bewertung der eEPK Modelle müssen also Rückschlüsse von den möglichen Instanzen des Modells auf die Qualität dessen vorgenommen werden.</td></tr>
</table>

Tabelle 12: Bibliographische Angaben zu Beitrag 3

Business process modeling is an important task in business transformation initiatives and information systems (IS) development projects. Process models visualize the working procedures of a company and pinpoint the way in which business value is created. Based on process models, requirements on IS are derived, decisions on IS investments are made and business processes are analyzed in terms of efficiency and effectiveness for instance. However, in case process models become too large in size, employees will hardly understand them, which restricts the potential benefits associated with business process modeling. Therefore, the decomposition of process models is a means of reducing their complexity by delineating corresponding subprocess models. However, there are no commonly accepted approaches for decomposing process models and the properties that characterize a well-done decomposition are unclear yet. We thus revert to the good decomposition model of Wand and Weber, which was initially 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 an extensive process model from a corporation project shows that the metrics provide a helpful way of objectively assessing the quality of decompositions in process modeling, by using the good decomposition model.

2.3.1 Introduction

Business process modeling has increasingly gained attention in business transformation initiatives 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 IT investments (Becker et al. 2010). As a consequence, different user groups in an enterprise have varying expectations regarding the level of abstraction of information captured in process models (cf. Harmon 2016).

However, creating process models is a highly subjective task (Pinggera et al. 2013); not only designing process models is usually challenging (cf. Rosemann 2006) but also evaluating their quality (cf. Fettke et al. 2012). Accordingly, different quality perspectives and various approaches for evaluating conceptual models are proposed in literature (cf. Mendling et al. 2010; Overhage et al. 2012).

While different factors influencing the understandability of process models exist (e.g., modeling expertise) (cf. Mendling et al. 2012b), 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. 2013). Information adapted for user groups are placed in selected subprocess models to reduce mental efforts of finding the facts users search for (Bobrik et al. 2007).

Though the benefits of decomposing business process models are commonly accepted, decomposition is often done in an “ad-hoc fashion” since generally acknowledged guidelines used for doing so are missing (Milani et al. 2016; Reijers et al. 2011; Burton-Jones & Meso 2008). Consequently, there is uncertainty regarding those properties that characterize a good decomposition in process modeling. In this context, the good decomposition model of Wand and Weber (1997) is promising for guiding modelers in decomposing process models purposefully, leading to an easier understanding of decompositions of process models (Recker et al. 2009). The good decomposition model originates from the information systems (IS) discipline and defines five conditions that characterize a “good” decomposition. In previous research we transferred these conditions to business process modeling and derived guidelines for a good decomposition accordingly. (cf. Johannsen & Leist 2012b).

Besides supporting model creation, the decomposition conditions also enable the evaluation of models that were already decomposed in an arbitrary manner (cf. Johannsen et al. 2014b). However, checking the conformance of a decomposed model with the decomposition requires tremendous cognitive effort, if done manually.

Considering this, a formal operationalization of Wand and Weber’s decomposition conditions in the form of metrics is beneficial for the following reasons. First, metrics allow to objectively judge as to how a decomposed process model adheres to the conditions as defined, reducing the subjectivity of user assessments. Second, we develop a software tool to automatically perform the calculation. For that purpose, the metrics’ variables are mapped to corresponding algorithms and procedures executing the evaluation of the process model.

Against this background, the paper describes the development of metrics that measure the coherence of a decomposed process model with Wand and Weber’s decomposition conditions. To provide the necessary level of detail, we restrict our research to Event-driven Process Chains (EPCs) because the ontological expressiveness of modeling languages differs (Recker 2011), which, in consequence, impacts the interpretation of the decomposition conditions for modeling techniques. To systematically define metrics, we follow the “Goal Question Metric (GQM)” approach (cf. Basili et al. 1994).

The contribution of our work is the following: first, we formally operationalize the decomposition conditions as a set of metrics. This establishes an objective base for assessing the quality of a decomposition regarding the decomposition model. The metrics revert to the conditions of Wand and Weber to unveil properties of well-performed decompositions that have not been identified for the process modeling discipline so far. Additionally, we develop a prototype that supports the calculation of the metrics’ values for a decomposed process model, eliminating calculation errors and manual efforts. 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 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 explains the relevance of the decomposition model for the research at hand. Afterwards, in the section “Metrics” (see section 2.3.3) we present our methodological procedure and introduce metrics for evaluating a decomposed process model. Then, a use case for the application of the metrics demonstrates the application of the metrics on a process model stemming from a cooperation project (see “Application of the metrics”, section 2.3.4). The prototypical implementation of a tool to support the calculation of the metrics is described in the section that follows. In the section “Discussion”, we discuss the results. The paper ends with a conclusion, limitations, and an outlook on future research.

2.3.2 Basics and related work

2.3.2.1 Event-driven Process Chains and process model quality

Event-driven Process Chains (EPCs) were developed in the early 1990s and are currently one of the most frequently used techniques for business process modeling (Adam et al. 2013; Mendling 2008). A flat EPC model comprises nodes and arcs while a node can be a function type (green rectangle), an event type (red hexagon), a connector type (grey circle) 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).

Figure 11 provides an example of an eEPC model. It shows an excerpt of a student matriculation process at a German university. More precisely, the depicted subprocess deals with the creation of a student account after entering the basic claims data into the database by an employee from the administration department.

The quality of eEPC models and process models in general is a much discussed and complex topic (e.g., Fettke et al. 2012; Mendling et al. 2010; Zugal et al. 2011). Literature differentiates between quality dimensions such as “complexity”, “coupling”, “cohesion”, “size”, or “modularity” that determine the quality of a process model (Vanderfeesten et al. 2007a). Nevertheless, a commonly accepted theory on process model quality is missing (Houy et al. 2014).

Therefore, a variety of approaches for determining process model quality exist (cf. Mendling et al. 2010). Quality frameworks, such as the “Guidelines of Modeling (GoM)” (Schütte & Rotthowe 1998) or the “3QM-Framework” (Overhage et al. 2012), introduce criteria or rules (e.g., avoidance of OR connectors) against which a process model is reflected to judge its quality. Further, quality metrics exist that allow to quantify process model quality in an objective manner (cf. Mendling 2008; Cardoso 2006; Gruhn & Laue 2007). More, pragmatic guidelines for creating easy-to-understand models have been specified for practitioners in special (cf. Pieterse 2005; Silver 2008). Additionally, several studies analyze the expressiveness and usability of modeling techniques to create process models of high quality (cf. Recker 2011; Sarshar & Loos 2005; Frank 1998).

Besides the aforementioned approaches, also the modeling purpose (e.g., process analysis) needs to be considered when judging the quality of a process model (Fettke et al. 2012). Depending on the purpose, only specific information captured in a process model may be relevant for a certain model user (Bobrik et al. 2007; Jeanneret et al. 2011).

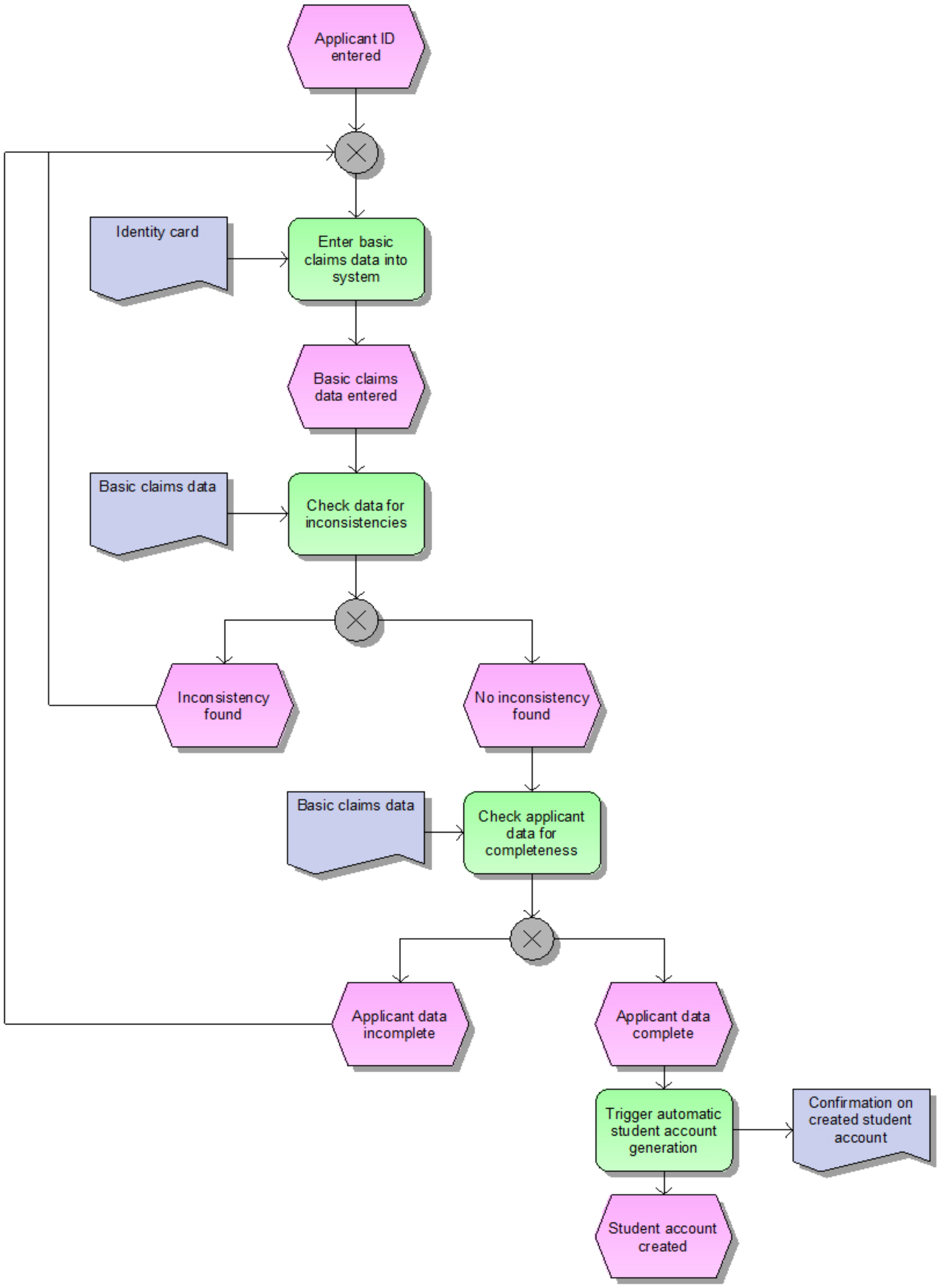


Figure 11: Subprocess model "trigger creation of student account" at a German university

2.3.2.2 Decomposition in process modeling and the relevance of Wand and Weber's decomposition model

Decomposition is a means of reducing the complexity and thus the quality of large process models (Zugal et al. 2013; Reijers et al. 2011). To this purpose, a process model is split into smaller subprocess models that are assigned to a level hierarchy (Reijers et al. 2011). Users may then focus on selected subprocess models on a certain model level to find the information relevant for their particular needs (Bobrik et al. 2007).

In that context, model abstraction and model fragmentation are differentiated (cf. Zugal 2013; Lara et al. 2013). During model abstraction, information is aggregated by designing an abstract process model whereas model fragmentation means to spread detailed information across several subprocess models (cf. Zugal 2013; Lara et al. 2013). Thus, a modeler 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 (Davis & Brabänder 2007). In the latter case, subprocess models are delineated by abstracting from the details (model abstraction) (Davis & Brabänder 2007). A variety of suggestions on how to do the decomposition are found in literature (e.g., Milani et al. 2016).

For example, "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). Another approach, which analyzes the connections between nodes of a process model, is called "graph-clustering" (Reijers et al. 2011). Accordingly, nodes that are strongly connected to each other should be captured within a subprocess model (Reijers et al. 2011). *Van der Aalst* (2013) investigates the decomposition of Petri nets in special, whereas *Ma et al.* (2015) propose an algorithm for an automatic decomposition of process models. *Milani et al.* (2016) analyze further decomposition approaches in a controlled experiment. It turned out 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 (Milani et al. 2016). More, *Basu and Blanning* (2003) present an approach to decompose processes by formally analyzing process structures.

Summing up, although there are approaches helping a modeler to decompose process models, none of them has established as a commonly accepted standard.

However, not only the process of decomposition lacks clear guidelines, also the characteristics that constitute a good decomposition remain unspecified in literature (cf. Zugal 2013; Reijers et al. 2011). Therefore, judging the quality of a decomposed process model is challenging and highly subjective. The development of metrics that focus on the quality of decomposed process models in special is a rather under-researched topic yet (cf. Reijers et al. 2011).

In this ongoing discourse, we build on the good decomposition model of Wand and Weber (cf. Weber 1997; Wand & Weber 1989a) to judge the quality of a decomposition in process modeling. The good decomposition model originates in the IS discipline and is part of the BWW ontology (Weber 1997). The potential benefits of the decomposition model to come to a manageable set of subprocess models in large modeling 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 modeling. In previous works, we transferred the decomposition conditions to business process modeling and specified them for eEPC models in particular (cf. Johannsen & Leist 2012b). Due to the diverging ontological expressiveness of modeling notations (Recker et al. 2009), the interpretation of the decomposition conditions varies depending on the modeling technique considered. We also showed that the perceived understandability of eEPC models strongly profits from the decomposition model (cf. Johannsen et al. 2014a). 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 as metrics. Adequate metrics precisely capture the constructs of the decomposition conditions as variables and thus facilitate an objective judgement as to what extent a decomposed process model adheres to the decomposition conditions. Our study addresses this gap and develops corresponding metrics.

2.3.2.3 Differences in terms between the good decomposition model and process modeling with eEPCs

The decomposition model builds on the so-called representational model of the 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 & Green 2002). Things can be grouped to systems and subsystems accordingly (Weber 1997). A detailed description of the BWW ontology is given in Weber (1997) 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.

However, due to its origin in IS, the terms and concepts used by the decomposition model differ from basic notions of process modeling with eEPCs. Additionally, business process models work on the type level whereas the BWW ontology focuses the instance level (Green & Rosemann 2000). This leads to challenges when using the decomposition model in the context of business process modeling. For example, the customer invoice “No. 463” in terms of a company’s sales process would be a “thing” according to the BWW ontology (e.g., Recker 2011). However, in an eEPC process model this invoice would be represented by a data object type “invoice”, and the invoice “No. 463” would be a concrete instance of this type.

Therefore, we present a short summary of the key concepts of the BWW model and the corresponding interpretation for our research.

First, the notion of a “system” requires clarification. A system is defined as a set of “things” by *Weber* (1997) with no equivalent counterpart existing for eEPC process models (cf. Green & Rosemann 2001). A system may thus comprise several organizational units or particular employees participating in the execution of a company’s business processes for instance (e.g., Green & Rosemann 2001). However, this perception is not compatible with eEPC process modeling, which puts the “business process”, defined as an arrangement of individual working activities, in the center of attention (cf. Hammer 2015). We therefore interpret a “system” as a self-contained business process with a clearly defined starting and ending point that is visualized as a corresponding eEPC model.

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. Nevertheless, they are the most suitable construct for representing "things" in eEPC modeling with a sequence of event types indicating a change of state (cf. Weber 1997) for the data object types (e.g., a change of state for the data object type "customer order" from "received" to "processed").

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 modeling 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 for eEPCs.

2.3.3 Metrics

2.3.3.1 Procedure

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 interpreted regarding the goal (Basili et al. 1994). For that purpose, a set of questions is used to specify the goal more precisely before metrics to quantitatively answer the questions are defined (Basili et al. 1994).

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

- (Q1) minimality,
- (Q2) determinism,
- (Q3) losslessness,
- (Q4) minimum coupling and
- (Q5) strong cohesion condition?

To answer the above questions, we derive metrics that help to evaluate the coherence of a decomposed eEPC process model in view of the decomposition conditions.

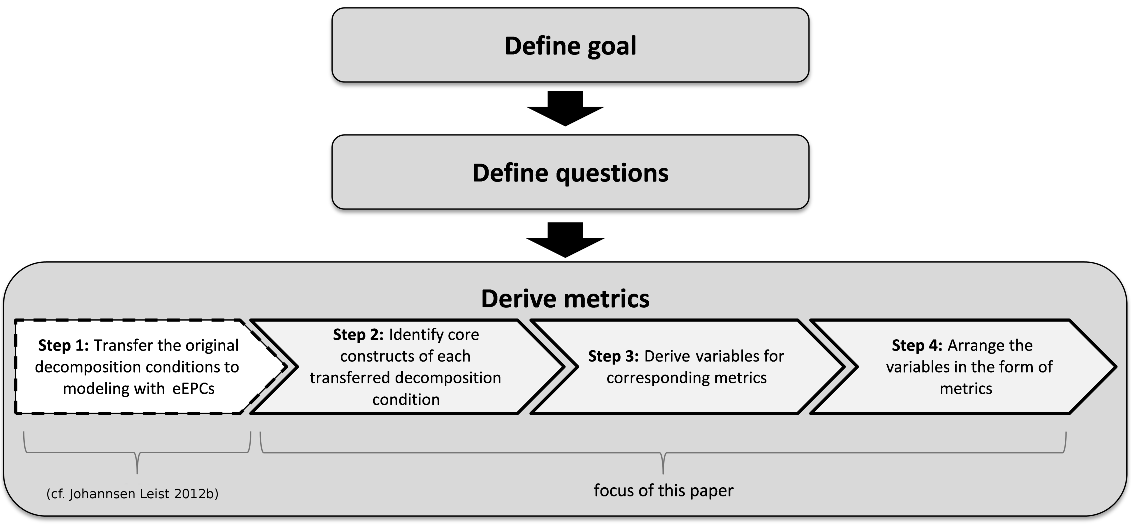


Figure 12: Procedure of the research

To specify metrics to answer the above questions, we follow a four-step approach. Our first step (step 1) is to explain the transfer of the decomposition conditions—as they were originally defined (cf. Weber 1997; Wand & Weber 1989a) – to modeling with eEPCs. This step was already performed in one of our previous works (cf. Johannsen & Leist 2012b), in which we built on the ontological analysis of modeling languages, (cf. Recker 2011; Green & Rosemann 2000), and mapped central concepts of the BWW ontology to corresponding modeling constructs. In a second step (step 2), we identify the central constructs of the newly specified decomposition conditions for eEPCs. Afterwards (step 3), 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 as to how a decomposed model adheres to these conditions (step 4). We summarize this procedure in Figure 12.

In the remainder of this chapter, we introduce each of the five decomposition conditions in more detail. After an interpretation of the decomposition conditions for eEPC models, we develop corresponding metrics for assessing the coherence of a process model to the conditions. 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. Fig. 19 shows that a decomposed process model has several levels with corresponding subprocess models.

2.3.3.2 Minimality

The minimality condition—as it was originally defined by *Weber* (1997, S. 153) - is the following: "*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*".

Explanation and transfer to eEPCs:

Step 1: The definition mentions "level structure", "(sub-)system", and "state variables" in particular. A system or subsystem is a collection of "things" (e.g., employee) that participate in a business process and interact with one another (cf. Weber 1997). A "thing" experiences several changes of state throughout its lifetime while potential states are determined by their state variables (cf. Weber 1997). Regarding the eEPC, we interpret data object types as types of representations for "things" that are characterized by attributes (see section 2.3.2.3). The attributes' values indicate the current state of an instance of the data object type. A change of state for an instance of a data object type is expressed by the triple "*event type → function type → event type*" in the eEPC (Keller et al. 1992; Hoffmann et al. 1993). Acknowledge that eEPC process models are allocated to the type level (section 2.3.2.3). As mentioned by the definition, eEPC event types represent the state variables. Following *Weber* (1997), state variables must not be redundant. This means that event types in an eEPC model indicating "states" of an instance of a data object type ("thing") that never occur in any process instance are to be avoided (Hoffmann et al. 1993). However, sometimes certain state variables are a prerequisite for the "change of state" of other variables (Weber 1997) and are thus not redundant while certain event types (e.g., "dataset locked") may be required for a profound system specification (Hoffmann et al. 1993).

Consequently, the following specification of the minimality condition for eEPCs emerged: "*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 (sub-)process on an instance level are to be avoided. Function types and modeling constructs from other ARIS (Architecture of Integrated Information Systems) views (e.g., of the organizational view, data view, output view) related to these must be scrutinized with respect to their necessity regarding this fact as well*" (Johannsen & Leist 2012b, S. 277).

<u>**Derivation of metrics to assess "minimality":**</u>

Step 2: Central constructs of this interpretation are "redundant event types" and "subprocesses". In general, a subprocess model may comprise none or several "redundant event types". However, since process modeling is a highly subjective task (Schütte & Rotthowe 1998), the identification of redundant event types in an eEPC model requires profound process and domain knowledge from the user's side. A user unfamiliar with the real-life circumstances will not be able to judge whether certain event types depicted in a process model might probably never occur and are thus "redundant".

Step 3: To define a metric, the variables "subprocess model" and "redundant event type" are derived regarding the aforementioned key constructs of the minimality condition. Further, because subprocess models are arranged in a level hierarchy (cf. Reijers et al. 2011), the variable "model level" is considered. This allows for differentiating whether "minimality" is to be judged regarding a certain model level or the holistic decomposition.

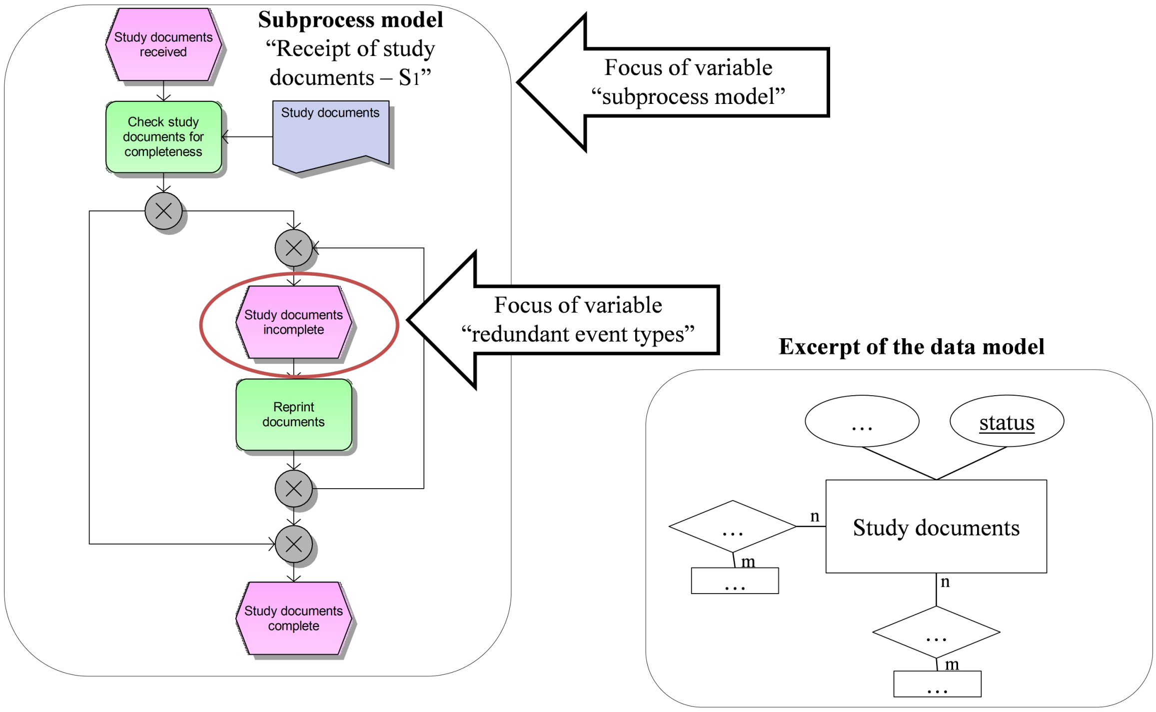


Figure 13: Example of the minimality condition

Figure 13 provides an example of a simple subprocess model "receipt of study documents—S_1". After a student has received the study documents, which characterize a "thing" in the BWW ontology and are represented as a data object type in the model (see section 2.3.2.3), these are checked for completeness. In the process model, the study documents are reprinted in case they are incomplete. Accordingly, in a data model, the data object type "study documents" may be depicted with a corresponding attribute (state variable) "status" amongst others, indicating whether the documents are complete or incomplete and need to be reprinted. Nevertheless, considering a real world situation, there might be an internal check by the university administration assuring that the study documents are definitely complete before they are handed over. Hence, the attribute value "incomplete" representing a potential "state" of the thing "study documents" will not occur in any process instance. The event type "study documents incomplete" would be a redundant event type in the process model.

The variable “redundant event types” explicitly focuses such event types and the variable “subprocess model” reverts to the whole model as shown (in our example in Figure 13 “receipt of study documents—S_1”). While our example focuses a subprocess model on a certain model level only, the model levels of a decomposition are generally considered by the variable “model level”.

Step 4: In total, we propose two metrics to measure “minimality” building on above-mentioned variables. The first metric determines the average ratio of “redundant event types” across all subprocess models (S_i) on a certain model level (M_j). By that, it becomes obvious how many event types are actually to be considered as redundant regarding all event types found across the subprocess models on a certain model level. This first metric is shown in Table 13.

$$Metric\ 1(M_j)=\frac{\sum_{i=1}^{\lvert S\rvert}\frac{number\ of\ redundant\ event\ types\ in\ subprocess\ model\ S_i}{number\ of\ event\ types\ in\ subprocess\ model\ S_i}}{number\ of\ subprocess\ models\ \lvert S\rvert\ on\ model\ level\ M_j\ considered}$$
Calculation/Measurement
This metric counts the “redundant event types” of a subprocess model S_i on a model level Mj. This number is divided by the total number of event types of that subprocess model Si. This is done for all subprocess models of the model level 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. Thus, 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.).
Interpretation
The final result represents the average ratio of redundant event types (regarding all event types modeled) for subprocess models on a model level M_j.

*Table 13: Average ratio of “redundant” event types for subprocess models on a specific model level—**metric 1***

In the above subprocess model (see Figure 13), there are three “event types” while one of these is “redundant”. The nominator of metric 1 thus has a value of “0.33”. In case the subprocess model shown would be the only one on that particular model level M_j, the overall value for metric 1 for that level would be “0.33” as well.

The second metric follows the same idea as metric 1 and is presented in Table 14. However, at this point, all model levels of a decomposition are considered. Thus, the focus is shifted from a particular model level to the holistic decomposition. Principally, different users might consider subprocess models on a specific level exclusively (e.g., management focusing on M_1, developers focusing on M_3, etc.). Considering the variety of potential model users in a company, it is important that the requirements of the decomposition conditions are not only carefully followed regarding certain selected model levels but also regarding the decomposition as a whole. By metric 2, an indicator for the overall adherence to minimality is received.

$Metric2 = \frac{\sum_{j=0}^{\lvert M \rvert} \frac{\sum_{i=1}^{\lvert S \rvert} \frac{\text{number of redundant event types in subprocess model } S_i}{\text{number of event types in subprocess model } S_i}}{\text{number of subprocess models } \lvert S \rvert \text{ on model level } M_j}}{\lvert M \rvert}$
Calculation/Measurement
In summary, the value for metric 2 is received by applying metric 1 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.
Interpretation
The value shows the average ratio of redundant event types (regarding all event types modeled) for subprocess models of a model level across a decomposition.

Table 14: Average ratio of "redundant" event types for subprocess models across model levels of a decomposition—metric 2

In summary, we propose two metrics to determine the "minimality" of decomposed process models. In both cases, the center of attention is directed towards redundant event types that need to be identified in a model. 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, as mentioned before, the challenge of applying the metrics lies in finding redundant event types of an eEPC model because process knowledge is required for that particular purpose.

2.3.3.3 Determinism

According to *Weber* (1997, S. 154), determinism is defined as follows: "*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*".

<u>Explanation and transfer to eEPCs:</u>

Step 1: The major constructs of this definition are “internal” and “external events”. Internal events are triggered by things in the (sub-)system itself, while things in the environment of a (sub-)system can cause external events (Weber 1997). Events lead to a change of state in a (sub-)system (Weber 1997). Following the definition, internal events need to be precisely defined. Hence, in case of a “change of state”, the emerging state of a state variable has to be determined unambiguously (cf. Weber 1997). As indicated in section 2.3, the term “event” needs to be treated with caution regarding a transfer of the decomposition condition to eEPCs. A change of state is expressed by the triple “event type → function type → event type” in the eEPC with the function type representing the transformation (Green & Rosemann 2000), (Green & Rosemann 2001). Thus, no uncertainties of the state that a process instance takes after the execution of a function must occur. Correspondingly, OR connectors should be avoided as already suggested by Mendling et al. (2010) or van der Aalst et al. (2002).

In addition, the definition reveals “external events”, which ought to be recognized by a modeler (cf. Weber 1997). Generally, external events are caused by circumstances that are beyond the scope of a company’s influence (Weber 1997). In this regard, an external event can be the “receipt of a customer order” or a “server crash at a supplier” for instance (cf. Weber 1997). Considering the second example (server crash), it becomes obvious that precisely specifying the impact of external events on process execution is challenging. Profound knowledge of an organization’s environment (e.g., suppliers, stakeholders, cooperation partners, etc.) is required for that purpose. Nonetheless, for all external events that an enterprise is aware of compensating activities on how to handle potentially unforeseen situations (e.g., server crash at supplier) can be defined to enhance the corresponding process models. Therefore, considering external events—as event types—is an important task for depicting well-defined process models and fulfilling the determinism condition. If a process model shows few external events, the corresponding process may either have been affected by environmental influences to a minimal degree only or the modeler may not have fully recognized the external circumstances.

Based on these findings, we propose the following interpretation of determinism for eEPCs: “*To fulfill the determinism condition, the conditions specifying which event(s) follow(s) the execution of a function during process execution need to be precisely defined. Rules have to be established and the event types to be labeled appropriately. OR connectors should be* *avoided. In addition, subprocesses are built around external events”* (Johannsen & Leist 2012b, S. 277).

<u>Derivation of metrics to assess “determinism”:</u>

Step 2: First, the “OR connector” is the central construct of this interpretation of “determinism”. As mentioned, OR splits have a high degree of imprecision and their existence in an eEPC model does not harmonize with the basic ideas of determinism. Therefore, the occurrence of OR connectors in a process model is to be checked by a corresponding metric and taken as a measure for determining the degree of conformance with the decomposition condition. In that context, OR splits negatively affect the metric value.

Second, “external events” are emphasized by the above definition. However, *Weber* (1997) explains that identifying external events is quite complex and a process accompanied by uncertainty. A process model can therefore acknowledge external events only to the degree as to which a modeler is able to anticipate the corresponding circumstances. Nevertheless, to judge determinism, the number of missing “external” event types in a decomposed process model—of which the model user is aware—represents a helpful indicator of whether the process model has accurately addressed environmental factors impacting process execution or not.

Step 3: “OR split” operations thus make up a first variable to be captured by a metric for assessing the fulfillment of “determinism”. Again, the model level is a further variable to be considered at this point because it enables to focus on the holistic decomposition or a certain model level only. As mentioned, this aspect is relevant considering the heterogeneous model user groups in a company. Further, the variable “subprocess model” is relevant to enable to accumulate results across subprocess models.

Additionally, a metric has to acknowledge the number of “missing” event types indicating external events, which represents a further central variable. However, to consider the size of the process model adequately (e.g., Mendling et al. 2007), all “external” event types (missing and explicitly modeled) should be acknowledged for the purpose of normalization (e.g., Hinrichs 2002), which is assured via the variable “external event type modeled”. As event types are spread across several subprocess models of a decomposition, the number of subprocess models is relevant in this context as well.

Figure 14 provides an example reverting to an excerpt of a subprocess model "check creditworthiness—S_1" of a financial institution. The creditworthiness of a customer is checked as soon as a customer's inquiry (e.g., for a loan) is received. This is done by a bank employee with the help of the CRM system that is run at an external provider as a software-as-a-service (SaaS) solution. The creditworthiness is either confirmed or denied. Figure 14 shows two alternative subprocess models of the real world situation described.

First, following the above specification of determinism, no uncertainties of the state a process instance takes after performing the creditworthiness check for a particular customer may occur. However, in the left alternative of the subprocess model, an OR split is found, which leads to ambiguity considering the further flow of the process at that point (cf. Mendling et al. 2010). The variable "OR split" explicitly focuses these split operations as these are not compatible with above described ideas of internal events according to *Weber* (1997).

Second, the external event types in a process model, hinting at external events in the sense of the BWW ontology, are emphasized by the variable "missing external event type". For instance, a server crash at the provider of the CRM system, run as a SaaS solution, may occur, which is beyond the influence of the financial service provider. Hence, the non-availability of the CRM system is an external factor influencing process execution. However, in the left subprocess model this circumstance is not modeled thus violating the determinism condition. An alternative design of the model considering this external influence is shown by the right subprocess model in Figure 14, which aligns with the real world situation. The "external event" (cf. Weber 1997) of a CRM system crash is acknowledged by the right control flow branch emerging from the XOR split and is initiated by the event type "CRM system crash occurred" representing an "external event type". The financial service provider may undertake efforts to handle the server crash, which is indicated by the function type "inform customer about delay", that can be seen in the rightmost branch of the alternative version (see Figure 14). Whereas the right alternative coheres to the determinism condition by adequately considering external events and the well-definition of internal events (no OR splits) (cf. Weber 1997), this does not hold true for the left model shown. Considering, the left subprocess model, the event type "CRM system crash occurred" is a "missing external event type" and captured by our variable.

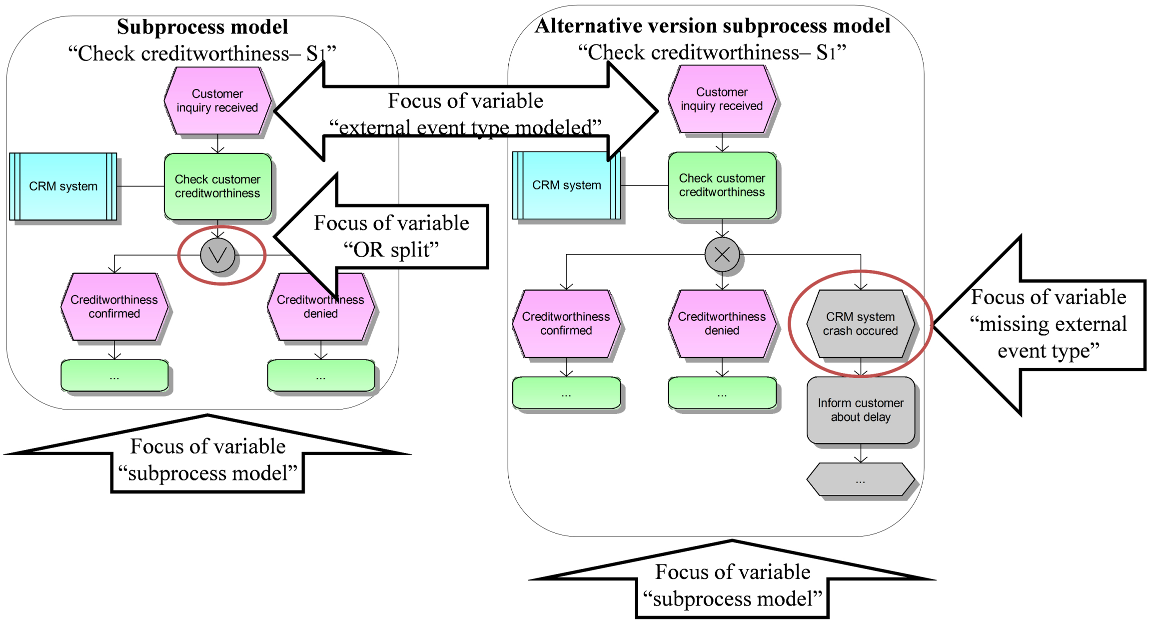


Figure 14: Example of the determinism condition

Step 4: Considering step 3, we propose three metrics to judge "determinism". Thus, both the key aspects of the determinism condition, namely the well-definition of internal events as well as the recognition of external events, are covered reverting to above variables.

The first metric, shown in Table 15, relates the number of OR splits to the total number of split operations (XOR, OR, AND) for all subprocess models (S_i) on a model level (M_j) and calculates an average value for the subprocess models.

Considering the left subprocess model in Figure 14, there is one OR split operation. As this is the only split operation in the model excerpt shown, the nominator for metric 3 would have a value of "1/1". In case the subprocess model would be the only model on level M_j, the metric's overall value would be "1" as well.

$$Metric\ 3(M_j)=\frac{\sum_{i=1}^{\|S\|}\frac{number\ of\ OR\ splits\ in\ subprocess\ model\ S_i}{number\ of\ split\ operations\ in\ subprocess\ model\ S_i}}{number\ of\ subprocess\ models\ \|S\|\ on\ the\ model\ level\ M_j\ considered}$$
Calculation/Measurement
This metric counts the number of OR splits of a subprocess model S_i on a model level 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 model level to achieve an average ratio.
Interpretation
The metric value represents the average ratio of OR split operations in regards to all split operations for subprocess models on a model level M_j.

*Table 15: Average ratio of OR splits of subprocess models on a specific model level—**metric 3***

The second metric follows the same idea but aggregates the values across all model levels to come to a final value for the holistic decomposition thus enabling a quality assessment regarding the determinism of the decomposition. Table 16 gives an overview.

$$Metric4=\frac{\sum_{j=0}^{\|M\|}\frac{\sum_{i=1}^{\|S\|}\frac{number\ of\ OR\ splits\ in\ subprocess\ model\ S_i}{number\ of\ split\ operations\ in\ subprocess\ model\ S_i}}{number\ of\ subprocess\ models\ \|S\|\ on\ the\ model\ level\ M_j\ considered}}{\|M\|}$$
Calculation/Measurement
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.
Interpretation
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.

*Table 16: Average ratio of OR splits of subprocess models across all model levels of a decomposition—**metric 4***

The third metric deals with external events and analyzes whether these were properly considered in the decomposition or not. Note that a metric that only assesses the coverage of "external events" for a certain model level would not be particularly practical regarding the fact that the assignment of information to subprocess models or model levels is rather subjective (cf. Reijers et al. 2011). Therefore, in one decomposition the information sought after might be found on level 1 (M_1) whereas in an equivalent decomposition the same information could be depicted on level 2 (M_2) accordingly. Thus, a holistic perspective on the decomposition is required for judging the fulfillment of determinism regarding the recognition of external events. Accordingly, we propose metric 5 as presented in Table 17.

$Metric\ 5=\frac{\sum_{j=0}^{\|M\|}\sum_{i=1}^{\|S\|}\frac{number\ of\ missing\ external\ event\ types\ in\ subprocess\ model\ S_i}{number\ of\ external\ event\ types\ modeled+number\ of\ missing\ external\ event\ types\ in\ subprocess\ model\ S_i}}{total\ number\ of\ subprocess\ model}$
Calculation/Measurement
Metric 5 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 modeled "external event types". 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 (across all model levels).
Interpretation
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. Acknowledge that the average value considers all subprocess models across all model levels without differentiating between model levels.

*Table 17: Ratio of missing "external event types" of a decomposition—**metric 5***

In Figure 14 (left subprocess model), the external event type "CRM system crash occurred" would be counted by the variable "missing external event types". The start event type "customer inquiry received" is an external event type as well, since the subprocess is triggered on the initiative of an external customer sending an inquiry. Thus, there is one explicitly modeled external event type and one missing external event type. Accordingly, the nominator of metric 5 has a value of "1/2". This calculation is similarly done for all subprocess models across the decomposition, the values are aggregated and the result is divided by the overall number of subprocess models.

Summing up, the first two metrics developed focus on the requirement that internal events of a decomposition need to be well-defined (cf. Weber 1997). Correspondingly, OR split operations in the subprocess models are the center of attention. These can be identified quite simply without requiring domain knowledge of the real world situation modeled. 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 5. The calculation underlies a certain subjectivity as only those external events can be used as a calculation base the user is actually aware of. All metrics have a value of "0" in an ideal decomposition.

2.3.3.4 Losslessness

According to *Weber* (1997, S. 155), a "*decomposition is good only if every hereditary state variable and every emergent state variable in a system is preserved in the decomposition*".

Explanation and transfer to eEPCs:

Step 1: This definition of losslessness explicitly emphasizes hereditary and emergent state variables that are described by properties according to the BWW representational model (cf. Weber 1997). A hereditary property can be assigned to a component of a "thing" (e.g., "long-term storage capacity" of a computer) whereas an emergent property is only meaningful if the thing (e.g., person, computer) as a whole (including all levels of the hierarchy) is considered (Weber 1997). The identification of emergent properties is quite complex and requires domain knowledge (Weber 1997). The "customer's level of satisfaction" in the context of the "customer order handling process" is an example of an emergent property. Following *Green and Rosemann* (Green & Rosemann 2000), the eEPC has drawbacks in representing the particular property types as introduced by *Weber* (1997). For instance, there are no means for expressing hereditary properties in special.

Due to this challenge of differentiating between hereditary and emergent properties, *Weber* (1997) suggests a more manageable interpretation of system decomposition by emphasizing that no properties should get lost at all. This sets aside the differentiation between the property types when decomposing a system or—for our purpose—a process model. Properties are represented by attributes of data object types in a data model and can be related to event types in eEPCs. Thus, all "non redundant" (cf. Weber 1997) event types that are required for visualizing a real world situation must be captured in a corresponding eEPC model.

Hence, considering decomposed eEPC models, losslessness means that no "*information must get lost during decomposition. The event types being related to attributes describing properties play a decisive role and should be preserved during decomposition. Function types associated to these event types together with modeling constructs from other ARIS (Architecture of Integrated Information Systems) views must be reflected upon—whether they can be found in* *corresponding subprocess models of a decomposition or not—as well”* (Johannsen & Leist 2012b, S. 277).

Derivation of a metric to assess “losslessness”:

Step 2: Event types are the key construct of the above definition. Non redundant event types (cf. Weber 1997) need to be fully preserved in the decomposed model. In case an initial flat model was decomposed, all “non redundant event types” of the original model must be found in the corresponding subprocess models afterwards. Otherwise, the losslessness condition is violated. However, a violation also occurs in case a decomposed model is created from scratch (cf. Davis & Brabänder 2007) and required non redundant event types are neglected by the modeler (cf. Hoffmann et al. 1993). As mentioned, these event types indicate a change of state for a data object—depicted in a data model—that participates in the process.

Step 3: The “missing non redundant event types” thus represent the central variable of a metric to judge losslessness. For considering the size of the process model for the purpose of normalization, all event types are to be acknowledged. Therefore, in addition to “non redundant event types” in particular, all types of events of the process model need to be referred to by an adequate metric. In general, the set of event types of a process model comprises “non redundant” and “redundant” event types.

Figure 15 shows an example referring to the aforementioned subprocess model “check creditworthiness—S_1” again. The left model violates the condition, whereas the right alternative coheres to losslessness. In a real world situation, the customer may be creditworthy or not. Thus, in a complementary data model, an attribute “creditworthiness” might exist for the customer taking the values “confirmed” or “denied”. By these, the state of the customer, who is a “thing” following the BWW ontology (cf. Weber 1997), is expressed. However, in the left model this circumstance is not properly considered, because only the case of a creditworthy customer is described. Considering the alternative version of the model (right model), a customer is informed by the financial institution in case the creditworthiness is denied, and the customer inquiry is archived, as can be seen in the rightmost branch. According to the BWW ontology (cf. Weber 1997), the state variable (attribute) “creditworthiness” of the thing “customer" would take the value of "denied" and the state variable (attribute) "status" of the thing "inquiry" the value "archived". This can be depicted via a corresponding data model and the entity types "customer" and "inquiry" (see Figure 15).

The variable "missing non redundant event types" explicitly focuses those event types that would be required to adequately capture the real world situation but are missing in the process model. In this respect, the event types indicate the states of state variables (attributes) particular "things" may take during process execution.

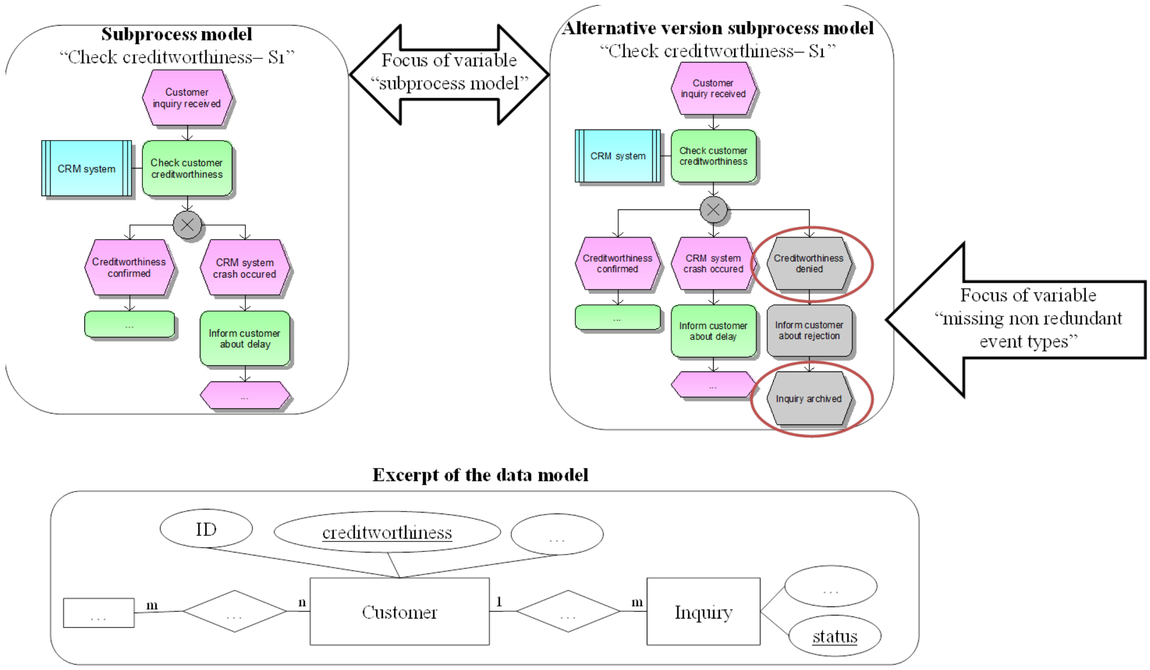


Figure 15: Example of the losslessness condition

Step 4: For judging losslessness, we propose a metric in Table 18 that counts the number of "non redundant event types" that are neglected by the modeler when decomposing an original flat model or designing a decomposed model from scratch. These event types are called "missing non redundant event types" in the following. Such event types would have been required in the process model for visualizing the real world situation correctly. However, they cannot be found in the model.

For the purpose of normalization, we suggest to set the number of "missing non redundant event types" in relation to the sum of all event types explicitly modeled and the "missing non redundant event types". Thus, the metric analyzes the missing event types in relation to those a modeler has actually considered and to event types that should have been modeled. This is done for all subprocess models (S_i) across all model levels (M_j) of a decomposition. The resulting value is normalized dividing it by the total number of all subprocess models of a decomposed model.

Again, losslessness cannot be adequately assessed for a particular model level only, as it is up to the modeler to assign information to subprocess models on certain model levels (cf. Reijers et al. 2011). Therefore, the decomposition needs to be considered from a holistic perspective in terms of losslessness.

$Metric\ 6=\frac{\sum_{j=0}^{\lvert M\rvert}\sum_{i=1}^{\lvert S\rvert}\frac{number\ of\ missing\ non\ redundant\ event\ types\ in\ subprocess\ model\ S_i}{number\ of\ event\ types\ in\ subprocess\ model\ S_i+number\ of\ missing\ non\ redundant\ event\ types\ in\ subprocess\ model\ S_i}}{total\ number\ of\ subprocess\ models\ of\ a\ decomposition\ across\ all\ model\ levels\ M_j}$
Calculation/Measurement
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 modeled 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).
Interpretation
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 modeled for subprocess models of a decomposition. Acknowledge that the average value considers all subprocess models across all model levels without differentiating between model levels.

*Table 18: Ratio of missing "non redundant event types" of a decomposition—**metric 6***

Taking the left model in Figure 15 as an example, there are two missing non redundant event types, which are unveiled by the right alternative. Further, the model excerpt (left model – Figure 15) comprises four explicitly modeled event types. Hence, the nominator of metric 6 takes the value of "2/(4+2)". Considering the overall value of the metric, this number is to be divided by the total number of subprocess models of the holistic decomposition.

In summary, profound 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 situation. User knowledge is mandatory in that context. 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 (cf. Weber 1997). Again, ideally, the application of the metric results in a value of "0".

2.3.3.5 Minimum coupling

A further condition of the good decomposition model is "minimum coupling": "*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, S. 161).

Explanation and transfer to eEPCs:

Step 1: The major constructs of this condition are "system", "coupling", and "input". The notion of a "system" was already dealt with in the context of the minimality condition (section 2.3.3.2). According to *Weber* (1997), "input" refers to all states of a system that arise due to actions of the environment. "Coupling" means that things of a system are related affecting each other (Weber 1997). As *Green and Rosemann* (2000) point out, there are no indications on how to express "input" or "coupling" in eEPC models. Therefore, a one-to-one mapping of these constructs to eEPC modeling is not possible. However, the basic idea of the "minimum coupling condition", namely to minimize "the total action of all environmental things on each subsystem in the decomposition" (Weber 1997, S. 159), can be drawn upon to derive characteristics of a well done decomposition.

First, the number of start event types for subprocess models is to be kept minimal because these point to external events (Green & Rosemann 2000). External events lead to a change of state for a (sub-)system and thus characterize a major environmental impact (Weber 1997). A reduction of start event types can be achieved by merging subprocess models for instance. However, further event types in a process model may be given hinting at external events (e.g., "crash of CRM system") as described in section 2.3.3.3.

Second, there should be minimum coupling between subprocess models. In this regard, different forms of coupling were identified for process models (e.g., Vanderfeesten et al. 2008c).

However, most of these suggestions are incommensurable with Wand and Weber’s primary intention of coupling, which builds on input states of a thing caused by environmental actions (cf. Weber 1997). Therefore, we specify minimum coupling in terms of eEPCs by focusing on data object types:

Two subprocess models “1” (S_1) and “2” (S_2) are coupled if a function type of S_2 receives a data object type as input that is produced as output by a function type in S_1 (cf. Johannsen & Leist 2012b). The data object type is thus shared between the subprocesses indicating that these are coupled. For example (see Figure 16), a model of a complaint management process comprises the subprocess models “complaint receipt” and “complaint handling” (cf. Stauss & Seidel 2004). During “complaint receipt”, a customer’s complaint—which represents a data object type—is documented in a complaint management software and is then transmitted to a responsible employee as output of a function type “transmit complaint to employee responsible”. During the “complaint handling” process, the complaint is received as input by a function type “analyze complaint reason”. The data object type “complaint” represents both an output data object type and an input data object type for two different subprocess models (S_1 and S_2). Therefore, these subprocess models are coupled accordingly. Following the minimum coupling condition, there should be minimal interchange of data object types between subprocess models. This interpretation of minimum coupling comes close to the idea of *Reijers and Vanderfeesten* (2004) to focus on common data elements of activities to determine a model’s coupling degree (“process coupling”).

Hence, the specification of the minimum coupling condition for eEPC models demands that “*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*” (Johannsen & Leist 2012b, S. 277).

Derivation of metrics to assess “minimum coupling”:

Step 2: The specification of minimum coupling for eEPCs 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.

As described above, 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. 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. These are ideally represented as start event types in an eEPC model (cf. Green & Rosemann 2000) but can also be modeled as intermediate event types.

Step 3: The variables derived from these central constructs to develop corresponding metrics are "coupled pair of function types" and "external event type". For normalization purposes, also the number of potential "pairs of coupled function types" is part of a corresponding metric. It expresses all possible constellations of coupling between function types that are possible. Further, the subprocess models (S_i) and model levels (M_j) are required once again to allow focusing single model levels or the holistic decomposition.

Take Figure 16 as an example, which shows the two subprocess models "complaint receipt—S_1" and "complaint handling—S_2". For exemplification purposes, 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 "analyze complaint reason". Further, the output data object type "customer letter" is given. Since both function types "transmit complaint to employee responsible" and "analyze complaint reason" stem from different subprocess models, one "coupled pair of function types", namely "*(transmit complaint to employee responsible; analyze 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. These are in the center of attention of the variable "external event type".

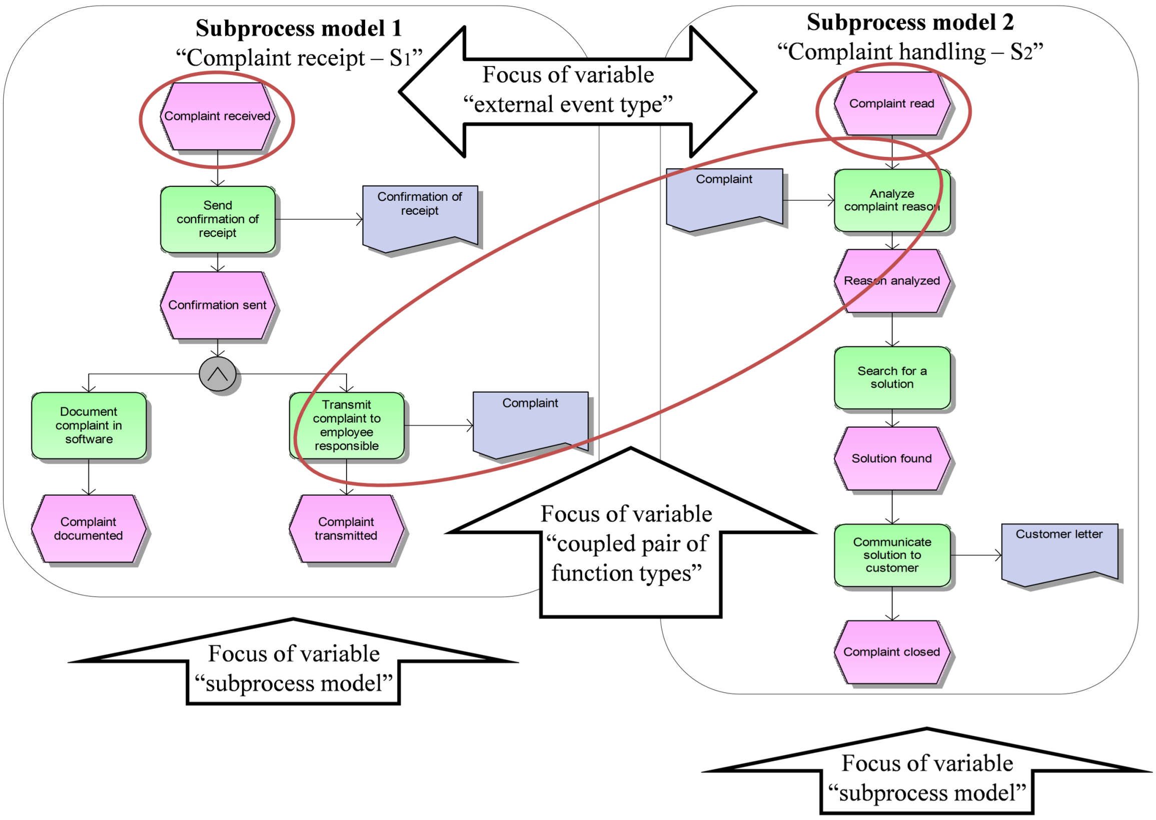


Figure 16: Example for the minimum coupling condition

Step 4: 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.

The first metric (metric 7—Table 19) addresses coupling between function types and focuses a certain model level of a level hierarchy 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 number resulting is divided by the total number of "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.

$\textit{Metric 7}(M_j) = \frac{\textit{number of coupled pairs of function types across all subprocess models on a model level } M_j}{\textit{number of potential pairs of coupled function types on a model level } M_j}$
Calculation/Measurement
This metric counts the number of "coupled pairs of function types" across all subprocess models S_i of a model level M_j and divides this number by the total number of "potential pairs of coupled function types" on that model level.
Interpretation
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).

*Table 19: Ratio of "coupled pairs of function types" across subprocess models on a model level—**metric 7***

Reverting to Figure 16, we acknowledged one pair of coupled function types as described above. 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 stemming from other subprocess models. This is done for all function types across all subprocess models for the model level under consideration. In Figure 16, 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. In total, this makes 18 "potential pairs of coupled function types". The value for metric 7 regarding Figure 16 thus is "0.056 (=1/18)".

$\text{Metric } 8 = \frac{\sum_{j=0}^{\lvert M \rvert} \frac{\textit{number of pairs of function types across all subprocess models on a model leve}}{\textit{number of potential pairs of coupled function types}}}{\lvert M \rvert}$
Calculation/Measurement
This metric counts the number of "coupled pairs of function types" across all subprocess models S_i of a model level M_j and divides this number by the total number of "potential pairs of coupled function types" on that model level. The result is divided by the total number of model levels of a decomposed process model.
Interpretation
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.

*Table 20: Average ratio of "coupled pairs of function types" for a model level—**metric 8***

All values calculated for the model levels of a decomposition via metric 7 can then be aggregated across all model levels to come to an overall value for the decomposed model as shown by metric 8 in Table 20.

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). Ideally, each subprocess model has one start event type only. The average number of start event types for subprocess models on a model level can be calculated by counting the start event types for all subprocess models and dividing it by the total number of subprocess models on that level. 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 9 (Table 21).

$Metric\ 9(M_j)=\frac{\sum_{i=1}^{\lvert S\rvert} number\ of\ external\ event\ types\ of\ subprocess\ model\ S_i}{\lvert S\rvert}$
Calculation/Measurement
This metric counts the number of external event types across all subprocesses S_i of a model level M_j and divides this number by the total number of subprocesses |S| on that model level.
Interpretation
The metric answers the question of how many external event types subprocess models on a model level M_j have in average.

*Table 21: Average number of external event types for subprocess models on a model level —**metric 9***

For example, there are two start event types in Figure 16 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 16, the external input is minimal.

To attain a value for the holistic decomposition, the values for metric 9 are aggregated for all model levels and then divided by the total number of all model levels existing, a procedure captured by metric 10 (Table 22). Both the metrics 9 and 10 result in values of "1" or higher, while "1" represents a perfect decomposition in that context.

$$Metric\ 10 = \frac{\sum_{j=0}^{\lvert M \rvert} \frac{\sum_{i=1}^{\lvert S \rvert} numer\ of\ external\ evvent\ types\ of\ subprocess\ model\ S_i}{\lvert S \rvert}}{\lvert M \rvert}$$
Calculation/Measurement
Metric 10 aggregates the values received by applying metric 9 across all model levels and divides the resulting sum by the total number of model levels |M|.
Interpretation
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 22: Average number of external event types for subprocess models of the decomposition—**metric 10***

In summary, minimum coupling is determined on the base of “coupled pairs of function types” as well as “external event types”. For both concepts, 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 7 and 8. This enables an automatized assessment of the coherence of a process model to the minimum coupling definition as presented. 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 –, subprocess models can be merged by a modeler, even though completely reducing the interaction between subprocess models may not be appropriate under all circumstances as the modeler might come up with one process model only, which counteracts the intention of decomposition.

2.3.3.6 Strong cohesion

Strong cohesion is defined by *Weber* (1997, S. 163) the following way: “*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, S. 42).

<u>Explanation and transfer to eEPCs:</u>

Step 1: The main constructs of the definition are “output” and “cohesion”. Cohesion is the counterpart of coupling and—in terms of IS—refers to the number of tasks performed by a single module within a software procedure (Weber 1997). Output specifies that things in the environment of the system are affected by the system itself (Weber 1997). However, there are no constructs to express these concepts in an eEPC model directly (cf. Green & Rosemann 2000).

Cohesion has not been explicitly dealt with in process modeling literature yet. Nevertheless, one particular interpretation is found in *Reijers and Vanderfeesten* (Reijers & Vanderfeesten 2004), which can be transferred to modeling with eEPCs and brought in line with Wand and Weber’s idea of strong cohesion. Therefore, data object types produced by function types as output or received as input are the centerpiece to reflect a process model against this condition. All function types generating a particular output (e.g., document type) by processing a set of input data object types must be captured in a subprocess model. Hence, the output data object types in a subprocess model functionally depend on the input data object types.

Consider Figure 17 as an example. Two subprocess models “complaint analysis” (S_3) and “complaint resolution” (S4) of a more comprehensive complaint management process model are depicted.

In S_3, the input data object types “complaint”, “confirmation of receipt”, and “analysis results” exist. “Complaint” and “confirmation of receipt” are required for producing the output data object type “analysis results”. The “complaint report” is created on the base of the “analysis results” in turn. In S_4, the input data object types “analysis results”, “complaint report” and “complaint” are differentiated, which are required for producing the output data object types “individual solution” and the “customized standard solution”.

This delineation of subprocess models violates the strong cohesion condition as all sets of output data object types in the subprocess models do not exclusively depend on the set of input data object types. For example, the “analysis results” and the “complaint” are elements of both sets of input data object types regarding subprocesses S_3 and S_4, which, however, contradicts the idea of the strong cohesion condition. An exclusive functional dependency of output data object types regarding input data object types—in the sense of the strong cohesion condition—requires the intersection of both sets of input data object types to be empty for all subprocess models. Recognize that this interpretation comprises both central notions of Wand and Weber’s original condition, namely “cohesion” and “output”, equally (cf. Weber 1997).

The strong cohesion condition for decomposed eEPC models is thus: “*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”* (Johannsen & Leist 2012b, S. 277).

<u>Derivation of metrics to assess “strong cohesion”:</u>

Step 2: Following the definition of “strong cohesion” for eEPCs, major emphasis is placed on input data object types, which thus represent a mandatory element of a corresponding metric. A violation of strong cohesion is given in case an input data object type can be found as input to a function type in more than one subprocess model (e.g., “analysis results” in Figure 17). In case two function types originate from different subprocess models but have the same input data object type, they make up a “duo of function types” in the following (e.g., “*(generate complaint report; search database for predefined solutions)”*). Further, “subprocesses” and “model levels” are key constructs of the definition as presented.

Step 3: Based on these insights, “duos of function types” – in the above sense—represent a central variable to calculate the coherence with the strong cohesion condition using an adequate metric. As normalization is striven for, the number of potential “duos of function types” – in terms of the strong cohesion condition—is required as well. It indicates all possible and conceivable constellations of function type duos. Additionally, the subprocesses (S_i) and model levels (M_j) enable to focus on single model levels or the holistic decomposition once again.

Consider the example shown in Figure 17. The function types “generate complaint report” and “search database for predefined solutions” share the input data object type “analysis results” and thus make up a duo “*(generate complaint report; search database for predefined solutions)”*. A further duo is “*(analyze complaint; develop solution)”* so that there are two corresponding duos in total. These are considered by the variable "duos of function types" as defined.

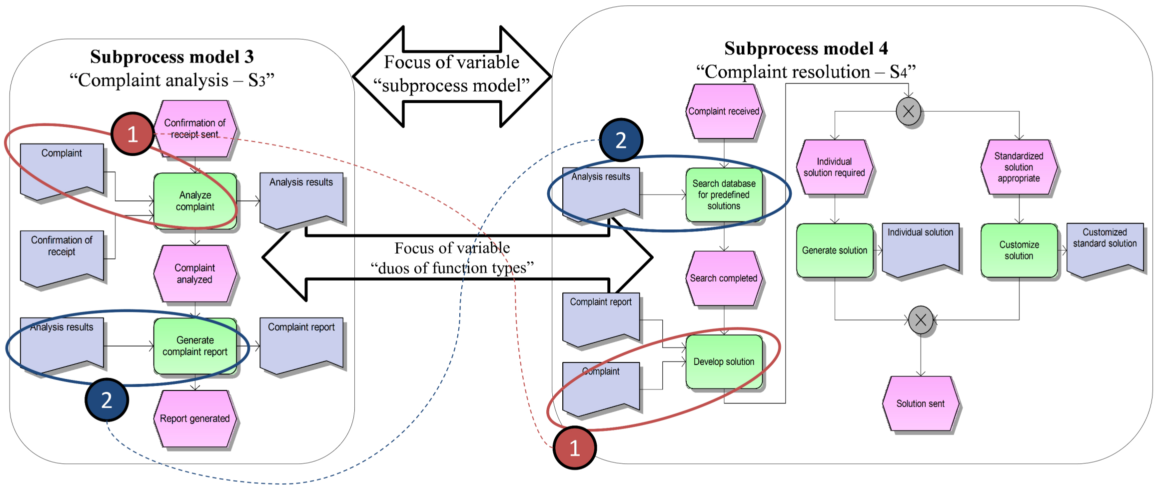


Figure 17: Example of the strong cohesion condition

Step 4: We introduce two metrics for assessing the degree as to which a decomposition adheres to the strong cohesion condition based on the above-mentioned variables. The first metric, as shown in Table 23 (metric 11), counts the amount of "duos of function types" that share a common input data object type but stem from different subprocess models. This number is then related to the number of potential "duos of function types" on a certain model level. An ideal metric value is "0".

$Metric\ 11(M_j)=\frac{number\ of\ duos\ of\ function\ types\ sharing\ a\ common\ input\ data\ object\ type\ across\ all\ subprocess\ models\ on\ a\ model\ level\ M}{number\ of\ potential\ duos\ of\ function\ types\ sharing\ a\ common\ input\ data\ object\ type\ on\ a\ model\ level\ M_j}$
Calculation/Measurement
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") 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.
Interpretation
The metric shows the ratio of function types sharing a common data input type in regards to all possible constellations of function types on a specific model level M_j.

*Table 23: Ratio of "duos of function types" sharing a common input data object type across subprocess models on a model level—**metric 11***

In Figure 17, S_3 holds two function types, S_4 comprises four function types and two duos of function types are given (1: "*(analyze complaint; develop solution)*" and 2: "*(generate com-*

plaint report; search database for predefined solutions)"). Thus, "8 (=4*2)" duos of function types in terms of strong cohesion can be established. Considering the example, the application of the metric results in a value of "0.25".

It needs to be mentioned that, in case two function types from different subprocesses shared more than one common data object type, the duo would only be counted once due to reasons of complexity reduction.

Once again, this perspective on a single model level is extended onto all model levels accordingly by metric 12 (Table 24).

$$\text{Metric } 12 = \frac{\sum_{j=0}^{|M|} \frac{\text{number of duos of function types sharing a common input data object type across all subprocess models on a model level } M}{\text{number of potential duos of function types sharing a common input data object type on a model level } M_j}}{|M|}$$
Calculation/Measurement
The ratio of the "duos of function types" sharing a common input data object type across subprocess models on a model level (see above) is calculated for all model levels, the partial results are aggregated and the result is divided by the total number of model levels.
Interpretation
The value stands for the average ratio of function types sharing 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 24: Average ratio of "duos of function types" sharing a common input data object type for a model level—**metric 12***

In summary, both metrics focus on structural aspects of a decomposition and analyze 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".

2.3.3.7 Summary

Following the GQM approach, twelve metrics were defined for measuring the coherence of a decomposed business process model with the decomposition conditions. Figure 18 provides an overview.

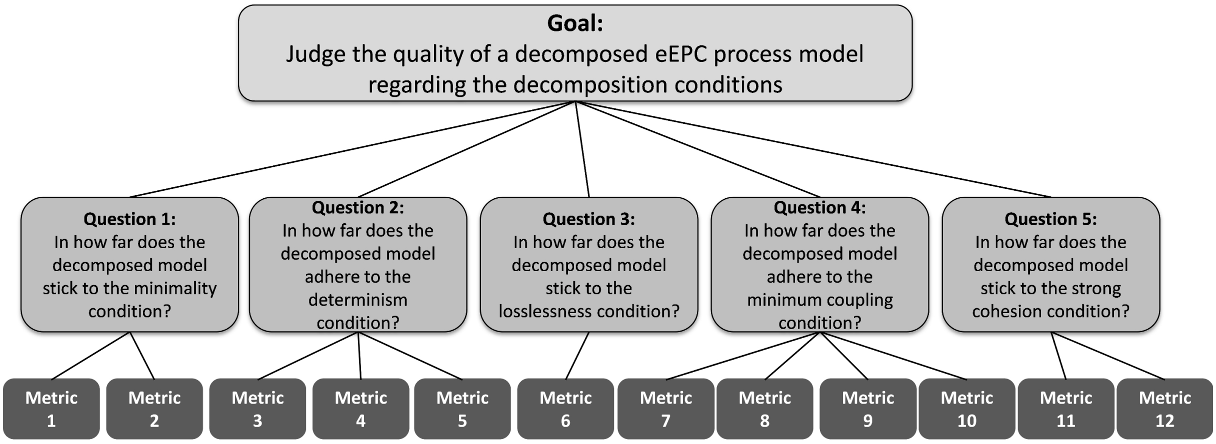


Figure 18: Assignment of metrics to cover the decomposition conditions

Most decomposition conditions allow to consider a decomposed model as a whole (metrics 2, 4, 5, 6, 8, 10 and 12) facilitating the comparison of alternative decompositions, but also enabling the analysis of model levels separately (metrics 1, 3, 7, 9 and 11). Judging losslessness (metric 6) requires a holistic view including all model levels and subprocess models to decide whether non redundant attributes have got lost.

Further, the calculation of the metrics 1, 2, 5, 6, 9 and 10 requires domain knowledge because the decomposed model is reflected against the real world circumstances. Therefore, the semantics is investigated closely whereas the results can be affected by subjective perceptions. The remaining metrics, on the contrary, focus on the structure of the decomposed model and the delineation of subprocess models in special. For all metrics defined, a low value indicates a strong coherence with the corresponding decomposition condition. Except for metrics 9 and 10, a value of "0" indicates a perfect decomposition in terms of the metrics, an ideal value for metrics 9 and 10 would be "1".

2.3.4 Application of the metrics

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. 2014a). In that context, three alternative decompositions "A", "B", and "C" of a process model depicting the "student enrollment process" at a German university, which violated the conditions to a varying degree, were created. The three decompositions were equivalent (cf. Weber 1997) as they described the identical functions to be performed during student enrollment. However, their arrangement in form of subprocess models is different. An overview of the material and the decomposed process model with detailed explanations is available at: *http://tinyurl.com/gqdoqy3*[5].

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. Hence, the delineation of subprocess models for these alternatives was similar. Alternative C violated all the decomposition conditions equally. In a previously conducted experiment, we found that models complying with the decomposition conditions are perceived as significantly easier to understand by model users (cf. Johannsen et al. 2014a). For this work, we reuse the process models and apply the newly developed metrics to see whether they indicate a difference as well. Figure 19 provides an abstract overview of how the decompositions differed from one another in terms of the number of subprocess models and their delineation. More details on the student enrollment process itself can be found under the above link.

The results of applying the metrics to the decomposed alternatives (A to C) of the student enrollment process are shown in Table 25. The calculation of each metric followed the steps as explained in sections 2.3.3.2 to 2.3.3.6. For demonstration purposes, we exemplarily show the calculation for metric 3 (determinism) in regards to alternative C and model level 1 (M_1), metric 9 (minimum coupling) in regards to alternative A and model level 1 (M_1) as well as metric 11 (strong cohesion) in regards to alternative C and the model level 3 (M_3).

5 Ein Abdruck des verlinkten Materials findet sich in Anhang 1

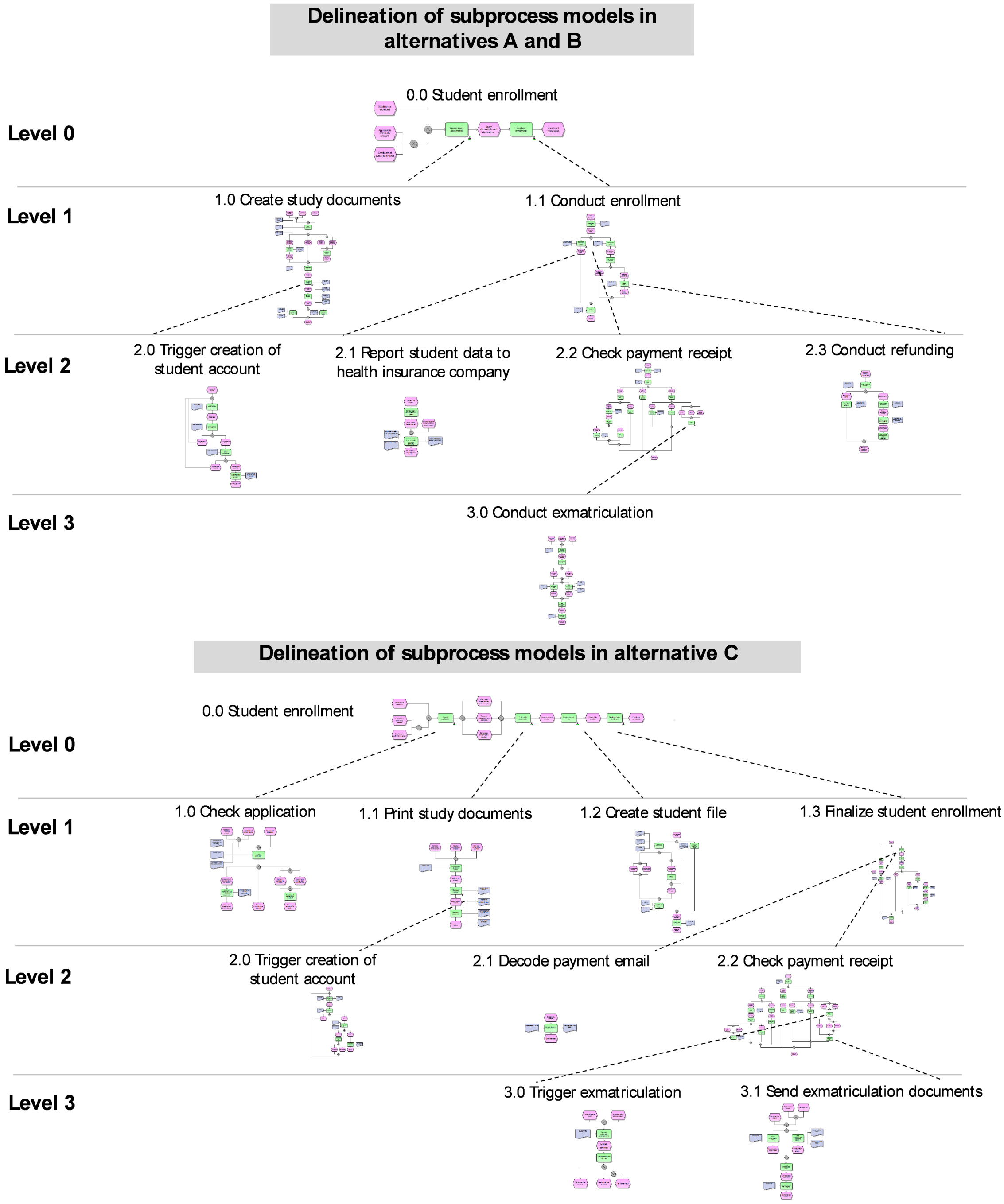


Figure 19: Overview of the decomposition alternatives in our use case

<u>*Calculation of metric 3 for M_1 in alternative C:*</u> On M_1, four subprocess models are given in alternative C (see Figure 19). The subprocess model "1.0 check application" has two split operations in total but no OR splits. While the subprocess model "1.1 print study documents" has no split operations at all, the model "1.2 create student file" is characterized by one OR

split and four split operations in total. The final subprocess model “1.3 finalize student enrollment” has three split operations but none of them is an OR split. Hence, the value for metric 3 is calculated as follows (see section 2.3.3.2):

$$Metric3(M_1)=\frac{\frac{0}{2}+0+\frac{1}{4}+\frac{0}{3}}{4}=0.063 \tag{1}$$

<u>*Calculation of metric 9 for M_1 in alternative A:*</u> In alternative A of the decomposition, two subprocess models “1.0 create study documents” and “1.1 conduct enrollment” are given on M1. The first subprocess model has three external event types in total, which indicate the influence of external factors on the subprocess. These are represented by start event types (“applicant is physically present”, “certificate of authority is given” and “deadline is not exceeded” – see supplementary material available at abovementioned link). The second subprocess model has one start event type indicating an external event (“study documents and information handed over”). Correspondingly, the application of metric 9 results in an overall value of “2” (see section 2.3.3.5):

$$Metric9(M_1)=\frac{\frac{3}{1}+\frac{1}{1}}{2}=2 \tag{2}$$

<u>*Calculation of metric 11 for M_3 in alternative C:*</u> On model level 3 (M_3) of alternative C, the data object type “student file” is a common input in both two subprocess models “3.0 trigger exmatriculation” and “3.1 send exmatriculation documents”. Correspondingly, the following duos of function types sharing this data object type can be found: “*(conduct automatic exmatriculation; create exmatriculation notice)*” and “*(conduct automatic exmatriculation; archive student file in registry)*”. As the first subprocess model contains two function types and the second one subsumes four function types, eight potential duos of function types can be established on M_3. The calculation of metric 11 thus is (see section 2.3.3.6):

$$Metric\ 11(M_3)=\frac{2}{4+2}=0.25 \tag{3}$$

Almost all of the metrics applied to alternative “A” result in ideal values for these particular measures (see section 2.3.3.7). However, some results emerge from applying the metrics that seem counter intuitive at first sight. They are highlighted by the coloration in Table 25 and further explained in the following.

First, the results for metrics 9 and 10 need explaining. Both metrics focus external event types of eEPC subprocess models. To minimize the external influence on a system as demanded by *Weber* (Weber 1997), each subprocess model should—ideally—only have one start event type initially, even though, the subprocess models in alternative “A” have more than one start event type on average. This holds true for each model level of the decomposition. That particular circumstance is also reflected by the values for the metrics 9 and 10. Essentially, it needs to be acknowledged that student enrollment is a complex process that is triggered by several external events, thus requiring the modeling of numerous start event types for the corresponding subprocess models. Building subprocess models around external events—to receive subprocess models with one start event type only—would not have been appropriate as interdependencies with the strong cohesion condition exist (cf. Johannsen & Leist 2012b). Correspondingly, the values for metrics 11 and 12 would have deteriorated in turn. Therefore, as the results indicate, a high quality decomposition might be given even if the application of the metrics does not deliver the ideal values.

Second, the results for metrics 3 and 4 (determinism) need clarifying. Regarding the application of metric 3, variant “B” performs worse than “C” for model levels M1 and M3, although alternative “B” was perceived as easier to understand by model users. Remember that metric 3 deals with the occurrence of OR splits, which should be avoided. Both alternatives, “B” and “C”, have only one OR split operation on M1. However, the delineation of subprocess models is different—two subprocess models are given in B and four subprocess models exist in C—which affects the results of the metric since an average value is calculated for subprocess models of one model level. The different number of subprocess models thus impacts the denominator of the metric. Similarly, this circumstance also affects the application of metric 4, aggregating the values for metric 3 across all model levels.

<table>
<tr><th>Condition</th><th colspan="2">Minimality</th><th colspan="3">Determinism</th><th>Losslessness</th><th colspan="4">Minimum Coupling</th><th colspan="2">Strong cohesion</th></tr>
<tr><th>Metric</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th></tr>
<tr><td colspan="13">Alternative A</td></tr>
<tr><td>Level 0 (M_0)</td><td>0</td><td rowspan="4">0</td><td>0</td><td rowspan="4">0</td><td rowspan="4">0</td><td rowspan="4">0</td><td>0</td><td rowspan="4">0</td><td>3</td><td rowspan="4">2.312</td><td>0</td><td rowspan="4">0</td></tr>
<tr><td>Level 1 (M_1)</td><td>0</td><td>0</td><td>0</td><td>2</td><td>0</td></tr>
<tr><td>Level 2 (M_2)</td><td>0</td><td>0</td><td>0</td><td>1.25</td><td>0</td></tr>
<tr><td>Level 3 (M_3)</td><td>0</td><td>0</td><td>0</td><td>3</td><td>0</td></tr>
<tr><td colspan="13">Alternative B</td></tr>
<tr><td>Level 0 (M_0)</td><td>0</td><td rowspan="4">0.049</td><td>0</td><td rowspan="4">0.148</td><td rowspan="4">0</td><td rowspan="4">0</td><td>0</td><td rowspan="4">0</td><td>3</td><td rowspan="4">2.063</td><td>0</td><td rowspan="4">0</td></tr>
<tr><td>Level 1 (M_1)</td><td>0.118</td><td>0.083</td><td>0</td><td>2</td><td>0</td></tr>
<tr><td>Level 2 (M_2)</td><td>0.079</td><td>0.177</td><td>0</td><td>1.25</td><td>0</td></tr>
<tr><td>Level 3 (M_3)</td><td>0</td><td>0.333</td><td>0</td><td>2</td><td>0</td></tr>
<tr><td colspan="13">Alternative C</td></tr>
<tr><td>Level 0 (M_0)</td><td>0</td><td rowspan="4">0.063</td><td>0</td><td rowspan="4">0.075</td><td rowspan="4">0</td><td rowspan="4">0</td><td>0</td><td rowspan="4">0.037</td><td>3</td><td rowspan="4">2.188</td><td>0</td><td rowspan="4">0.065</td></tr>
<tr><td>Level 1 (M_1)</td><td>0.143</td><td>0.063</td><td>0.012</td><td>2.25</td><td>0.012</td></tr>
<tr><td>Level 2 (M_2)</td><td>0.107</td><td>0.236</td><td>0.011</td><td>1</td><td>0</td></tr>
<tr><td>Level 3 (M_3)</td><td>0</td><td>0.25</td><td>0.125</td><td>2.5</td><td>0.25</td></tr>
</table>

Table 25: Calculation results

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 (cf. Johannsen et al. 2014a) 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 reach ideal values and perfectly adhere to each condition in practice, e.g., due to the complexity of the real world situation modeled.

2.3.5 Prototypical implementation

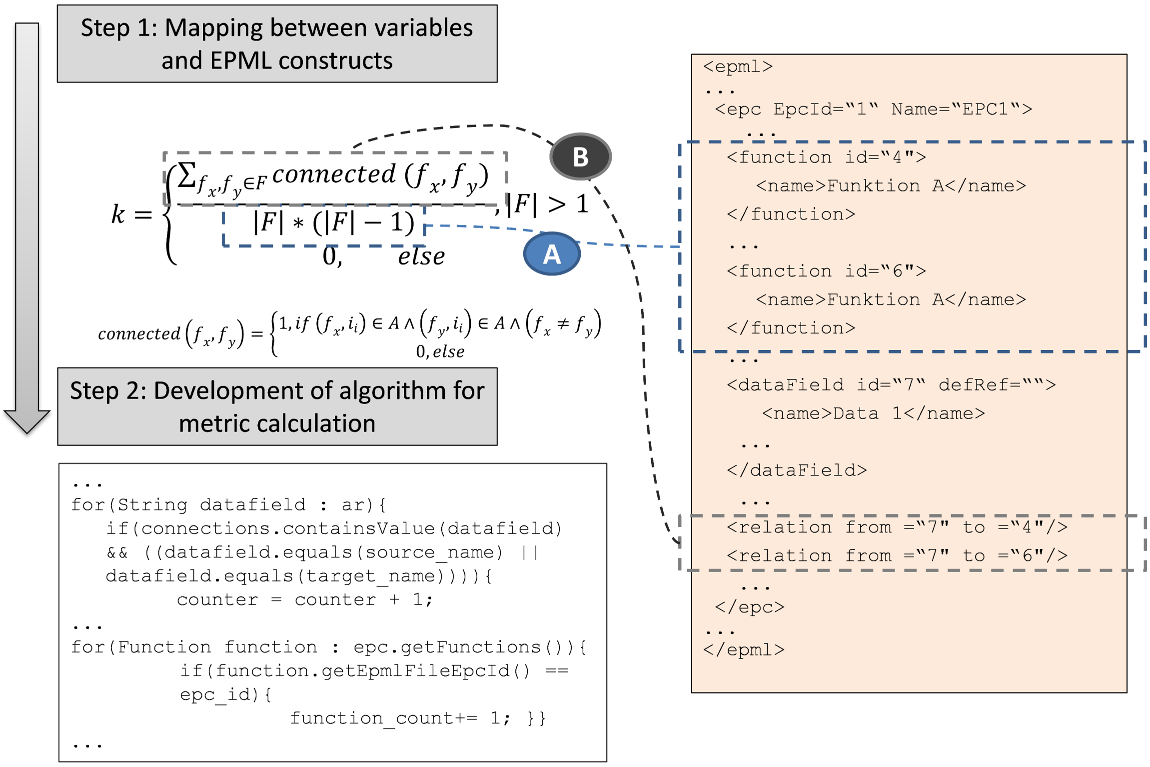


Figure 20: Implementation procedure

Many of the metrics presented here require a laborious calculation. Especially with regard to the high number of decomposed process models, which can typically be found in process architectures, modelers need means to support the calculation. In most practical settings, process models are available in electronic form in tools as e.g., ARIS or MS Visio. Thus, we used the formalization of the metrics to implement an automatic calculation to be used by modelers evaluating a large number of models.

For the metrics 3, 4, 7, 8, 11 and 12, the implementation was straight forward as indicated by Figure 20. First, we mapped the variables from the metrics to eEPC modeling elements (step 1). Then, we implemented the calculation procedures as algorithms (step 2).

For the implementation, we used the ProM process mining and analysis framework *(http://www.promtools.org/doku.php)*. 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 & Nüttgens 2006), which is an open standard supported by many frequently used modeling tools. The source code of our implementation is available at: *https://svn.win.tue.nl/trac/prom/browser/Packages/QualityMetrics*.

Currently, there is no implementation for a fully automatized calculation of the metrics 1, 2, 5, 6, 9 and 10 because these require information about redundant event types (metric 1 and metric 2), missing external event types (metric 5), missing non redundant event types (metric 6) or external event types in general (metrics 9 and 10). As of now, users have to identify them using their process knowledge or the advice of experts.

2.3.6 Discussion

2.3.6.1 Reflecting the metrics against Wand and Weber's decomposition conditions

In this research, the decomposition conditions of Wand and Weber have been transformed into metrics to evaluate the quality of decomposed eEPC models. In that context, the degree to which the metrics capture the initial ideas of Wand and Weber's decomposition conditions is to be discussed.

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 (see also section 2.3.2.3). Thus, for unambiguously determining redundant event types, which is a prerequisite for correctly calculating the metrics' values, a modeler 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 considering the labeling of nodes in a process model as well, because inadequate labels might lead to ambiguities in process execution in addition counteracting the ideas of *Weber* (1997). On the other hand, metric 5 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 (see section 2.3.2.3). 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 modeling notation. Nevertheless, this circumstance is negligible because our metric builds on the requirement to preserve all types of properties during decomposition (cf. Weber 1997) and counts the missing non redundant event types of a process model accordingly. Hence, the primary idea of the decomposition condition was enhanced considering the application for eEPC modeling.

To operationalize the input to or the environmental influence on a subsystem, in regards to the minimum coupling condition, data object types and external event types of subprocess models are focused by our metrics. Considering the ontological expressiveness of eEPCs (cf. Green & Rosemann 2000), these modeling constructs are most appropriate to define coupling for subprocess models from a structural perspective and to receive an interpretation that comes close to the primary ideas of Wand and Weber.

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 modeling as introduced by *Reijers and Vanderfeesten* (2004) and was adapted for the research at hand. Actually, the interpretation of cohesion on the base of data object types is most appropriate to capture the condition's initial purpose as described in section 2.3.3.6. However, our metrics do not consider event types as means to express input state or output state variables, as an in-depth analysis of the semantics attached to each event type of an eEPC model would have been required, a procedure largely influenced by subjectivity. By means of our metrics, we were able to avoid this subjective assessment of strong cohesion by focusing on the data object types of an eEPC model as mentioned.

2.3.6.2 Benefits and Restrictions

Judging the quality of a decomposition by reverting 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 (e.g., Malinova et al. 2013), the results can currently only be thoroughly interpreted in case equivalent decompositions are compared to each other. So, for every metric, the values have to be regarded separately because a general proposition as to whether some conditions are more important than others cannot be done. Much more, the modeler has to decide whether certain conditions and metrics should be prioritized considering a modeling project or not. Generally, the aggregation of all metrics to come to an overall value across all decomposition conditions remains an open issue.

Second, the application of some of the metrics requires, to some degree, users’ process knowledge. This is because the metrics do not exclusively consider structural aspects of a model (e.g., metrics 1, 2 or 5).

Third, the “distance” between the values of the metrics cannot be interpreted, which means that it is hard to determine how much effort (time or resources) would be necessary to improve a value of “0.148” for metric 4 to a value of “0.005” for example (see Table 25).

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 regards, the metrics for the minimum coupling condition, the strong cohesion condition, and the determinism condition (for dealing with internal events) provide clear 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 (metrics 3, 4, 7, 8, 11 and 12). Further, we proposed ideal values for the metrics (see section 2.3.3.7). The “distance” of the values received from applying the metrics to the “ideal values” as proposed enable the initial estimation of the coherence of a decomposed process model with the decomposition conditions (e.g., values “0.148” and “0” regarding metric 4 in Table 25). The subjectivity when manually judging a decomposed process model against the decomposition conditions is systematically reduced that way.

Second, the calculation can partly be automatized by the implementation of the metrics as a tool, which considerably speeds up the quality assessment procedure and reduces cognitive efforts and the likelihood of errors of a manual calculation. Corresponding means to assess the quality of decompositions are missing yet. Therefore, the user is strongly supported in choosing an alternative from a set of equivalent decompositions.

Third, we introduce metrics for both the model level and the holistic decomposition (see section 2.3.3), which is decisive because excerpts from a large process model may be relevant for particular users only. 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.

Last, we could demonstrate the interdependency among the metrics for minimum coupling and strong cohesion. Whereas “minimum coupling” generally suggests to merge subprocess models, “strong cohesion”, on the contrary, tends to delineate subprocess models. Consequently, the application of the metrics showed that the decomposition achieved optimal values for strong cohesion only for the cost of raised values for minimum coupling (e.g., alternative A in Table 25).

Summing up, despite the restrictions as described, the metrics are beneficial means to objectively assess the quality of a decomposition, supporting modelers in delineating subprocess models properly.

2.3.7 Conclusion and outlook

In our work, we develop metrics to judge the coherence of a decomposed process model with the decomposition conditions of Wand and Weber. Since the ontological expressiveness of modeling languages differs affecting the interpretation of the conditions, we only consider eEPCs in this paper to provide an adequate level of detail.

Decomposition is of high relevance for practice as it helps to reduce the complexity of large process models (Reijers et al. 2011). However, a theory on good decomposition for process modeling is missing just as generally accepted guidelines for delineating subprocess models are. In that context, we build on the decomposition model of Wand and Weber promoted in literature as a promising field for research to come to decompositions of high quality and understandability.

In a previous work, we were able to show that adhering to the decomposition conditions positively influences model understandability. However, a formal specification of the conditions to perform a corresponding process model evaluation was missing so far. We close this gap by introducing metrics that reduce the subjectivity in model assessment and enable a calculation of the degree as to which a process model adheres to the decomposition conditions.

Our work is beneficial for research and practice alike: With our set of metrics, based on the decomposition conditions of Wand and Weber, we contribute to the academic discussion on how to decompose process models purposefully and as to what properties characterize a well-performed decomposition (cf. Milani et al. 2016). Corresponding means to evaluate decomposed process models are not found in literature yet. Therefore, we provide a better understanding of those characteristics of a decomposed process model that influence model understandability.

Our research provides practitioners with the means to judge the quality of decomposed models, which are created in modeling or process improvement projects for instance. To this purpose, we implemented a prototype partly automating the calculation. That way, the tremendous cognitive effort required to manually assess the quality of a decomposition is drastically reduced. In addition, the likelihood of calculation errors is eliminated by an automated quality assessment. Further, the clear advice provided by the metrics regarding determinism, minimum coupling and strong cohesion can support a modeler 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—remains an open issue. Further, a fully automatized calculation is not possible as domain knowledge is needed for properly calculating some of the metrics (see section 2.3.3.7). Additionally, to receive a manageable set of metrics, not all the ideas as captured in the original decomposition conditions can be mapped to modeling with eEPCs (see section 2.3.6.1). Moreover, the practical experience of applying the metrics is limited to the aforementioned modeling project with the university administration so far. Correspondingly, in future, further applications will have to be realized.

In a future work, we will develop metrics for other modeling languages, e.g., BPMN, as well. In addition, we intend to investigate the intervals of metric values for each condition more closely. That way, precise threshold values may be deducted for the metrics indicating whether a decomposition has been well-performed or not. A quantification of the contribution of following the guidelines and metrics for process model understandability may help to assess the value of the improvement effort. Beyond that, we will further develop the software prototype to better match the requirements of practitioners. For that purpose, it will be used in several modeling projects in practice and enhanced by a more intuitive user interface. Further, a link to process mining tools will be established. By that, an automatic match between the log-files received from process execution and the conceptual process model will be performed. That way, redundant event types can be identified straight away enabling a fully automatized calculation of losslessness and minimality.

2.4 Beitrag 4: Analysing the Contribution of Coupling Metrics for the Development and Management of process architectures

<table>
<tr><td>Adressierte Forschungsfrage</td><td>Forschungsfrage 4: Wie kann Coupling im Kontext von Geschäftsprozessmodellen definiert werden und was sind mögliche Einsatzszenarien für und Anforderungen an Coupling Metriken im praktischen Einsatz?</td></tr>
<tr><td>Erscheinungsort</td><td>23rd European Conference on Information Systems, ECIS 2015, Münster, Deutschland, Mai 26-29, 2015 (VHB Jourqual 3: B)</td></tr>
<tr><td>Autoren</td><td>Daniel Braunnagel 80%
Dr. Florian Johannsen 10%
Prof. Susanne Leist 10%</td></tr>
<tr><td colspan="2">Der Beitrag klassifiziert 12 Metriken in 11 Cluster, stellt diesen Anwendungsmöglichkeiten aus dem Management von Prozessarchitekturen gegenüber und entwickelt daraus eine Aufstellung von aktuell unterstützten Anwendungen und zu entwickelnden Metriken. Weiter wird die Klassifikation herangezogen, um die Definition von Coupling zu präzisieren. So wird aufgezeigt, dass sich Metriken allgemein in zwei Gruppen unterteilen lassen: in solche, die Aussagen über das Prozessmodell treffen, sowie in solche, die Aussagen über den modellierten Prozess machen. Metriken dieser Gruppen unterscheiden sich grundlegend im Bezugsobjekt, in der Verwendung und Interpretation, weshalb es gerechtfertigt erscheint, diese mit separaten Definitionen zu beschreiben.

Die Einsatzmöglichkeiten zeigen auf, welche Zielsetzung mit der Entwicklung des Artefakts verfolgt werden kann und welcher Nutzen daraus resultiert. Der Beitrag lässt sich daher im DS Zyklus den ersten beiden Entwicklungsschritten zuordnen.

Die besondere Herausforderung des Vorhabens ist es, ein geeignetes Klassifikationsframework zu finden. Die theoretischen Grundlagen der Metriken unterscheiden sich deutlich und reichen von Anforderungen eines konkreten Projekts (vgl. Vanderfeesten et al. 2008c) über die kognitive Psychologie bis hin zum Information Retrieval. Es wurde anhand der Literatur ein neues Framework entwickelt, in dem elf Cluster aufzeigen, welche Anwendungsfälle durch existierende Metriken unterstützt werden und für welche Anwendungen neue Metriken entwickelt werden sollten.</td></tr>
</table>

Tabelle 26: Bibliographische Angaben zu Beitrag 4

Currently, the development and modeling of enterprise architectures is an intensively discussed topic in both science and practice. Process architectures represent a core element in recent enterprise architecture frameworks. With process models being a central means for communicating and documenting the process architectures, both their quality and understandability are decisive. However, the concept of process model quality is still not fully understood. The recent development has highlighted the role of coupling in models. Coupling is expected to represent an important dimension of quality for conceptual models. Still, this perspective is hardly understood and its definition vague. Therefore, this work collects diverse coupling interpretations in the field of process modelling and integrates them to a common and precise definition. Once introduced and formally specified, the metrics serve as a basis for a discussion on coupling and on how the future development in respect to coupling could look like. The main findings are that currently metrics evaluate either the documentation of the process architecture regarding its understandability or they contribute to the individual applications of process architectures. These findings support practitioners selecting metrics for a particular task and scientists to identify research gaps for further development.

2.4.1 Introduction

The systematic development and modeling of enterprise architectures is not only an intensively discussed topic in literature (Zachman 1987), but also an important subject in practice (cf. Jung 2005; Uhl 2004; Davis 2013; Harmon & Wolf 2014). Against the background of constantly changing markets, the fast development of new technologies, and increasing customer demands, it is necessary to implement changes throughout all levels of the organizations quickly (Harmon & Wolf 2014). Above all, enterprise architectures support the change in companies by describing the most important objects as well as interdependencies among them. Further, typical areas of application are analyses of interfaces (Aier et al. 2008), e.g. to identify dependencies which have to be considered by changes or to foster uniformity. Likewise, enterprise architectures are used to define standards, e.g. as a consistent specification for processes or IT functions, to counteract historically grown and often very heterogeneous landscapes (Buhl & Heinrich 2004). A core element of an enterprise architecture is the process architecture (Leist & Zellner 2008), which is defined as the type of processes it contains and the relationships among them (Barros 2007). The importance of process architectures is illustrated by the fact that they can be found in many enterprise architecture frameworks. In practice, processes are also seen as the most important objects within enterprise architectures (Aier et al. 2008).

Practitioners and scientists identified the documentation, which is the essential prerequisite for an overall transparency of the enterprise's operations or for the ability to support change or to define standards, as one of the main purposes for architectures and accordingly for process architectures (Aier et al. 2008; Malinova et al. 2013)). Architectures are usually described by means of models (Ferstl & Sinz 2006). Every model provides a specific perspective of the architecture. For instance, the process model takes the perspective of an enterprise's operations and contains its decomposition into a set of activities linked by a logical and temporal flow. In addition, it includes a link to the data model by specifying which activities interact with which information objects. To fulfil the documentation purpose, namely the quality, by means of its ease of understanding of every single model as well as the interdependencies of all models, is of utmost importance.

To address the issue of evaluating the ease of understanding of conceptual models, previous research systematized the knowledge on conceptual evaluation in categories such as complexity, modularity, size or cohesion. These categories were supported by corresponding metrics (cf. Vanderfeesten et al. 2007a; Gruhn & Laue 2006b). Also, works focused on alternative approaches like top-down frameworks (cf. Becker et al. 2000), pragmatic guidelines (cf. Sharp & McDermott 2009), and empirical studies (cf. Briand & Wust 2001; Zugal et al. 2013). Recently, "coupling" has been introduced as another perspective on the quality of process models which focuses the effect of dependencies among processes (cf. Vanderfeesten et al. 2007a; Vanderfeesten et al. 2008c; Khlif et al. 2010). E.g. Vanderfeesten et al. (2007a) define coupling in process modeling as the connectedness of elements. Coupling does not only evaluate the quality of single process models, there is a strong focus on interdependencies among models as well. Further, many operationalizations of coupling can be calculated automatically. This is a mandatory prerequisite for the use with process architectures, which frequently encompass a very large number of process models (Malinova et al. 2013). Thus the concept of coupling is very suitable for an application in the context of process architectures.

Ultimately, we aim to provide means which support practitioners in managing the complexity of process architectures. In the current work we follow this aim by investigating the contribution of "coupling" to the evaluation of the ease of understanding of process models as part of a process architecture and to the use of process architectures in general. Our focus is on coupling metrics which can be calculated automatically for process architectures. We thus pose the following research questions:

Q1: What is the current state of knowledge on coupling and how is the coupling concept operationalized using metrics?

Q2: Which perspectives on "coupling" can be distinguished and how do these perspectives help to assess the ease of understanding of process models or process architectures, respectively?

Our analysis requires detailed descriptions of the coupling instantiations. In the literature on coupling in process modeling which we found and which had a precise explanation of coupling, the authors always present metrics measuring coupling in process models. These metrics are usually specified for a specific notation. However, the metrics we found use elements appearing in many common process-modeling languages (cf. Recker et al. 2005; Recker et al. 2009; Braunnagel & Johannsen 2013; Braunnagel et al. 2014). The contribution of our paper is as follows: first, the current body of knowledge of coupling is analyzed and different perspectives on coupling are derived, generating a better understanding of the coupling concept and of how it helps to judge the ease of understanding of process models as part of a process architecture. Second, the research uncovers perspectives on coupling which have not properly been investigated or operationalized yet (e.g. by means of metrics) but bear promising potential for future research. Third, we introduce a prototypical tool for an automatic calculation of coupling metrics.

We follow a bottom-up approach. We analyze in detail how the existing work on coupling operationalizes the concept by collecting and comparing actual operationalizations. The criteria comprise the internal view, i.e. the measurement, and the external view, i.e. the indication, on each metric (cf. Purao & Vaishnavi 2003). Based on the results we interpret and discuss the application of coupling in process architectures.

The paper unfolds as follows: in section 2.4.2, basics and related work on enterprise architectures, process architectures and coupling are presented. Subsequently, the methodology of this research is explained. In section 2.4.4, current operationalizations of coupling in the form of metrics are analyzed. Based on that, section 2.4.5 introduces a classification of coupling metrics showing different perspectives on the coupling concept. The implementation of a tool for calculating the metrics is highlighted in section 2.4.6. Then, we derive an interpretation for coupling in the context of process architectures and conclude our paper with a summary and an outlook on future research.

2.4.2 Basics and related work

2.4.2.1 Enterprise Architecture and Process Architecture

Despite the fact that the term “architecture” is not consistently used in literature (e.g., Schekkerman 2006; Ross 2003) most definitions are based on the IEEE standard (IEEE, 2000). According to that standard an architecture is the fundamental organization of any kind of socio-technical system embodied in its components and their relationships, as well as construction rules for the generation and the design of the system (1471). The definition comprises two main aspects of an architecture: (1) the specification of the components of the system (e.g. data, processes), abstracting from insignificant aspects, often represented by models of the “as-is” resp. the “to-be” state and (2) construction rules guiding the design and evolution of the system by its components from an “as-is” into a “to-be” state. Due to the complexity—especially of business systems—the architectures need to facilitate the structuring of the business system by describing different architecture levels (e.g. business data, entity types, business processes, functions) or taking different views on these levels (e.g. designers or builders view), as is the case e.g. with the Zachman Framework (cf. Zachman 1987).

Practitioners recently judged the process architecture as one of the most important components of an enterprise architecture (Aier et al. 2008). Normally, they are allocated to a distinct process or organization level and modelled together with organization units, roles, and information objects (Leist & Zellner 2008). The process itself consists of functions (or tasks) and their relationships. Functions are carried out to develop, produce, and distribute services to customers (Österle 1995). Due to a variety of potential views on processes (e.g. structure or behavior), different process models are organized within the framework of a process architecture. The framework for a very simple process architecture for instance would define the hierarchical decomposition of a process into sub processes and the architecture would be presented with coarse-granular towards fine-granular process models. In an empirical analysis, Malinova et al. (2013) found two important construction rules for process architectures. First, process models are related through a hierarchical decomposition. Therefore, an activity is explained in more detail in a separate process model. Second, in a service oriented architecture, distinct services are combined into one process. Thus, the detailed descriptions of the services are reused in the resulting process. As both these approaches present different levels of granularity, they are referred to as vertical decomposition. (cf. Malinova et al. 2013; Zugal et al. 2013)

Process models as part of an enterprise architecture have an important transfer function because they are used as the documentation upon which to realize requirements of business strategies for operational processes. In addition, they illustrate restrictions and potentials of the IT infrastructure for business strategies (cf. Österle 1995; Frank 1994). Beyond that, they can be used for many further purposes, as for instance for quality management, during the implementation of standard software, for compliance management, within sourcing decisions, or for process optimization (Aier et al. 2008). Another important use of process architectures becomes handy in case of major restructuring efforts, e.g. in case of a company merger. Architectures ease the identification of redundant functionality among the companies' processes or the assignment of responsibilities and tasks. They further serve as a planning basis for the restructuring of processes and not at last to identify the impact of changes in individual processes on further, dependent, processes. Thus, process architectures support the planning of structural changes in a company. (cf. Aier et al. 2008; Malinova et al. 2013)

The empirical analysis of Malinova et al. (2013) supports the importance of process architectures. Especially practitioners highlight the improved transparency of processes due to the documentation of architectures as most important and further stress that the process architectures help to plan necessary changes in processes due to mergers (cf. Aier et al. 2008).

However, these use cases suffer if a process architecture is overly complex. On the one hand, a process documentation is just as good as a model user is able to understand the documentation. A high degree of complexity is known to impair the ease of understanding. Further, dependencies among processes easily become obfuscated under a high number of complicated, possibly hidden, relations among processes. Coupling metrics are statedly used to control the complexity of conceptual models and to highlight possibly hidden dependencies. Further, their formalization allows an automatized application, which comes handy with the high number of process models that architectures frequently cover. They thus appear to offer potential to be applied to process architectures.

2.4.2.2 Coupling and Coupling metrics

Coupling in process management stems from the domain of Software-Engineering. Its current understanding is based on a transfer of knowledge from Software-Engineering to process modeling as well (cf. Vanderfeesten et al. 2007a; Khlif et al. 2009; Braunnagel & Johannsen 2013; Braunnagel et al. 2014). For example, Vanderfeesten et al. (2007a) introduce the concept as originating in Software-Engineering: “Coupling is measured by the number of interconnections among modules. Coupling is a measure for the strength of association established by the interconnections from one module of a design to another. The degree of coupling depends on how complicated the connections are and on the type of connections.” Thus, coupling is measurable and the measurement uses modules and interconnections as input. Further, the measurement indicates complexity. However, it is not clear what a module and an interconnection exactly constitute in process models and what “complicated” means in this context.

Based on one particular metric, Vanderfeesten et al. (2007b) explain coupling in greater detail. “Coupling measures the number of interconnections between the activities in a process model. The degree of coupling depends on how complicated the connections are and also on the type of connections between the activities.” This explanation is precise towards a particular metric. Here, the input of the calculation only constitutes activities in the limited scope of one model.

However, e.g. the conceptual coupling metric considers the semantic overlap of different models as coupling, whereas the Coupling Between Objects (CBO) metric counts the control flow connections between different models (cf. Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Chidamber & Kemerer 1994). Generally, metrics provide definitions of coupling that are specific for each case. Thus, these definitions express a specific understanding and cannot serve as a general definition of coupling in process modeling.

The above explanations define coupling as a measure for the number of connections among model elements that indicate the complexity of a process model. The first definition references Software-Engineering and thus needs to be interpreted with regard to process modeling. The metrics express specific views on coupling. Thus, the question of what coupling is in process architectures needs further elaboration. The concept of coupling is operationalized by a variety of coupling metrics. Each metric expresses a specific and individual understanding of the concept of coupling. E.g. Cardoso et al. (2006) transfer the software metrics by McCabe (1976), which quantify the paths through a model into process modeling (cf. Cardoso et al. 2006). Further Vanderfeesten et al. (2007b) present a coupling metric called “CP”, which is also influenced by software metrics. The metric “CP” evaluates all pairs of nodes averaging their value over all pairs. Another metric by Vanderfeesten et al. (2008a) is the cross-connectivity metric, which analyses the number of different possible paths in a process model.

Finally, González et al. (2010) present a review of measurement in process management. Regarding measures on process models, they argue that authors wrongly categorize their metrics as coupling or complexity metrics due to the lack of a precise definition.

In summary, the development of coupling stems from the research on software metrics. This development has resulted in a plethora of metrics for business processes. However, it has also caused confusion about the meaning of the terms and their contribution to process architectures has not yet been investigated.

2.4.3 Methodology

The first step of our methodology (Figure 21) was a literature review. For the review, the electronic databases Google Scholar, Computer.org, AISeL and Emerald Insight were queried using the term pair '"coupling metrics" "business process model"' and "coupling metrics" (cf. Elliott et al. 1999; Vom Brocke et al. 2009). The hits, 45 peer-reviewed results, which were considered as relevant on the basis of their title or abstract, consist of 31 conference papers, nine journal papers, four technical reports and one book.

Applicable sources (Allen et al. 2001; Cardoso 2006; Chidamber & Kemerer 1991; Gui & Scott 2006; Poshyvanyk & Marcus 2006; Reijers & Vanderfeesten 2004; Vanderfeesten et al. 2007a; Vanderfeesten et al. 2008a)
Inapplicable sources
Use cases. These works discuss the application of metrics but do not present new metrics. (Arshad et al. 2007; Beyer et al. 2001; Binkley & Schach 1998; Chowdhury & Zulkernine 2010; El-Emam 2001; González et al. 2010; Harrison et al. 1998; Lee & Chan 2001; Markovic et al. 2009; Meyers & Binkley 2007; Vanderfeesten et al. 2007a; Wahler & Küster 2008)
Not applicable. These metrics cannot be applied on process models in the design phase. (Allier et al. 2010; Allen et al. 1999, 1999; Birkmeier 2010; Briand et al. 1999; Briand et al. 1997; Cho et al. 1998; Gui & Scott 2008; Halstead 1977; Hitz & Montazeri 1995; Joshi & Joshi 2010; Kazemi et al. 2011; Orme et al. 2006; Green et al. 2009; Perepletchikov et al. 2007; Qian et al. 2006; Quynh & Thang 2009; Rajaraman & Lyu 1992; Sunju & Joongho 2009)
Known metrics. These works discuss metrics from previous works. (Chidamber & Kemerer 1994; Chen et al. 2009; Khlif et al. 2009); (Khlif et al. 2010; Rosenberg & Hyatt 1997; Sandhu & Singh 2005; Újházi et al. 2010)

Table 27: Grouped literature review results

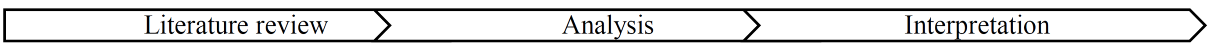


Figure 21: Method

Upon close inspection, we divided the literature in four groups (see Table 27). "Applicable sources" comprises literature that defines those metrics that are used in the remainder of our analysis. The remaining works turned out to be inapplicable for the following reasons. The literature group "use cases" contains works which discuss use cases for the metrics, but do not present new metrics in comparison to the first group. E.g. (Binkley & Schach 1998) discuss relations of coupling and run-time failures. The group "not applicable in the design phase" presents coupling metrics that require information unavailable in the design phase of process architectures and thus cannot be used for our analysis. E.g. (Green et al. 2009) present an approach involving runtime information which is not available in conceptual models. Lastly, we found the literature group which we named “Known metrics”. Here we classified works which discuss or present metrics that are already covered by those in the group “applicable sources”, but did not provide/deliver new metrics for our analysis. E.g. Khlif et al. (2009) transfer metrics to BPMN. We refer to the original description. A more detailed presentation of transferred and not transferred metrics can be found in Braunnagel and Johannsen (2013).

Our second step (see Figure 21) is the analysis of the interpretations of coupling. We systematically assess the characteristics of the interpretations with the aid of five criteria. Since all of the works on coupling we found describe metrics, we select the criteria according to (Purao & Vaishnavi 2003). The criteria are specifically designed to compare quality metrics for conceptual models and distinguish the internal and the external perspective on metrics.

The internal perspective ensures the scientific base of the measurement. This perspective operationalizes the entity-attribute view for metrics of Fenton (1998), which expects a thorough description of those attributes that the calculation of the metric takes into account. Thus, we introduce the categories input and scope. Input describes the elements of the process model that make up the variables of the metric calculation. E.g. a metric may count the number of arcs, or the number of nodes, or only nodes of certain types like functions or events. Scope lists the selection criteria for elements. More precisely, it describes whether a metric considers elements from only one model or several models and what the selection criterion is. E.g. a metric may consider only nodes from the one model in focus, or those arcs that connect different models, or all functions and events from all models that describe the same process. The external perspective captures the intended use of the metric. This perspective is in accordance with the goal-question-metric approach of Basili (1992). Thus, this perspective describes the implication and utilization of the metric. The implication states what a high or low metric value implies about the model, while the utilization exemplifies how to utilize this indication.

Finally, a mathematical function maps the perspectives. In essence, a function needs to transform the elements from the internal perspective into a value that reflects the indication stated in the external perspective. Our category “strategy” generally classifies the different approaches. We discuss the meaning and indications of the categories more thoroughly together with the respective metrics in the section “Analysis”.

The third step of our approach (see Figure 21) is the interpretation of the results of the preceding analysis. A summary of the analysis is the subject of the classification of the works, which also serves as a source of discussion on the state of and the gaps in the currently available set of metrics and their contribution to process architectures. In doing so, we demonstrate for every metric in which way it supports the aims or purposes of the process architecture.

2.4.4 Analysis

The following analysis bases on the coupling metrics shown in Table 27. The metrics are analyzed according to the criteria outlined before. In our analysis, we found that many metrics have the same value regarding the implication criterion, which according to our analysis are either the **complexity or the length of the control flow**, the **process flexibility**, **functional redundancy** or **model complexity**. To support the clarity, we use these values to group metrics in the following discussion.

Complexity of the control flow. Table 28 shows the similarity of the metrics Cross Connectivity (CrC) (Vanderfeesten et al. 2008a), Weighted Coupling (WC) (Vanderfeesten et al. 2007a) and Control Flow Complexity (CFC) (Cardoso 2006) (cf. Braunnagel & Johannsen 2013; Braunnagel et al. 2014). This similarity is a consequence of the implication the metrics have in common, which is the complexity of the control flow. The complexity of the control flow impairs the understanding of the model. Thus, the metrics are utilized to evaluate the quality of the process model. To assess this complexity, the metrics count the states that the control flow of the model can have. The number of states results from the number of branches in a control flow that are active after a connector. After an “AND”-connector, all succeeding branches are triggered, thus there is only one possible state. An “XOR”-connector can trigger as many different states as there are outgoing branches. After an “OR”-connector any combination of branches can be triggered, thus there are as many states as there are combinations of outgoing branches. The authors argue that a high number of states is complex and thus difficult to understand.

These three metrics use this common principle but differ in their calculation. First, the CFC metric counts and adds the states (Cardoso 2006). Second, the WC metric counts the number of states and divides it by the number of functions and events (Vanderfeesten et al. 2007a). Finally, the CrC metric calculates the states between any two pairs of functions and averages them over all nodes (Vanderfeesten et al. 2008a). Consequently, the metrics use either only connectors, connectors and functions, or additionally events as input. However, all three metrics use only elements of the control flow as their input. They are selected exclusively from single, isolated models. Thus, the metrics' scope is local. Furthermore, the determining step of the metrics, thus their strategy, is to count the elements of the input.

Metric	Implication	Input	Scope	Use
WC	Possible number of states due to choices indicating the complexity of control flow	Functions, Events, Connectors	Local	Model Evaluation
CrC	Possible number of states due to choices indicating the complexity of control flow	Functions, Connectors	Local	Model Evaluation
CFC	Possible number of states due to choices indicating the complexity of control flow	Connectors	Local	Model Evaluation
DC	Length of the control flow, indicated by the number of connected models	Functions, Process References	Global	Model evaluation
IC	Length of the control flow, indicated by the number of connected models	Functions, Process References	Global	Model evaluation
TC	Length of the control flow, indicated by the number of connected models	Functions, Process References	Global	Model evaluation
RFC	Length of the control flow, indicated by the number of connected models	Functions, Process References	Global	Model evaluation
CBO	Length of the control flow, indicated by the number of connected models	Process references	Global	Model evaluation
PC	Process flexibility, indicated by share information	Information elements, Functions	Local	Process evaluation
CoC	Functional redundancy, indicated by semantic overlap	All terms	Global	Process evaluation
CM	Model complexity, indicated by the graphs entropy	Nodes and arcs	Global	Model evaluation
ICM	Model complexity, indicated by the graphs entropy	Nodes and arcs	Local	Model evaluation

Table 28: Metrics and their main features

Length of the control flow. As per Table 28, there is also a high similarity of the Direct Coupling (DC), Indirect Coupling (IC) and Total Coupling (TC) metric (cf. Gui & Scott 2006; Braunnagel & Johannsen 2013; Braunnagel et al. 2014). These metrics evaluate the process model. However, their implication refers to the length of the control flow, instead of its number of states, as before. The authors argue that an increasing length of the control flow impairs the reading and understanding of the model. To assess the control flow length, these metrics follow the strategy of counting connections. The operationalization is different for each metric.

For the Coupling Between Objects metric (CBO), the number of connections is counted which a model has in common with other process models. As an extension, the Response For a Class (RFC) metric includes the length of the control flow in the connected models by including the number of functions. (cf. Chidamber & Kemerer 1994; Braunnagel & Johannsen 2013; Braunnagel et al. 2014) In addition, the DC metric counts the number of connections in relation to the number of functions of a model. The IC metric extends the DC metric by paths between any two models. The TC metric applies this principle on all models of a process. (cf. Gui & Scott 2006; Braunnagel & Johannsen 2013; Braunnagel et al. 2014)

Since these five metrics only differ in their actual calculation, the input elements are mostly the same. The CBO metric counts process references, the remaining metrics additionally refer to the number of functions. In any case, the number of elements determines the value of the metric, thus metrics have a counting strategy. Since the metrics account for connections to multiple models their scope is global. Finally, as their implication refers to the model, their utilization is the model evaluation.

Process flexibility. The Process Coupling (PC) metric evaluates a process instead of its models (cf. Reijers & Vanderfeesten 2004; Braunnagel & Johannsen 2013; Braunnagel et al. 2014). More precisely, the metric implicates one aspect of how flexible the execution of a process is. The aspect that is implicated refers to the dependency of functions in a process due to commonly used information. If a function depends on information that is created in a second function, then the first function cannot be executed before the latter function is finished. Thus, if the second function fails, this affects all those functions that require its information. To calculate the metric, the pairs of dependent functions are counted. The dependent pairs are those pairs that use common information elements. The number of dependent function pairs is then divided by the number of all function pairs indicating the maximum possible number of dependent functions. Thus, the metric uses a counting strategy. The authors define the metric for all functions of a single process. Its scope is local in relation to one single process. The model topology is not a concern of this metric. Consequently, the input of the metric consists of information elements and functions. (cf. Reijers & Vanderfeesten 2004; Braunnagel & Johannsen 2013; Braunnagel et al. 2014)

Functional redundancy. The Conceptual Coupling (CoC) metric evaluates processes (cf. Marcus & Poshyvanyk 2005; Braunnagel & Johannsen 2013; Braunnagel et al. 2014). More precisely, the metric implicates the pairwise similarity of two processes. The similarity of the processes bases on the process functionality that the metric discovers by decomposing the process into semantic concepts. These concepts are the result of a latent semantic analysis, which is an analysis technique from information theory. In consequence, we label the strategy as “information theory”. The latent semantic analysis uses a term-model matrix with a mathematical transformation, called singular-value-decomposition, to create clusters of correlated terms. Thus, the metric groups term-labels that are frequently used together to indicate semantic concepts. Their overlap implicates redundant functionality. (cf. Marcus & Poshyvanyk 2005; Braunnagel & Johannsen 2013; Braunnagel et al. 2014) Since the metric uses only textual content, its input are node labels. Further, as the metric compares multiple processes, the metrics’ scope is global. Thus, the model topology is irrelevant for this metric.

Model complexity. Finally, the metrics Coupling of a module (CM) and IntramoduleCoupling of a module (ICM) are utilized to evaluate the process model, again (cf. Allen et al. 2001; Braunnagel & Johannsen 2013; Braunnagel et al. 2014). These metrics implicate the complexity of the model graph. Therefore, they consider all nodes, irrespective of the node type. Further, the authors use an information theoretic concept, the entropy, to define complexity. The entropy quantifies information content. The authors argue that an exceeding amount of information impairs the understanding of the model. The CM metric applies the entropy calculation to nodes and arcs that connect different process models. Consequently, the metric’s input are nodes and arcs and its scope is global. The ICM metric applies the metric onto nodes and arcs inside a model. Thus, its scope is local. (cf. Allen et al. 2001; Braunnagel & Johannsen 2013; Braunnagel et al. 2014)

2.4.5 Interpretation and Discussion

2.4.5.1 On the application of coupling metrics

The previous analysis has shown the individual traits of the different interpretations of coupling. The following classification summarizes the analysis and is the first step towards integrating and interpreting the various interpretations. In Table 28 the external perspective is displayed on the X-axis and the internal perspective on the Y-axis. This classification supports the identification of common criteria among the metrics. In addition to the clustering of available metrics, the classification reveals research gaps and starting points for future works.

The first utilization in Table 28 covers the evaluation of process models in an architecture as to their ease of understanding. The metrics support a practitioner in designing the process architecture by indicating which models will be hard to understand due to its high complexity. A high value thus indicates that a model's documentary function is impaired by its design. Looking closer, the metrics cover different views on a model's ease of understanding.

Cluster A: The metrics CrC, WC and CFC assess a model's complexity depending on is branching structure. They thus indicate whether the level of granularity is chosen well regarding the ease of understanding, or not. As Malinova *et al.* (2013) explain, the level of granularity is an important issue for practitioners. However, process architectures are used to promote the complete overview of processes, which is not covered by the current metrics. Such metrics would be found in *Cluster B.*

Cluster C/D: Another perspective on the process model's ease of understanding is the length of the control flow, which is covered by the metrics CBO, RFC, DC, IC, and TC. These metrics assess the size of the process presentation. They indicate the difficulty in reading the overall process due to the fine-grained granularity. In addition, they calculate the size of a module or the number of interconnections to further processes as well.

Cluster G/H: The metrics previously discussed assess the ease of understanding only on the basis of the number of states or nodes and arcs. A first attempt for a more elaborate assessment can be found with the metrics ICM and CM. They use the information entropy of the models as a complexity measure to indicate how properly the models, either individually (ICM), or as a group (CM), support the documentary function of a process architecture. Another reason to control the complexity of modules is mentioned by Aier and Schönherr (2004). Accordingly, modules need to be constructed in a way that their complexity does not prohibit its handling regarding a company's flexibility (cf. Aier & Schoenherr 2004). In this context it appears feasible to measure the complexity of modules. Further, the field of IS knows more than this one theoretical approach to measure the ease of understanding of conceptual models, which seem promising to offer a more reliable measurement (cf. Houy et al. 2014). Such metrics would be found in *Cluster E.*

			Strategy: Count Scope: Local	Strategy: Count Scope: Global	Strategy: Information Theory Scope: Local	Strategy: Information Theory Scope: Global
Utilization	Model evaluation	Implication: Complexity o.t. control flow	CrC, WC, CFC Cluster A	Cluster B	Cluster E	
Utilization	Model evaluation	Implication: Length o.t. control flow	CBO Cluster C	RFC, DC, IC, TC Cluster D	Cluster E	
Utilization	Model evaluation	Implication: Model complexity	Cluster F		ICM Cluster G	CM Cluster H
Utilization	Process evaluation	Implication: Flexibility o.t. process	PC Cluster I	Cluster J		
Utilization	Process evaluation	Implication: Functional similarity				CoC Cluster K

CBO	CBO metric	(Chidamber & Kemerer 1991)	DC	Direct Coupling	(Gui & Scott 2006))
RFC	RFC metric	(Chidamber & Kemerer 1991)	TC	Total Coupling	(Gui & Scott 2006))
CrC	Cross Connectivity	(Vanderfeesten et al. 2008a)	CFC	Control flow complexity	(Cardoso 2006)
WC	Weighted Coupling	(Vanderfeesten et al. 2007a)	IC	Indirect Coupling	(Gui & Scott 2006)
CoC	Conceptual Coupling	(Poshyvanyk & Marcus 2006)	ICM	Intramodule Coupling of a module	(Allen et al. 2001)
PC	Process Coupling	(Reijers & Vanderfeesten 2004)	CM	Coupling of a module	(Allen et al. 2001)

Table 29: Classification of coupling metrics

Cluster F indicates further potential research. As of now, the metrics refer only to the control flow to analyze a process architecture's ease of understanding. However, architectures offer several further views, such as an information or organization view, which should also be accounted for. Such metrics would be found in *Cluster F.*

The second utilization, namely the process evaluation, covers those metrics that do not evaluate a process model regarding is ease of understanding, but a process instead. These metrics focus the control and use of the process architecture towards business needs other than process clarification.

Cluster I: The first metric measures the dependency among process functions due to shared information, which finds its use in the assessment of the flexibility of process architectures. Aier and Schönherr (2006) discuss the issue of a high number of data interfaces that become difficult to manage. Further, Aier and Schönherr (2004) explain the contribution of modularization to a company's flexibility. This first metric can indicate dependencies between modules in the process architecture stemming from the information view. Also, the metric can be used to indicate information-intensive processes which would justify a higher priority regarding the analysis for IT support of the processes (Aier et al. 2008). Vanderfeesten et al. (2008c) used the metric to assess the flexibility of individual processes, indicating in how far it is possible to change functions in the process without breaking dependencies with functions that occur due to shared information. In the worst case, required information would not be provided any longer for a certain function. However, the current specification of the metric refers to process models in isolation. A possible extension would apply the metric onto all models of a process or a complete process architecture instead (*Cluster J*).

Cluster K: The second metric of this group, the CoC metric, attempts to measure the similarity of functions between processes, thus supporting the use of process architectures in several ways. First, the metric allows the identification of similar processes, indicating potential candidates to be merged into a standardized process (cf. Malinova et al. 2013). Further, similar functionality is also an indicator for potential modules in an architecture. Thus, processes and parts of processes with a high functional similarity are identified by the metric and highlighted as candidates for being integrated to a module. This supports the flexibility of the process architecture (cf. Aier & Schoenherr 2004). Finally, Aier *et al.* (Aier et al. 2008) discuss the issue of redundancies among process architectures, e.g. regarding their IT support. In case different systems support similar functionality, the information on potential redundancies would tremendously support the architecture's development.

However, process models have a wider application. As mentioned in the introduction, models are used among others for process improvement initiatives or system requirements specification. We would thus expect further metrics to be developed that aid indicating issues or effort for the respective use.

The previous analysis addresses our research question Q1, showing the current state of knowledge regarding coupling and its operationalizations, and serves to draw the following summary. We have a set of 12 coupling metrics addressing five different use cases, e.g. evaluating the flexibility of processes or evaluating the complexity of the control flow in process models. Practitioners may choose to apply a particular metric on the elements of their process architecture if they suspect a particular problem. Alternatively, since the metrics can be computed automatically, they can be applied arbitrarily as well. In that case, a practitioner can analyze the results for outliers which justify special attention regarding further analysis. Researchers may have particular interest in the gaps that can be seen in Table 28 in order to develop further metrics. In the case of process architectures, metrics with a global scope seem to be highly relevant, as they cannot only be applied on isolated process models but can evaluate their interaction with their environment as well. At last, practitioners and researchers need to cooperate to develop further metrics. Practitioners need to identify further relevant use cases for an automated evaluation while researchers need to analyze whether these use cases can be linked to the effects of coupling and measured with respective metrics.

The classification demonstrates that, while the relationship of the indication and the use are distinctive, there are metrics for all combinations of the scope and strategy criteria. Thus, we argue that metrics with the same utilization share more traits. In addition, while we expect further metrics to introduce additional implications, i.e. further dimensions of model quality and process characteristics, we do not expect further utilizations to turn up. For both these reasons, we decided to treat coupling for process model evaluation and coupling for process evaluation separately.

2.4.5.2 On the definition of coupling

The current discourse on coupling metrics is based on the definition of coupling as presented by Vanderfeesten *et al.* (2007a) "Coupling is measured by the number of interconnections among modules. Coupling is a measure for the strength of association established by the interconnections from one module of a design to another. The degree of coupling depends on how complicated the connections are and on the type of connections.", which we fully agree with. However, as González *et al.* (2010) explain, there is currently some confusion in research about the classification of concepts like coupling due to the lack of clarity in the concepts. As a contribution to research, we propose to direct the attention to the important distinction to be made in regard of the utilizations within the group of coupling metrics. The distinction that metrics either evaluate the process architecture regarding its ease of understanding by its individual models on the one hand, or the dependencies among processes on the other hand, shifts the understanding of the concept as well as the interpretation of the definition.

In the first case, the measurement refers to the syntax of process models and their decomposition. Thus, the measurement neglects the semantic of a process. Interconnections in the sense of Vanderfeesten *et al.* (2007a) are arcs in and between related process models or process model elements. They are connected to indicate a perspective of complexity of the process models and their respective decomposition.

In summary, if coupling is used to evaluate process model quality, it measures the dependency of process model elements based on the syntax of a process model's decompositions. Correspondingly, the measurement value indicates the complexity of a process architecture's models.

In the second case, if metrics are used to evaluate e.g. the flexibility of a process, the measurement refers to the process, i.e. the semantic of a model, oblivious to its decomposition. Interconnections in the sense of Vanderfeesten *et al.* (2007a) refer to functional dependencies among the processes, e.g. shared information, as opposed to connections between process models and their elements. They are used to indicate dependencies in the different views of the process architecture.

Regarding Q2, we can summarize that, if coupling is used to evaluate the processes in a process architecture, it measures the dependency of processes, and the measurement value indicates the dependency of processes regarding a particular architectural view.

2.4.6 Implementation

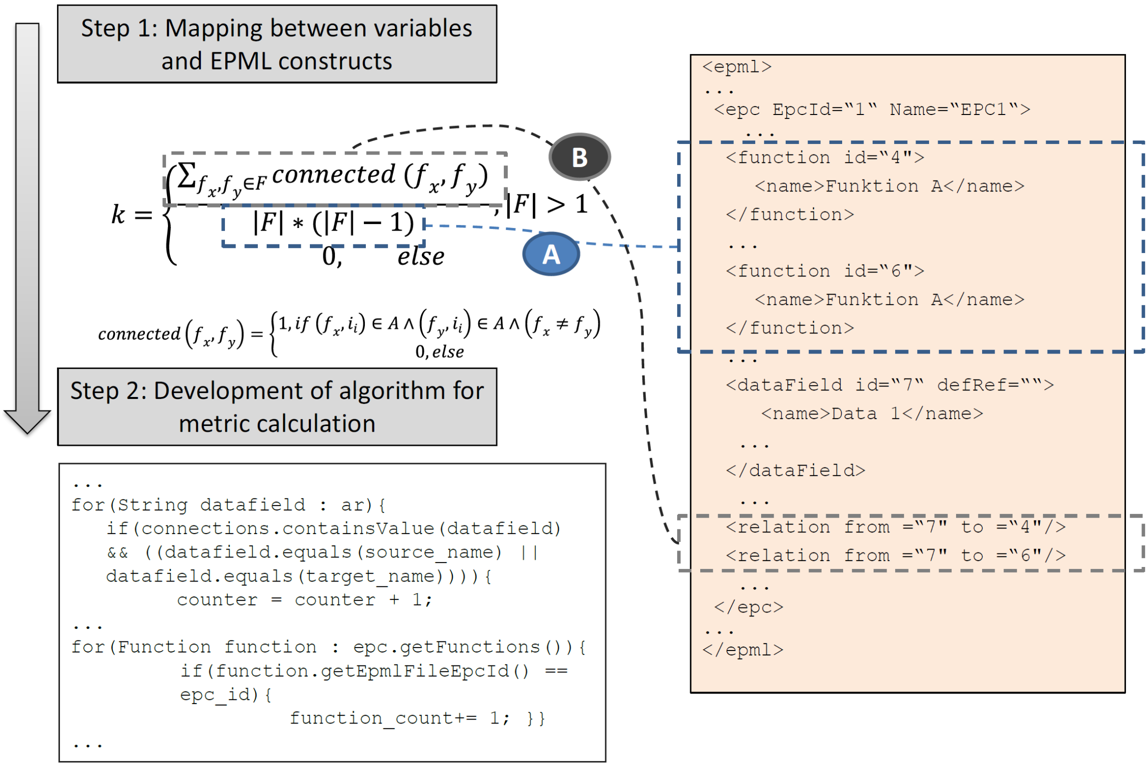


Figure 22: Implementation procedure (Example for the metric "Process Coupling")

The metrics, as described above, need to be applied onto a process architecture in order to be useful. Since many of the calculations are laborious and architectures naturally contain a high number of models, it is not feasible to calculate these metrics by hand for a complete architecture. We thus created a tool to support the calculation.

We opted for implementing the metrics as a Plug-In for the process analysis framework ProM 6. ProM is a framework offering several techniques for process mining and model analysis (cf. Dongen et al. 2005). Thus, our plugin can use existing means to import process models in different file formats, interact with workflow-enabled systems, and automatize the use.

Since the metrics are fully formalized in formulas as described in (Braunnagel & Johannsen 2013; Braunnagel et al. 2014), we were able to implement the calculations without introducing further assumptions or adaptions. Figure 22 exemplarily shows the development procedure for the metric “Process Coupling”. First, the variables of the metric and the corresponding EPML (EPC markup language) constructs were mapped. EPML is an XML based exchange format for eEPC models and is supported by a wide range of modeling tools (Mendling & Nüttgens 2006). The calculation of the metrics is performed on the EPML representation of a process model. As the example shows, the <relation> construct in the EPML representation indicates a relation between a function and an information object in a process model (marker “B”). Further, by focusing on the <function> construct, the number of functions of a model can be determined (marker “A”). Based on these findings, an algorithm to calculate the process coupling metric was derived in a second step. The code excerpt shows how the number of function pairs in a process model, coupled via a common information object, can be determined. Further, a code snippet to count the total number of functions which is required by the denominator of the metric is shown.

For implementing the metrics, we extended the EPML import Plug-In for ProM. The implementation is written in Java, whereas some of the mathematical functionality refers to the mathematics software R. The Plug-In is available as source code at https://svn.win.tue.nl/trac/prom/browser/Packages/CouplingMetrics/. By using the ProM framework, a user may work within a familiar process analysis environment, combine the analysis of these metrics with other means of analysis, and benefit from a high degree of automation. Once the models of the process architecture are imported, the end user chooses from the available metrics and is presented with the results without having to take any further interaction.

Currently, we are developing an interface with the frequently used ARIS Framework (Scheer 2000). Thus, the metrics will be available in the familiar from to analyze process models and architectures in ARIS, which are reports.

2.4.7 Summary, limitations and outlook

In our paper, we investigate the contribution of coupling metrics in the context of process architectures. To do so, we follow a bottom-up approach and analyze existing coupling metrics derived from a literature review. Each metric is assessed regarding the criteria “input”, “scope”, “strategy”, “implication” and “use”. The majority of metrics found evaluate the complexity of the control flow within a single process model. Regarding the uses of process architectures, these metrics can be used to support the documentary function of the architecture as they measure the ease of understanding of process models. Several other metrics also support the control of the model’s ease of understanding, based on different approaches though. Currently, the theoretical foundations of these metrics are basic. The operationalization of further theories could improve the reliability of the measurement. Considering the pivotal importance of the ease of understanding with regard to the documentary function of the process architecture, additional research is advisable in this regard.

Works that evaluate the process itself exist. These metrics contribute to the individual applications of process architectures. One metric measures the dependency among processes from an information view, while another one accounts for the functional similarity and thus the potential redundancy among processes. Further, metrics of this group could account for process dependencies from the organizational view or due to IT-systems and thus support the assessment of the flexibility of the process architecture.

We discuss what current metrics imply regarding the understanding of coupling in research. We also discuss an important distinction among the current metrics. Metrics refer to either the syntax or the semantic of process models. This distinction shifts the notion of coupling as well as the interpretation of the current definition.

The benefits of this research are twofold. From a scientific perspective, first, the current body of knowledge regarding coupling in process architectures is analyzed and categorized. This helps to better understand current works and metrics on coupling, since they are compared with common perspectives, as shown in Table 28. In addition, their contribution to evaluating process architectures is investigated. Second, the results support the development of new metrics. The classification shows perspectives which are currently not covered, but which are nonetheless worth investigating to assess dependencies among processes. E.g. the flexibility of an architecture is subject to organizational dependencies as well, which could be indicated by means of further metrics. Practitioners benefit from this work, first, by metrics indicating a number of different aspects to improve process architectures, and, second by the automatic evaluation of the process architecture, since coupling metrics are formalized.

However, there are some limitations. The completeness of the literature review regarding all works on coupling for process models cannot be guaranteed. Nevertheless, we supplemented the forward search by a corresponding backward search. We only considered our collection of literature sources appropriate for the investigation until after both the forward as well as the backward search did not provide any new sources. Also, at the current state, the discussion is purely theoretical. As to support a practitioner in the development and management of process architectures, we need to evaluate our artefact in a practical setting.

In future works, the metrics found will be tested empirically. The influence of the metrics' value on model complexity, or model readability, respectively, needs to be analyzed. Based on these insights, guidelines for designing process models in terms of coupling may be developed. And last but not least, metrics will be developed covering all those perspectives which are still underrepresented in current research.

2.5 Beitrag 5: Coupling and Process Modelling – An analysis at hand of the eEPC

<table>
<tr><td>Adressierte Forschungsfrage</td><td>Forschungsfrage 5: Wie und welche Coupling Metriken, bzw. der jeweils gemessene Qualitätsaspekt, aus verwandten Disziplinen lassen sich für Prozessmodelle operationalisieren?</td></tr>
<tr><td>Erscheinungsort</td><td>Modellierung 2014, Wien, Österreich, März 19-21, 2014 bzw. Lecture Notes in Informatics P-255, 2014 (VHB Jourqual 3: C)</td></tr>
<tr><td>Autoren</td><td>Daniel Braunnagel 85 %
Dr. Florian Johannsen 5 %
Prof. Susanne Leist 10 %</td></tr>
<tr><td colspan="2">Der Beitrag erläutert den Transfer und die Formalisierung von neun Coupling Metriken aus dem Software-Engineering für die eEPK Notation. Hierfür wurden zunächst die Informationen ermittelt, die für die Berechnung der Metrik notwendig sind. Anschließend werden Elemente der eEPK identifiziert, die, bezogen auf den Effekt, der laut ursprünglicher Metrik von Coupling hervorgerufen wird, äquivalente Informationen ausdrücken. Damit können neue Metriken formuliert und deren Aussage bezogen auf das Coupling in eEPK Modellen genauer beschrieben werden. Zuletzt wird eine Implementierung der Metriken angesprochen.

Da in diesem Beitrag das eigentliche Artefakt, die formalisierten Metriken, entsteht, lässt er sich dem dritten Schritt des DS Entwicklungszyklus zuordnen.

Die besondere Schwierigkeit des Vorhabens folgt, wieder, aus der heterogenen, theoretischen Grundlage der Metriken. Zwar operationalisieren diese alle Coupling in konzeptuellen Modellen, jedoch unterscheiden sich die Metriken deutlich dahingehend, was sie unter Coupling verstehen bzw. was ursächlich für Coupling ist. Allen Metriken gemein ist die Annahme, dass Coupling Verbindungen im mentalen Modell, das sich ein Nutzer während des Verständnisprozesses von einem konzeptuellen Modell bildet, beschreibt, die vom Nutzer zusätzlich verstanden werden müssen, wodurch das mentale Modell komplexer und schwieriger zu verstehen wird. Ursprung der mentalen Verbindungen können funktionale Redundanzen, Verbindungen im Kontrollfluss oder Datenflussbeziehungen sein, die in eEPK Modellen wiedergefunden werden müssen. Diese Heterogenität führt zu einer eigens entwickelten Methode für den Transfer der Metriken.</td></tr>
</table>

Tabelle 30: Bibliographische Angaben zu Beitrag 5

Business process modeling is a fundamental aspect in BPM initiatives. Being a central means of communication and documentation, both the quality and understandability of process models are decisive. However, the concept of process model quality is still not fully understood. The recent development has highlighted the role of coupling in models. Coupling is expected to represent an important dimension of quality for conceptual models. Still, contrary to Software-Engineering, this perspective is hardly understood or adapted in form of metrics in process modeling. Therefore, this work collects diverse coupling metrics in the field of Software-Engineering and transfers them to the eEPC modeling language. Once introduced and formally specified, the metrics serve for a discussion on coupling, process model quality with respect to coupling, and for their implementation.

2.5.1 Introduction

Business process modeling has gained considerable attention in BPM initiatives in recent years (Mendling et al. 2010; Becker et al. 2010; Polyvyanyy et al. 2008). Process models help a business analyst in documenting and analyzing a company's business processes properly (Becker et al. 2010). Based on thorough process documentation, improvement initiatives can be triggered whereas process simulation may be used for identifying weaknesses in the current process design and for evaluating alternative should-be process designs (van der Aalst et al. 2010). Further, process models serve as a means for communication between stakeholders and software developers (Gruhn & Laue 2006a). Therefore profound decisions on IT-investments are possible, indicating whether software is to be developed individually or standard software is to be bought for supporting a business process (Aguilar-Savén 2004; Becker et al. 2010). Process models help to derive requirements software has to meet in a systematic way (Becker et al. 2000).

However the described benefits of process modeling become blurred in case the process models cannot be understood by its users (see Gruhn & Laue 2006a; Houy et al. 2012; Becker et al. 2000). A high quality of the process models is thus decisive for BPM initiatives as well as for software development projects. Nevertheless, quality and understandability of process models are poorly understood concepts yet (see Houy et al. 2012; Moody 2005). A process model is a "construction of the mind" which makes its quality hard to judge (Moody 2005).

As a consequence, evaluating conceptual models usually is an “art” and does not follow systematic guidelines (Moody 2005).

For assessing the quality of process models, a variety of quality dimensions, such as complexity, modularity, size or cohesion have been introduced and corresponding metrics have been developed (see e.g. Vanderfeesten et al. 2007a; Mendling 2008; Gruhn & Laue 2007). Further, top-down frameworks (see e.g. Becker et al. 2000), pragmatic guidelines (see e.g. Silver 2008) and empirical studies (see e.g. Recker et al. 2011) can be found as approaches for operationalizing process model quality (Mendling et al. 2010). Recently “coupling” has been presented as a quality dimension for business process modeling (see Vanderfeesten et al. 2007a; Vanderfeesten et al. 2008a; Khlif et al. 2010). While “coupling” is a well-established quality characteristic in information systems development, research has only just begun to investigate the “coupling” concept in the context of process model quality.

In the current understanding, coupling is generally defined as the connectedness of elements. It is generally used as a means to improve the understandability and maintainability of processes and respective models (Vanderfeesten et al. 2007a). The actual way to achieve this goal, however, is subject to different implementations of the concept. As an example, *Vanderfeesten et al.* use coupling on the one hand to evaluate the variety of a process. Therefore they analyze whether or not a process allows so many alternatives that it becomes difficult to understand all of them (Vanderfeesten et al. 2008a). On the other hand, *Vanderfeesten et al.* also use coupling as means to balance the alignment of parts of a workflow between an overly flexible or rigid structure (Vanderfeesten et al. 2008c). The diversity of available applications underlines the multiplicity of interpretations of the concept of coupling for process modeling.

In addition to the two above examples, a couple of further publications deal with the topic of coupling in process modelling (see section 2.5.2.1). Even though each of these publications introduces another interpretation of coupling, the currently available literature does not cover the definition extensively. As a consequence the understanding of what constitutes the quality of a process model from the perspective of coupling is limited. Also the means to measure and control the understandability and maintainability of processes or process models respectively remain limited.

The objective of the current paper is therefore to supplement the range of interpretations of coupling and its means of determining it by introducing new ways of measuring coupling in the field of process modelling.

A thorough discussion and analysis as well as a practical application of coupling in process modeling require a detailed and precise interpretation. The preferred means of the available publications (see section 2.5.2.1) are metrics, which are described either formally or semi-formally. Their specification describes precisely which elements of a process model and which connections are taken into account and how inferences on the quality of models are made upon them. Consequently this work uses metrics as means of introducing new ways to measure coupling in process modeling. Further, since metrics are necessarily language-dependent and in order to retain an insightful level of detail we focus on the modeling language eEPC.

The contributions of this paper are as follows. We supplement the current body of knowledge on coupling in process modelling with further interpretations of the concept. We therefore continue the work of discovering new factors determining the quality of processes and process models from the perspective of coupling. We provide precise definitions for each interpretation in the form of measures which are the means for a thorough discussion of what constitutes coupling in process modelling and for measuring and controlling the quality of process models.

The paper is structured as follows: In the following section we provide an overview of related work and basic terms. After introducing the methodology of transferring the metrics to EPC models (section 2.5.3), we present corresponding metrics in section 2.5.4. Section 2.5.5 explains the implementation of the metrics. The paper ends with a summary of the results, limitations and an outlook on future research.

2.5.2 Basics and Definitions

2.5.2.1 Coupling

The current literature on coupling in process modeling is preceded and influenced by literature on Software-Engineering (Vanderfeesten et al. 2007a). There, coupling is operationalized in the form of metrics to predict measure and control the quality of software code and its conceptual models respectively. Each metric implicitly defines a particular interpretation of coupling. E.g. one definition focuses the graph representation of software systems, i.e. the way nodes are connected by arcs, whereas another definition uses information theory to account for reused code (Chidamber & Kemerer 1994). Some further definitions can be found together with multiple metrics interpreting each of them (see section 2.5.3).

In process modeling, *Vanderfeesten et al.* present a definition for the concept of coupling: “Coupling is measured by the number of interconnections among modules. Coupling is a measure for the strength of association established by the interconnections from one module of a design to another. The degree of coupling depends on how complicated the connections are and on the type of connections.” (Vanderfeesten et al. 2007a, S. 181). Here, coupling is generally considered as measurable and its key concept is the connections qualified by additional concepts (e.g. number, strength, etc…). As a means to improve the quality of conceptual models, reducing coupling is expected to improve the structure towards more understandable models. (Vanderfeesten et al. 2007a)

This definition founded several coupling metrics in process modeling. E.g. *Vanderfeesten et al.* present the coupling metric CP evaluating all pairs of nodes averaging their value over all pairs (Vanderfeesten et al. 2007b). Another metric by *Vanderfeesten et al.* is the cross connectivity metric analyzing the number of different possible paths in a process model (Vanderfeesten et al. 2008a). Other authors use already available metrics from Software-Engineering as starting point for their work. E.g. *Cardoso et al.* (2006) transfer metrics developed by *Halstead* (1977) that use information theory to quantify code reuse. They further transfer metrics by *McCabe* (1976) that quantify the paths through a model (Cardoso 2006). The fan-in/fan-out metric, quantifying branches, developed by *Henry/Kafura* (1981), is transferred by *Mendling* (2006) and *Cardoso et al.* (2006). Although these metrics exist, they do not exhaust the definition by *Vanderfeesten et al.* (2007a). Further, the range of existing definitions already demonstrates how vague the current understanding of coupling is and that an extensive range of metrics with their precise definitions is necessary to render more precisely the currently fuzzy understanding. Further each distinctive metric introduces an additional application scenario. We therefore continue the previous work by transferring further metrics.

2.5.2.2 EEPC modeling

Event-Driven Process Chains (EPCs) are a popular standard for business process modeling (Scheer et al. 2005; Mendling 2008). EPC models can be extended by additional information in different views (e.g. data view, organization view, etc.) (see Scheer et al. 2005) in which case literature then speaks of enhanced Event-Driven Process Chains (eEPCs). For the current work, relevant aspects of the eEPC can be formalized as follows (see van Hee et al. 2005; Mendling 2008).

An extended enhanced Event-Driven Process Chain (eEPC) is defined as weakly connected Graph $g=(N, A)$, fulfilling:

1. The set of nodes N is the union set of the four disjoint sets E, F, C, P and R where

- E is the set of events $E=E_s \cup E_f \cup E_i$ and E_s, E_f and E_i are the disjoint sets of start-, final- and intermediate events with $|E_s| \geq 1$ and $|E_f| \geq 1$.
- $F \neq \emptyset$ is the set of functions.
- C is the set of connectors,
- P is the set of process interfaces.
- R is the set of resources I encompasses the information elements: $I \subseteq R$

2. Each arc A in $A \subseteq (E \cup F \cup C \cup P \cup R) \times (E \cup F \cup C \cup R)$ connects two different nodes:

- $|n \bullet| = 1$ for each $n \in F \cup E_i \cup E_s$ and $|\bullet n| = 1$ for each $n \in F \cup E_i \cup E_f$.
- Resources are connected with undirected arcs.

3. Process interfaces have either an incoming or an outgoing arc: $\forall p \in P : (|\bullet p| = 1 \wedge |p \bullet| = 0) \vee (|\bullet p| = 0 \wedge |p \bullet| = 1)$

A hierarchical eEPC $eEPC_H = (G, h)$ is a set of eEPCs $g \in G$ and a partial relation $h : D \rightarrow G$ of the set D of decomposed functions or process interfaces in $Z : D \subset \coprod_{g \in G} (P, F)$. For a node $d \in D$ where $h(d) = g$, g is called subprocess of d or process referenced by d.

The above definition covers the notation which will be used later on. A more exhaustive definition of the eEPC modeling language can be found in (van Hee et al. 2005; Mendling 2008).

2.5.3 Methodology

Figure 23 summarizes our methodology. First, conducting a literature review, we search for already existing coupling metrics in both, Software-Engineering and process modeling. Second, we transfer discovered metrics from Software-Engineering to process modeling. This step is detailed in figure 24. The work ends with discussing the results. The conceptual approach behind this work is presented in (Braunnagel & Johannsen 2013). There we present the idea as well as the expected results of the transfer.

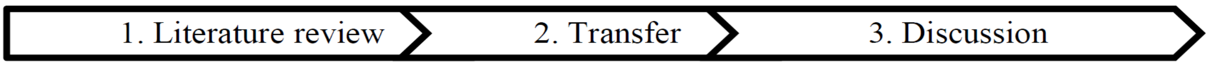


Figure 23: Methodology

For the review, the electronic databases Google Scholar, Computer.org, AISeL and Emerald Insight were queried (cf. Elliott et al. 1999; Vom Brocke et al. 2009). The hits, 46 peer-reviewed results were considered as relevant on the basis of their title or abstract, consist of 32 conference papers, nine journal papers, four technical reports and one book. Five sources defined metrics that are transferred and presented in this work. The remaining literature can be grouped as follows.

A first group discusses use cases, resp. consequences of high coupling. E.g. (Binkley & Schach 1998) discuss relations of coupling and run-time failures in software. The second group presents metrics that cannot be transferred to process models. E.g. (Green et al. 2009) present an approach involving runtime information which is not available in conceptual models. Third, sources discuss coupling metrics that were originally developed or transferred for eEPCs, (e.g. Cardoso et al. 2006; Mendling 2006; Vanderfeesten et al. 2007b; Vanderfeesten et al. 2008a). The existing metrics will be discussed more thoroughly in section 2.4.5. Finally, the fourth group of literature is redundant. These sources discuss metrics that are already part of the above groups. E.g. *Khlif et al.* transfer metrics to BPMN (2009). We refer to the original description. A more detailed presentation of transferred and not transferred metrics can be found in (Braunnagel & Johannsen 2013).

Use cases	(Arshad et al. 2007; Beyer et al. 2001; Binkley & Schach 1998; Chowdhury & Zulkernine 2010; El-Emam 2001; González et al. 2010; Harrison et al. 1998; Lee & Chan 2001; Markovic et al. 2009; Meyers & Binkley 2007; Rajaraman & Lyu 1992; Vanderfeesten et al. 2007b; Wahler & Küster 2008)
Not transferred	(Allier et al. 2010; Birkmeier 2010; Briand et al. 1998; Briand et al. 1999; Briand et al. 1997; Cho et al. 1998; Green et al. 2009; Gui & Scott 2008; Hitz & Montazeri 1995; Joshi & Joshi 2010; Orme et al. 2006; Perepletchikov et al. 2007; Qian et al. 2006; Quynh & Thang 2009; Rajaraman & Lyu 1992; Sunju & Joongho 2009)
Already specified	(Cardoso et al. 2006; Henry & Kafura 1981; Vanderfeesten et al. 2007b; Vanderfeesten et al. 2008a){
Redundant	(Chen et al. 2009; Khlif et al. 2009; Khlif et al. 2010; Rosenberg & Hyatt 1997; Sandhu & Singh 2005; Újházi et al. 2010)
Transferred	(Allen et al. 1999, 2001; Chidamber & Kemerer 1994; Gui & Scott 2006; Kazemi et al. 2011; Mendling 2006; Poshyvanyk & Marcus 2006; Reijers & Vanderfeesten 2004)

Table 31: Grouped literature review results

The remaining metrics were transferred as shown in figure 24.

Figure 24: Transfer

First, we identified the concepts of each metric's variables. Then, equivalent concepts in the eEPC notation were identified. Finally, the original concepts in the metric's definition were replaced to reformulate the metric.

2.5.4 Coupling metrics in the context of eEPC modeling

2.5.4.1 Process Coupling

Reijers/Vanderfeesten present the Process Coupling metric (see Reijers & Vanderfeesten 2004). Its objective is the delineation of functions that are to be executed en block. Since overly large work units turn processes inflexible and overly small work units increase the number of handovers making processes failure-prone, the balanced delineation of functions in a workflow is a means for its improvement. The functions size is measured by the number of connected information elements. (Reijers & Vanderfeesten 2004)

Identify concepts. The metric was originally defined for a graph of nodes and arcs representing information elements and operations respectively. The structure focuses the processing of information elements and is called information element structure. It is delineated into partitions representing activities. The metric calculates the quotient of the number of activities actually coupled and the number of activities possibly coupled. Activities that involve one or more common information elements are considered as “coupled”. (Reijers & Vanderfeesten 2004)

Identify equivalent concepts. To transfer the metric to eEPC modeling language, the procedure in section 2.5.3 was used. First, involved concepts were identified which are information element, activity and oper*ation*. Equivalent concepts were identified using the original description. Information elements exist in both domains with similar meaning. Activities express behavior and possess information elements as do functions in eEPCs. Operations, expressing the way information elements are combined at a very high level of detail, could not be matched with eEPC concepts. However, the calculation does not require them.

Reformulate metric. Adapted to the eEPC language and formalization from section 2.5.2.2, process coupling for eEPCs can be calculated as follows. The degree of Process Coupling *k* is the sum of coupled pairs of information elements divided by the maximum possible number of pairs.

- $$k=\begin{cases}\sum_{f_x,f_y\in F} connected\,\frac{(f_x,f_y)}{|F|*(|F|-1)}, |F|>1\\ 0, else\end{cases}$$

Functions are connected with each other if they share a common information element.

- $$connected(f_x,f_y)=\begin{cases}1, if\,(f_x,i_i)\in A\wedge(f_y,i_i)\in A\wedge(f_x\neq f_y)\\ 0, else\end{cases}$$

Application. The metric quantifies the interdependence of activities regarding information elements. To achieve a low degree of coupling, one reduces the number of coupled pairs, i.e. splitting tasks in such a way that information elements are grouped in the same function, or one increases the number of functions without introducing new pairs. A process with perfectly low coupling would use any information element only once as in- or output. A process with perfectly high coupling would be such that every step in a workflow depends on one and the same information. In such a process every step would come to a halt in case this one information was missing or the one person processing the information was ill, indicating a highly inflexible process design. However, it remains difficult to interpret the difference of two values, e.g. what is the impact of 10% more coupling? In summary, the metric has a special purpose, namely to quantify the dependency degree of process steps. It allows comparing different process designs and also gives a rough indication of how good or bad a design is regarding the coupling of activities.

2.5.4.2 Coupling of a module, intramodule coupling of a module

Allen et al. (2001) present a pair of metrics, the coupling of a module and the intramodule coupling of a module. They use information theory to quantify the amount of information in the structure with a special focus on connections between eEPCs. The authors argue that the cognitive limitation of a model user is a reason for misunderstandings and erroneous application if the model exceeds this limit. Therefore, the measure is a means of controlling the amount of information in the presented model. (Allen et al. 2001)

Identify concepts. The metrics focus a graph with modules that partition nodes. Nodes from different partitions can be connected. The coupling of a module assesses the graph structure connecting different modules. Therefore, the graph is reduced to arcs connecting nodes from different modules. Second, the arcs are used to build a predicate table, i.e. the incidence pattern, for each node. Third, the relative frequency of each predicate is used to calculate its entropy. Finally, the entropy values are summed up. The second metric, the intramodule coupling of a module, follows the same procedure with arcs connecting nodes within eEPCs.

Identify equivalent concepts. The transfer focuses the graphs of eEPCs. Accordingly, nodes in an eEPC, i.e. functions, connectors, resources, etc. are considered as nodes here. Further, arcs from an eEPC are considered arcs here. Modules group nodes and arcs; therefore we use an eEPC for modules. However, the eEPC notation has no arcs between eEPCs. Therefore we propose using process references and decompositions as the extension of the control flow, i.e. as arcs connecting eEPCs.

Reformulate metric. As a consequence of the previous step, the definition from section 2.5.2.2 is extended in the context of this metric by arcs between eEPCs:

B is the set of intermodule arcs:

- B_p is the set of process references from eEPCs referencing each other.
- B_{es} is the set of pairs of decomposed function and start-events of the referenced models.
- B_{ef} is the set of pairs of an end-event of a referenced eEPC and a decomposed function referencing the eEPC.
- Then B is defined as: $B = B_p \cup b_{es} \cup B_{ef}$. Each tuple in B is a directed arc called intermodule arc.

The intermodule sub-graph S_i^* consists of all the nodes of a group of eEPCs and arcs connecting nodes from different eEPCs with nodes from an eEPC *i*.

Inc_i^* is an incidence matrix of S_i^* : $Inc_i^* = (inc_{n,a}) \in S_i^*$ with $inc_{n,a} = \begin{cases} 1, if\ n_i \subset a \\ 0, else \end{cases}$

A pattern pat_J is a sequence of 0 and 1 of line vectors of the matrix. Its probability $Prob(pat_j)$ is its frequency over the number of distinct patterns.

The information content of a sub-graph S_i^* is defined as:

- $I(S_i^*) = \sum_{j=0}^{n} (-\log_2 Prob(pat_j))$

Finally, the coupling of a module is defined as:

- $Coupling(m|MS) = \sum_{i \in m} I(S_i^*)$

The metric intramodule coupling of a module follows the same steps, although instead of arcs connecting nodes from different eEPCs, with arcs connecting nodes from the same eEPC for the intramodule sub-graph S_i^* . The metric is defined as:

- $IntramoduleCoupling(m_k|MS) = \sum_{i \in m_k} I(S_i^*)$

Application. The metrics build on information theory and calculates the entropy of arcs as means of their complexity. It is therefore an ambitious attempt to quantify the cognitive load imposed on a model reader. The authors explain that a simpler structure is better understandable and indicated by a lower metric value (Allen et al. 2001). The practical application, however, is limited. For once, the metric does not account for the amount of information stemming from the nodes semantics. Further, the metric is constructed in a way that it is essentially driven by the number of nodes. Also, without any indicator about the actual cognitive limits of model readers, any calculated metrics value remains without reference and has therefore a weak indicational value. The metrics may therefore be used to compare two alternative layouts but do not allow any inference to be drawn about minimal, optimal or maximal values. Finally, a user will face trouble trying to understand what the metric actually does and why low values are important in this case. In summary, the metric is an interesting attempt to use information theory as a means of assessing the complexity of conceptual models. Nonetheless, the lack of reference values and complicated construction make the metric difficult to apply.

2.5.4.3 CBO, RFC

Chidamber and Kemerer (1994) introduced the CBO and RFC metric for object-oriented systems analyzing how classes are connected. They argue that highly connected classes are hardly reusable and difficult to change (Chidamber & Kemerer 1994).

Identify concepts. The CBO metric counts the connections of one class with other classes, the RFC metric also considers the number of methods in the source class.

Identify equivalent concepts. These metrics (and the following one) use the concepts *software program*, *class*, and *method*. Previously published literature transferred them in (Vanderfeesten et al. 2007a) and (Khlif et al. 2009). Further, (Green & Rosemann 2000) mapped eEPC constructs onto the ontology of (Weber 1997) and (Evermann & Wand 2005, 2009) mapped programming constructs onto (Weber 1997). Therefore *Weber's* ontology is used as mediator to compare both domains. Table 32 summarizes the transfer for the current context.

Method. The method in object-oriented programming expresses the behavior of classes (Armstrong 2006). For their transfer to BPMN, *Khlif et al.* (2009) suggest the analogy to tasks. Further, *Vanderfeesten et al.* (2007a) propose the analogy with operation elements. Since operations have no equivalent in eEPCs, functions are the best fit (cf. section 2.5.4.1). Ontological analyses of (Green & Rosemann 2000; Evermann & Wand 2005, 2009) suggest that functions have their ontological equivalents in transformations and therefore their object-oriented equivalent in operations. As before, the degree of detail of operations is not shown in eEPCs. The ontological equivalent of methods is lawful transformations, subsets of all possible transformations. Nonetheless, considering the lawfulness being negligible here, the analogy of method and function fits close enough. (Vanderfeesten et al. 2007a; Khlif et al. 2009)

Class. In object-orientation, classes group methods into logical units (Armstrong 2006). *Khlif et al.* map classes onto processes and sub-processes (Khlif et al. 2009). *Vanderfeesten et al.* relate classes to activities arguing along the hierarchy of methods and classes (Vanderfeesten et al. 2007a). Consequently, we suggest the equivalence of classes and sub-processes, since activities are already mapped onto functions. The ontological analyses of *Green/Rosemann* and *Evermann/Wand* (see Green & Rosemann 2000; Evermann & Wand 2005, 2009) suggest that classes find their ontological equivalent in functional schemas. They describe the temporal order of states, as is also done by process models. The ontological concept of a “process”, as mentioned by *Green/Rosemann*, could not be found. Therefore, the current mapping relates a class onto a sub-process diagram.

Software program. The software program is a set of classes (Armstrong 2006). The concept is ignored by *Khlif et al.* (2009). However, *Vanderfeesten et al.* (2007a) argue along the hierarchy of concepts to map programs onto business processes. We follow their suggestion. (Vanderfeesten et al. 2007a; Khlif et al. 2009)

Object orientation	eEPC notation
Software program	All eEPCs of a process
Class	Sub-process diagram
Method (private)	Function
Method (public) /Interface	Process interface, decomposition

Table 32: Conceptual Mapping

Reformulate metric. CBO is calculated as the number of connections from one eEPC to another eEPC.

- $CBO=|C \cup P|$

RFC counts the number of process interfaces and decomposed functions plus the number of functions in the eEPC.

- $RFC=|C \cup P|+|F|$

Application. The CBO metric quantifies the number of connections a model has with another model. The RFC metric additionally takes the number of functions of a model into account. Lower numbers indicate more readable models. The metrics from *Chidamber/Kemrerer* are well known and have been subject to empirical research (c.f. Harrison et al. 1998). Their application and interpretation is easy. They do, however, capture the complexity of the models only partly, e.g. they count the number of connections but do not evaluate them, and further do not incorporate all nodes, arcs and their meaning within models. Further, information about levels that constitute “easy” or “difficult” models is not available. In summary, the metrics are an easy and transparent way to analyze the number of connections between eEPCs. Still, without any information about the levels of the metric, the interpretation of a value is difficult. It remains to compare two alternative models.

2.5.4.4 Direct Coupling, Indirect Coupling, and Total Coupling

Gui/Scotts’ intention is to improve the CBO and RFC metric incorporating transitive relations (Gui & Scott 2008).

Identify concepts. The calculation takes three steps. First, the direct coupling between two classes is calculated as the quotient of commonly used methods to all methods in the first class. Second, the indirect coupling between two classes is calculated as the product of all direct coupling values on the longest path in between. Finally, the total coupling is calculated as the quotient of the sum of all indirect coupling values and the number of pairs of classes.

Identify equivalent concepts. Building upon the metrics CBO/RFC, the transfer of concepts in table 32 can be used again. EEPCs are used for classes, references for public methods, and functions for private methods.

Reformulate metric. The direct coupling metric calculates the quotient of process references between two eEPCs g_1 and g_2 and the functions and process interfaces in eEPC g_1. This is formalized:

- $$CoupD(g_{1,}g_2)=\frac{|D_{g_1,g_2}|}{|F_{g_1}\cup P_{g_1}|}$$

For a pair of eEPCs g_1 and g_2 connected by a path π (the longest available), the indirect coupling metric calculates the product of direct coupling values on the path:

- $$CoupT(g_{1,}g_{2,}\pi)=\prod_{g_{3,}g_4\in\pi} CoupD(g_{3,}g_4)$$

The metric is aggregated over all eEPCs in a system calculating the average indirect coupling among all eEPCs G:

- $$WTCoup=\frac{\sum_{i,j\in G} CoupT(i,j)}{|G|^2-|G|}$$

Application. The metrics extend the CBO metric by Chdiamber/Kemerer by paths over several connected eEPCs. It presents an indicator for the length of a process model and for how many different eEPCs need to be referred to in order to understand all paths in a process, where shorter lengths (a lower value) indicate a lower complexity. The metric is more sensitive than counting the number of eEPCs, since it takes into account which part of a process is reachable after all. I.e. a low value is reached if the parts are connected linearly so that a reader can follow the eEPCs in sequence. The value will rise if the parts are connected in circles and a reader has to refer to eEPCs repeatedly to follow a path through the process.

2.5.4.5 Conceptual coupling

Poshyvanyk/Marcus (2006) present the conceptual coupling metric that uses semantic information to calculate how far methods in object-oriented programming refer to the same semantic concept. A high semantic overlap indicates dependency causing complexity and should thus be avoided (Poshyvanyk & Marcus 2006).

Identify concepts. The metric references information retrieval techniques to decompose a set of classes into semantic concepts. *Poshyvanyk/Marcus* combine vector space retrieval and latent semantic indexing on the source code of classes as text corpus. First, the source code of the methods is transformed into a term-method matrix showing the frequency of a term in a method. Second, the matrix is transformed using latent semantic indexing, analyzing which terms are highly correlated forming a semantic concept. The values allow the calculation of the distance of two classes, judging how close their concepts are (cf. Poshyvanyk & Marcus 2006).

Identify equivalent concepts. The transfer takes special consideration of the authors' original intention. Therefore the transfer analyzes the role of the textual corpus. The role of a method is taken by an eEPC, whereas, instead of a class, the calculation is done with a group of eEPCs from the same process. In place of the terms from the source code, the redefined metric uses node labels.

Reformulate metric. Calculating the metric begins with building the term-eEPC matrix showing for each eEPC and each term its respective frequency. Second, a latent semantic analysis is applied on the matrix, reducing the matrix to its main components. The first metric, the conceptual similarity between eEPCs, CSM, uses the cosine of the vectors of two eEPCs in the reduced matrix as measure of distance.

- $$CSM(g_k, g_j) = \begin{cases} \frac{\vec{g_k}' * \vec{g_j}}{\|\vec{g_k}\| * \|\vec{g_j}\|} \; if \; \frac{\vec{g_k}' * \vec{g_j}}{\|\vec{g_k}\| * \|\vec{g_j}\|} \geq 0 \\ 0, else \end{cases}$$

The second measure is the similarity of an eEPC *g* with a group of eEPCs *gg*. Therefore, the average conceptual similarity of one eEPC with all eEPCs of the group is calculated:

- $$CSMMg(g_i, gg_j) = \sum_{m_j \in mv_j} CSM \frac{(g_i, g_j)}{|g_j \in gg_j|}$$

Third, the conceptual similarity of an eEPC group with another eEPC group is calculated as the average CSMMg of their eEPCs:

- $$CSMgMg(gg_i, gg_j) = \frac{\sum_{m_j \in mv_j} CSMMg(g_i, gg_j)}{|g_i \in gg_i|}$$

Finally, the conceptual coupling of an eEPC group can be calculated as the average coupling of a group with all other eEPC groups:

- $$CCMg(\mathrm{gg}_i) = \frac{\sum_{mg_j \in MG} CSMgMg(gg_i, \mathrm{gg}_j)}{n-1}$$

Application. The conceptual coupling metric uses an information retrieval technique that discovers semantic concepts and evaluates the degree of redundancy in the concepts, resp. terms, among eEPCs. It therefore analyzes whether either nodes are labeled ambiguously or similar tasks appear in different contexts and models. High values indicate a high semantic overlap, i.e. many common terms. The same terms reused in different contexts impair understandability. Our adaption does not define the construct of a group of eEPCs strictly, since it depends on the use case. The groups should be formed by domain, i.e. groups of processes that are supposed to deal with the same terms or not, as for example eEPCs for processes that belong together.

2.5.5 Implementation

In the previous sections the metrics were presented, transferred, re-specified, and their contribution to the assessment of process model quality was discussed. However, as can be taken from the definition of some of the metrics, the complicated calculation of some of the metrics makes their practical application tedious. E.g. the conceptual coupling metric requires a singular value decomposition of a term-model matrix over all terms used. As an application aid, we implemented the metrics presented in this work in the form of a Plug-In (available at https://svn.win.tue.nl/trac/prom/browser/Packages/CouplingMetrics/) for ProM. ProM is a framework offering several techniques for process mining and model analysis (cf. Dongen et al. 2005). The implementation assumes to find eEPC elements as defined but remains oblivious to their source format. To ensure the functionality, we also extended the EPML-Interface of ProM for eEPC elements that are required for the metrics. Contrary to proprietary formats such as ARIS-XML or VDX, EPML is a platform-independent XML-schema with a publicly available schema-definition (cf. Mendling & Nüttgens 2006). We used the plugin with twelve different eEPC models to gain a first impression about the applicability of the metrics and the plugin. It showed that though the implementation produced values for each metric and model, their application suffers from a lack of reference. Thus it remains unclear how strong the effect onto the reader is if models perform e.g. 10% better or worse regarding a certain metric. Nonetheless, the metrics serve for the comparison of two models, giving a rough indication if one model performs better or worse than another in respect to a metric (c.f. Braunnagel & Johannsen 2013).

2.5.6 Summary, limitations and outlook

This work discusses the topic of “coupling” in process modeling. Even though it is recognized as an important quality dimension (see Vanderfeesten et al. 2007a; Vanderfeesten et al. 2008c) for process models it has not been explained in detail yet. Coupling metrics exist, especially in neighboring disciplines such as Software-Engineering, based on individual and heterogeneous perceptions of coupling, while the understanding of coupling in process modeling is sparse and vague. Our research addresses this gap by analyzing and transferring ideas on “coupling” from the field of Software-Engineering to gain a better understanding and application of this ill-defined concept. Thus our contribution is the transfer of a well-established means of controlling and managing quality from systems development to process modeling. Therefore, our work supplements the metrics allowing the measurement and management of the coupling of process models. Next to their application, the metrics provide additional definitions of the concept of coupling. They constitute elementary groundwork for the discussion of coupling in process models as well as for the fuzzy concept of process model quality and understandability in general.

However, there are limitations. First, our understanding of coupling builds on preliminary work on coupling (see section 2.5.3). Future developments regarding coupling might bring new interpretations requiring our transfer to be repeated. Second, the transfer was influenced by subjectivity regarding the interpretation of equivalent concepts. However subjectivity was mitigated by two researchers conducting the procedure and consolidating the results. Finally, we focused eEPCs to provide a reasonable level of detail. The metrics’ interpretation will differ for other languages such as e.g. BPMN or UML.

In future work, the metrics will be evaluated empirically. We will analyze which metrics, and thus underlying perspectives, influence process model understandability most. Based on these insights, guidelines for producing process models that are easy to understand (regarding coupling) can be formulated. They will then be tested with practitioners and adapted to their specific needs.

2.6 Beitrag 6: Applying evaluations while building the artifact—Experiences from the development of process model complexity metrics

Adressierte Forschungsfrage	**Forschungsfrage 6**: Erfüllt die Konstruktion der Metriken bekannte Kriterien aus der Forschung?
Erscheinungsort	48th Hawaii International Conference on System Sciences, HICSS 2015, Kauai, Hawaii, January 5-8, 2015 (VHB Jourqual 3: C)
Autoren	Daniel Braunnagel 90% Prof. Susanne Leist 10%
Der Beitrag leistet zweierlei. Zum einen werden die Metriken anhand der Kriterien von Weyuker (1988) evaluiert. Für die festgestellten Verstöße werden deren Auswirkung im praktischen Einsatz aufgezeigt und Lösungsmöglichkeiten vorgeschlagen. Zum anderen wird das DS evaluation pattern nach Sonnenberg und vom Brocke (2012a) angewendet und aufgezeigt, dass, und unter welchen Bedingungen, das Pattern dem DS Prozess zugutekommt. Der Beitrag erläutert den ersten Teil der Evaluation des Artefakts. Es werden der Entwicklungsprozess hinterfragt und die Metriken anhand der Kriterien nach Weyuker (1988) evaluiert.	

Tabelle 33: Bibliographische Angaben zu Beitrag 6

The Design Science Research method is decisive for the quality of the resulting solution. Thus, many discussions focus the evaluation of the solution at the end of the Design Science cycle. But design, implementation and evaluation of artifacts are laborious and need to be repeated if the artifact does not meet the evaluation criteria. Thus, recent works have proposed to conduct additional evaluations early in the Design Science process to possibly reduce the number of repetitions of the research process. However, such early evaluations may also be an unnecessary burden. Therefore, this work presents a case where these additional evaluations are applied ex-post in a practical research project which developed process model complexity metrics and the outcomes are compared. Once compared, benefits and limitations of early evaluations are discussed.

2.6.1 Introduction

The application of Design Science Research (DSR) in Information Systems Research is currently an often discussed topic. Especially researchers are interested in improving the rigorous application of the method (cf. Hevner et al. 2004; Pries-Heje et al. 2008; Sonnenberg & Vom Brocke 2012a). Many publications in DSR focus on the evaluation of the artifact. They define techniques or criteria to prove whether the developed artifact meets defined requirements and can be considered an optimal or adequate solution for the problem. Occasionally, researchers essentially follow a search process to find an effective solution for a problem. This search process forces them to conduct the build and evaluation phases in many iterations as the developed artifact has to be evaluated to identify whether the problem is solved (Hevner et al. 2004). If the proof fails, a new solution has to be developed, which will be evaluated again to see if it is more satisfying or optimal. This is referred to as the design cycle (Hevner & Chatterjee 2010; Hevner et al. 2004).

For the aim of reducing the number of design cycles, a promising approach could be to focus on the Build phase and identify techniques or criteria to support the development of an optimal or at least satisfying solution. On the other hand, identifying and using these techniques or criteria is time consuming and only few researchers have been dealing with the definition, application or usefulness of such approaches to improve the build phase.

Against this background, the aim of the paper is to investigate in how far such an effort in the build phase can contribute to the development of the artifact and e.g. can reduce the number of design cycles.

We do so at hand of a design science (DS) project, which was previously conducted in the traditional way. For the current work, we repeat the build phase with the aid of evaluation techniques and discuss the impact techniques and criteria to support the development in the build phase have in this case.

In the previous DS project, we developed so called coupling metrics. They are used to assist process modelers with guidance about the quality of the models. The perspective of coupling evaluates the understandability in particular. (cf. Braunnagel et al. 2014)

For our evaluation at hand, this particular DS project is especially interesting. First, since the metrics are fully formalized and can be described within a paper, the influence which the different methods have on the artifact can be discovered clearly. Further, the development of the metrics is complex and the current research provides little guidance for design decisions, because of which the results of evaluations are not foreseeable during the build phase. *Sonnenberg and vom Brocke* (2012a) call this situation the emergent nature of knowledge in DS projects, because of which they propose to evaluate early and often along the DS process (cf. 2012a). And since we had recently built the metrics with a traditional build and evaluate approach, we were highly interested if and how the results differ with another evaluation strategy.

The structure of our paper is as follows. Section 2.6.2 explains the basics, which comprise the Design Science and coupling metrics. Section 2.6.3 presents the methodology which we followed originally, the methodology which includes the additional early evaluation activities and the corresponding evaluation criteria. In section 2.6.4 we explain how the evaluation was conducted in detail, as well as its result. Section 2.6.5 discusses the implications of performing additional, early evaluations and section 2.6.6 concludes our work.

2.6.2 Basics and related work

2.6.2.1 Design Science

The design science paradigm seeks for the enhancement of human and organizational capabilities by creating new and innovative artifacts. Emanating from engineering and the sciences of the artificial (Simon 2008), it is concerned with the design, development, implementation, and use of socio-technical systems in organizational contexts. Design scientists produce and apply the knowledge about tasks or situations in order to create effective artifacts (March & Smith 1995). Thus, DSR is fundamentally a problem solving paradigm. A challenge in design science results from the fact that an artifact's performance depends on the environment in which it is used. An incomplete understanding of the environment can lead to inappropriately designed artifacts (March & Smith 1995). In consequence, the evaluation of the designed artifacts is particularly important.

Currently a variety of different approaches for the conduct of design science research can be found (cf. Klecun & Cornford 2005), which all describe a process organized in the two phases build and evaluate (cf. Offermann et al. 2009; Sonnenberg & Vom Brocke 2012a). As a prominent example, *Peffers et al.* developed an approach which represents the synthesis of design science processes from IS and other disciplines and comprises six steps: (1) identify problem & motivate, (2) define objectives of a solution, (3) design & development, (4) demonstration, (5) evaluation, and (6) communication of the results (see Table 34) (Peffers et al. 2007).

Whereas the first three steps are part of the build phase of the design science research method, the last three steps are assigned to the evaluation phase. The build phase is especially in design science projects of great importance and researchers spend much time on designing and constructing the artifact (Sonnenberg & Vom Brocke 2012a). Accordingly, many publications on design projects lay their focus on the build phase, while the evaluation phase is often either neglected or only described as an evaluation concept. Only a minority of these publications do in fact evaluate the developed artifact (Griesberger 2014). A possible explanation for the high emphasis on building an artifact could be that it is a less satisfying duty for researchers to check whether all their efforts to strengthen the applicability and usefulness during the construction of the artifact does actually hold truth value during its evaluation (Sonnenberg & Vom Brocke 2012a).

In addition, many publications discuss the use of the design science research method theoretically, mostly without focusing a specific DS project (e.g. Gregor & Jones 2007; Pries-Heje & Baskerville 2008; Hevner et al. 2004). In these publications, the main question is how to conduct the design science research process more rigorously in order to provide guidance for the researchers. In contrast to practical publications of applied design science projects, theoretical works often emphasize the evaluation phase (e.g. Pries-Heje et al. 2008) and introduce several methods and techniques for evaluation (cf. Klecun & Cornford 2005; Venable et al. 2012). E.g. *Venable et al.* (2012) present an evaluation framework, assisting a DS researcher in the selection of methods for ex ante and ex post evaluations. They provide detailed guidance for the evaluations themselves, though without alignment with current DSR processes. (cf. Pries-Heje et al. 2008) Also, only few theoretical publications focus on the construction of the artifact (cf. Pries-Heje et al. 2007). Since the practical publications are necessarily case specific, only rather rudimental guidance is provided for the Build phase, and almost none of these publications consider the different types of artifacts (Griesberger 2014). The best known publications for the build phase are the following.

- *Vaishnavi and Kuechler* (2007) define patterns which describe techniques that can be applied to support the construction of the artifact. Similarly, *Sonnenberg and vom Brocke* (2012a) describe patterns which are used to evaluate the results of different steps during the build and evaluation phase of a design science project.

- Some authors define activities or guidelines which describe in more detail tasks to conduct in the build phase (e.g. Hevner et al. 2004; Offermann et al. 2009; Fischer & Gregor 2011)

- Other authors define requirements an artifact has to meet, which should already be considered during the construction of the artifact (e.g. Iivari 2007; Gregor & Jones 2007).

- *Gericke* suggests approaches to support the construction for three artifact types (Gericke 2009).

- *Sein et al* (2011) developed an approach which conducts the activities during the Build and Evaluate phase concurrently to immediately reflect the progress achieved and to trigger artifact revisions early within a design process.

2.6.2.2 Coupling Metrics

The subject of the DS project upon which we conduct the analysis is the development of so called “Coupling Metrics” which are used to support business process modeling in assessing and managing the quality of process models (Vanderfeesten et al. 2007a). Coupling does not only evaluate the quality of single process models, there is a strong focus on interdependencies between models as well. Further, many operationalizations of coupling can be calculated automatically. This is an obligatory prerequisite for the use in practice, which frequently encompass a very high number of process models (Malinova et al. 2013).

Coupling in process management was preceded by coupling in Software-Engineering. There, it was recognized as an indicator for the complexity of conceptual models. As such, the indicator is used to predict areas of high complexity, since the complexity of a system is known to cause implementation errors. Thus, coupling in conceptual models in Software-Engineering is assessed with the intention to avoid errors in the conceptual stage, prior to their implementation when it is more difficult and expensive to correct them. (Vanderfeesten et al. 2007a)

In process management, the means to measure coupling in conceptual process models are based on transferring knowledge from Software-Engineering to process modelling (cf. Vanderfeesten et al. 2007a; Khlif et al. 2009; Braunnagel & Johannsen 2013; Braunnagel et al. 2014). For example, *Vanderfeesten et al.* (2007a) introduce the concept as originating in Software-Engineering: “Coupling is measured by the number of interconnections among modules. Coupling is a measure for the strength of association established by the interconnections from one module of a design to another. The degree of coupling depends on how complicated the connections are and on the type of connections.” Thus, coupling is measurable and the measurement uses modules and interconnections as input. Further, the measurement indicates complexity (cf. Vanderfeesten et al. 2007a). As such, the measurement is again conducted with the intention to identify areas of high complexity in conceptual process models. Just as in Software-Engineering, it is expected that highly complex processes lead to errors during their implementation. Further, due to their formalized description, the metrics can be computed automatically without user intervention (Braunnagel et al. 2014). They thus extend the currently existing means to measure and control the quality of conceptual models in business process management.

2.6.2.3 Development of Coupling Metrics

To make the concept of coupling available for end users, we transferred coupling metrics from the Software-Engineering domain and specified them for the use in process models (cf. Braunnagel & Johannsen 2013; Braunnagel et al. 2014). In order to guide the transfer, we followed the activities of the Design Science Research (DSR) method. Therefore, DSR supports the development of our coupling metrics for process models and additionally helps to ensure the applicability and usefulness of the developed metrics.

From a DS perspective, our artifacts are the metrics' implementation in a process modeling environment, where they are supposed to assist process modelers in creating models that are easier to understand for a user. The metrics measure the complexity of process models providing guidance to improve the models, for which, however, the metrics need to be easily accessible and provide useful information. The respective quality of the metrics is regularly gained through an evaluation in a practical setting, e.g. in a case study, and serves to re-design, re-implement and re-evaluate the artifact, which, however, is laborious and expensive. To reduce the number of design cycles, an ex-ante evaluation as proposed by *Sonnenberg and vom Brocke* (2012a) seems promising to prevent potential design flaws which would otherwise surface either during the construction or, worse, during the practical evaluation.

2.6.3 Methodology

2.6.3.1 DSR methods

The well-known methods to conduct Design Science Research have general similarities (Fischer & Gregor 2011). Table 34 shows a comparison of prominent DSR methods, adapted from *Fischer and Gregor* (2011) where similar steps are shown in the same line. While we can assume a general sequence of steps in the DSR process, we omitted returns for reasons of clarity. While some of the authors defined returns for their DSR process, the trigger was not always obvious. In this work, however, the focus are returns due to the result of an evaluation and it can be assumed that in every method the result of an evaluation can be a reason to repeat the previous steps in a DSR cycle.

For our purpose, Table 34 highlights the evaluation steps of each method and also includes the DSR method by *Sonnenberg and vom Brocke* (2012a). It can clearly be seen that while the currently prominent methods propose to explicitly evaluate the artifact only at the end of one cycle, the latter method proposes to evaluate after each step to avoid causes of repetition beforehand.

Our investigation aims to contribute to the discussion whether early evaluations in the build phase are beneficial or superfluous. To provide practical insights, we repeat a previous DS project in which we followed the Build and Evaluate approach. Now, we perform the Build phase with early evaluation activities in addition and compare results. This allows us to show whether in this case the gains, both in the quality of the solution and in the reduced DS cycles outweighed the additional evaluation effort, or not.

(1) Novelty/ Anomaly	(1) Construct a conceptual framework	(1) Important and relevant problems	(1) Identify problem and motivate (2) Define objectives of a solution	(1) Awareness of problem	(1) Identify problem (2) Eval1
(2) Generation of conjectures	(1) Construct a conceptual framework (2) Develop a system architecture (3) Analyze and design the system	(2) Iterative search process	(2) Define objectives of a solution (3) Design and development	(2) Suggestion	(3) Design (4) Eval2
(3) Generation of hypotheses	(3) Analyze and design the system (4) Build the prototype system	(3) Evaluate	(3) Design and development (4) Demonstration	(3) Development	(5) Construct (6) Eval3
(4) Empirical testing of the hypotheses	(5) Observe and evaluate the system	(4) Communicate	(5) Evaluation (6) Communication	(4)Evaluation (5) Conclusion	(7) Use (8) Eval4
Idealized Research Model (Fischer & Gregor 2011)	Nunamaker et al. (Nunamaker, Jay, F., Jr. 1990)	Hevner et al. (Hevner et al. 2004), also Peffers et al. (Peffers et al. 2007)	Peffers et al. (Peffers et al. 2007)	Takeda et al. (Takeda et al. 1990), Kuechler & Vaishnavi (Kuechler & Vaishnavi 2008)	Sonnnenberg & vom Brocke (Sonnenberg & Vom Brocke 2012a)

Table 34: Comparison of different DS approaches (cf. Fischer & Gregor 2011)

2.6.3.2 Previous procedure

The focus of our paper is not the evaluation of approaches for the Build phase in general. Instead, we want to show the usefulness of evaluations in the Build phase to reduce the overall effort. Thus, we present the following discussion on the basis of the DS methodology by *Sonnenberg and vom Brocke* (2012a), which is applicable for the construction of our coupling metrics.

So far, we pursued the research of coupling metrics in a traditional Build-Evaluate approach. Table 34 visualizes our approach as an adaption of the method by *Peffers et al.* (2007).

(1) First, as part of our research on process model understandability, we discovered that the currently available means to control process model understandability were insufficient, while e.g. Software-Engineering successfully used metrics to analyze the complexity of conceptual models.

(2) In the second step, we defined the objectives for the metrics in more detail. E.g. we decided to limit our work on metrics which can be applied in the design phase of business process management, as such metrics would serve to avoid issues prior to their implementation.

(3) Third, we designed the actual metrics. To do so, we conducted a literature review with the objective of discovering existing coupling metrics for conceptual models in e.g. Software-Engineering. The review was aligned to the review method by *Cooper (1999)*. The well-known literature databases Google Scholar, Computer.org (IEEE Computer Society), AISeL and Emerald Insight, that offer a wide range of different electronic sources were queried using the term pair '"coupling metrics" "business process model"' as well as "coupling metrics". We then transferred the metrics from their original domain to process modeling. For the transfer itself, we distinguished between the cause of the complexity impairing the users' understanding and the artifacts upon which the cause was calculated by each metric. (cf. Flood & Carson 1993) E.g. one metric would compute upon the number of connections between two randomly chosen artifacts in the model (cf. Gui & Scott 2008), whereas another one would focus the hierarchical decomposition of the modelled artifacts (cf. Chidamber & Kemerer 1994). This allowed us to search for artifacts within process modelling fulfilling an equivalent role regarding e.g. the number of steps in the control flow for any nodes of the control flow or the hierarchical decomposition of nodes in a process model hierarchy. Then, with these artifact candidates we were able to redefine the metrics by replacing the original artifacts with those from the domain of process modeling. At this stage, the coupling metrics were defined, based on the theoretical description of the respective process modelling language. At this stage, we had a theoretical definition for each metric, the artifacts' design. To make this design accessible

for the end-user, we still needed to actually construct the metrics, as so far we had only implemented the metrics in a process modeling environment as a prototype. A more detailed presentation of our research work can be found in (Braunnagel & Johannsen 2013; Braunnagel et al. 2014). The resulting metrics are presented in sect. 2.6.3.5.

Originally, the subsequent activities were the (4) demonstration and (5) evaluation of the artifact, which in case of the metrics would be twofold. First, we planned to conduct a laboratory experiment to verify that after the adaption the metrics still do indicate complexity and a degraded understandability. Second, the usability of an actual implementation needs to be evaluated in an organization, eg. by means of a case study. Both evaluations are laborious and would have to be repeated if they uncovered reasons to alter the design of the metrics.

For later reference, it is important to point out that this method does not refer to intermediate evaluations. The first systematically manifested feedback on the artifact is expected after the “Design & Development” and “Demonstration” steps. In the preceding steps, reasons to reconsider the design of the artifact in special or the method in general surface only by coincidence and not due to a systematic assessment.

2.6.3.3 Applying the DSR Evaluation pattern

In order to assist researchers in their DS projects, *Sonnenberg and vom Brocke* (2012a) propose the so called “General DSR Evaluation Pattern”. The pattern is a high level description of a DS process, which takes into account the emergent nature of DS artifacts by introducing additional evaluation steps. In detail, the described process consists of the four DS activities “Identify problem”, “Design”, “Construct”, and “Use” each of which is followed by an evaluation activity “Eval1” to “Eval4”. Depending on their position in the pattern relative to the construction activity, they are considered Ex-Ante or Ex-Post evaluations. (cf. Sonnenberg & Vom Brocke 2012a)

The evaluation activities of the general DSR Evaluation Pattern are described in more detail by separate patterns. There, Eval1, which follows the identification of the problem, is termed Justify. It serves to show that the current DS project is a meaningful one. Next to its objective, the description of the evaluation activity also shows possible methods to do so, e.g. an assertion, a literature review or a review of practitioner initiatives. (cf. Sonnenberg & Vom Brocke 2012a)

The second evaluation activity of the process, Eval2, follows the design activity. It is meant to evaluate the design and to show that the design can bear a possible solution of the DS problem. Possible methods of this activity are an assertion, a mathematical proof, or logical reasoning, etc. (cf. Sonnenberg & Vom Brocke 2012a)

The Ex-Post evaluation activities follow the construction of the artifact as well as its use. Eval3, performed after the construction, is meant to initially demonstrate the artifact, e.g. by a prototype demonstration or experiment. Eval4 evaluates the artifact in its environment to show that it is practically useful, e.g. with a case study or a field experiment. (cf. Sonnenberg & Vom Brocke 2012a)

Comparing our previous procedure and the Evaluation Pattern by *Sonnenberg and vom Brocke* (2012a), Table 34 shows which additional steps we performed ex-post when we repeatedly developed the metrics with the new method.

(1) We started with the identification of the problem, which led us to the same problem statement as the previous DS project had.

(2) Second, the pattern suggests to evaluate the identified problem, by means of e.g. a literature review. To do so, we use the review method by *Cooper* (2006), proposing five steps to specify a systematic conduct of the review. Starting with the problem statement and the search criteria, the method lead us to the same review we had conducted for the previous DS project. This review gave us feedback on a potential solution, we consider this early evaluation done. In fact, the literature we found underlined the relevance of the problem we identified once again and also provided feedback about the objectives because of which we focused on fully automatized metrics.

(3) Third, the pattern suggests to design the artifact. Since, up to this point, we had no additional knowledge with respect to design decisions, we came up with the same metrics as in our previous DS project.

(4) Fourth, after the design, the pattern proposes another evaluation, Eval2, by means of e.g. an assertion. However, originally we did not evaluate the design of our artifact (Eval2) before we implemented a prototype of the metrics. Thus, we do so in the following.

(5)-(8) Following the Eval2, the pattern suggests to construct the artifact. To do so, it is planned to implement the metrics in a process modelling software where they are accessible for modelers in their daily work. This implementation does then require an additional evaluation (6), before the use phase (7) and its evaluation (8), which provides feedback for further problems. These steps are subject for future work, which depends on the results of the early evaluations (2) and (4).

To analyze the benefit of early evaluations in the build phase, we instantiated the general DSR evaluation pattern. Thus, we performed the additional Eval2. It is to our benefit that the experiences from a long history of research on complexity metrics for conceptual models is documented in Software-Engineering by *Weyuker* (1988). We could thus base our informed argument (cf. Sonnenberg & Vom Brocke 2012a) on extensive previous knowledge and benefit from the rigorous documentation of prescriptive knowledge on previous instantiations of the DSS methodology.

2.6.3.4 Evaluation criteria

To evaluate the design of our artifact (Eval2), we adapted the criteria by *Weyuker* (1988) for process models, a set of desirable properties of complexity metrics in relation to their calculation:

P1: A metric should not rate all models equally complex, regardless of differences in their content.

The notion behind this property is that if one metric assigns the same level of complexity to each and every process model, it has no value to a user. Further, such a metric would contradict both our intuition and our empirical knowledge that differences in the complexity of process models do exist.

P2: A metric should not divide all models in only a few complexity classes.

This property extends **P1**. If a metric assigns e.g. only two or three different classes of complexity to process models of each and every size and shape, this is only of limited value to a user. Such metrics would not be suitable to prioritize a large set of models to be reworked, as many models of highly different complexity would be assigned the same class.

P3: A metric should allow for different models with equal complexity.

To explain this property, let us imagine a metric violating this property. It would assign a unique degree of complexity to each model, leading to an order of absolutely all possible process models. Since different processes lead to different models, this would imply that most processes cannot be modelled in a simple fashion. This is unrealistic. A sufficiently high degree of abstraction would lead to a simple model, which would still differ albeit only by the names of the nodes. Here, we preclude the case of infinite decimal places, as in practical settings users would most probably ignore very small differences anyway and thus assume different models to have equal complexity.

P4: A metric should allow for different models with the same semantic with different complexity.

The same process can be displayed in models which differ, despite having an equal level of detail and same information. Such a case can be devised by decomposing process models or aggregating functions differently. Different decompositions of the same process can lead to different degrees of complexity as is shown in empirical work (cf. Johannsen et al. 2014a)

P5: A metric should be monotone, thus the complexity of two concatenated models cannot be lower than the complexity of either of the two individually.

Complexity, as a property of the artifact, can further be disaggregated into the number of elements and their connections. Thus, if the number of elements increases, the complexity of a model will increase as well. Further, experience has shown that if one combines two previously separated processes into one common model, especially when done poorly, a reader will be even more overwhelmed.

P6: A metric should account for that two models with the same complexity may interact with a third model in different ways and thus have different complexities if concatenated.

Again, mind the decomposition of process models. If the concatenation is performed upon sub-models of a process model, it makes a difference if, otherwise identical, models are e.g. either attached to the end of the parent model or if the concatenation extends an already complex branching structure.

P7: A metric should account that permutations of one model may lead to different complexities.

This property reflects the motivation to decompose process models in the first place. If a process model is both very detailed with an extensive branching structure and very large, its complexity may challenge a reader. Therefore, modelers may break the model into parts with different decomposition. Overall, this will not change the model, but only permute its nodes over different parts. The resulting degree of complexity, however, depends on the actual decomposition. It is not hard to imagine, that a decomposition that tears apart closely related parts of a process model will increase the overall complexity (cf. Zugal 2013)

P8: A metric should result in the same complexity if models are renamed.

We cannot make up a case where the naming of a process model increases or decreases its complexity.

P9: A metric should allow that the complexity of two models united is more than the sum of the individual models.

Imagine that processes with previously separated resources are joined into one common process. This will introduce interaction between the processes due to shared resources which did not exist before. Thus, the coupling encompasses not only the sum of the two models but also the newly introduced connections.

P10: A metric should not allow for the complexity of two united models to be lower than the sum of individual models.

This property extends **P5**, complexity is considered the result of elements and their connections. Assuming that by uniting two models the resulting number of nodes equals at least the sum of the individual numbers, the complexity of two models united cannot be lower than the sum of the individual ones.

We use these criteria to conduct the evaluation of our metrics' design, which was not part of our previous methodology.

2.6.3.5 Selected metrics

A short description of the metrics upon which the evaluation is done can be found in the following.

Coupling of a module This metric uses information theory to quantify the amount of information in the model graph within a sub model. An exceedingly high amount of information is expected to correlate with a decreased understanding and indicate complexity. (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Allen et al. 2001) **Intramodule Coupling of a module** This metric quantifies the amount of information as well, but does so upon the graph of arcs which connect sub models instead of the model graph itself. (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Allen et al. 2001) **CBO** The argumentation behind the CBO metric is that sub models with a high number of connections are suspect of more external influence with unpredictable behaviour and are thus more difficult to understand. Therefore, the metric counts the connections a model has with other models to assess this form of complexity. (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Gui & Scott 2008) **RFC** The RFC metric extends the CBO metric. Here, the metric further asses the size of a model by the number of functions. (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Gui & Scott 2008) **Direct Coupling** For the Direct Coupling metric, complexity is the result of connections between models in relation to the number of functions. (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Chidamber & Kemerer 1994) **Indirect Coupling** The Indirect Coupling metric extends Direct Coupling over transitive connections. Thus, the strength of the connections between two randomly chosen submodels is calculated. (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Chidamber & Kemerer 1994) **Total Coupling** The Total Coupling metric aggregates the prior over all sub models to indicate the overall complexity of a set of models. (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Chidamber & Kemerer 1994) **Process Coupling** Its objective is the delineation of functions that are executed in one block. Since overly large work units render processes inflexible and overly small work units increase the number of handovers, making processes failure-prone, the balanced delineation of functions in a workflow is a means for its improvement. For this metric, a function is large if it refers to many information elements and functions are coupled if they share a common information element. (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Vanderfeesten et al. 2008c)

2.6.4 Evaluation

In the following, for the purpose of illustration, we discuss the properties with one of the metrics, and Table 35 shows the result for the remaining metrics. For more information readers may refer to (Braunnagel et al. 2014) to verify our application of the properties onto all metrics.

2.6.4.1 Exemplary application

The **Process Coupling** metric was originally invented by (Vanderfeesten et al. 2008c). It compares different process designs regarding the alignment of tasks in functions in a process. As the dependence between two functions due to shared information increases the number of handovers and possible failures, the metric calculates the fraction of functions which depend on the same information. We adapted the artifacts to the information elements and the functions in a business process. Thus, it is calculated as follows:

$$k = \begin{cases} \dfrac{\sum_{f_x, f_y \in F} connected(f_x, f_y)}{|F| * (|F| - 1)}, |F| > 1 \\ 0, else \end{cases}$$

If the set of functions (F) in a process model is greater than 1, then the degree of coupling (k) is calculated by the quotient of the sum of connected function pairs (f_x, f_y) to all possible function pairs (|F|*(|F|-1)).

$$connected(f_x, f_y) = \begin{cases} 1, if\ (f_x, i_i) \in A \wedge (f_y, i_i) \in A \wedge (f_x \neq f_y) \\ 0, else \end{cases}$$

A pair of functions (f_x, f_y) is connected, if there is an arc (f_x, i_i) between the function f_x and the information element i_i and an arc (f_y, i_i) between the function f_y and the information element i_i in the set of arcs A. Originally, the authors argue that with a high degree of shared information elements, a workflow becomes less flexible. We argue, that this principle applies to business process modelling as well. Evaluating our design against the previous ten properties might present further insight into whether the implementation of the metric is a worthy pursuit.

Regarding **P1** and **P2**, it can be easily seen that the ratio of connected to all function pairs is different for models with either a different number of functions, connected functions, or both. Also, since there is no syntactical limit to the number of functions in a process model, the number of possible coupling degrees is unlimited within the range of [0, 1] as well. Thus, the properties **P1** and **P2** are fulfilled by the metric.

P3, different models with equal complexity, is fulfilled as well. Since the metric refers only to the number of functions and information elements, but not to their semantic, one may easily replace the actual elements, and thus create new models with the same complexity. Also, one may alter further nodes or the size of a model. As long as the ratio of connected and unconnected function pairs remains constant, all different models have the same degree of complexity.

The metric was originally created to assist companies in creating flexible processes by aligning tasks differently. Aggregating tasks into functions in such a fashion that collects those tasks which require the same information elements leads to processes with a low degree of **Process Coupling**, notwithstanding that another composition of the same process might lead to another degree of coupling. Thus, **P4** is fulfilled.

P5, two models together may not be less complex than any of the individual ones, is violated due to the scaling. For two models, with the one model having coupled function pairs, and the second one not having coupled functions, the resulting **Process Coupling** Degree will be lower than that of the first model. In fact, the current construction of the metric tempts a user to create larger process models (i.e. include more functions) to reduce the degree of **Process Coupling**. Thus, the alignment of functions and information elements remains unchanged, even though it was identified as a source of inflexibility in the first place. The solution therefore is to omit the scaling by |F|*(|F|-1). This alternation would not violate the previous conditions, for the same reasons as before.

Regardless of whether we omit the scaling or not, **P6** is fulfilled by the metric. If a model is joined with one of two other models, it may or may not happen that functions from both models share information elements and that thus the number of paired functions increase for more than the additional model. Therefore, further interaction may affect the metric value, depending on whether this scenario happens or not.

The original intention of the metric was to point out the alignment of functions and information elements leading to the lowest sharing of information elements, thus encouraging process designer to permute functions and information elements in such a way as to reduce **Process Coupling**. Thus, **P7**, a metric should account that permutations lead to different complexities, is fulfilled.

The same can be said for **P8**. Since the name of the process is not considered in the calculation, it does not change the coupling value.

The **Process Coupling** metric allows for the complexity of two models united to exceed the sum of the individual models (**P9**) if the scaling is omitted. This can be shown by example, when two models are joined which share common information elements, the number of function pairs may rise beyond the sum of the pairs. If the scaling is still done, the calculated value will either decrease or remain unchanged, which appears counter-intuitive to us.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Coupling of a module (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Allen et al. 2001)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Intramodule Coupling of a module (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Allen et al. 2001)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
RFC (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Gui & Scott 2008)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
CBO (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Gui & Scott 2008)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Direct Coupling (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Chidamber & Kemerer 1994)	Y	Y	Y	Y	N	Y	Y	Y	N	N
Indirect Coupling (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Chidamber & Kemerer 1994)	Y	Y	Y	Y	N	Y	Y	Y	N	N
Total Coupling (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Chidamber & Kemerer 1994)	Y	Y	Y	Y	N	Y	Y	Y	N	N
Process Coupling (Braunnagel & Johannsen 2013; Braunnagel et al. 2014; Vanderfeesten et al. 2008c)	Y	Y	Y	Y	N	Y	Y	Y	N	N

Table 35: Measures and properties

As a consequence of the scaling, the metric as originally presented systematically violates **P10**. If two models are united, the degree of **Process Coupling** will either decrease or remain the same. Again, as a solution, one may omit the scaling in (2).

2.6.4.2 Result

The above discussion has shown that our current design of the **Process Coupling** metric cannot fulfill three of the desirable properties. They are all related to the scaling of the original design which causes a side effect that we did not anticipate. Originally, the metric was supposed to aid a practitioner in evaluating his process design, regarding dependencies among functions which result from shared resources. A lower metric value is supposed to indicate a better design regarding the resource coupling of functions which indicates a higher flexibility in the process' execution since fewer functions depend on each other due to shared information. However, due to the scaling, a lower metric value can also be achieved by e.g. merging different models or otherwise by introducing additional functions without any link to a resource. As a result, the number of coupled functions would not decrease and the actual flexibility of a process would not improve. Instead, the model would either be filled with irrelevant information or merged unnecessarily. In any case, its understandability would be degraded. To avoid this side effect, we alter the metric by omitting the scaling.

Table 11 shows the performance of the metrics in our current design regarding the desirable properties. Our adaption of the Process Coupling metric violates the desirable properties **P5**, **P9** and **P10** due to a scaling function and so do the metrics **Direct Coupling, Indirect Coupling, Total Coupling** and **Conceptual Coupling**.

The evaluation framework by *Venable et al.* (2012) distinguishes between naturalistic evaluations (e.g. action research) and formalistic evaluations (e.g. criteria based). The latter are generally less costly. Following the method of *Sonnenberg and vom Brocke* (2012a), we performed a formal (criteria based) evaluation after the design step, prior to the more costly evaluations, and identified potential design issues.

In our original research method, we had planned to evaluate this design in a laboratory experiment and a practical setting. We suspect that the design issue would have surfaced in the latter evaluation, too but would have triggered another design cycle in addition. Thus, the altered design would have required another costly evaluation, both in a laboratory and a practical setting.

Of course the early evaluation cannot guarantee that every design flaw was uncovered, and thus no further cycles are necessary. However, it will ease the identification of the causes to fail the practical evaluation, since less issues will cause less confusing interplay.

2.6.5 Discussion

The DSR method of *Sonnenberg and vom Brocke* suggests two additional evaluation activities for each phase of the general Build and Evaluation phases (Sonnenberg & Vom Brocke

2012a). Especially the two evaluations in the Build phase (Eval1 and Eval2) seem to be an interesting and novel recommendation. In this context, the aim of our paper was to investigate in how far the evaluation of the artifact during the Build phase can contribute to the quality of the research by means of reducing the number of cycles in the whole DS project. On the one hand, the conduct of additional evaluation activities during the Build phase should generally contribute to the quality of the artifact. On the other hand, additional evaluation activities are time consuming and the required effort must be adequate regarding the gains in quality.

Sonnenberg and vom Brocke suggest a first evaluation after the problem identification which is meant to ensure that a meaningful design science research problem and a meaningful statement is formulated (Sonnenberg & Vom Brocke 2012a). In our research project, Eval1 was based on a literature review. The review served not only as proof of relevance for the problem, but also for the refinement of the problem definition. We identified that coupling is especially relevant for process architectures because automatically computable coupling metrics are of great help for the design and development of process models in a process architecture.

The second evaluation activity (Eval2) serves to show that an artifact design provides the solution to the stated problem as well as to ensure the solution’s quality. The object of this evaluation is the artifact as a concept and not the finished solution. We did not evaluate the concept of our artifact in our primary investigation (see section 2.6.3.2). Therefore the emphasis in our investigation was on examining the contribution of Eval2. To perform our evaluation, we instantiated the “Assertion” pattern and presented an informed argument with all metrics as concepts and criteria in the form of desirable properties, which we found in literature. As a result, we identified four metrics which were not able to meet all desirable properties. We could show that the original design which neglects the missing properties would have misled practitioners. The metrics indicated improvements which actually degraded the models. Originally, the metrics gave e.g. a better rating to a model if it was inflated with unnecessary information and the source of harmful coupling remained unaltered. Such a model would be more difficult to read, implement or maintain and generally more difficult to use. To avoid this effect, we altered the metric by omitting the scaling. Therefore, the benefit of conducting Eval2 was twofold: we did not implement misleading metrics and thus saved time otherwise spent on needless implementation efforts. In addition we could further improve these four metrics in an early phase and eliminate their defects.

All in all, we could demonstrate with our investigation that the application of Eval2 reduced the cycle time of our research project, and we improved the quality of the artifact at an early stage during the design science project. Further, we are certain that in our case the additional evaluation was more efficient in comparison to additional evaluation and implementation cycles. However, this was much due to the easily available evaluation criteria in our project. If *Weyuker* (1988) had not documented the experiences from decades of metric development, the efforts to find applicable evaluation criteria or apply another evaluation technique would have been larger and they might possibly even have outweighed the additional implementation cycles. In summary, despite our promising results, we cannot declare the early evaluations in the Build phase to be generally reasonable for all DS projects, but we do argue that early evaluations of the concept are generally a worthwhile consideration. As we have shown in our case, they can spare DS cycles and improve the concept and thus the solution, as well. Also, as experience from Software-Engineering shows, resolving issues in an early phase of the SE cycle (e.g. during the analysis phase) is less time-consuming and cost-intensive than later e.g. during implementation. (cf. Sonnenberg & Vom Brocke 2012a).

2.6.6 Conclusion

Our paper deals with the application of the DS methodology by *Sonnenberg and vom Brocke* (2012a), resp. with the application of early evaluations in the build phase. In comparison to traditional DS approaches, this puts particular emphasis on the evaluation after each step of the DS method. The additional evaluations can be either superfluous or helpful to uncover pitfalls which otherwise enforce additional DS cycles. Thus, we compare the procedure of one of our research projects where we follow a traditional DS approach with a procedure including early evaluations in the build phase.

In our case, the additional evaluation uncovered, and also helped to mitigate, design flaws that would have forced us to repeat the laborious evaluation in the organization, had they remained undiscovered. The effort of the additional evaluation profits from the availability of evaluation criteria. We benefited from ready-to-use criteria from complexity metrics development.

Without them the evaluation would have been either less effective or more laborious. Thus, we encourage the research community to document prescriptive knowledge (cf. Sonnenberg & Vom Brocke 2012b) as support for DS projects.

As stated, the evaluation of the developed artifact is missing in many publications about DS projects (cf. Griesberger 2014). This has been discussed in literature, too (cf. Sonnenberg & Vom Brocke 2012a). Apart from the different possible explanations to this observation, this underlines the necessity to apply the evaluation patterns for the Build phase, as they provide the possibility to assert the artifact's concept and to show its usefulness and superiority.

2.7 Beitrag 7: The Effect of Coupling in EPC Models – A Laboratory Experiment

Adressierte Forschungsfrage	**Forschungsfrage 7**: Zeigen die Coupling Metriken eine verminderte oder verbesserte Verständlichkeit von Prozessmodellen an?
Erscheinungsort	25th European Conference on Information Systems, ECIS 2017, Guimarães, Portugal, 5-10 Juni, 2017 (VHB Jourqual 3: B)
Autoren	Daniel Braunnagel 100 %
In diesem Beitrag wird mit einem Laborexperiment ermittelt, dass die bessere oder schlechtere Verständlichkeit der verwendeten Prozessmodelle von Metriken angezeigt wird. Anhand von zwei praxisnahen Problemstellungen kann gezeigt werden, dass Modelle mit schlechteren Couplingwerten schwerer verständlich bzw. die Aufgaben für diese Modelle schwerer lösbar sind. Damit erfüllen die Metriken ihre ursprüngliche Zielsetzung in einer Laborumgebung. Der Beitrag nimmt die zweite Evaluation vor und nimmt damit die fünfte Stelle im DS Entwicklungszyklus Coupling Metriken ein. Die besondere Schwierigkeit des Vorhabens resultiert daraus, dass sich kognitive Prozesse nicht direkt beobachten lassen. Entsprechend der Cognitive Load Theory erhöht Coupling die Auslastung des Arbeitsgedächtnisses, bis es zu einer Überforderung und zu Verständnisproblemen kommt. Weil die Auslastung des Arbeitsgedächtnisses nicht direkt messbar ist, kann diese Theorie nur über das Auftreten von Verständnisproblemen bestätigt werden. Folglich muss aber sichergestellt werden, dass eben jene Probleme aus dem Coupling resultieren und nicht von anderen Faktoren stammen. Daher wurden zwei Messinstrumente verwendet bzw. neu entwickelt, die den Teilnehmern unterschiedliche mentale Modelle abverlangten. Weiter musste eine große Zahl externer Einflüsse, vom Wissensstand bis zur Ermüdung der Teilnehmer, berücksichtigt werden, wodurch der Versuchsaufbau deutlich komplizierter wurde.	

Tabelle 36: Bibliographische Angaben zu Beitrag 7

Business process modeling is a fundamental aspect in BPM initiatives. As a central instrument of communication and documentation, both the quality and understandability of process models are decisive. However, the means to support modelers in creating process models of high quality are still insufficient. Recently, coupling metrics have been presented as a tool to create more understandable process models. Whether they do capture the effect of coupling on model understandability has not yet been shown conclusively, though. Therefore, this work presents results from a laboratory experiment showing that the theory of coupling found in the control flow of models as well as the metrics operationalizing this theory are promising means for measuring the impact of coupling on the understandability of process models. Further, we use the empirical data to discuss the differences between the metrics and also why different metrics are necessary, and finally also discuss different aspects to consider when developing new metrics.

2.7.1 Introduction

For many enterprises, business process modelling has become an indispensable tool to pace with ever changing markets. Companies create process models as a means to support the communication and the planning of business processes for the purpose of, e.g., the development of information systems, process improvement, IT investment decisions or enterprise architecture design. (cf. Becker et al. 2010; Mendling et al. 2010) To make full use of their potential, process models need to be understood by the modeller and the model user alike. However, the subjectivity involved in both tasks, creating conceptual models and understanding them, makes this goal difficult to pursue. (cf. Gruhn & Laue 2006a; Houy et al. 2012; Becker et al. 2000)

To assist modellers in producing understandable models, practitioners and researchers alike published different approaches, e.g., best practices that work in the cases they were designed for (e.g. Sharp & McDermott 2009) or guidelines that are highly flexible because the user derives the measures for each setting (e.g. Schütte & Rotthowe 1998) Process model quality metrics are another approach. Often calculated automatically, their application is easy. And, because the user judges if a value is acceptable, they allow a flexible application. (cf. Vanderfeesten et al. 2007a)

Though an established systematization has not emerged yet, metrics that evaluate the understandability of conceptual models were developed for categories such as complexity, modularity size or cohesion. Recently, coupling has been added as new category and respective metrics were developed e.g. to improve a process's flexibility (Reijers & Vanderfeesten 2004) or to predict the number of modelling errors (cf. Vanderfeesten et al. 2008a).

While the understanding of coupling is diverse in literature—and so are the metrics—they commonly refer to the effect of dependencies in conceptual models. Vanderfeesten et al. (2007b) define coupling in process modelling as the connectedness of elements, which is a typical contributor to complexity. Flood and Carson (1993) explain the characteristics of complex conceptual systems, among which they list the number of elements and their dependencies, characteristics that are, in fact, addressed by coupling. Previous publications developed metrics for model evaluation, systematized the metrics and analysed possible use cases. Braunnagel et al. (2015) highlight three different metrics that use coupling as a means to quantify the complexity of the control flow, a possible reason why process models are difficult to understand and use.

The aforementioned publications discuss coupling in theory. Empirical work on coupling metrics and process model understandability document mixed results (Mendling et al. 2006; Mendling et al. 2007; Vanderfeesten et al. 2008a). One of the metrics shows a promising relationship with model understandability but the authors also report a limited statistical significance (Vanderfeesten et al. 2008a). A second metric does not show the expected relationship with process model understandability at all (Mendling et al. 2007). Thus it is uncertain whether coupling in general and the metrics in particular can be a helpful tool to support the design of more understandable process models. This work contributes the results of a laboratory experiment which is designed to isolate the effect of coupling in conceptual process models that use the EPC notation and answer the following research questions:

- A. Does coupling impact process model understandability?
- B. Do coupling metrics indicate the understandability of process models?
- C. Which are the differences between the metrics?

The remainder of this work is organized as follows. This introduction is followed by a section on the "Basics" of coupling for EPC models. "Related Work" presents previous results. "Method" explains how the results of this study, subsequently presented in "Results", were achieved. Before the work closes with a summary and an outlook, a “Discussion” of the results is presented.

2.7.2 Basics

While the control flow constructs of established process modelling languages are comparable in general, this study’s metrics focus event-driven process chains (EPC) in particular. EPC models are well accepted by users and supported by tools, e.g. the ARIS (Architecture of Integrated Information Systems) framework (Mendling 2008). Relevant symbol types of the control flow are the event, which represents a point in time or a condition, the function, which represents a work-related activity, the connector types AND, OR, XOR, which denote alternatives in the control flow and the sequence. (cf. Mendling 2008) The focus on one modelling language only is necessary to mitigate the influence of different process model notations (see Sec. Instruments).

In previous literature, Braunnagel et al. (2015) present a wide range of different understandings of coupling and metrics, which operationalize coupling in different ways and pursue different goals. The systematization groups those metrics that are common as to their application and as to what they indicate about the process model. One of those groups is meant to evaluate the understandability of single process models under the aspect of how coupling takes place in the control flow. As such, the metrics focus on coupling as the product of the control flow within a model instead of dependencies among models. For these two reasons, this group of metrics promises the best potential to provide useful tools for modellers, which is why this study focuses upon them.

The metrics of this group share the idea that, in a nutshell, an increasing number of control flow states goes with an increasing cognitive load of understanding the model, which, in turn, increases the likelihood of misunderstanding the model. The theoretical basis for this idea can be found in the cognitive load theory (CLT) (Sweller 1988), a frequently used theory on cognition in process model understandability research (Houy et al. 2014). In the domain of conceptual modelling, Zugal (2013) introduces the CLT, which can be summarized as follows.

The CLT describes by means of the three cognitive processes Search, Recognition and Inference how users read and understand conceptual process models. During Search, a user locates the elements that seem necessary for the task at hand. Here, the layout and visual properties of the model have a major impact. Another cognitive process is Recognition. Recognition focuses relationships between elements to recognize patterns in the model, e.g., a sequence or a branching. This cognitive process is strongly influenced by previous knowledge, and also by the presentation of the model, because information need to be presented in a highly explicit manner to be recognized by a user. A third process is Inference, which solves problems that cannot be read from the model easily. (Zugal 2013)

Central to the cognitive processes is the interplay of the short term and long term memory. The short term memory is capable of storing and processing information to solve problems, but its capacity is strictly limited to about seven elements. Further, stored elements are lost after about 30 seconds, if not rehearsed. The number of elements stored in the short term memory at any given time is referred to as cognitive load. (Zugal 2013) The idea of the metrics is that if the cognitive load, which is necessary to solve a problem, exceeds the user's cognitive capacity, the user will fail to understand the model correctly. And, further, the number of control flow states in a process model increases the cognitive load. (Vanderfeesten et al. 2007a)

The number of control flow states results from the type of connector in the context of the three metrics. An AND connector activates all outgoing paths with no alternatives, thus having exactly one possible succeeding control flow state. An XOR connector activates alternatively one of all outgoing paths, thus the number of possible succeeding control flow states equals the number of outgoing paths. Last, an OR connector can activate any combination of outgoing paths, thus the number of possible succeeding control flow states equals the number of the permutations of the outgoing arcs. Since a model reader, so it is assumed, needs to mentally store and process all of the alternative states in parallel while reading a model, the increasing number of states raises the model's complexity which leads to a higher chance of making mistakes. Thus such, a model is potentially less suitable as a communication and planning tool the higher the number of states. (cf. Cardoso 2005; Vanderfeesten et al. 2007b; Vanderfeesten et al. 2007a; Vanderfeesten et al. 2008a)

The three metrics of the abovementioned group, the Control Flow Complexity (CFC) metric (Cardoso 2005), the Weighted Coupling (WC) metric (Vanderfeesten et al. 2007b) and the

Cross Connectivity (CC) metric Specification of the CC metric (Vanderfeesten et al. 2008a), operationalize the theory of coupling from control flow states in different ways.

$$CFC(P)=\sum_{i\in\{XOR\text{-split of }P\}} CFC_{XOR\text{-split}}(i)+\sum_{j\in\{OR\text{-splits of }P\}} CFC_{OR\text{-split}}(j)+\sum_{k\in\{AND\text{-splits of }P\}} CFC_{AND\text{-split}}(k)$$

$$CFC_{XOR\text{-split}}(i)=\#\,outgoing\,branches(i)$$

$$CFC_{OR\text{-split}}(j)=2^{\#\,outgoing\ branches(j)}-1$$

$$CFC_{AND\text{-split}}(k)=1$$

Legend: P: Process Model, i: XOR splits, j: OR splits, k: AND splits

Figure 25: Specification of the CFC metric (Cardoso 2005)

The Control Flow Complexity (CFC) metric calculates the sum of the possible states resulting from connectors in a model (see Figure 25). The metric defines the number of states as the number of alternative combinations with which outgoing branches can be activated. For an AND split, this is exactly one state. For an XOR split, this is the number of outgoing branches. For an OR split, this is the combinations of all outgoing paths. This design results in a higher value indicating a more complex model. (Cardoso 2005)

$$CP=\frac{\sum_{t_1,t_2\in T} connected(t_1,t_2)}{|T|*(|T|-1)}$$

$$connected(t_1,t_2)=\begin{cases} 1, & if\,(t_1\rightarrow t_2)\wedge(t_1\neq t_2)\\ 1, & if\,(t_1\rightarrow AND\rightarrow t_2)\wedge(t_1\neq t_2)\\ \frac{1}{(2^m-1)*(2^n-1)}+\frac{(2^m-1)*(2^n-1)-1}{(2^m-1)*(2^n-1)}*\frac{1}{m*n}, & if\,(t_1\rightarrow OR\rightarrow t_2)\wedge(t_1\neq t_2)\\ \frac{1}{m*n}, & if\,(t_1\rightarrow XOR\rightarrow t_2)\\ 0, & if\,(t_1=t_2)\end{cases}$$

Legend: n: #outgoing branches, m: #incoming branches, t1, t2: nodes in T, T: Set of all nodes

Figure 26: Specification of the WC metric (Vanderfeesten et al. 2007b)

The design of the Weighted Coupling (WC) metric follows the same idea, but with the following differences. First, the metric calculates the likelihood of a node preceding a connector and a node succeeding a connector being activated in the same instance. Second, the metric scales the result by the number of nodes in the model (see Figure 26). A lower value indicates a more complex model. (Vanderfeesten et al. 2007a)

$$w(n)=\begin{cases} 1 & ,if\ n\in C\wedge n\ is\ of\ type\ AND \\ \frac{1}{d} & ,if\ n\in C\wedge n\ is\ of\ type\ XOR \\ \frac{1}{2^d-1}+\frac{2^d-2}{2^d-1}*\frac{1}{d} & if\ n\in C\wedge C\ is\ of\ type\ OR \\ 1 & ,if\ n\in T_{Tasks} \end{cases}$$

$$W(a)=w(src(a))*w(dest(a))$$

$$v(p)=W(a_1)*W(a_2)*...*W(a_n)$$

$$v(n_{1,}n_2)=max_{p\in P_{n1},n2}v(p)$$

$$CC=\frac{\sum_{n_1,n_2\in N} V(n_1,n_2)}{|N|*(|N|-1)}$$

Legend: d=#incoming branches + #outgoing branches, n: Node

Figure 27: Specification of the CC metric (Vanderfeesten et al. 2008a)

The last of the three metrics is the Cross Connectivity (CC) metric which extends the approach of the CP metric for any two nodes in a model lying on the same path from the start to the end of the control flow, and aggregates all pairs of nodes (see Figure 27). (Vanderfeesten et al. 2008a)

In this study, the number of control flow states, operationalized by the three metrics, is the independent variable. The study's dependent variable is model understandability, which Houy et al. define as "[…] related to the ease of use, respectively the effort for reading and correctly interpreting a conceptual model, which is a cognitive process of assigning meaning to the different parts of a conceptual model" (Houy et al. 2012, S. 66). The metrics and their underlying theory operationalize this definition as follows.

First, the cognitive process is described by means of the cognitive load theory and its three cognitive processes: Search, Recognition and Inference. The effort of reading and correctly interpreting is operationalized by the cognitive load itself. The actual number of control flow states or the different variants, respectively, are not explicitly visible in the model, especially for a less experienced model user. Thus, one can suspect that for answering questions regarding the control flow relationship between elements, an unexperienced user makes heavy use of the Inference process. Therefore, the experiment needs to control the influence of previous knowledge by using an abstract domain (see "Instruments"), which means that "understandability", in the current context, refers to syntactical knowledge instead of domain knowledge. For such designs, Gemino and Wand (2004) propose the term "comprehension" instead of "understanding", which, however, is considered a synonym by Houy et al. (2012). Last, in the definition understanding happens for parts of the model. The metrics of this study focus an aspect of the conceptual model, namely the control flow and its influence on the understandability, which impact the experiment's design.

2.7.3 Related Work

The CFC and the CC metrics have been addressed in empirical work before. A study on different indicators for the understandability of process models that have an impact during model creation, including the CFC metric, hypothesizes that if models are very complex, formal errors are more likely to occur during model creation (Mendling et al. 2006). The authors explain that the expected relationship between the CFC values and the number of errors in the model is not supported by their results. Unfortunately, the authors do not discuss possible reasons for this outcome in detail. It remains unclear whether, e.g., the formal errors do not reflect the models' understandability, or it is that the metrics do not relate to model creation, or whether the metric is generally unsuitable to predict modelling errors. The study has no particular focus on coupling and understanding during model reading. (Mendling et al. 2006)

An evaluation of the CC metric reuses the previous approach (cf. Mendling et al. 2006) and checks the correlation of the metric's value with formal errors. (Vanderfeesten et al. 2008a)

They find a strong, highly significant, correlation between the metric's indication and the actual occurrence of automatically discovered soundness violations. (Vanderfeesten et al. 2008a)

Another study analyses factors that influence the understanding while reading a process model, one of which is the CFC metric. (Mendling et al. 2007) This time, students participate in a laboratory setting and answer questions regarding the control flow. The results, however, cannot support the hypothesis that an increase of the CFC metric reduces the model users' understanding. The reasons for this result are discussed briefly. (Mendling et al. 2007)

Further research follows this second approach as well to evaluate the CC metric, finding support for their hypothesis that the CC metric indicates a reduced understanding of process models. (Vanderfeesten et al. 2008a)

The CFC metric is the subject of an evaluation where participants rate the control flow complexity (Cardoso 2006). It shows that the participants perceive models with a higher CFC value as more complex but not whether this perception has an impact on the understandability of the process model.

All three of the metrics count the possible control flow states in a process model as a means of indicating process model understandability. While their computations differ in detail, they share theory that the cognitive load increases with the number of alternative flows, and with it the possibility of misunderstanding. This makes the differences in the evaluation interesting. For the CFC metric, the publications do not show support during either model creation or model use. The CC metric is shown to be helpful during both model creation and use. The WC metric has not yet been evaluated in a comparable setting. Unfortunately, the published evaluations were not designed with a focus on coupling which is the focus of this work.

2.7.4 Method

To answer the research questions, this work draws on data from a laboratory experiment. The following subsections explains the preparation procedure (Figure 28) that implements the methodical suggestions of (Wohlin et al. 2012) and (Shadish et al. 2002).

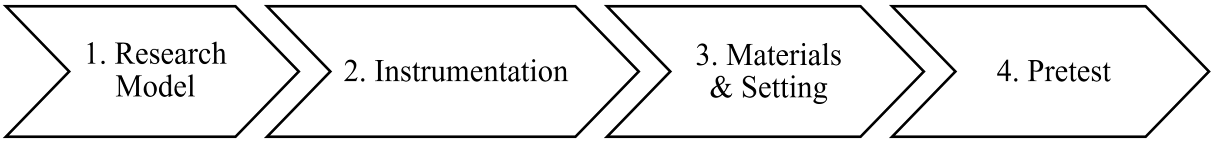


Figure 28: Preparation procedure and outline of the section

2.7.4.1 Research Model

The metrics in question share a common foundation, namely that increasing coupling, here defined as the number of control flow states, increases a reader's cognitive load which, in return, manifests in a higher chance of misunderstandings. The first pair of hypotheses tests this theory in accordance with research question A:

- H_{a0}: A higher number of control flow states does not decrease the understandability of the process models.
- H_{a1}: A higher number of control flow states does decrease the understandability of the process models.

Research question B asks whether the coupling metrics' measurement does reflect the impact of understandability on process models. Thus, this question requires to analyse the relationship of each metric's measurement on the process model understandability independently. The resulting research model and hypotheses are:

- H_{b0}: A higher value of the CFC metric does not decrease the understandability of the process models.
- H_{b1}: A higher value of the CFC metric does decrease the understandability of the process models.

- H_{c0}: A lower value of the CC metric does not decrease the understandability of the process models.
- H_{c1}: A lower value of the CC metric does decrease the understandability of the process models.

- H_{d0}: A higher value of the WC metric does not decrease the understandability of the process models.
- H_{d1}: A higher value of the WC metric does decrease the understandability of the process models.

Research question C asks about the differences of the metrics. This study analyses the differences of the metrics using the empirical data of this experiment. For this research question, we do not state testable hypotheses, instead we discuss the differences of the metrics performance with respect to the instruments.

2.7.4.2 Instruments

This experiment operationalizes two concepts, coupling as an independent variable and understandability as a dependent variable, and a set of control variables.

The first concept, coupling, is operationalized by the number of control flow states on three treatment levels. The first model has 33 possible control flow states, the second model has 65 possible states, and the third model has 110 possible states[6]. As the design of the models presents a major influence during the Search and Recognition process, it addresses several influencing factors which can be found in previous works, e.g., the model's size (i.e., the number of nodes), its notation and structural aspects, as for instance its density. The influence of the notation was controlled by keeping it constant for all models. To mitigate the influence of model size and structure, each treatment level is constructed from the same process model, which we modify very carefully. First, for each level, we replace the connectors to achieve a different number of control flow states. Second, we rearrange the branches of each model and change their position on the modelling canvas so that the similarity of the models is no longer visible, while, at the same time, the structure of the models does not change. To further obfuscate any similarity, we change the nodes' labels for each model. And, while the tasks do refer to nodes in about the same area (i.e., nodes within the same branch), they never refer to nodes at the same position.

6 Materials and further information are available online: http://tinyurl.com/z3r8wa5
Ein Abdruck der verlinkten Seite findet sich in Anhang 2.

For the second concept, understandability, this study uses two instruments for measuring understandability to avoid a mono-method bias (cf. Shadish et al. 2002). As coupling refers to the dependency among process model elements, so do the instruments.

The first instrument to measure understandability is a set of eight questions on the relationship between nodes (cf. Mendling et al. 2007). The questions ask for instance "If function DL has been executed, can function AA be executed as well in the same run?". For an answer, the model user has to consider possible control flows between the functions. If a higher number of alternative states decreases the understandability, we expect that this kind of question will be more difficult to answer. To mitigate their influence, the number of nodes between those nodes addressed by the questions remains constant over all treatment levels. Further, the questions address long and short control flows equally. Originally, the instrument counts the number of correct answers (called understanding effectiveness). As a further control, the time required for performing this instrument accounts for the fact that using more time to answer the questions could compensate a decrease in understandability. In practical settings, this kind of questions appear as well. When a model user wants to know whether, e.g., a particular checking function is performed before every mode of delivery, or why processes terminate unsuccessfully, s/he has to analyse the dependency between the check and delivery functions or the different events leading to a termination, in a similar way. Thus, in summary, the first instrument assesses the understanding effectiveness with regard to answering questions about the dependency of two or more nodes in a process model.

The first instrument is the basis for previous empirical work (see "Related Work"). The second instrument asks the subjects to mark the longest instance in the model. There, to analyse which branches of a connector lead to the path with the highest number of functions until the end of the process model, it is not sufficient to only consider the directly following branch of a connector, since a directly following long branch may deactivate another even longer branch after another connector. Instead, the subject is forced to consider all possible combinations of control flow states. In a practical setting, users perform a similar task e.g., during performance analysis. To discover the costliest, fastest or least efficient process instance, a user has to follow all different combinations of process branches, depending on the cost and time requirement of functions. Thus, instrument two assesses the understanding effectiveness with regard to finding the longest path in a model.

For both these instruments, it is possible that subjects do not understand the questions or tasks. Thus, 12 questions on eEPC modelling assess whether a subject actually has the necessary background knowledge. Also, a short example demonstrates the tasks. Then, prior to the three treatments, the subjects answer a training treatment for verification and to familiarize themselves with the tasks without influencing the results of the experiment. Last, at the end of the experiment, we ask the subjects for feedback on the tasks and models. If either of these checks indicate a threat to validity, we remove the subject from the analysis.

Another threat to validity is the fatigue of the students. First, at the end of the three treatments, the subjects perform an additional run for verification which is significantly smaller than the previous ones and not used for the analysis. Outliers indicate a drop in either concentration or motivation and can thus be removed. Second, at the end of the experiment, subjects report if they experience a decrease in concentration because of which we remove the subjects. Last, the experiment follows an in-subject design where every subject solves the tasks for every treatment. The order of the treatments is randomized, thus an equal number of subjects perform the treatments in the order 1-2-3, 1-3-2, 2-1-3, 2-3-1, 3-1-2- and 3-2-1.

Another threat to validity is the subjects' domain knowledge If subjects are familiar with the process domain, they might recognize the answer to the questions based on their background knowledge without inferring the control flow states. Thus, to counteract this threat, we follow the idea of (Mendling et al. 2007) and use an abstract labelling for our nodes. The nodes are labelled with arbitrary combinations of letters, which supports our analysis in two ways. First, since we analyse coupling in isolation, abstract labels avoid the impact of domain knowledge. Second, the instruments work better, as questions on the relationship between abstractly labelled nodes do not require interpretation. Last, coupling the current context is understood as a property of the syntax, foremost.

2.7.4.3 Materials, Setting and Pre-test

All subjects of the experiment received a printout starting with an introduction, explaining in short the general goal and the structure of the experiment, followed by instructions for the tasks, as well as an example as to how the material should be used. The section background information asks about the modelling experience of the subject and the syntax of eEPC models, to verify a subject's suitability. The background information is followed by a verification run to examine whether a subject understands how to perform the experiment. Afterwards the three treatments follow in mixed order, resulting in six different cases. A consistent sample size for every case compensates possible repercussions of learning or fatigue.

The experiment was conducted with university students in two settings. First, 47 bachelor students from a course on enterprise modelling and, second, 16 master students from a course on workflow modelling participated. A tutor from each of the courses ensured that the students had the necessary knowledge and granted extra credits on the course's exam as a motivation to participate in the experiment, which took place in the last week of the semester, shortly before the exams.

In total, 63 students participated in the experiment and returned their materials 14 of which were either incomplete or failed the instruments' verifications. Thus, 49 returns, with three treatments each, remained supporting our analysis with N=147 treatments in total.

Prior to the experiment, the materials were pretested by six students who had attended the courses before. They provided feedback on the instruments and helped timing the tasks for a 90 minutes course session.

2.7.5 Results

Research question A asks about the impact of coupling, i.e., the number of control flow states on process model understandability. The respective null hypothesis is "A higher number of control flow states does not decrease the understandability of the process models" which is addressed by two instruments.

The first instrument, which asks a subject to reproduce the dependency of two nodes, and the second instrument, where the subject has to consider the states following from each connector to identify the longest possible path, show significantly more wrong answers if the number of states increases. The bounds of the confidence intervals have the same sign, a significance level higher than 99% is reached for the first instrument. The bounds are mostly positive and a still high significance level of 74% is reached for instrument 2. (see Table 37) Thus, we reject the null hypotheses and conclude that more control flow states render questions on the dependency between functions and on path tracing more difficult to answer. The models' R-value are 0.536 and 0.466.

This result is interesting. First, we could reproduce the effect of coupling independent of other studies. Second, it seems the effect is not limited to questions that focus the dependency between two nodes but can be found in further cognitive processes.

Research question B. From the previous results, we follow that the number of control flow states does decrease the understandability of process models. Question B asks whether this impact is shown by all three of the coupling metrics (see Table 37)

	Model	Unstand. Coeff.		Sig.	95.0 % Conf. Int. for B	
		B	Std. Error		Lower B.	Upper B.
Instr. 1	(Constant)	.720	.031	.000	.658	.781
	#States	.002	.000	.000	.003	.001
Instr. 2.	(Constant)	.031	.047	.000	.937	1.125
	#States	.004	.001	.000	.005	.003

Table 37: Hypothesis Tests for H_a

All three metrics show a significant relationship with the subjects' performance in both instruments. Their bounds do not change their sign and the significance levels are above 99%. Thus, we reject the null hypothesis for the three metrics.

Research question C. Since all three of the metrics indicate a significant change in understandability, we are left to discuss the nature of the differences as well as the impact they have. The correlations for all three metrics are between 0.70 and 0.99, confirming the close similarity of the metrics.

		Model	Unstand. Coeff.		Sig.	95.0% Conf. Int. for B.	
			B	Std. Error		Lower B	Upper B
Instrument 1	H_b	(Constant)	.720	.031	.000	.658	.781
		CFC Metric	-.002	.000	.000	-.003	-.001
	H_c	(Constant)	-.115	.069	.099	-.251	.022

		WC Metric	122.827	12.230	.000	98.538	147.117
	H_d	(Constant)	.488	.013	.000	.462	.515
		CC Metric	1.271	.131	.000	1.010	1.532
Instrument 2	H_b	(Constant)	1.031	.047	.000	.937	1.125
		CFC Metric	-.004	.001	.000	-.005	-.003
	H_c	(Constant)	.009	.141	.950	-.271	.288
		WC Metric	134.200	25.097	.000	84.427	183.973
	H_d	(Constant)	.660	.025	.000	.610	.711
		CC Metric	1.532	.255	.000	1.026	2.038

Table 38: Hypothesis Tests for H_b, H_c, H_d

To analyse the impact of the differences in regard to process model understandability, we calculate the different R-Values for the metrics and the instruments (see Table 38). It becomes evident that the WC metric explains most of the variance in the understandability effectiveness as measured by instrument 1 with an R-value of 0.723, whereas the CFC metric explains least of the variance (R-value: 0.491) and the CC metric resides in the middle (R-value: 0.710). For instrument 2, it is the other way round. WC explains the least (R-value: 0.466), CC resides in the middle (R-value: 0.509) and CFC explains most of the variance (R-value: 0.536). To explain the reasons for this result, one needs to discuss the designs of both the metrics and the instruments.[7]

R-Values	Instrument 1	Instrument 2
WC Metric	0.723	0.466
CC Metric	0.710	0.509
CFC Metric	0.491	0.536

Table 39: R-Values for the Coupling Metrics and Instruments

2.7.6 Discussion

To answer the questions in instrument 1, the subject first searches for a path in the model that connects both nodes. Second, the subject analyses each connector along this path to decide whether both functions can, must, or must not be activated in the same instance. We assume that this decision becomes more difficult as the connectors introduce more control flow states

7 Further statistical information is available online: http://tinyurl.com/z3r8wa5
Ein Abdruck der verlinkten Seite findet sich in Anhang 2.

because a subject has to memorize and process all states and their combinations which increases the cognitive load. Overall, the previous results support this assumption, however, the metrics' R-values differ.

While the WC's and CC's R-values are essentially the same for instrument 1, the CFC metric explains less of the first instrument's variance. The CFC metric's design differs from that of the other metrics in two aspects. First, the CFC metric only calculates the number of control flow states that result from outgoing branches from split connectors. In contrast, the WC and CC metrics' designs do incorporate join connectors as well. To answer the questions of instrument 1, a subject has to analyse all connectors. Split connectors decide if the branch towards a node can be activated, but the join connectors' pre-conditions decide if a path can be followed as well. Thus, the cognitive load for this particular question is measured more precisely by the WC and CC metric, as can be seen in the R-values.

Further, the CFC metric, in contrast to the WC and CC metrics, rate OR connectors with the combination of outgoing branches. Thus, the CFC metric punishes the appearance of the OR connector of all connectors the most strongly. Instead, the WC and CC metrics use the number of states to calculate the likelihood that a particular branch gets activated. For an OR connector, this likelihood is higher than after an XOR connector, because a particular branch can become activated in an exclusive setting as well as in many combinations. Thus, the OR connector is punished less strongly. This approach comes closer to how subjects address the questions of instrument 1. Since the subjects are only interested in the one branch that leads to the node in question, they do not consider and process all possible states with the same amount of attention. The subjects only need to know whether combinations exist in which the branch can be activated, which is more in line with the WC/CC metrics' approaches.

Instrument 2 makes subjects search and mark the instance in which the most functions are activated. To do so, the subjects decide at each connector which combinations of outgoing paths the connector allows and which combination activates the most functions in the directly following branch. Even more, they have to consider all subsequent connectors for this decision, because a longer path in direct succession might deactivate subsequent connectors that would allow for combinations with more functions in sum. Thus, the problem cannot be solved locally and it is more relevant to consider the total number of control flow states than the likelihood of branches in isolation, which would explain why, in this setting, for this instrument, the R-value of the CFC metric exceeds the values for the WC and CC metrics.

The previous argument does not explain the different values of the WC and the CC metrics for instrument 2, since they both refer to the likelihood of the arcs being activated in the same way. They do, however, differ in their treatment of paths. The WC metric calculates the likelihood for any two nodes in a model being activated in the same instance, if, and only if, they are connected with exactly one connector in-between. The metric does not consider pairs of functions that are connected via a path with functions in-between like the CC metric does. This seems to better explain the results of instrument 2, where subjects have to analyse paths from the beginning to the end of a model.

The metrics differ in one more aspect. The CFC metric's value has no upper bound, while the WC and CC metrics' upper bound is one, because they are both scaled by the number of nodes. The scaling follows the idea that a larger model should allow for more complexity. And, without bounds, it is difficult to judge whether a value is high or not. This limitation has special relevance when reference values for a good design are missing. On the other hand, scaling can introduce counterintuitive behaviour of the metrics. If modellers attempt to improve a model's design using these metrics, they can do so by blindly introducing additional functions thus decreasing its understandability, nonetheless improve the metric's values. Another way would be to merge or concatenate the model with models that have a low metric value. Both actions would improve the metric's values and at the same time counteract known means to improve the understandability of process models.

The above results show that situations exist where coupling decreases the understandability in the laboratory, which leads to discussing its impact in practical settings.

2.7.7 Implications

First, the practical relevance of coupling metrics depends on the means that are available to change their values. Considering, that users create process models with the intention of transporting information about a business process, connectors are necessary, and they result from the process' structure, not from the modellers free choice. An option is to replace OR connectors with a combination of XOR and AND connectors. This would clarify which combinations of outgoing branches of an OR connector are permissible, possibly reducing its number. However, the resulting modelling construct would be very confusing if many combinations were permissible. A third option is process model decomposition, which is already well researched and known to improve the understandability of process models (Johannsen et al. 2014a). Decomposition, in this context, means to model on different levels of detail. Thus, complex structures of a process with a high degree of coupling can be modelled in detail on one level, whereas higher levels aggregate these details into a single function. As a benefit, the same process may be shown with different degrees of coupling and users may choose between detailed but complex and simplified presentations, depending on their needs. This supports the idea that more experienced users are capable of dealing with more model complexity than inexperienced users. (cf. Milani et al. 2016)

The preceding discussion shows that the effect of coupling and the quality of its measurement are closely tied to specific scenarios of using the model. For instrument 1, which asks about the relationship of functions within one instance, a practical case is documented by (Falk et al. 2013), who report findings from a business process improvement project. That project deals with the improvement of a student matriculation process. When the project team analyses which functions actually share logical dependencies, they find that the matriculation process has less dependencies than initially assumed. Thus, the team splits up the process into several, independent sub-processes that perform in a less error-prone fashion and with a higher ability to adapt to changing requirements. (cf. Falk et al. 2013) Instrument 2 makes subjects search the instance of a process that activates the most functions. Essentially, subjects produce all different instances and compare them. In a practical setting, model users have to do the same, when they analyse the process's performance, e.g., by finding the instance with the longest or shortest cycle time, or the highest processing costs. Such a case can be found in (Fill & Johannsen 2016), where the authors describe the process improvement of the "end-of-terms" process at an automotive bank where long cycle times lead to customer complaints. In their case, only particular process variants show the problem of long cycle times, and, once identified, the project team discusses options to improve the performance of these variants. However, process models are used in more scenarios than just these two.

While metrics are tightly bound to particular scenarios for which a process model is used, none of the three metrics in this study is generally better or worse. Instead, they all suit—

more or less—a scenario, as discussed in section 2.7.5. Many metrics are designed to assess process model understandability in general, which, as can also be seen in this work's results, is not a homogeneous concept.

It is for this dependency of metrics and the use case that the ideal coupling metric for all cases will be difficult to find. Metrics assess the quality of process models as a result of the users' cognitive process working with them. This process is different for different use cases, which affects the design of the metrics.

Thus, in the future, we need to analyse the different use cases for process models from a cognitive perspective and coupling to develop the respective metrics and reference values. For example, when a process is modelled in order to support the development of information systems or to support IT investment decisions, the metric needs to focus the respective model elements, for instance data in- and output or interfaces.

The two previously presented instruments made a start with two cases.

2.7.8 Summary, limitations and outlook

This paper investigates the impact of coupling on process model understandability. In a laboratory experiment, 49 students answer questions and perform tasks for two instruments and three levels of treatment. The first hypothesis tests the relationship between the number of control flow states in a process model and the process model understandability. The results show a significant impact, a higher number of control flow states increases the likelihood of misunderstandings, according to both instruments. The second hypothesis tests the three existing coupling metrics, which operationalize the number of control flow states as an indicator, the CFC metric (Cardoso 2005), the WC metric (Vanderfeesten et al. 2007b) and the CC metric (Vanderfeesten et al. 2008a). Each of the three metrics indicate an incline or a decrease in model understandability, which confirms their design empirically. The third research question addresses the differences of the three metrics. By means of the empirical results and a theoretical discussion of the metrics' design, we show which impact the differences have and that the metrics are tightly bound to the cognitive processes of use cases. We also show that the different purposes of process modelling, e.g., performance analysis or systems planning, require different metrics. Thus, we present aspects that need to be considered when designing and developing new metrics.

The benefits of this research are twofold. Scientifically, we were able to confirm that an increasing number of control flow states increases the likelihood of misunderstandings, confirm the design of three metrics operationalizing this theory, verify an established instrument for measuring understandability, and also present a new instrument. In addition, we discuss aspects that aid researchers in developing new metrics. Practitioners benefit from the results as these underline the significance of coupling measurement to create process models that suit their respective purpose.

The experiment demonstrates coupling in a laboratory. While the two instruments imitate the cognitive processes of two practical settings and show an impact, it is difficult to tell how much of that effect would be left outside the laboratory. Thus, we need to evaluate the metrics in a practical setting as well. Still, the laboratory experiment shows that coupling is a promising topic for a practical evaluation.

In future works, coupling will be evaluated in practical settings. The metrics need to be implemented in the practitioners' modelling environment and combined with guidelines that help to improve process models. Combined, this toolset can be installed in a process modelling project where its applicability may be evaluated.

3 Schlussbetrachtung

3.1 Zusammenfassung und Fazit

Die Zielsetzung dieser Arbeit ist die Entwicklung von Artefakten zur Messung der Prozessmodellverständlichkeit. Im Schwerpunkt „Good Decomposition“ wurden hierfür fünf Conditions evaluiert und zwölf Metriken entwickelt. Im Bereich „Coupling“ wurden hierfür neun Metriken entwickelt und jene drei der Kontrollflusskomplexität evaluiert. Insbesondere aber wurden für jeden Schritt des Entwicklungszyklusses von den verschiedenen Theorien bis zu den fertigen Metriken Methoden aufgezeigt und exemplarisch angewendet. Diese Methoden können herangezogen werden, um die Entwicklung weiterer Metriken zu erleichtern.

Die Zielsetzung wird mit einem Design Science Vorgehen verfolgt, anhand dessen sich auch die Forschungsfragen, Beiträge und jeweiligen Ergebnisse gliedern und wie folgt zusammenfassen lassen.

- **Forschungsfrage 1**: Zu welchem Grad setzen Unternehmen BPM Maßnahmen ein und welche Faktoren motivieren oder behindern den Einsatz von BPM in Unternehmen?

In dem Beitrag wurde der State of the Art für den BPM-Einsatz in bayerischen Unternehmen empirisch erhoben. Hierfür wurde auf eine Methodenkombination zurückgegriffen, die aus 10 Fallstudien und einer Umfrage mit 114 Teilnehmern besteht. Für die Entwicklung des Messinstruments wurden Maßnahmen zur BPM-Maturity Messung adaptiert, die sich in die Kategorien „Strategie“, „Zielerfüllung“ und „Zuverlässigkeit“, „Dokumentation“, „Capabilities“, „Leistungsmessung“ und „Redesign“ unterteilen lassen. Die Erhebung zeigt Verbesserungspotentiale auf, sowie die Gründe warum diese bisher nicht genutzt werden. So schätzen viele Unternehmen ihre Prozesse im alltäglichen Betrieb als zielführend und zuverlässig ein, während dennoch eine überraschend hohe Zahl an Unternehmen von nicht-zielführenden Prozessen (7 %) oder sehr unzuverlässigen Prozessen (10,5 %) berichtet.

Zur strategischen Positionierung des BPM wurde bspw. ermittelt, dass ca. ein Drittel der untersuchten Unternehmen BPM für die strategische Entwicklung sehr wichtig einschätzt. Dagegen ordnet eine ähnlich hohe Zahl von Unternehmen dem BPM bei der strategischen Entwicklung eine untergeordnete Relevanz zu. Bei den verfolgten Zielsetzungen des BPM-Einsatzes dominieren die Standardisierung, Produktivität und die Qualitätssteigerung. Andere Ziele wie bspw. das In- und Outsourcing werden kaum verfolgt.

Auf dieser Basis wurden Vorschläge für Forschung und Praxis entwickelt, um die Nutzung der Potentiale zu erleichtern. Dies wiederum soll die Konkurrenzfähigkeit der Unternehmen steigern.

Im Kontext der Geschäftsprozessmodellierung konnten ebenfalls interessante Ergebnisse ermittelt werden. So erklären mehr als zwei Drittel der Unternehmen, ihre Prozessdokumentation mehr als nur sporadisch zu aktualisieren, was die hohe Relevanz der Geschäftsprozessmodellierung für Unternehmen unterstreicht. Weitergehende Maßnahmen der Prozessmodellierung, wie bspw. eine Prozesslandkarte, werden dagegen von weniger Unternehmen verwendet.

Insgesamt zeigt sich, dass der Verbreitungsgrad von BPM-Maßnahmen in den Unternehmen polarisiert. Während viele Unternehmen BPM-Maßnahmen konsequent und umfangreich einsetzen, finden sich sehr viele Unternehmen, die den BPM-Einsatz nahezu gänzlich verweigern und grundlegendste Maßnahmen nicht umsetzen.

An Gründen lässt sich insbesondere anführen, dass der Aufwand für BPM gescheut wird bzw. der Nutzen grundlegender Maßnahmen wie der Prozessmodellierung für darauf aufbauende Maßnahmen wie bspw. der Definition von Performance-Indikatoren nicht klar wird. Durchgängige BPM-Methoden finden ebenfalls keinen Anklang in den Unternehmen. Dies hat die inkonsistente, isolierte Umsetzung von BPM-Maßnahmen zur Folge.

Darauf bauen die entwickelten Vorschläge für Forschung und Praxis auf. Lösungen können bspw. so aussehen, dass einfachere, zielgruppenspezifische BPM-Methoden entwickelt werden, oder aber das Bewusstsein für die konsistente Umsetzung der Maßnahmen in den Unternehmen geschärft wird. Ein anderer Vorschlag ist, schlicht ausreichend Ressourcen für die Umsetzung von BPM-Maßnahmen zu schaffen.

Letztlich zeigt der Beitrag auf, dass Forschung im Bereich BPM weiterhin notwendig ist und dass dies auch für den Teilbereich „Geschäftsprozessmodellierung“ zutrifft.

- **Forschungsfrage 2**: Haben Verstöße gegen die Good Decomposition Conditions nach (Johannsen & Leist 2012b) einen Einfluss auf die Verständlichkeit von Prozessmodellen?

Die zweite Forschungsfrage behandelt die Evaluation der Good Decomposition Conditions nach (Johannsen & Leist 2012b, S. 283). Diese operationalisieren das Bunge-Wand-Weber Good Decomposition Modell (vgl. Wand & Weber 1989a) für eEPK Modelle in fünf Conditions. Es wurde in einem Laborexperiment die Annahme getestet, dass ein Einhalten der Good Decomposition Conditions einen positiven Einfluss auf die Verständlichkeit von Geschäftsprozessmodellen habe, was von den Resultaten letztlich unterstützt wird. Damit erscheinen die Good Decomposition Conditions als eine geeignete Grundlage für die Entwicklung weiterer Artefakte, aber womöglich auch als Mittel für die Messung der Verständlichkeit von Geschäftsprozessmodellen im praktischen Einsatz.

Besonders erwähnenswert ist der starke Einfluss auf die wahrgenommene Verständlichkeit. Hier zeigen die Ergebnisse, dass Schwächen bezüglich der Prozessmodellverständlichkeit nicht zwangsläufig zu mehr Fehlern in der Interpretation der Modelle führen, weil sie vom Modellnutzer kompensiert werden können. Diese Kompensation beeinflusst den Verständnisprozess jedoch sehr wohl negativ, was sich an der Dauer der Problemlösung und an der subjektiv wahrgenommenen Verständlichkeit zeigt.

Weiter zeigen die Ergebnisse einen starken Einfluss von Fragmentierung und Kohäsion der Modelle auf. Stark fragmentierte Modelle erschweren die Verständlichkeit erheblich, wenn im Gegenzug die Kohäsion niedrig ist. In einer solchen Situation sind zusammengehörige Informationen über verschiedene Subprozessmodelle verteilt, wodurch der Split-Attention Effekt (vgl. Zugal et al. 2013) merklich negativen Einfluss nimmt.

Insbesondere zeigt diese Evaluation die Tauglichkeit der Good Decomposition Conditions als Grundlage für einen weiteren Entwicklungszyklus auf, in dem Metriken als Artefakt zur Bewertung von Dekompositionen entwickelt werden.

- **Forschungsfrage 3**: Welche Metriken lassen sich aus den Good Decomposition Conditions ableiten, um Modellierer bei der Dekomposition von Prozessmodellen zu unterstützen?

Der Beitrag zur dritten Forschungsfrage entwickelt anhand der fünf Good Decomposition Conditions zwölf Metriken zur Bewertung von Dekompositionen.

Für die erste Condition, „Minimalität“, werden zwei Metriken entwickelt. Die erste Metrik ermittelt den Anteil redundanter Ereignistypen in den Modellen einer Detailebene der Dekomposition, während die zweite Metrik den Anteil redundanter Event-Typen über alle Ebenen einer Prozessmodelldekomposition eruiert. Der Idealwert beider Metriken ist null, jedoch ist die Identifikation redundanter Ereignistypen in Prozessmodellen nicht einfach. Zusätzliche Informationen aus der Prozessausführung, welche Ereignistypen bei ausreichend langer Beobachtung in keiner Prozessinstanz instanziiert werden, können Hinweise auf redundante Event-Typen liefern. Ohne weitergehende Informationen welche Ereignistypen redundant sind, lässt sich die Metrik auch nicht automatisiert ermitteln.

Für die Condition „Determinismus“ werden drei Metriken entwickelt. Zunächst ermittelt Metrik drei den relativen Anteil an OR-Konnektoren, die ein Hinweis auf Determinismusverstöße sind, an allen Split-Konnektoren einer Dekompositionsebene. Metrik vier ermittelt den Anteil der Regelverstöße für die gesamte Dekomposition. Metrik fünf bezieht die fehlenden externen Ereignistypen mit ein, die ebenfalls zu einem Regelverstoß führen. Während die ersten beiden dieser Metriken anhand der Prozessmodelle automatisiert ermittelt werden können, werden für die Ermittlung fehlender externer Ereignisse wieder weitergehende Informationen benötigt, bspw. aus der Erfahrung des Modellierers oder Laufzeitinformationen. Alle drei Metriken haben idealerweise den Wert null

Die Verlustfreiheit wird mit einer Metrik, Metrik sechs, operationalisiert. Hierbei wird die Anzahl fehlender, nicht-redundanter Ereignistypen für die gesamte Dekomposition ermittelt. Das Ermitteln fehlender (interner) Ereignistypen kann wieder nicht alleine anhand der Modelle erfolgen. Der Idealwert der Metrik liegt bei null.

Die vierte Condition, „minimales Coupling“, wird anhand von vier Metriken operationalisiert. Die Erste dieser vier Metriken berechnet das Verhältnis der gekoppelten Funktionstypenpaare zu den möglichen Funktionstypenpaaren auf einer Detailebene. Gekoppelt ist ein Paar an Funktionstypen, wenn sie mit demselben Datentypen verbunden sind. Die zweite der vier Metriken, Metrik acht, setzt diesen Wert zusätzlich ins Verhältnis zu der Anzahl an Detailebenen. Der Idealwert beider Metriken ist null. Metrik neun operationalisiert die Condition anhand der externen Events. Diese werden für alle Modelle einer Detailebene ermittelt und mit der Anzahl an Modellen auf der Ebene ins Verhältnis gesetzt. Zuletzt ermittelt die zehnte Metrik die externen Ereignisse im Verhältnis zu allen Modellen einer Dekomposition. Für diese beiden Metriken ist der Idealwert eins.

Zuletzt operationalisiert der Beitrag das Kriterium „starke Kohäsion“ in zwei Metriken mit dem Idealwert null. Die erste Metrik, Metrik elf, ermittelt die Anzahl an Funktionstypenpaaren, welche dieselben Datenobjekttypen als Input haben im Verhältnis zu allen Funktionstypenpaaren auf einer Detailebene. Metrik zwölf ermittelt den Wert für alle Modelle einer Dekomposition.

Neben der Entwicklung der Metriken werden diese auch demonstriert. Dazu werden die Werte für drei alternative Dekompositionen eines Modells berechnet. Für die Alternativen wurden vormals empirisch Unterschiede in der Verständlichkeit ermittelt. Anhand der Ergebnisse werden Abhängigkeiten zwischen den Metriken diskutiert. Außerdem wird eine prototypische Implementierung vorgestellt.

Die Forschung zu den Good Decomposition Conditions in dieser Arbeit endet mit der Entwicklung der Metriken. Die nächsten Forschungsfragen behandelten die Entwicklung der Coupling Metriken.

- **Forschungsfrage 4**: Wie kann Coupling im Kontext von Geschäftsprozessmodellen definiert werden und was sind mögliche Einsatzszenarien für und Anforderungen an Coupling Metriken im praktischen Einsatz?

Die vierte Forschungsfrage beginnt den Design Science Zyklus zur Entwicklung der Coupling Metriken. Gegenstand des zugehörigen Beitrages ist die Analyse von Metriken, die in der Literatur bereits vorgestellt wurden. Diese werden dahingehend untersucht, welchen Beitrag sie zum Management von Prozessarchitekturen leisten können.

Im ersten Schritt wird eine systematische Literaturrecherche vorgestellt, mit welcher die existierenden Metriken ermittelt werden. Diese werden auch dahingehend selektiert, dass sie auf konzeptuelle Modelle anwendbar sind. Im zweiten Schritt werden die Metriken klassifiziert. Als Klassifizierungsmerkmale werden Kriterien ermittelt, die zwischen der externen und internen Sicht auf die Metriken unterscheiden. Die Klassen werden in einem dritten Schritt empirisch ermittelten Problemstellungen im Management von Prozessarchitekturen gegenübergestellt. Ermittelt wird dadurch, welche Anwendungsfälle durch Klassen an Metriken unterstützt werden können. Dies führt zu einer Diskussion, wie die Metriken in den Klassen den Anwendungsfall unterstützen, bzw., im Fall von unbesetzten Klassen, welche Metriken eine sinnvolle Ergänzung sein könnten.

Mit diesem Vorgehen wurden insgesamt elf Klassen ermittelt. Exemplarisch erwähnt sei bspw. eine Klasse an Metriken, welche die Komplexität des Kontrollflusses von einzelnen Modellen der Prozessarchitektur bewerten und damit einen Hinweis auf möglicherweise schwer verständliche Modelle liefern. Die nächste Klasse, welche dieselbe Analyse auf die gesamte Prozessarchitektur ausweitet, ist aktuell unbesetzt. Eine weitere erwähnenswerte Klasse deckt eine Metrik ab, welche potentielle Redundanz von Funktionen in einer Prozessarchitektur bewertet und damit bspw. Hinweise für die Standardisierung von Prozessen liefert.

Weiter verwendet der Beitrag die Ergebnisse der Analyse, um den Begriff „Coupling“ im Kontext von Geschäftsprozessmodellen spezifischer zu definieren. So schlägt der Beitrag zwei Definitionen vor. Damit wird erstmals berücksichtigt und explizit beschrieben, dass Coupling, je nach Auffassung des Autors, eine Aussage über das Geschäftsprozessmodell oder aber über den Geschäftsprozess trifft.

- **Forschungsfrage 5**: Wie und welche Coupling Metriken, bzw. der jeweils gemessene Qualitätsaspekt, aus verwandten Disziplinen lassen sich für Prozessmodelle operationalisieren?

Der vorherige Beitrag zeigt die Anwendungsfälle von Coupling Metriken für Geschäftsprozessmodelle auf. Der Beitrag zu Frage fünf transferiert die Metriken auf Geschäftsprozessmodelle und nimmt damit die konkrete Entwicklung im Sinne des Design Science Zyklus vor. Grundlage für den Transfer sind Metriken aus anderen Disziplinen, bspw. dem Software-Engineering, die zur Messung von Coupling in konzeptuellen Modellen entwickelt wurden. Ziel ist es, das individuelle Verständnis des jeweiligen Autors darüber, wie Coupling in einem konzeptuellen Modell in Erscheinung tritt, wenn sinnvoll möglich, auf Geschäftsprozessmodelle zu übertragen.

Im ersten Schritt wird folglich die Literaturrecherche vorgestellt, die dazu dient, für den Transfer geeignete Metriken zu finden. Der zweite Schritt behandelt den eigentlichen Transfer. Hierfür wird zunächst aufgelistet welche Konzepte des Modells von der Metrik im Original adressiert werden. Anschließend werden äquivalente Konzepte für Geschäftsprozessmodelle gesucht, wobei ausschlaggebend berücksichtigt wird, wodurch, nach Auffassungen der Autoren der ursprünglichen Metrik, Coupling, entsteht.

Zunächst betrachtet die Metrik von (Reijers & Vanderfeesten 2004) Coupling als die Abhängigkeit betrieblicher Aufgaben aufgrund von Informationen, die von einer Aufgabe dokumentiert und von einer anderen Aufgabe konsumiert werden. Entsprechend wird für den Transfer berücksichtigt, wie ein solcher Sachverhalt in Geschäftsprozessmodellen abgebildet wird. Im Vergleich dazu ermittelt die Metrik nach (Allen et al. 2001) Coupling als Eigenschaft des dem Modell zugrundeliegenden Graphen, weshalb für den Transfer der Graph der eEPK herangezogen wird. Mit diesem Verfahren werden neun Metriken transferiert und für die eEPK formal definiert. Eine dritte Variante behandelt Metriken, die ursprünglich für Konzepte der objektorientierten Programmierung entwickelt wurden. Als Bindeglied zu den Konzepten der Geschäftsprozessmodellierung wird die bereits erwähnte Ontologie nach Bunge-Wand-Weber herangezogen. Ontologische Analysen für den Vergleich von Modellierungssprachen (vgl. Green & Rosemann 2000; Evermann & Wand 2005, 2009) erlauben es, ontologisch äquivalente Konzepte zu identifizieren. Eine besondere Rolle nimmt die Metrik nach (Poshyvanyk & Marcus 2006) ein. Diese wurde zwar ursprünglich ebenfalls für die objektorientierte Programmierung entwickelt. Sie verwendet aber ein Textanalyseverfahren aus dem Information Retrieval und bezieht sich auf Textfragmente, bspw. auch Kommentare und Dokumentation, im

Programmcode. Entsprechend wird für den Transfer also der textuelle Inhalt von Prozessmodellen herangezogen.

Die Definitionen der Metriken eigenen sich als Grundlage für die Implementierung als Tool, auch weil sie automatisiert berechenbar sind. Eine solche Implementierung wird vorgenommen und im Beitrag vorgestellt.

Zusammenfassend behandelt der Beitrag zu Forschungsfrage fünf also das Design und die Demonstration der Metriken im Forschungsschwerpunkt Coupling.

Mit Forschungsfrage sechs beginnt die Evaluation des Artefakts im Forschungsschwerpunkt Coupling:

- **Forschungsfrage 6**: Erfüllt die Konstruktion der Metriken bekannte Kriterien aus der Forschung?

Hierbei werden zum einen das Vorgehen zur Konstruktion, als auch die dabei entstandenen Metriken betrachtet.

Etablierte Methoden der Design Science sehen vor, das entwickelte Artefakt im letzten Schritt zu evaluieren. Dagegen schlagen Sonnenberg und vom Brocke (2012a) vor, nach jedem Schritt im DS Prozess zu evaluieren. Dies kann einerseits die Qualität des Artefakts erhöhen und ggf. zusätzliche Entwicklungszyklen vermeiden, wenn Probleme früher entdeckt werden. Andererseits führen die zusätzlichen Evaluationen stets zu zusätzlichen Arbeiten. Im Beitrag wird anhand der Entwicklung der Coupling Metriken erklärt, welche Auswirkungen es haben kann, eine solche Evaluationsstrategie wie die nach Sonnenberg und vom Brocke (2012a) anzuwenden.

Die erste zusätzliche Evaluation führt zu wenig zusätzlichem Aufwand, da eine ohnehin notwendige Literaturrecherche herangezogen werden konnte. Die zweite, zusätzliche Evaluation folgt auf das Design des Artefakts, vor dessen Implementierung. Sie wird im Beitrag anhand der Kriterien nach *Weyuker* (Weyuker 1988) durchgeführt. Diese Kriterien beschreiben wünschenswerte Eigenschaften von Komplexitätsmetriken. Bei dieser Evaluation wird festgestellt, dass nicht alle Metriken alle wünschenswerten Eigenschaften erfüllen. Gründe und Auswirkungen dieser Verstöße werden im Anschluss diskutiert.

Die Evaluation anhand der wünschenswerten Eigenschaften zeigt bspw., dass die Skalierung einiger Metriken im praktischen Einsatz zu Missverständnissen führen kann. So ist es möglich einen augenscheinlich besseren, niedrigeren, Wert für die Metrik zu erhalten, indem der Skalierungsfaktor erhöht wird, bspw. indem Modelle mit irrelevanten Informationen künstlich aufgebläht werden oder aber unzusammenhängende Sachverhalte in ein Modell zusammengefasst werden. Beides läuft der ursprünglichen Zielsetzung, die Prozessmodellqualität zu erhöhen, zuwider. Die Metriken zeigen also eine Verbesserung im Modelldesign nicht zwingend zuverlässig an. Andererseits erlaubt es die Skalierung, Modelle unterschiedlicher Größe zu vergleichen. Dieser Umstand muss vor einem praktischen Einsatz berücksichtigt werden.

Die oben diskutierte Einschränkung ist im bisherigen Entwicklungsprozess nicht aufgefallen. Sie ist jedoch potentiell geeignet, einen neuen Entwicklungszyklus zu forcieren, falls in einer praktischen Evaluation herauskäme, dass Anwender, im Vertrauen auf einen besseren Wert der Metrik, Modelle künstlich vergrößern.

Im Ergebnis evaluiert der Beitrag das Design der Metriken und zeigt Verbesserungspotentiale auf. Weiter wird im Beitrag der Entwicklungsprozess hinterfragt und aufgezeigt zu welchem Aufwand die zusätzlichen Evaluationen führen können.

Die nächste Forschungsfrage behandelt die empirische Evaluation der Metriken, die als Laborexperiment durchgeführt wird.

- **Forschungsfrage 7**: Zeigen die Coupling Metriken eine verminderte oder verbesserte Verständlichkeit von Prozessmodellen an?

Der Beitrag zu Forschungsfrage sieben berichtet über ein Laborexperiment, dass Aufschluss darüber liefert, dass Coupling Metriken eine veränderte Verständlichkeit von Prozessmodellen messen können. Die theoretische Grundlage für das Design des Experiments ist die Cognitive Load Theory (vgl. S. 184). Diese besagt, dass übermäßig komplexe Prozessmodelle ihre Nutzer überfordern, was wiederum die Wahrscheinlichkeit für Missverständnisse erhöht. Weil das Design des Experiments, insbesondere die Materialien, auf die jeweilige Theorie hin zugeschnitten sein muss, wird der Fokus auf jene drei Metriken gelegt, die auf der Cognitive Load Theory fußen.

Für jede der drei Metriken wird zunächst ein Hypothesenpaar definiert, dass ein höherer bzw. niedrigerer Wert als Indikator für ein komplexeres Modell, mit einer niedrigeren Verständlichkeit des Modells einhergeht.

Im Weiteren werden die Instrumente vorgestellt, mit denen das Verständnis der Nutzer gemessen wird, bzw. inwieweit Nutzer infolge des kognitiven Prozesses in die Lage versetzt werden auf Problemlösungen zu schließen. Die Teilnehmer des Experiments sollen zwei dafür entwickelte bzw. angepasste Problemstellungen lösen. Für das erste Messinstrument sollen Teilnehmer Fragen zur Beziehung zwischen zwei Knoten des Modells beantworten, die sich aus dem Kontrollfluss ergibt. Hierfür müssen die Teilnehmer die möglichen Zustände des Kontrollflusses zwischen den zwei Knoten erkennen, speichern und dahingehend untersuchen ob es zu unerlaubten Zuständen kommt. Für das zweite Instrument werden Fragen gestellt, für die Teilnehmer die möglichen Zustände des Kontrollflusses vom ersten zum letzten Knoten erkennen, speichern und evaluieren müssen.

Diese beiden Instrumente sind auf die Cognitive Load Theory zugeschnitten und sind gleichzeitig Grundlage für die entwickelten Materialien des Experiments. Diese umfassen fünf Prozessmodelle, mit drei Behandlungsstufen und berücksichtigen gleichzeitig zahlreiche Kontrollvariablen. Jeder Teilnehmer durchläuft alle drei Behandlungsstufen. 49 Teilnahmen können zur Auswertung herangezogen werden.

Die statistische Auswertung ergibt, dass die Nullhypothese gegen den Einfluss für jede Metrik abgelehnt werden muss.

Im Anschluss an die Beziehung zwischen Modellverständnis und Metrik werden die Unterschiede zwischen den Metriken diskutiert. Sie haben unterschiedliche Erklärungswerte für die jeweiligen Instrumente. Diese werden diskutiert, indem der spezifische, dem Instrument zugrundeliegende kognitive Prozess mit der Metrik verglichen wird. So kann erstmals gezeigt werden, dass Metriken mehr oder weniger geeignet sind, wenn sich der Anwendungsfall für das Prozessmodell und damit der kognitive Prozess der Modellnutzung ändern. Entsprechend zeichnet sich ab, dass zukünftig Metriken spezifisch für Modellanwendungsfälle entwickelt werden müssen, bzw. deutlich mehr Metriken als in dieser Arbeit behandelt wurden.

3.2 Kritische Würdigung

Die bisher aufgezeigten Ergebnisse benötigen einer kritischen Einordnung, die zunächst an den Methoden der Beiträge vorgenommen werden können.

Im ersten Beitrag wurden Fallstudien und eine Umfrage vorgenommen. Insbesondere im Zusammenhang mit der Fallstudie besteht die Gefahr von Subjektivität bei der Interpretation der Ergebnisse. Folglich wurden die Fallstudien stets von zwei Forschern durchgeführt und mit zwei weiteren diskutiert, um dieses Risiko zu minimieren. Auch wurden sie den Ergebnissen der Umfrage gegenübergestellt. Bei Umfragen wiederum besteht die Gefahr, dass die Teilnehmer das Ergebnis verzerren, insbesondere, dass nur Unternehmen an der Umfrage teilnehmen, die ein Interesse an BPM haben. In diesem Fall wäre davon auszugehen, dass deutlich mehr Unternehmen als beschrieben das Potential von BPM nicht oder nur beschränkt nutzen.

Im zweiten Beitrag wurden die Good Decomposition Conditions evaluiert. Hier ist kritisch anzumerken, dass die Conditions in der Arbeit für die eEPK Notation entwickelt wurden. Grund hierfür ist, dass die Unterschiede der Sprachen Einfluss auf die Conditions nehmen und deswegen bei der Entwicklung berücksichtigt werden müssen um, die Conditions konkret beschreiben zu können. Eine sprachübergreifende Beschreibung wäre abstrakt und interpretationsbedürftig und damit aus Sicht von Modellierern aufwendiger anzuwenden. Eine weitere Einschränkung ist, dass das Experiment bisher nur mit Studenten vorgenommen wurde. Für Modellierer und Modellnutzer mit einem anderen Erfahrungs- und Wissensstand können die Conditions womöglich einen stärkeren oder schwächeren Einfluss haben. Dieser Frage muss mit weiteren Experimenten und ggf. Evaluationen in praktischer Umgebung nachgegangen werden.

Im dritten Beitrag werden die vormals evaluierten Conditions nun in Form von Metriken operationalisiert. Hierbei ist zu kritisieren, dass die Aggregation der Metriken für eine umfassende Bewertung der Dekomposition noch nicht behandelt wurde. Weiter ist anzumerken, dass nicht alle Metriken automatisiert berechenbar sind, da nicht alle dafür notwendigen Informationen alleine aus Modellen gewonnen werden können. Hierfür wäre zu untersuchen, inwieweit diese Informationen bspw. mit Process Mining ermittelbar sind. Zuletzt wurden die Metriken bisher nur auf einen Fall angewandt, sodass eine praktische Evaluation zukünftig noch vorgenommen werden muss.

Der nächste Beitrag behandelt Anwendungsfälle für Coupling Metriken. Der Beitrag fußt auf einer Literaturrecherche, die stets das Risiko mit sich bringt, relevante Literatur zu übersehen. Um dieses Risiko zu minimieren, wurde die Vorwärtssuche mit einer Rückwärtssuche ergänzt, die jedoch zu keiner zusätzlichen, relevanten, Literatur führt.

Beitrag fünf beschreibt den Transfer, also die eigentliche Entwicklung der Metriken. Hierbei besteht die Gefahr, dass die Interpretation äquivalenter Konzepte durch die Subjektivität der Forscher beeinflusst wurde. Um das Risiko zu minimieren, wurde der Transfer von zwei Forschern vorgenommen, die ihre Resultate am Ende konsolidierten. Außerdem ist auch hier der Fokus auf die eEPK zu kritisieren, der gesetzt werden musste, um eine formale Definition der Metriken entwickeln zu können.

Der Beitrag zu Forschungsfrage sechs evaluiert das Design der Coupling Metriken und präsentiert Erfahrungen mit der Evaluationsstrategie früher Evaluationen. An Einschränkungen sind hier vor allem zwei vorzunehmen. Einerseits konnte zwar gezeigt werden, dass frühe Evaluationen sinnvoll sein können. Dennoch ist jedes DS Projekt für sich anders, weshalb die präsentierten Erfahrungen nicht zwingend gemacht werden müssen. Auf der anderen Seite wurde der Einfluss der Verstöße gegen die wünschenswerten Eigenschaften nur theoretisch diskutiert. Es wurde nicht nachgeprüft, ob die möglichen, diskutierten, Einschränkungen bei der Verwendung der Metriken in der Praxis auch tatsächlich auftreten.

Der letzte Beitrag evaluiert die Coupling Metriken in einem Laborexperiment und zeigt auf, dass die Metriken Fälle verminderter Verständlichkeit anzeigen können. In der Natur von Laborexperimenten liegt es, dass deren Sachverhalt nicht zwingend auf den praktischen Einsatz übertragbar ist. Wie häufig die Behandlungsstufen sich auch in alltäglich verwendeten Prozessmodellen wiederfinden ist nicht bekannt. Jedoch ist der Funktionsbeweis im Idealfall eine zwingende Voraussetzung dafür, dass die Metriken im weniger idealen Praxisfall funktionieren können. Das Laborexperiment ist folglich eine zwingende Voraussetzung für die Evaluation der Metriken in der Praxis, bspw. in Form von Fallstudien.

3.3 Ausblick

In dieser Arbeit wurden die gestellten Forschungsfragen wie o.g. beantwortet. Dabei kamen im Laufe des Forschungsprozesses weitere Fragestellungen auf, die zwar außerhalb des Fokus dieser Arbeit lagen, jedoch nicht weniger relevant sind.

Offensichtlich sollte die in Kapitel 3.2 geäußerte Kritik aufgegriffen werden, um die methodische Grundlage weiter zu stärken. So sollte bspw. die experimentellen Evaluationen der Decomposition Conditions und der Coupling Metriken mit Praktikern wiederholt und in Fallstudien evaluiert werden. Ebenfalls sollten die Metriken beider Schwerpunkte auf andere Modellierungssprachen, wie bspw. die BPMN erweitert werden. Ein Anfang hierfür wurde in Johannsen et al. (2014b) gemacht. Die Literaturrecherchen und die Transfers der Konzepte sollten von weiteren Forschern wiederholt werden, um dem Ziel der Vollständigkeit näher zu kommen. Und zuletzt kann die Generalisierbarkeit der Ergebnisse zum Stand des BPM im ersten Beitrag und zur Evaluationsstrategie im sechsten Beitrag durch wiederholte Anwendung in weiteren Sachverhalten gestärkt werden.

Darüber hinaus lassen sich die behandelten Schwerpunkte, Decomposition Conditions und Coupling Metriken, im Sinne der übergreifenden Zielsetzung, Mittel und Möglichkeiten zur Steuerung und Verbesserung der Verständlichkeit von Geschäftsprozessmodellen, fortführen, erweitern, und vertiefen.

Eine mögliche Fortführung kann darin bestehen, die beiden, bisher separat behandelten, Schwerpunkte zu kombinieren. Die Coupling Metriken geben einen Hinweis darauf, welche Modelle für einen Nutzer schwer verständlich sein könnten, schlagen aber keine Lösung zur Verbesserung des Modells vor. Die Modelldekomposition wird eingesetzt, um komplexe Prozessmodelle verständlich darzustellen und die Decomposition Conditions bewerten die Güte einer Dekomposition. Sie geben aber wiederum keine Hinweise darauf, wo eine Dekomposition vorgenommen werden sollte. Eine Kombination beider Mittel könnte folglich Hinweise darauf geben, wo Verbesserungsbedarf besteht, wie die Verbesserung erreicht wird und ob sie erfolgreich war. Jedoch muss hierfür sichergestellt werden, dass das von den Coupling Metriken aufgezeigte Problem auch tatsächlich durch eine Dekomposition gelöst wird, dass Conditions und Metriken nicht im Widerspruch stehen und vor allem muss eine solche Kombination in einer Art und Weise implementiert werden, dass sie für Modellierer einfach zu verwenden ist. Grundlage hierfür können die in dieser Arbeit entwickelten Artefakte sein. Dies lässt sich wie folgt als Forschungsfrage formulieren:

- **Forschungsfrage 8**: Wie können verschiedene Werkzeuge zur Messung und Steuerung der Prozessmodellverständlichkeit integriert werden?

In dieser Arbeit wurde der DS Entwicklungszyklus umfassend umgesetzt, wofür die zu entwickelnden Artefakte eingegrenzt werden mussten. So wurde für den ersten Schwerpunkt die Ontologie nach Bunge-Wand-Weber als eine von verschiedenen möglichen theoretischen Grundlagen (vgl. Houy et al. 2014) herausgegriffen. Insbesondere aber im zweiten Forschungsschwerpunkt wurden zunächst Metriken der Dimension „Coupling“ herausgegriffen und Dimensionen wie bspw. Dichte und Komplexität ausgeschlossen. Für die Evaluation wurden auch noch solche Coupling Metriken betrachtet, die auf der Cognitive Load Theory fußen. Dabei gibt der vierte Beitrag bereits Grund zu der Vermutung, dass für den praktischen Einsatz die Entwicklung einer großen Zahl verschiedener Metriken sinnvoll ist. Folglich sollten zukünftig weitere Metriken entwickelt werden. Hierfür leistet die vorliegende Arbeit in zweierlei Hinsicht einen erheblichen Beitrag.

Zum einen wurden in der Arbeit Methoden für den Transfer von Metriken entwickelt, die für die Entwicklung weiterer Metriken eine erhebliche Hilfestellung leisten. In Forschungsfrage fünf wird eine Methode aufgezeigt, wie Metriken verschiedenster theoretischer Grundlagen transferiert und für die eEPK spezifiziert werden können. Dabei ist die entwickelte und demonstrierte Methode weder auf die Qualitätsdimension „Coupling“ noch auf die eEPK Notation beschränkt. So lassen sich mit der Methode auch zweifelsohne bspw. Dichtemetriken für die BPMN spezifizieren. Die Evaluation in Beitrag sechs eignet sich für solche Metriken ebenfalls. Und letztlich sind auch die Klassifikationsmerkmale des Frameworks in Beitrag vier generisch genug, um für weitere Metriken wiederverwendet zu werden.

- **Forschungsfrage 9**: Welche weiteren Metriken zur Messung und Steuerung von Prozessmodellqualität lassen sich, ggf. unter Zuhilfenahme der vorgestellten Methoden, entwickeln?

Zum anderen leistet die vorliegende Arbeit einen Beitrag zur Entwicklung zusätzlicher Metriken, indem die Klassifikation Lücken aufzeigt, in denen Klassen an Metriken bisher unbesetzt sind. Zusammen mit den diskutierten Anwendungsfällen findet sich hier die Grundlage für die Entwicklung weiterer Metriken.

- **Forschungsfrage 10**: Welche weiteren Metriken zur Messung und Steuerung von Prozessmodellqualität lassen sich anhand der unbesetzten Klassen der Klassifikation entwickeln?

Der vierte Beitrag zeigt noch eine weitere Grundlage für die Entwicklung weiterer Metriken auf. So werden die Klassen anhand von Anwendungsfällen des Prozessarchitekturmanagements diskutiert. Hier erscheint es naheliegend und notwendig weitere Anwendungsfälle in die Entwicklung miteinzubeziehen.

- **Forschungsfrage 11:** Welche Metriken zur Messung und Steuerung von Prozessmodellqualität lassen sich für weitere Anwendungsfälle im Geschäftsprozessmanagement entwickeln?

Eine Vertiefung lässt sich bspw. im Schwerpunkt „Decomposition Conditions" vornehmen. Prinzipiell konnte der zweite Beitrag zeigen, dass Verstöße gegen die Conditions zu weniger verständlichen Dekompositionen führen, jedoch nehmen nicht alle Conditions den gleichen Einfluss. Weiter zeigt Beitrag drei auf, dass zwischen den Metriken zu den Conditions Wechselwirkungen existieren. Daher kann es für den praktischen Einsatz notwendig sein, zu entscheiden, welche Verstöße eher in Kauf genommen werden. Hierfür ist es jedoch notwendig, zu ermitteln, welche Verstöße schwerwiegender sind, und weiter vertieft bspw. ob dieses Verhältnis zueinander und in Abhängigkeit der Nutzergruppen konstant ist.

- **Forschungsfrage 12:** Welche Auswirkungen haben Wechselwirkungen zwischen den Decomposition Metriken auf den praktischen Einsatz?

Anhang

Anhang 1: Materialien zum Dekompositionsexperiment

http://wi3.uni-regensburg.de/fileadmin/user_upload/Questionnaire.pdf

Overview of decompositions – student enrollment process

The following process model shows the “student enrollment process” at a German university. The model stems from a project to document the processes of the university administration. Due to its size, the process model was decomposed into subprocess models that were assigned to model levels of a hierarchy (e.g., level 0, level 1).

The following alternatives A to C represent three different decompositions for the aforementioned student enrollment process model. Each alternative comprises four model levels (level 0 – level 3). The number of subprocess models is identical for alternatives A and B (eight subprocess models), whereas alternative C comprises ten subprocess models.

Alternative A

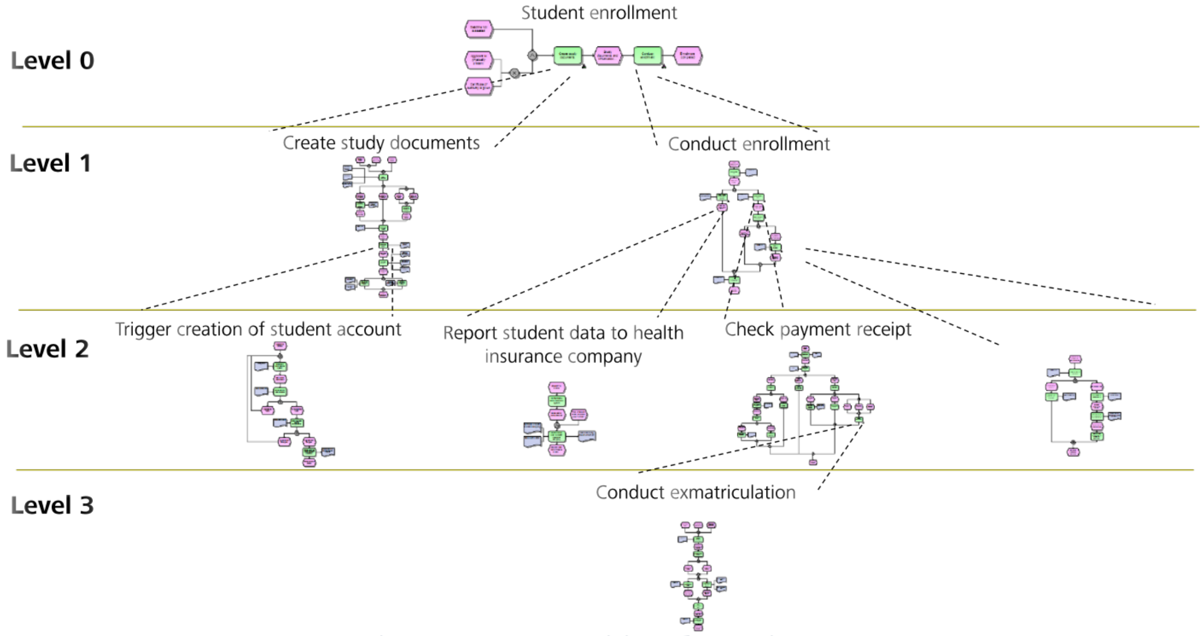


Figure 1: Decomposition alternative A

Alternative B

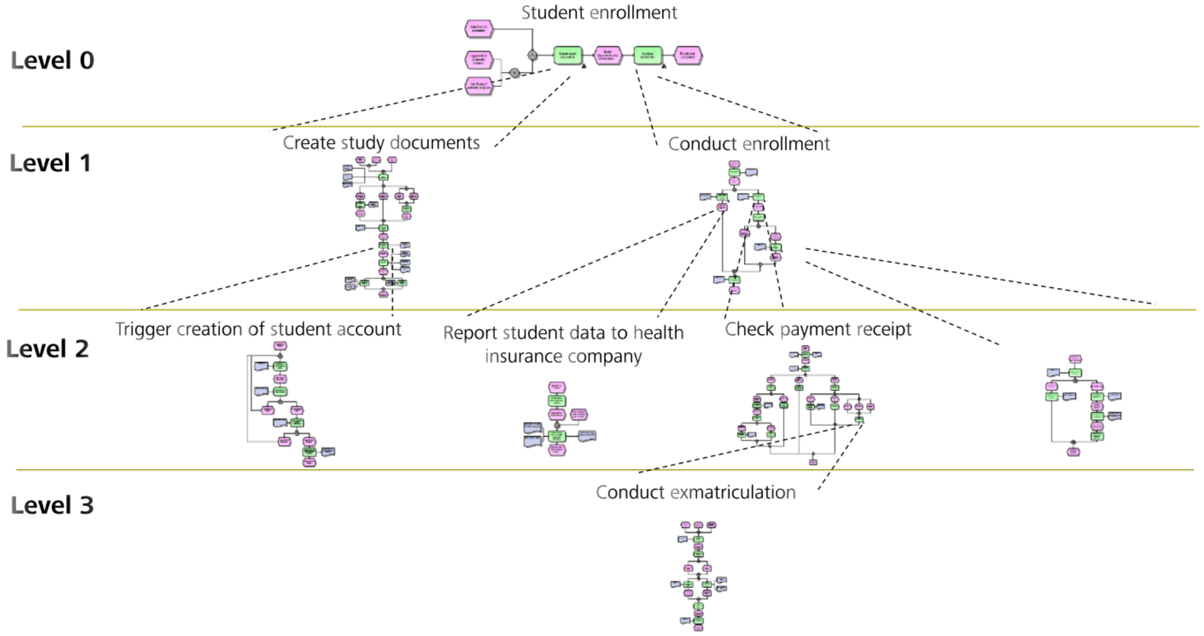


Figure 2: Decomposition alternative B

Alternative C

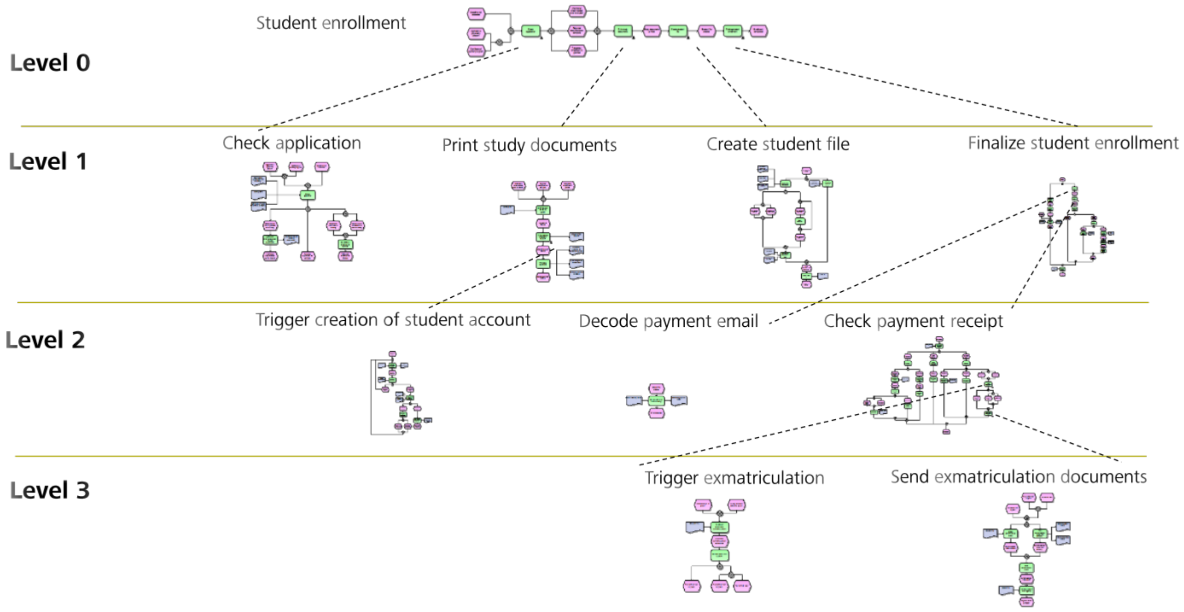


Figure 3: Decomposition alternative C

For the purpose of illustration, we exemplarily describe the decomposition "alternative A" in more detail. This alternative fulfills all decomposition conditions as described in the paper.

Models (detailed) for alternative A

The most upper level of the hierarchy (level 0) shows the general student enrollment process from a high-level perspective.
To trigger the process, it is required that the deadline for enrollment has not exceeded yet. Additionally, the applicant or a replacement person (e.g., parent, having a certificate of authority) has to be physically present. Only then, the study documents which officially certify the enrollment are created and the enrollment is conducted from an administrative point of view.

Level 0: Student enrollment:

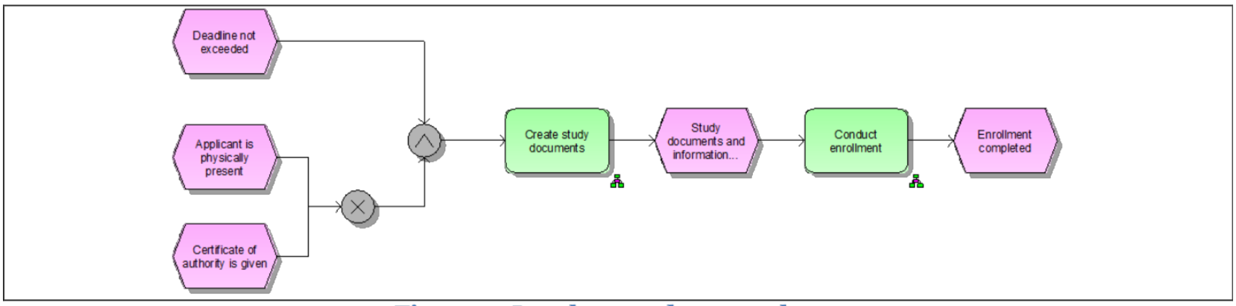


Figure 4: Level 0 - student enrolment

The activities “create study documents” and “conduct enrollment” were each specified in corresponding subprocess models assigned to level 1 of the hierarchy.

Create study documents: Prior to the creation of the study documents, the application needs to be checked. In case all mandatory documents are attached, the generation of a student account in the corresponding student database is started. Otherwise, depending on the relevance of the missing documents, only a temporary enrollment is triggered.
The main documents which are handed over to the applicant are a brochure on the course of studies, an acknowledgement of enrollment as well as an information brochure on the introductory courses offered.

Conduct enrollment: After the study documents were handed over to the applicant, the enrollment needs to be closed from an administrative perspective. This includes the creation of a student file, which is archived later on, the report of the student data to the health insurance company as well as the check of payment regarding the course fees. In case the enrollment turns out to be invalid (e.g., admission conditions not fulfilled), a refunding of the payment is conducted.

Level 1: Create study documents

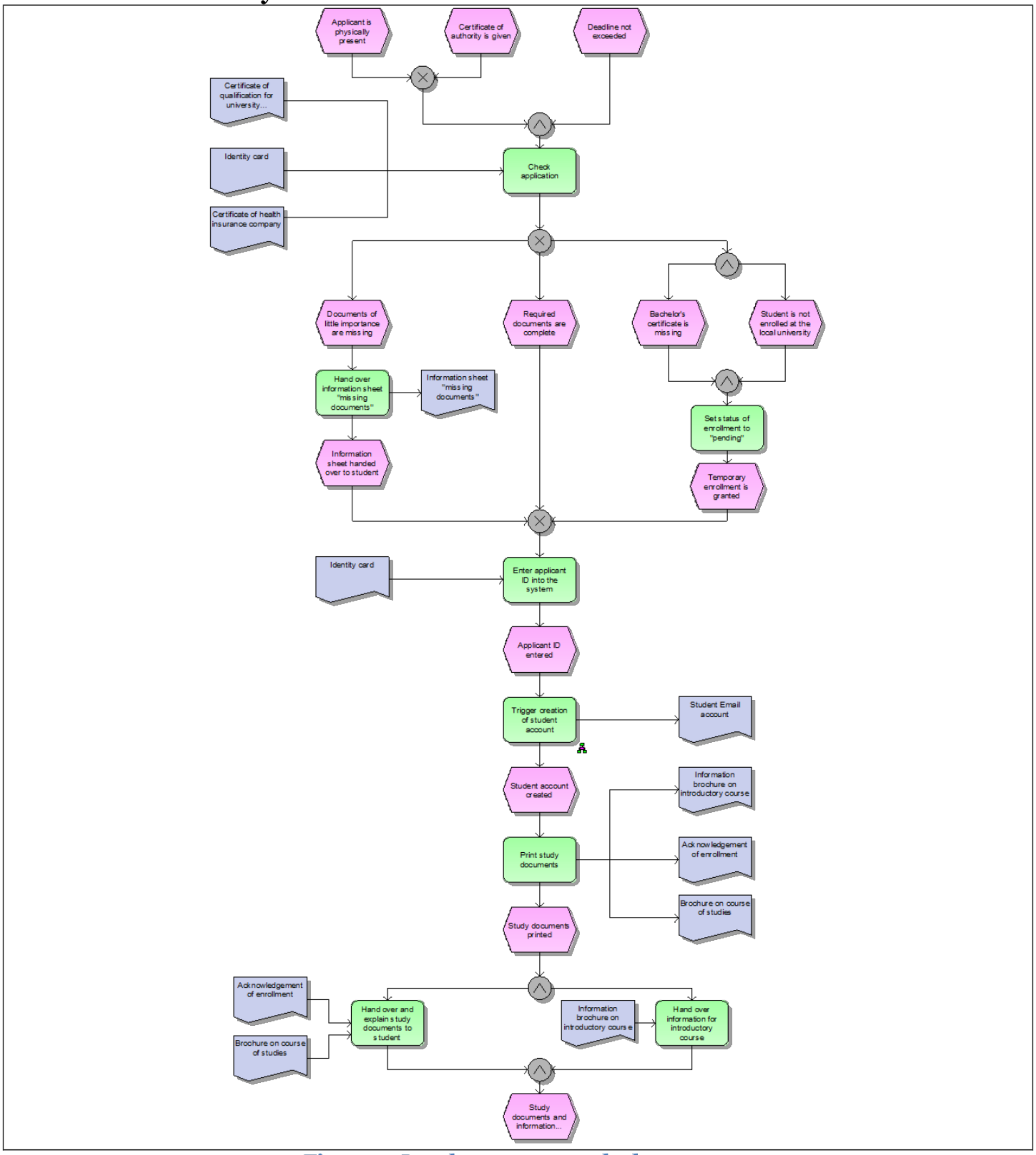


Figure 5: Level 1 – create study documents

Level 1: Conduct enrollment

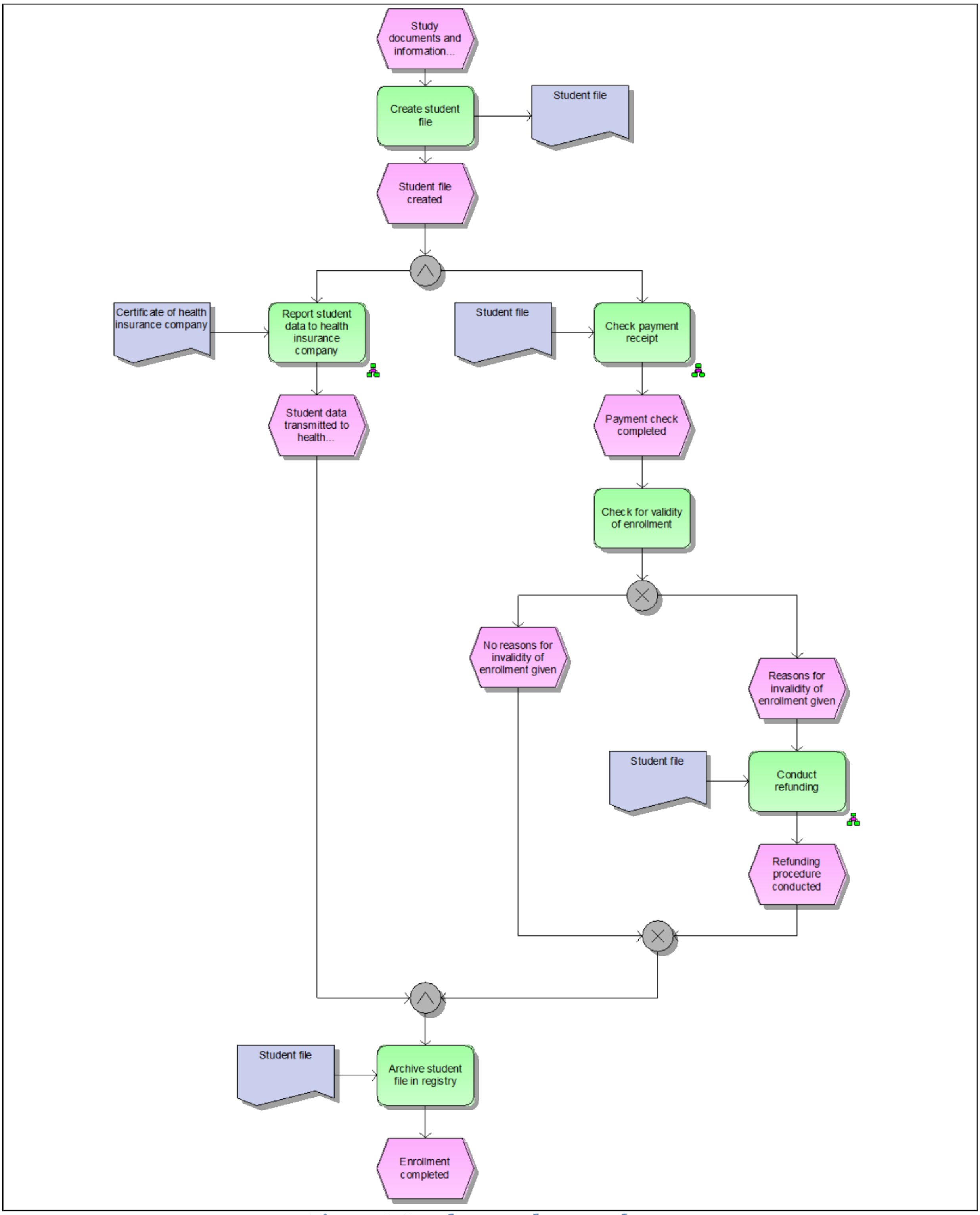


Figure 6: Level 1 – conduct enrolment

Both processes comprise complex working procedures that were described using subprocess models again. This concerns (1) the creation of a student account in the student database, (2) the report of the student data to the health insurance company, (3) the check of the payment receipt and (4) the refunding procedure in case of an invalid enrollment.
As a result, four subprocess models were created and assigned to level 2 of the hierarchy. These subprocess models provide more detailed information on specific tasks of the enrollment process that is relevant for different user groups. For example, the "trigger creation of the student account" subprocess model holds information significant for employees of the university's computer center. They are responsible for adapting the IT-systems to the needs of the administrative staff amongst others. Therefore, the computer center's employees need to be aware of the process steps that are to be supported by corresponding systems.

Trigger creation of student account: The successful creation of a student account requires a consistency check of the basic claims data at first. Further, the data need to be complete. Only then, the automatic triggering of an account can be started.

Report student data to health insurance company: To be able to generate a letter for the health insurance company confirming the successful student enrollment, a proof of the health insurance cover needs to be submitted by the applicant. Based on this proof and the basic claims data, which are extracted from the database, the confirmation is created.

Check payment receipt: To check the payment status of the applicant, the student office receives an email from the state treasury holding the corresponding payment data. Depending on private circumstances of the applicant, exemptions from payment may be given. In case the student does not pay, even after a reminder was sent, a rejection of the enrollment is performed. The same holds true in case of an undercharge. Due to internal working errors, a valid payment transaction may have been erroneously declared as invalid in the email of the state treasury. In that case, the transaction is booked manually by the student office employees.

Conduct refunding: A refunding is triggered in case reasons for an invalid enrollment are given. For example, it may turn out in retrospect, that an applicant did not fulfill the admission conditions for a certain course of studies. A payment transaction that has already been performed is then refunded. This is done by the budgeting department after the validity of the refunding has been carefully checked once again.

Level 2: Trigger creation of student account

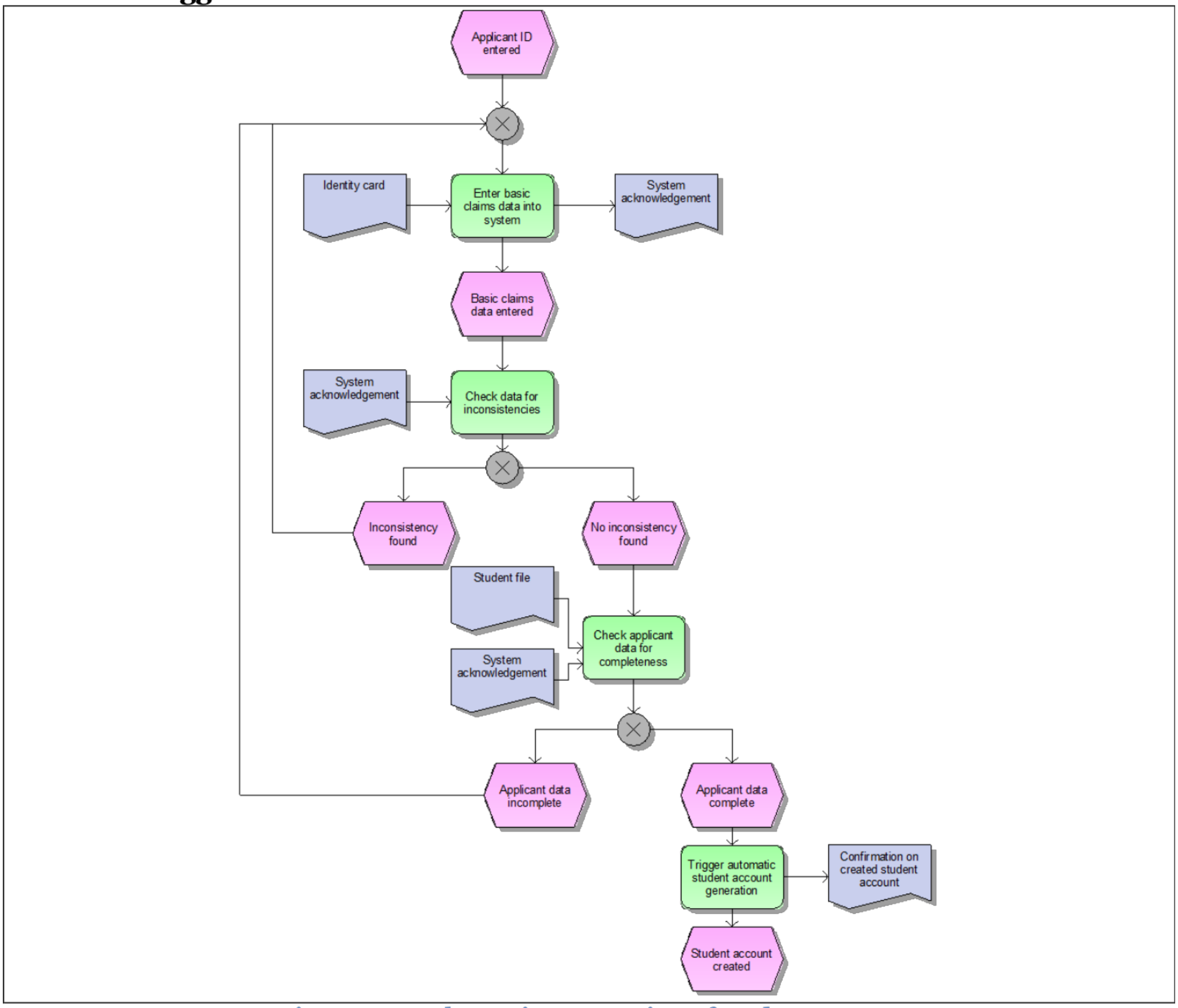


Figure 7: Level 2 – trigger creation of student account

Level 2: Report student data to health insurance company

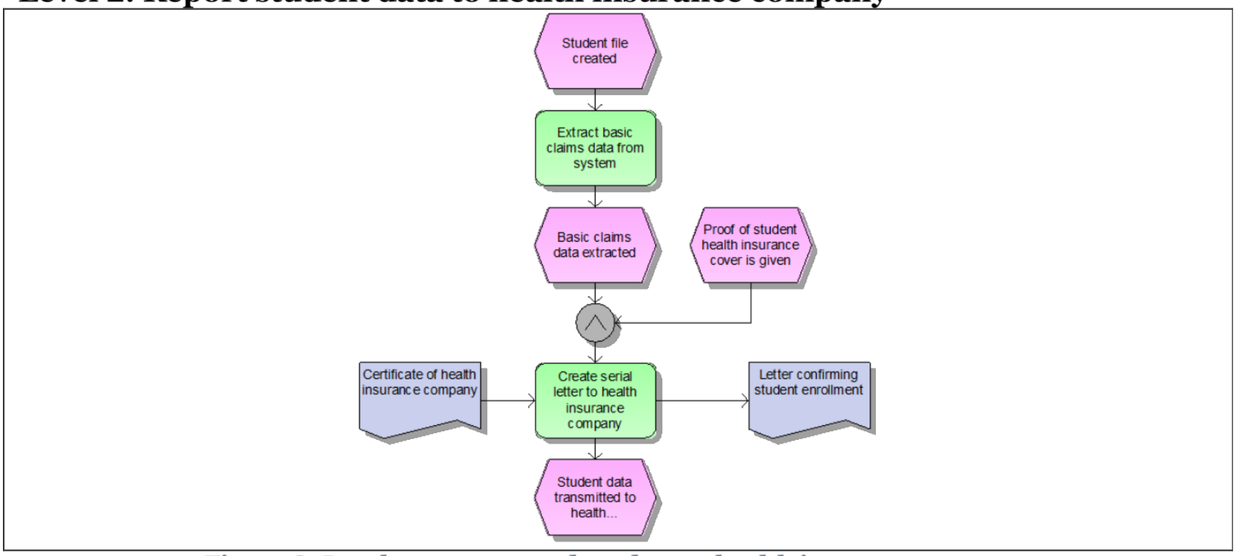


Figure 8: Level 2 – report student data to health insurance company

Level 2: Check payment receipt

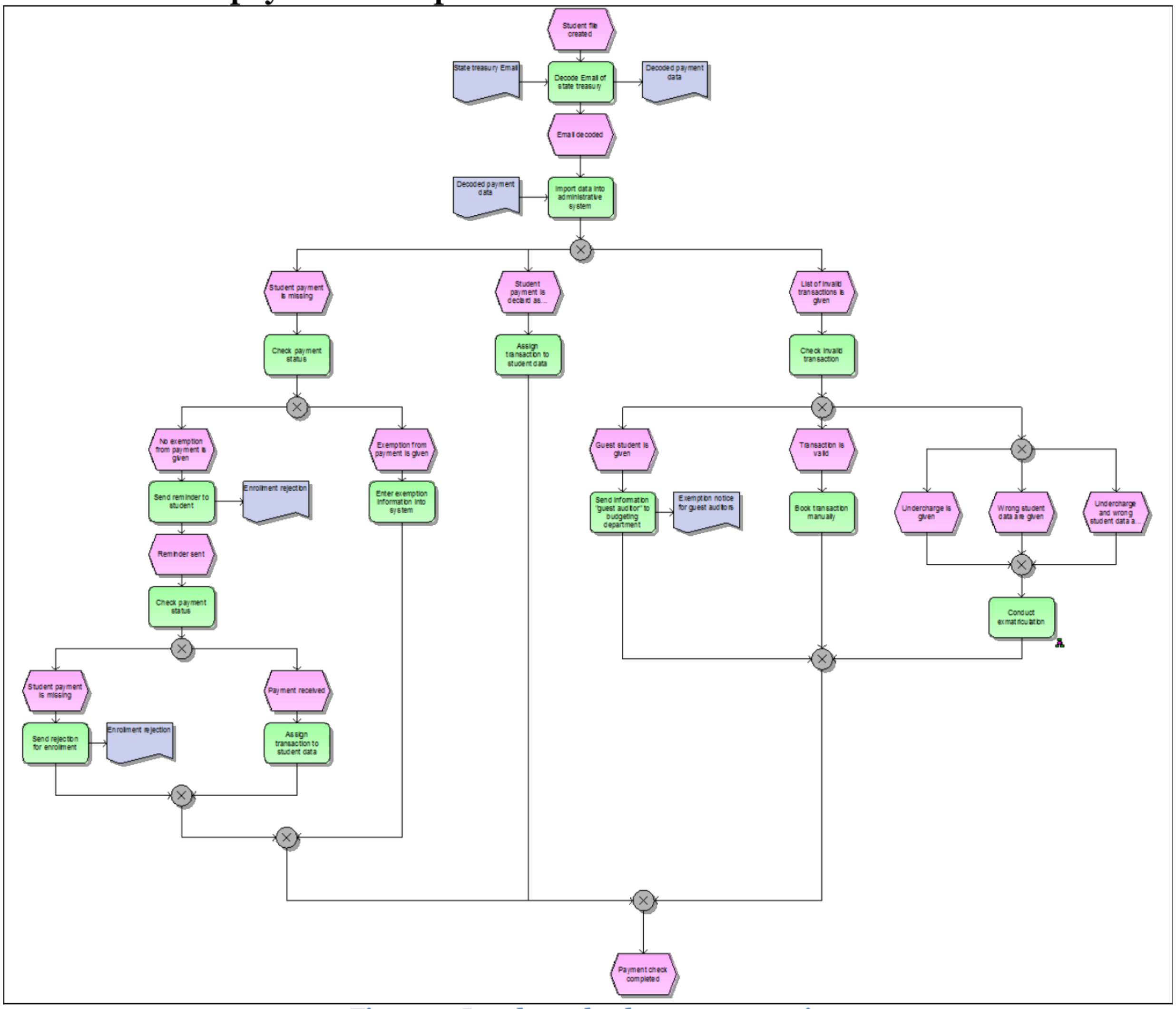


Figure 9: Level 2 – check payment receipt

Level 2: Conduct refunding

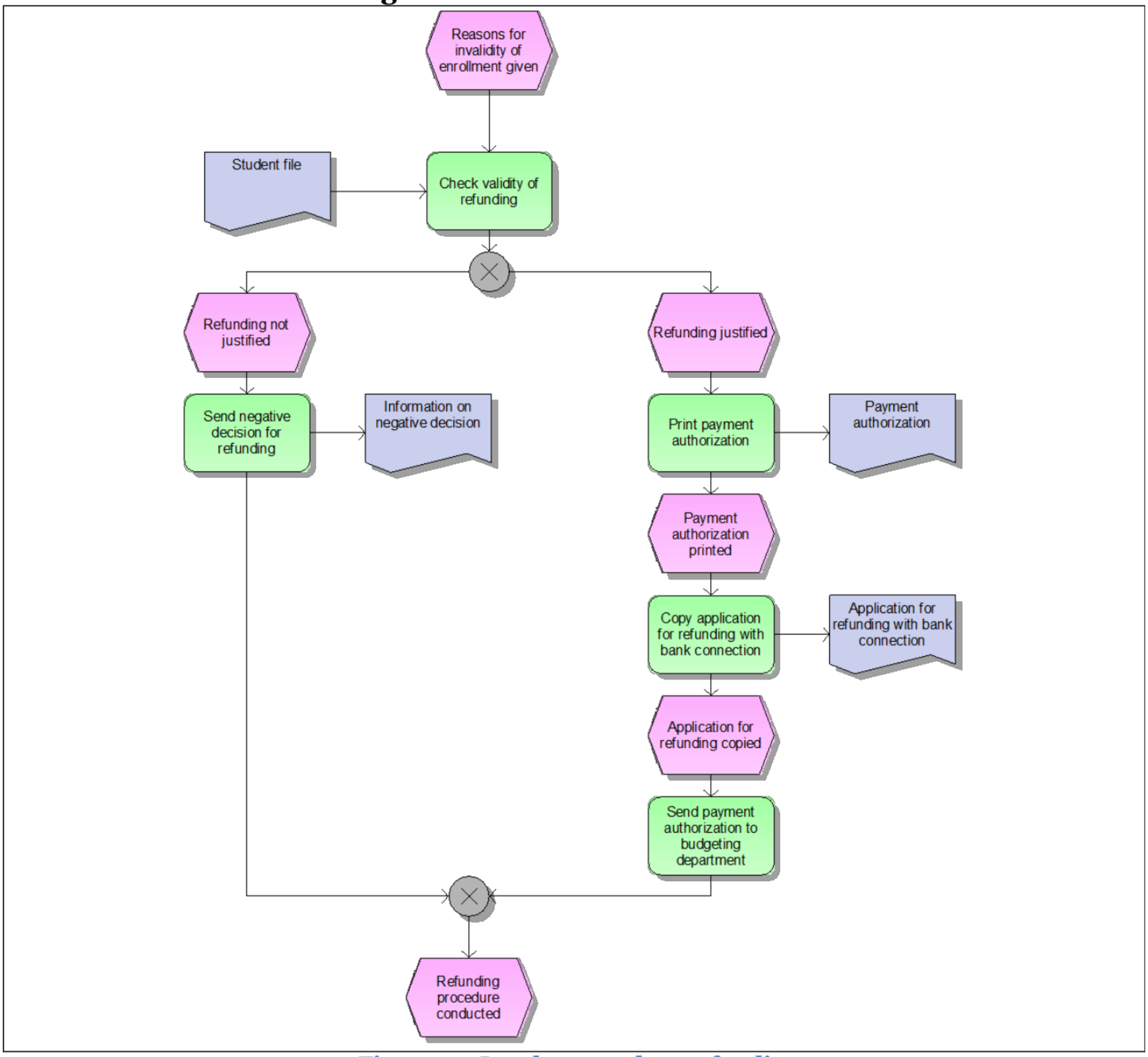


Figure 10: Level 2 – conduct refunding

The activity "conduct exmatriculation" in the subprocess "check payment receipt" is further specified by a subprocess model assigned to level 3 of the level hierarchy.

Conduct exmatriculation: In case an undercharge of payment or wrong student data are given, or both, the automatic exmatriculation is triggered. In that context, an exmatriculation notice and an annuity assurance policy are created.

Level 3: Conduct exmatriculation

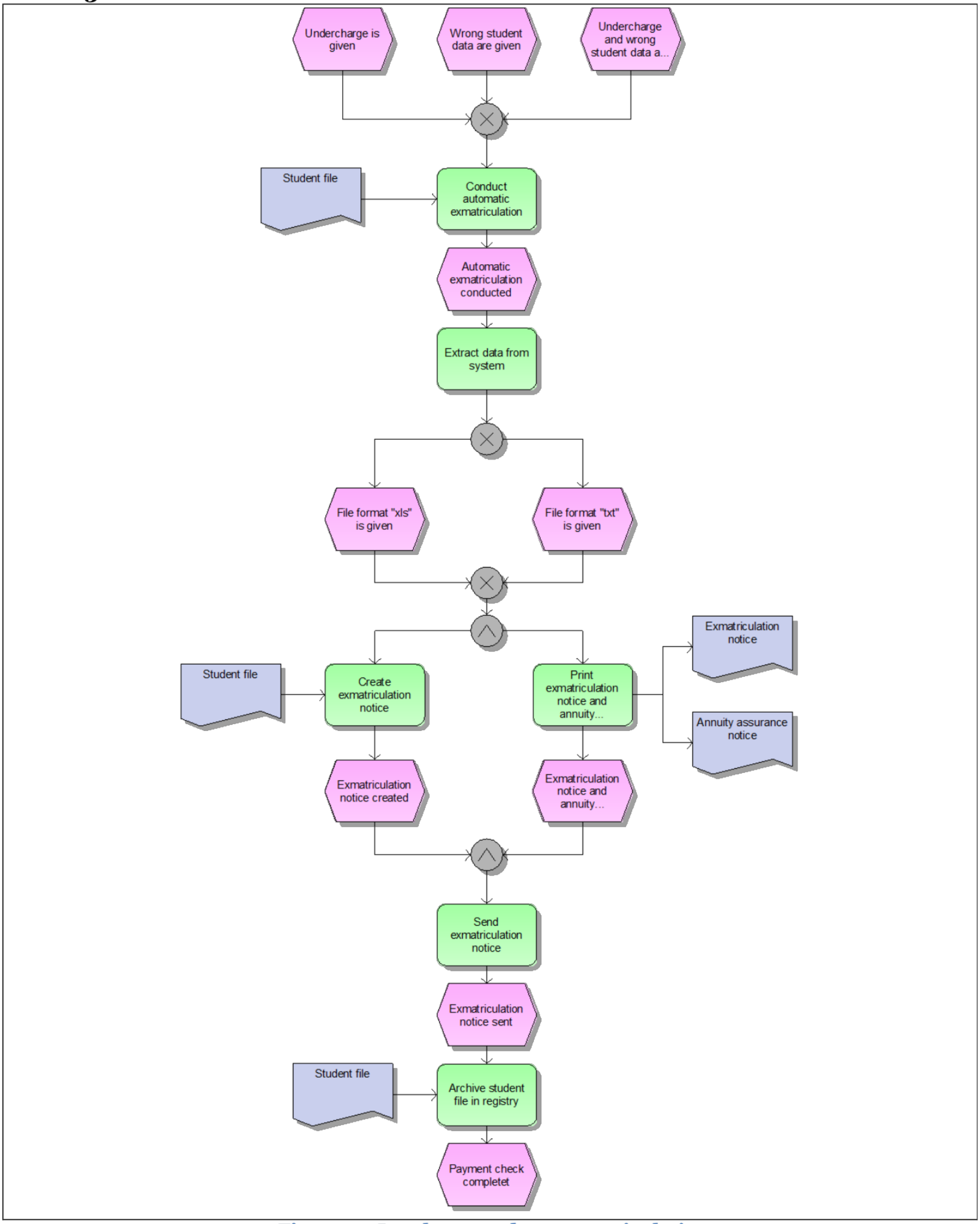


Figure 11: Level 3 – conduct exmatriculation

Differences in the alternatives A, B and C

In the following, we provide a brief summary of the differences for the alternatives A, B and C. As mentioned in the paper (see section “Materials of the Experiment”), alternative A fulfills all decomposition conditions, whereas alternative B shows moderate violations (violation of minimality, determinism and losslessness) and alternative C severely violates all five conditions.

As becomes obvious from Figure 3 (see this document), the delineation of subprocess models in alternative C differs from the delineation in alternatives A and B. For example, contrary to A and B, alternative C holds four subprocess models on level 1. This difference stems from a violation of the conditions “minimum coupling” as well as “strong cohesion”. By that, document types that serve as input in one subprocess model can be output of an activity type in another subprocess model for example (violation of minimum coupling – see Table 2 in the paper). This holds not true for alternatives A and B, both adhering to the minimum coupling and strong cohesion condition.

Further, there are no event types in the decomposed model A, indicating a state that never occurs in any process instance (minimality). Additionally, no relevant event types as well as attached function types and corresponding document types are missing in the decomposed model (losslessness). Also, there are no connectors (e.g., OR) or imprecise labels for the event types, which might lead to uncertainties during process execution in alternative A (determinism). Table 1 summarizes the violations that occur in alternatives B and C.

Alternative B
Subprocess model “create study documents”
Redundant event types “study documents complete”, “study documents printed properly”, “study documents incomplete” and “study documents complete” *(violation of minimality)*.
Subprocess model “conduct enrollment”
Missing input document types “certificate of health insurance company” and “student file” *(violation of losslessness)*.
Subprocess model “trigger creation of student account”
Redundant event types “applicant data fully incomplete” and “applicant data partly incomplete” *(violation of minimality)*. Splitting/joining OR-connector before/after event types “applicant data fully incomplete” and “applicant data partly incomplete” *(violation of determinism)*. Missing input document type “student file” *(violation of losslessness)*.
Subprocess model “check payment receipt”
Splitting/joining OR-connector before/after event types “student has withdrawn payment” and “student payment is missing” *(violation of determinism)*. Redundant event types “exemption entered into system” and “transaction assigned” *(violation of minimality)*. Splitting/joining OR-connector before/after event types “undercharge is given” and “wrong student data are given” *(violation of determinism)*. Splitting/joining OR-connector before/after event types “file format ‘xls’ is given” and “file format ‘csv’ is given” *(violation of determinism)*.
Subprocess model “conduct exmatriculation”
Joining OR-connector after event types “undercharge is given” and “wrong student data are given” *(violation of determinism)*. Splitting/joining OR-connector before/after event types “file format ‘xls’ is given” and “file format ‘csv’ is given” *(violation of determinism)*.
Alternative C
Subprocess model “check application”
Input document type “identity card” is input in subprocess model “print study documents” as well *(violation of strong cohesion)*. Input document type “certificate of health insurance company” is input in subprocess model “finalize student enrollment” as well *(violation of strong cohesion)*.
Subprocess model “print study documents”
Input document type “identity card” is input in subprocess model “check application” as well

(violation of strong cohesion). Output document type “information brochure on introductory course” is input in subprocess model “create student file” as well *(violation of minimum coupling)*. Output document type “acknowledgement of enrollment” is input in subprocess model “create student file” as well *(violation of minimum coupling)*. Output document type “brochure on course of studies” is input in subprocess model “create student file” as well *(violation of minimum coupling)*.
Subprocess model “create student file”
Input document type “information brochure on introductory course” is output in subprocess model “print study documents” as well *(violation of minimum coupling)*. Splitting/joining OR-connector before/after event types “study documents complete” and “study documents printed properly” *(violation of determinism)*. Redundant event types “study documents complete”, “study documents printed properly”, “study documents incomplete” and “study documents complete” *(violation of minimality)*. Output document type “student file” is input in subprocess model “finalize student enrollment” as well *(violation of minimum coupling)*.
Subprocess model “finalize student enrollment”
Input document type “certificate of health insurance company” is input in subprocess model “check application” as well *(violation of strong cohesion)*. Input document type “student file” is output in subprocess model “create student file” as well *(violation of minimum coupling)*. Missing input document types “certificate of health insurance company” and “student file” *(violation of losslessness)*.
Subprocess model “decode payment email”
Output document type “decoded payment data” is input in subprocess model “check payment receipt” as well *(violation of minimum coupling)*.
Subprocess model “trigger creation of student account”
Redundant event types “applicant data fully incomplete” and “applicant data partly incomplete” *(violation of minimality)*. Splitting/joining OR-connector before/after event types “applicant data fully incomplete” and “applicant data partly incomplete” *(violation of determinism)*.
Subprocess model “check payment receipt”
Input document type “decoded payment data” is output in subprocess model “decode payment email” as well *(violation of minimum coupling)*. Splitting/joining OR-connector before/after event types “student has withdrawn payment” and “student payment is missing” *(violation of determinism)*. Redundant event types “exemption entered into system” and “transaction assigned” *(violation of minimality)*. Splitting/joining OR-connector before/after event types “undercharge is given” and “wrong student data are given” *(violation of determinism)*. Splitting/joining OR-connector before/after event types “file format ‘xls’ is given” and “file format ‘csv’ is given” *(violation of determinism)*.
Subprocess model “trigger exmatriculation”
Input document type “student file” is input in subprocess model “send exmatriculation documents” as well *(violation of strong cohesion)*. Joining OR-connector after event types “undercharge is given” and “wrong student data are given” *(violation of determinism)*. Splitting OR-connector before event types “file format ‘xls’ is given” and “file format ‘csv’ is given” *(violation of determinism)*. Output document type “data file” is input in subprocess model “send exmatriculation documents” as well *(violation of minimum coupling)*.
Subprocess model “send exmatriculation documents”
Joining OR-connector after event types “file format ‘xls’ is given” and “file format ‘csv’ is given” *(violation of determinism)*. Input document type “data file” is output in subprocess model “trigger exmatriculation” as well *(violation of minimum coupling)*. Input document type “student file” is input in subprocess model “trigger exmatriculation” as well *(violation of strong cohesion)*.

Table 1: Violation of conditions in alternatives B and C

Questionnaire

Block 1

Task:
Please answer the following questions! [1]

How often do you deal with process models within your studies? (e.g. lecture, etc.)	
☐	Never
☐	Less than once a month
☐	Several times a month
☐	Almost each day

How often do you deal with process models besides your studies? (e.g. internship, etc.)?	
☐	Never
☐	Less than once a month
☐	Several times a month
☐	Almost each day

When did you first encounter process models in your studies or besides your studies?	
☐	A month ago at the most
☐	A year ago at the most
☐	Three years ago at the most
☐	Five years ago at the most

How often have you already enrolled at university for a certain course of studies?	
☐	Never
☐	Once
☐	Twice
☐	Several times
☐	No statement

[1] Derived and adapted for the research at hand from the proposal of: Mendling, J., Strembeck, M., Recker, J.: Factors of process model comprehension - Findings from a series of experiments. Decision Support Systems 53, 195-206 (2012)

Did you gather information which documents are needed by the registry office before registering at your university?	
☐	Yes
☐	No

Are you familiar with the procedure of the student enrolment process and the activities that need to be performed in that context from an administrative perspective?	
☐	No
☐	Partially
☐	Yes

Have you ever dealt with the matriculation process at a university in a lecture?	
☐	Yes
☐	No

Overview of decompositions – student enrollment process

The following process model shows the "student enrollment process" at a German university. The model stems from a project to document the processes of the university administration. Due to its size, the process model was decomposed into subprocess models that were assigned to model levels of a hierarchy (e.g., level 0, level 1).

The following alternatives A to C represent three different decompositions for the aforementioned student enrollment process model. Each alternative comprises four model levels (level 0 – level 3). The number of subprocess models is identical for alternatives A and B (eight subprocess models), whereas alternative C comprises ten subprocess models.

Alternative A

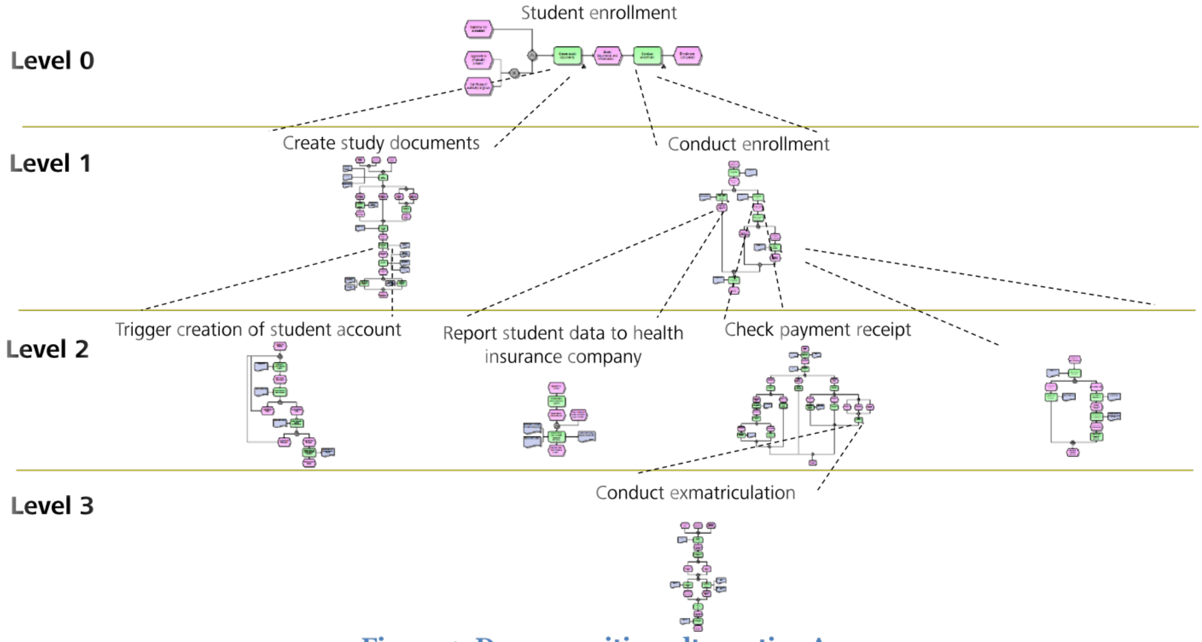


Figure 1: Decomposition alternative A

Questions on the process model

Block 3

Task:
Please answer the following questions at hand of the given process model – and not from your own experience.

Please answer in the given fields and hand back this questionnaire in the end.

a) Which documents does a student have to provide if he wants to matriculate?

b) In which cases is a reminder sent or the student removed from the student register in case of a negative payment check?

c) Which information does the encrypted mail from the state treasury contain?

d) How is to proceed with the creation of the student account if the applicant data happen to be incomplete?

e) Which preconditions are required for a refunding? And which documents are necessary therefore?

Block 4

Task:
In the following you will find statements about the process models shown. Please verify whether these statements are correct or not from your point of view.
Please decide by checking the boxes whether you accept or reject the statement. Please give a short explanation for your decision.

a) The study documents are only handed over after the applicant data have been entered into the system.

	Accept	*Reject*	*Reason*
Please check	☐	☐	

b) The matriculation process can be continued even if the student does not bring a bachelor's certificate.

	Accept	*Reject*	*Reason*
Please check	☐	☐	

c) The payment authorization is an obligatory precondition to verify the justification of the refunding.

	Accept	*Reject*	*Reason*
Please check	☐	☐	

d) The automatic exmatriculation is only triggered in case of an undercharged tuition fee.

	Accept	*Reject*	*Reason*
Please check	☐	☐	

e) In case the student did not pay the tuition fee, he will be sent a reminder before the matriculation possibly gets rejected.

	Accept	***Reject***	***Reason***
Please check	☐	☐	

Block 5

Task:
In the following you find a textual description of the process with gaps. Please fill the gaps to make the text describe the process correctly.

In order to start the matriculation process, it is necessary that, ____ ____ ________________________.

At the first check of the application, it may be found, that ____ ____ ________________________. Afterwards the student account is created. Therefore the basic claims data are entered into the system. If the check of the data brings up inconsistencies, then ____ ____ ________________________.

After the student account is created, the study files and the ____ _____________are printed and handed to the student. Afterwards, the student file is being assembled.

In order to execute the health insurance reporting ordinance, the student needs to hand in ____ _____________ so that a serial letter can be created, confirming the matriculation of the student.

Block 6

Task:
In the following, you will be asked about your personal rating of the depicted process model. For that you will find four statements.

Please note for every statement, in how far you agree with the statement.
The value "1" means that you fully disagree with the statement.
The value "7" means that you fully agree with the statement.
With the values in-between you may grade your agreement.

a) The pictured process model visualizes the control flow in a comprehensible way.

	I disagree completely						**I agree completely**
Please check	1	2	3	4	5	6	7

b) Understanding the process model requires enormous mental effort.

	I disagree completely						**I agree completely**
Please check	1	2	3	4	5	6	7

c) The decomposition of the process model improved its understandability.

	I disagree completely						**I agree completely**
Please check	1	2	3	4	5	6	7

d) The delineation of the subprocesses has been done properly in the process model as depicted.

	I disagree completely						**I agree completely**
Please check	1	2	3	4	5	6	7

Anhang 2: The Effect of Coupling in EPC Models

http://pc58397.uni-regensburg.de/ECIS17/Coupling/

The Effect of Coupling in EPC Models

A Laboratory Experiment

The submission Coupling in EPC models presents results from an experiment on the relationship of coupling, coupling metrics and the chance of misunderstanding process models as a result of high coupling. The submission provides context and interpretation. This page presents the experiment's details, the setup and results.
Enjoy!

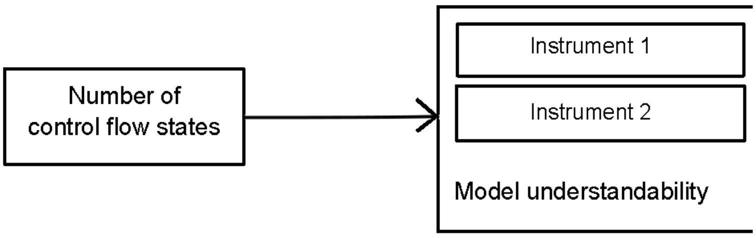
The author wants to thank the Institute for Information Systems (IWi) at the German Research Center for Artificial Intelligence (DFKI) and Saarland University (UdS) for providing the RefMod Miner as a Service (http://rmm.dfki.de/), which helped greatly to achieve the following results.

Hypotheses and Research Model

Research Question A

Hypothesis A

Ha0: A higher number of control flow states does not decrease the understandability of the process model.
Ha1: A higher number of control flow states does decrease the understandability of the process model.

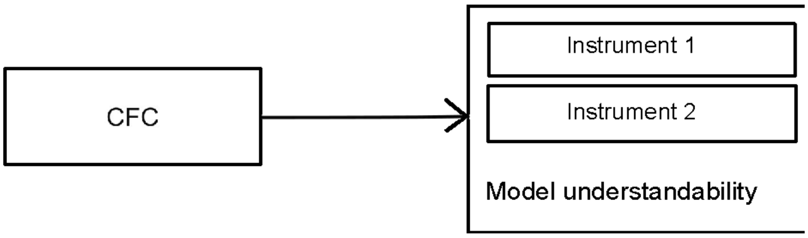


Research Question B

Hypothesis B

Hb0: A higher value of the CFC metric does not decrease the understandability of the process model.

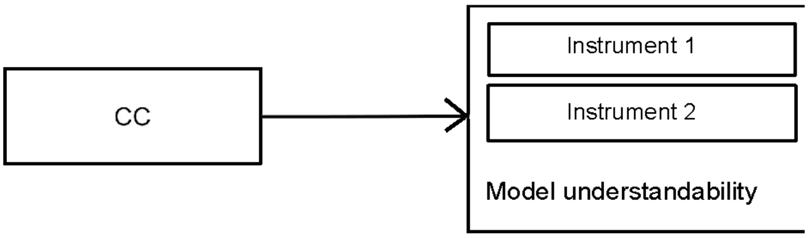
Hb1: A higher value of the CFC metric does decrease the understandability of the process model.



Hypothesis C

Hc0: A higher value of the CC metric does not decrease the understandability of the process model.

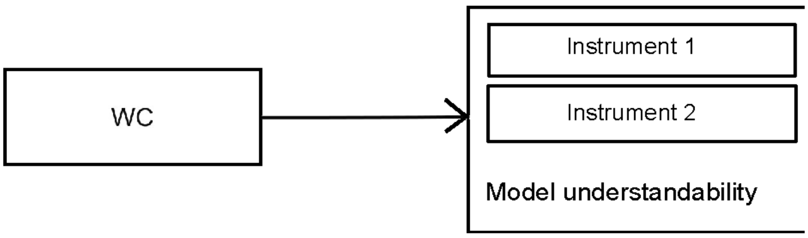
Hc1: A higher value of the CC metric does decrease the understandability of the process model.



Hypothesis D

Hd0: A higher value of the WC metric does not decrease the understandability of the process model.

Hd1: A higher value of the WC metric does decrease the understandability of the process model.



Research Question C

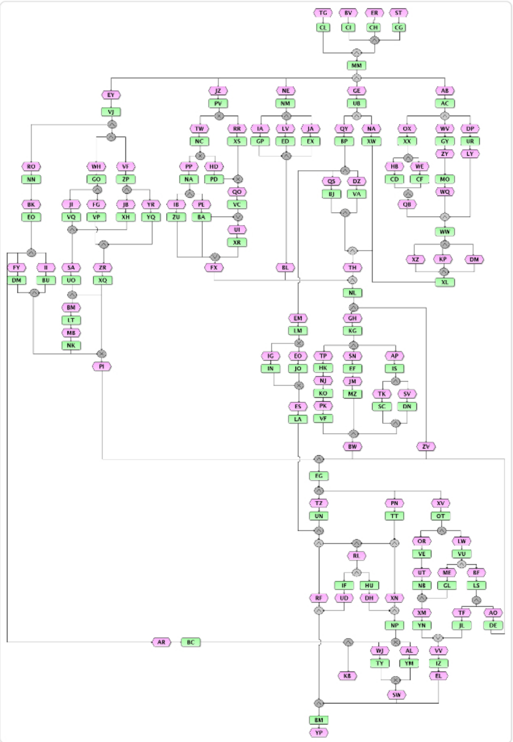
Research question C asks about the differences of the metrics. For this research question, we do not state testable hypotheses, instead we discuss the differences of the metrics performance with respect to the instruments.

Materials and Models

Process Models

Process Model 1

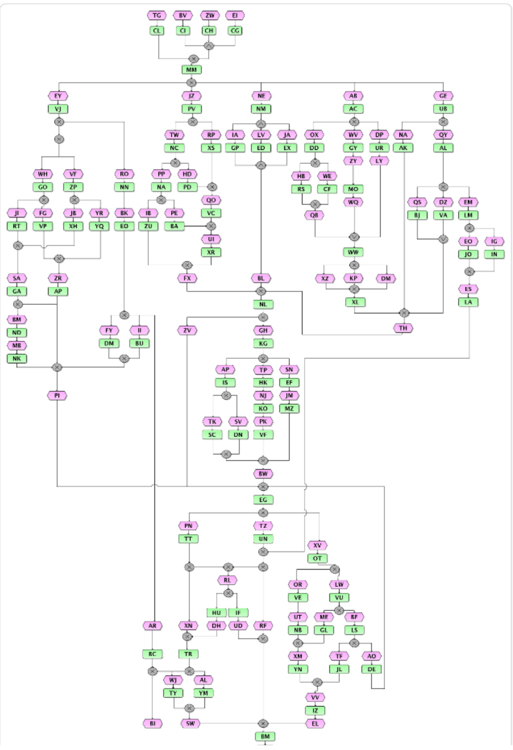
Start Events	4
Internal Events	87
End Events	2
All Events	93
Functions	81
AND splits	24
AND joins	22
XOR splits	4
XOR joins	4
OR splits	0
OR joins	3
Connectors	57
All Nodes	231
All arcs	269



(downloads/Modell_1.jpeg)

Process Model 2

Start Events	4
Internal Events	87
End Events	2
All Events	93
Functions	81
AND splits	1
AND joins	2
XOR splits	27
XOR joins	25
OR splits	0
OR joins	2
Connectors	57
All Nodes	231

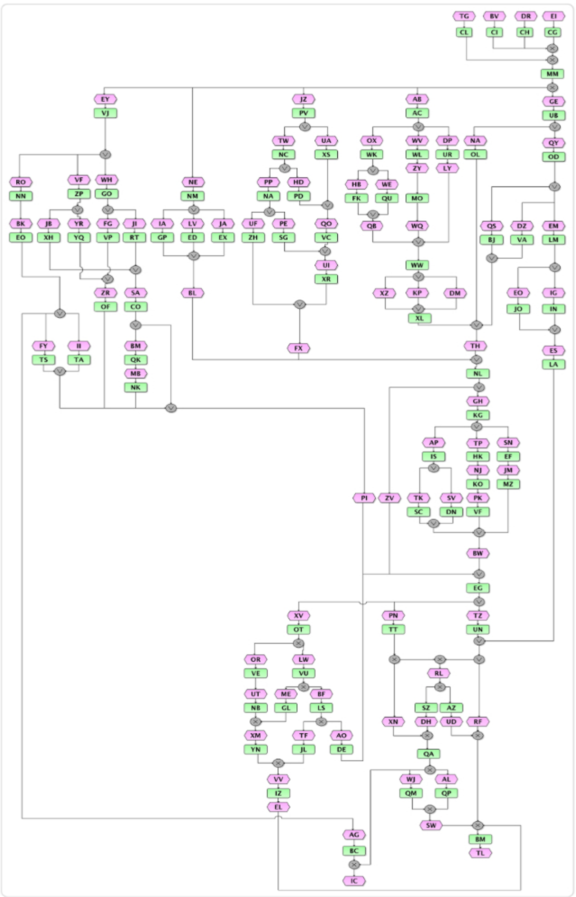


All arcs	268

(downloads/Modell_2.jpeg)

Process Model 3

Start Events	4
Internal Events	87
End Events	2
All Events	93
Functions	81
AND splits	0
AND joins	0
XOR splits	7
XOR joins	10
OR splits	20
OR joins	19
Connectors	56
All Nodes	230
All arcs	267

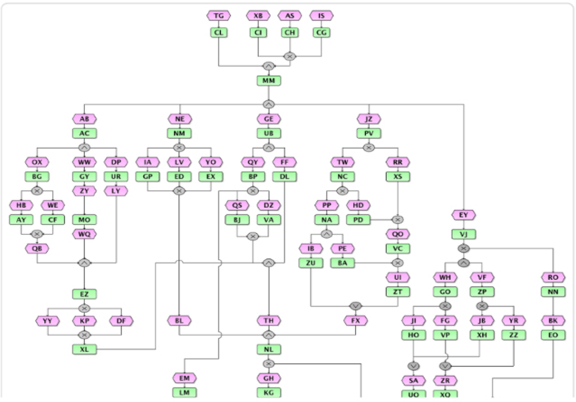


(downloads/Modell_3.jpeg)

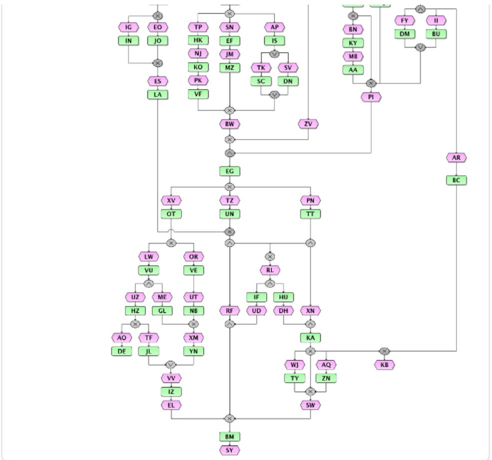
Process Model 0

Process Model 0 was not used as treatment. Instead, the model was used to verify that participants understood the tasks. Participants were dropped if results for this model indicated so.

Start Events	4
Internal Events	87
End Events	2
All Events	93
Functions	81
AND splits	10



AND joins	7
XOR splits	17
XOR joins	17
OR splits	1
OR joins	6
Connectors	58
All Nodes	232
All arcs	269

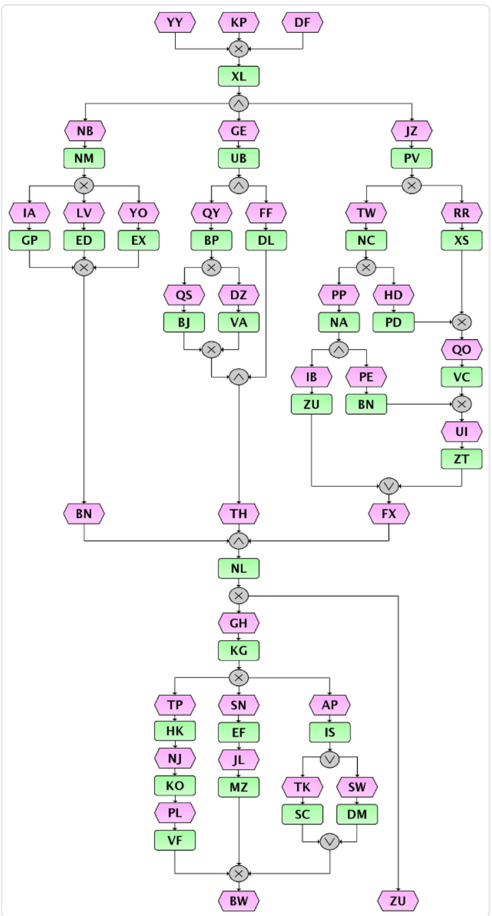


(downloads/Modell_0.jpeg)

Process Model 4

Process Model 4 was not used as treatment. Instead, the model was used to verify that participants understood the tasks. Participants were dropped if results for this model indicated so.

Start Events	3
Internal Events	30
End Events	2
All Events	35
Functions	29
AND splits	3
AND joins	2
XOR splits	6
XOR joins	6
OR splits	1
OR joins	2
Connectors	20
All Nodes	84
All arcs	95



(downloads/Modell_4.jpeg)

Metric values

Metric	validation only Model 0	Model 1	Model 2	Model 3	validation only Model 4
density_1	0,005	0,005	0,005	0,005	0,014
density_2	0,037	0,039	0,038	0,04	0,072
coefficient_of_connectivity	1,159	1,165	1,16	1,161	1,131
coefficient_of_network_complexity	311,901	313,251	310,926	309,952	107,44
cyclomatic_number	38	39	38	38	12
avg_connector_degree	3,397	3,421	3,404	3,429	3,35
max_connector_degree	6	6	6	6	4
separability	0,057	0,048	0,048	0,048	0,122
sequentiality	0,316	0,312	0,317	0,318	0,316
depth	5	4	4	4	3
mismatch	19	12	13	4	5
heterogeneity	0,845	0,549	0,324	0,559	0,853
token_splits	16	34	2	28	5
diameter	60	59	59	59	33
join_complexity	67	45	71	127	23
CFC Metric	54	33	65	110	20
WC Metric	0,005	0,006	0,004	0,005	0,013
CC Metric	0,014	0,188	0,006	0,006	0,032

These results were calculated by the RefMod-Miner as a Service (http://rmm.dfki.de/) at the Institute for Information Systems (IWi) at the German Research Center for Artificial Intelligence (DFKI) and Saarland University (UdS).

Download the models

AML	EPML
Model 0 (downloads/Modell_aml0.aml)	Model 0 (downloads/Modell0.epml)
Model 1 (downloads/Modell_aml1.aml)	Model 1 (downloads/Modell1.epml)

Model 2 (downloads/Modell_aml_2.xml)	Model 2 (downloads/Modell2.epml)
Model 3 (downloads/Modell_aml_3.xml)	Model 3 (downloads/Modell3.epml)
Model 4 (downloads/Modell_aml_4.xml)	Model 4 (downloads/Modell4.epml)

Questionnaire

Structure of the questionnaire

Introduction

- Goal & Structure of the experiment
- Repetition of EPC notation and connectors in particular
- Instructions regarding questions and tasks
- Examples of questions and tasks and how to fill the questionnaire

Background Information

Personal Information

- Current semester and previous studies
- Modelling experience

Modelling knowledge

- 12 questions on the syntax of EPC models

Verification 1 fixed position

Coupling

- Model 0
- 93 Events 81 Functions 34 XOR 7 OR 17 AND 54 Control Flow States

Understandability

- Effectiveness (1): 8 Questions, Duration
- Effectiveness (2): 2 Tasks, Duration

Control

- Subjective rating of model difficulty

In the experiment, the order of the three treatments is randomized, so that six versions of the questionnaire exist. They were distributed in the same strength, and differed only by the order of the treatments and respective process models.

The example here shows the order 1-2-3. So for first treatment covers Model 1 and the respective questions. Treatment two covers Model 2 and its questions and treatment three covers model three and its questions.

Subjective rating of task difficulty

Treatment 1 random position

Coupling

Model 1

93 Events 81 Functions 17 XOR 3 OR 46 AND 33 Control Flow States

Understandability

Effectiveness (1): 8 Questions, Duration

Effectiveness (2): 2 Tasks, Duration

Control

Subjective rating of model difficulty

Subjective rating of task difficulty

Treatment 2 random position

Coupling

Model 2

93 Events 81 Functions 52 XOR 2 OR 3 AND 65 Control Flow States

Understandability

Effectiveness (1): 8 Questions, Duration

Effectiveness (2): 2 Tasks, Duration

Control

Subjective rating of model difficulty

Subjective rating of task difficulty

Treatment 3 random position

Coupling

Model 3

93 Events 81 Functions 17 XOR 39 OR 0 AND 110 Control Flow States

Understandability

Effectiveness (1): 8 Questions, Duration

Effectiveness (2): 2 Tasks, Duration

Control

Subjective rating of model difficulty

Subjective rating of task difficulty

Verification 2fixed

Coupling

Model 4

36 Events 29 Functions 12 XOR 3 OR 5 AND 20 Control Flow States

Understandability

Effectiveness (1): 8 Questions, Duration

Effectiveness (2): 2 Tasks, Duration

Control

Subjective rating of model difficulty

Subjective rating of task difficulty

Verification 3fixed

Questions about the subject's experience in the experiment

Did you concentration decrease?

Did you have problems with particular tasks or questions?

Which models were particularly difficult?

What else did you not like when answering?

Download the Questionnaires

Questionnaire 1-2-3 (downloads/Fragebogen1-2-3.pdf)

Questionnaire 1-3-2 (downloads/Fragebogen1-3-2.pdf)

Questionnaire 2-1-3

The downloads are available in German since the original experiment was conducted with German speaking participants. An English translation can be made available upon (downloads/Fragebogen2-1-3.pdf) request.

Questionnaire 2-3-1 (downloads/Fragebogen2-3-1.pdf)

Questionnaire 3-1-2 (downloads/Fragebogen3-1-2.pdf)

Questionnaire 3-2-1 (downloads/Fragebogen3-2-1.pdf)

Statistical Analysis and Hypotheses Tests

Pearson Correlation between WC, CFC, and CC

	-(Z CFC)	**(Z WC)**	**(Z CC)**
-(Z CFC)	1		0.813**
(Z WC)	0.702**	1	0.985**
(Z CC)			1

Sig: 0.001 (2-tail)

The table shows the correlation between the metrics. For its calculation we centralized the values and changed the sign of the CFC, so that it runs in the same direction as the other two.

R-Values for the Coupling Metrics and both instruments

	Instrument 1	**Instrument 2**
WC Metric	0.723	0.466
CC Metric	0.710	0.509
CFC Metric	0.491	0.536

The table shows the relationship with both the instruments treatments. The R-Values show how much of the variance in one variable is explained by the other variable, or in other words, whether, if one variable increases by a certain fraction, the second variable increases by the same fraction. A higher R-Value makes an instrument a more reliable predictor.

Hypothesis Tests

Hypothesis Test for Ha

Dependent Variable: Understanding Effectiveness for Questions

	Unstandardized Coefficients		Standardized Coefficients			95.0% Confidence Interval for B	
Model	**B**	**Standard Error**	**Beta**	**t**	**Sig.**	**Lower B**	**Upper B**
(Constant)	0.720	0.031		23.320	0.000	0.658	0.781
Number of control flow states	-0.002	0.000	-0.491	-5.406	0.000	-0.003	-0.001

Dependent Variable: Understanding Effectiveness for Tasks

	Unstandardized Coefficients		Standardized Coefficients			95.0% Confidence Interval for B	
Model	**B**	**Standard Error**	**Beta**	**t**	**Sig.**	**Lower B**	**Upper B**
(Constant)	1.031	0.047		21.828	0.000	0.937	1.125
Number of control flow states	-0.004	0.001	-0.536	-6.440	0.000	-0.005	-0.003

Hypothesis Test for Hb

Dependent Variable: Understanding Effectiveness for Questions

	Unstandardized Coefficients		Standardized Coefficients			95.0% Confidence Interval for B	
Model	**B**	**Standard Error**	**Beta**	**t**	**Sig.**	**Lower B**	**Upper B**
(Constant)	0.720	0.031		23.320	0.000	0.658	0.781
CFC Metric	-0.002	0.000	-0.491	-5.406	0.000	-0.003	-0.001

Dependent Variable: Understanding Effectiveness for Tasks

	Unstandardized Coefficients		Standardized Coefficients			95.0% Confidence Interval for B	
Model	B	Standard Error	Beta	t	Sig.	Lower B	Upper B
(Constant)	0.660	0.025		26.037	0.000	0.610	0.711
CC Value	1.532	0.255	0.509	6.005	0.000	1.026	2.038

Literaturverzeichnis

Adam, Sebastian; Riegel, Norman; Imal, Selcuk; Jeswein, Thomas; Koch, Matthias (2013): Studie – BPM Suites 2013. Hg. v. Fraunhofer IESE. Fraunhofer IESE. Kaiserslautern (IESE, 075.013/D). Online verfügbar unter https://www.iese.fraunhofer.de/content/dam/iese/de/dokumente/oeffentliche_studien/Fraunhofer_IESE_Studie_BPM-Suites2013.pdf, zuletzt geprüft am 19.12.2016.

Aguilar-Savén, Ruth Sara (2004): Business process modeling. In: *International Journal of Production Economics* 90 (2), S. 129–149.

Aier, Stephan; Riege, Christian; Winter, Robert (2008): Unternehmensarchitektur – Literaturüberblick und Stand der Praxis. In: *WIRTSCHAFTSINFORMATIK* 50 (4), S. 292–304.

Aier, Stephan; Schoenherr, Marten (2004): Flexibilisierung von Organisations- und IT-Architekturen durch EAI. In: Stephan Aier (Hg.): Enterprise application integration. Flexibilisierung komplexer Unternehmensarchitekturen. Berlin: GITO-Verl. (Enterprise architecture, 1), S. 1–59. Online verfügbar unter https://www.researchgate.net/profile/Marten_Schoenherr/publication/44939293_Flexibilisierung_von_Organisations-_und_IT-Architekturen_durch_EAI/links/0deec5284a5a18b003000000.pdf.

Aier, Stephan; Schönherr, Marten (2006): Status quo geschäftsprozessorientierter Architekturintegration. In: *WIRTSCHAFTSINFORMATIK* 48 (3), S. 188–197.

Allen, Edward B.; Khoshgoftaar, Taghi M.; Chen, Ye (1999): Measuring coupling and cohesion. An information-theory approach. In: 6th Int. Software Metrics Symposium. METRICS. Boca Raton, USA, 04.11. IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 119–127.

Allen, Edward B.; Khoshgoftaar, Taghi M.; Chen, Ye (2001): Measuring Coupling and Cohesion of Software Modules. An Information-Theory Approach. In: 7th Int. Software Metrics Symposium. METRICS. London, UK, 04.04. IEEE: IEEE Comput. Soc. Press.

Allier, Simon; Vaucher, Stephane; Dufour, Bruno; Sahraoui, Houari (2010): Deriving Coupling Metrics from Call Graphs. In: 10th Working Conf. on Source Code Analysis and Manipulation. SCAM. Timisoara, Rumania, 12.- 13.09. IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 43–52.

Armstrong, Deborah J. (2006): The quarks of object-oriented development. In: *Communications of the ACM* 49 (2), S. 123–128.

Arshad, Fahad; Khanna, Gunjan; Laguna, Ignacio; Bagchi, Saurabh (2007): Distributed Diagnosis of Failures in a Three Tier E-Commerce System. Purdue University (ECE Tech. Reports, 354).

Barros, Oscar (2007): Business Process Architecture and Design. Hg. v. BPTrends. Online verfügbar unter http://www.bptrends.com/publicationfiles/05-07-ART-Business %20Processes%20and%20Design-Barros.pdf, zuletzt geprüft am 19.12.2016.

Basili, Victor R (1992): Software modeling and measurement. The Goal/Question/Metric paradigm. Hg. v. University of Maryland at College Park. University of Maryland at College Park. College Park, MD, USA, zuletzt geprüft am 20.04.2014.

Basili, Victor R; Caldiera, Gianluigi; Rombach, H. Dieter (1994): Goal Question Metric (GQM) Approach. In: John J. Marciniak (Hg.): Encyclopedia of software engineering. New York: Wiley.

Basu, Amit; Blanning, Robert W. (2003): Synthesis and Decomposition of Processes in Organizations. In: *Information Systems Research* 14 (4), S. 337–355.

Becker, Jörg (Hg.) (2000): Prozessmanagement. Ein Leitfaden zur prozessorientierten Organisationsgestaltung ; mit 34 Tabellen. 2. Aufl. Berlin et al.: Springer.

Becker, Jörg; Rosemann, Michael; Uthmann, Christoph von (2000): Guidelines of Business Process Modeling. In: Wil M. P. van der Aalst, Jörg Desel und Andreas Oberweis (Hg.): Business Process Management. Models, Techniques, and Empirical Studies, Bd. 1806. Berlin et al.: Springer (LNCS, 1806), S. 241–262.

Becker, Jörg; Thome, Irina; Weiß, Burkhard; Winkelmann, Axel (2010): Constructing a Semantic Business Process Modelling Language for the Banking Sector. In: *Enterprise Modeling and Information Systems Architectures* 5 (1), S. 4–25.

Beyer, Dirk; Lewerentz, Claus; Simon, Frank (2001): Impact of Inheritance on Metrics for Size, Coupling, and Cohesion in Object-Oriented Systems. In: Reiner Dumke und Alain Abran (Hg.): Proc. of 10th Int. Workshop on Software Measurement. New approaches in software measurement. IWSM. Berlin, 04.10- 06.10. Berlin: Springer (LNCS, 2006), S. 1–17.

Binkley, A.B; Schach, S.R (1998): Validation of the coupling dependency metric as a predictor of run-time failures and maintenance measures. In: Koji Torii, Kokichi Futatsugi und Richard A. Kemmerer (Hg.): Proceedings of the 1998 Int. Conf. on Software Engineering. Forging New Links. ICSE. Kyoto, Japan, 19.04-25.04. IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 452–455. Online verfügbar unter http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=671604.

Birkmeier, Dominik (2010): On the State of the Art of Coupling and Cohesion Measures for Service-Oriented System Design. In: Martin Santana, Jerry N. Luftman und Ajay S. Vinze (Hg.): 16th Americas Conf. on Information Systems. Sustainable IT Collaboration Around the Globe. ACIS. Lima, Peru, 12.08-15.08. AIS: AISeL. Online verfügbar unter http://aisel.aisnet.org/amcis2010/296.

Bobrik, Ralph; Reichert, Manfred; Bauer, Thomas (2007): View-Based Process Visualization. In: Gustavo Alonso, Peter Dadam und Michael Rosemann (Hg.): Business Process Management, Bd. 4714. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 88–95.

Bodart, François; Patel, Arvind; Sim, Marc; Weber, Ron (2001): Should Optional Properties Be Used in Conceptual Modelling? A Theory and Three Empirical Tests. In: *Information Systems Research* 12 (4), S. 384–405.

Branco, Moisés Castelo; Xiong, Yingfei; Czarnecki, Krzysztof; Küster, Jochen; Völzer, Hagen (2014): A case study on consistency management of business and IT process models in banking. In: *Software & Systems Modeling* 13 (3), S. 913–940.

Braunnagel, Daniel; Johannsen, Florian (2013): Coupling Metrics for EPC Models. In: Rainer Alt und Bogdan Franczyk (Hg.): Int. Conf. on Wirtschaftsinformatik // Proceedings of the 11th International Conference on Wirtschaftsinformatik (WI2013). Feb. 27 - Mar. 01.2013, Leipzig, Germany. WI2013. Leipzig, 27.02 - 01.03.2013. Internationale Tagung Wirtschaftsinformatik; Internationale Conference on Wirtschaftsinformatik; WI 2013. Leipzig: AISeL, S. 1797–1811.

Braunnagel, Daniel; Johannsen, Florian; Leist, Susanne (2014): Coupling and process modelling. An Analysis at hand of the eEPC. In: Hans-Georg Fill, Dimitris Karagiannis und Ulrich Reimer (Hg.): Proceedings of Modellierung 2014. Modellierung. Wien, Österreich, 19.03 - 21.03. Gesellschaft für Informatik e.V. Bonn: Köllen (LNI, P-225), S. 121–137.

Braunnagel, Daniel; Johannsen, Florian; Leist, Susanne (2015): Analysing the contribution of coupling metrics for the development and management of process architectures. In: Jörg Becker, Jan Vom Brocke und Marco de Marco (Hg.): Proceedings of the European Conference on Information Systems // ECIS 2015 completed reseaech papers. ECIS. Münster, May 26-29. ERCIS; ECIS. AIS Electronic Library, S. 19.

Briand, Lionel C.; Daly, J.W; Porter, V.; Wust, J.K (1998): A comprehensive empirical validation of design measures for object-oriented systems. In: 5th Int. Software Metrics Symposium. METRICS. Bethesda, USA, 20.03- 21.03. IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 246–257. Online verfügbar unter http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=731251.

Briand, Lionel C.; Daly, J.W; Wust, J.K (1999): A unified framework for coupling measurement in object-oriented systems. In: *IEEE Transactions on Software Engineering* 25 (1), S. 91–121. Online verfügbar unter http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=748920.

Briand, Lionel C.; Devanbu, Premkumar T.; Melo, Walcélio L. (1997): An Investigation into Coupling Measures for C++. In: Richards W. Adrion, Alfonso Fuggetta, Richard N. Taylor und Anthony I. Wasserman (Hg.): Proc. of the 19th Int. Conf. on Software Engineering. Pulling together. ICSE. Boston, USA, 17.05.1997 - 23.05.1997: ACM Press, S. 412–432.

Briand, Lionel C.; Wust, J. (2001): The impact of design properties on development cost in object-oriented systems. In: 7th Int. Software Metrics Symposium. METRICS. London, UK, 04.04. IEEE: IEEE Comput. Soc. Press, S. 260–271.

Brucker-Kley, Elke; Kykalova, Denisa; Pedron, Claudia; Luternauer, Theresa; Keller, Thomas (2014): Business Process Management 2014. Status quo und Perspektiven eines ganzheitlichen Geschäftsprozessmanagements : Ergebnisse einer branchenübergreifenden empirischen Studie + Themenfokus "Business Process Management in der Immobilienwirtschaft". Zürich. Online verfügbar unter https://digitalcollection.zhaw.ch/bitstream/11475/1027/1/3675_Business-Process-Management-2014_OA.pdf.

Buhl, H. U.; Heinrich, B. (2004): Unternehmensarchitekturen in der Praxis. Architekturdesign am Reißbrett vs. situationsbedingte Realisierung von Informationssystemen. In: *WIRTSCHAFTSINFORMATIK* 46 (4), S. 311. Online verfügbar unter http://epub.uni-regensburg.de/23201/1/heinrich.pdf.

Burton-Jones, Andrew; Meso, Peter (2002): How Good Are These UML Diagrams? An Empirical Test of the Wand and Weber Good Decomposition Model. In: Francesc MIralles und Josep Valor (Hg.): Proceedings of the International Conference on Information Systems. ICIS. Barcelona, Spain, December 15-18. AIS: AISeL. Online verfügbar unter http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1102&context=icis2002.

Burton-Jones, Andrew; Meso, Peter N. (2006): Conceptualizing Systems for Understanding. An Empirical Test of Decomposition Principles in Object-Oriented Analysis. In: *Information Systems Research* 17 (1), S. 38–60. Online verfügbar unter http://pubsonline.informs.org/doi/pdf/10.1287/isre.1050.0079.

Burton-Jones, Andrew; Meso, Peter N. (2008): The Effects of Decomposition Quality and Multiple Forms of Information on Novices' Understanding of a Domain from a Conceptual Model. In: *Journal of the Association for Information Systems* 9 (12), S. 748–802. Online verfügbar unter http://content.ebscohost.com/ContentServer.asp?T=P&P=AN&K=40623831&S=R&D=buh&EbscoContent=dGJyMNXb4kSep7I4wtvhOLCmr06ep69Ssau4S7GWxWXS&ContentCustomer=dGJyMPGrt0y1rLVMuePfgeyx44Dt6fIA.

2011: Business Process Model and Notation™ (BPMN™). Online verfügbar unter http://www.omg.org/spec/BPMN/2.0/.

Cardoso, Jorge (2005): Evaluating the process control-flow complexity measure. In: IEEE Int. conf. on web services. ICWS. Orlando, USA, 12.07- 15.07. IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 803–804. Online verfügbar unter http://eden.dei.uc.pt/~jcardoso/Research/Papers/Old%20paper%20format/IEEE-ICWS-05-CFC-Weyker-Properties.pdf.

Cardoso, Jorge (2006): Process control-flow complexity metric: An empirical validation. In: IEEE International conference on Services Computing. SCC, 18. - 22.09. IEEE: IEEE Comput. Soc. Press, S. 167–173.

Cardoso, Jorge; Mendling, Jan; Neumann, Gustaf; Reijers, Hajo A. (2006): A Discourse on Complexity of Process Models. In: Johann Eder und Schahram Dustdar (Hg.): Business process management workshops. BPM 2006 International Workshops, BPD, BPI, ENEI, GPWW, DPM, semantics4ws. BPM. Wien, Österreich, 04.09. Berlin: Springer (LNCS, 4103), S. 117–128.

Chen, Jianguo; Yeap, Wai K.; Bruda, Stefan D. (2009): A Review of Component Coupling Metrics for Component-Based Development. In: WRI World Congress on Software Engineering. WCSE. Xiamen, China, 19.05.2009 - 21.05.2009. WRI. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 65–69.

Chidamber, Shyam R.; Kemerer, Chris F. (1991): Towards a metrics suite for object oriented design. In: *ACM SIGPLAN Notices* 26 (11), S. 197–211.

Chidamber, Shyam R.; Kemerer, Chris F. (1994): A Metrics Suite for Object Oriented Design. In: *IEEE Transactions on Software Engineering* 20 (6), S. 476–493. Online verfügbar unter http://web.cs.wpi.edu/~gpollice/cs562-s05/Readings/CKMetrics.pdf.

Cho, Eun Sook; Kim, Chul Jin; Kim, Soo Dong; Rhew, Sung Yul (1998): Static and Dynamic Metrics for Effective Object Clustering. In: 5th Asia-Pacific Software Engineering Conf. APSEC. Taipei, Taiwan, 02.12-04.12. IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 87.

Chong, Sandy (2007): Business process management for SMEs. an exploratory study of implementation factors for the Australian wine industry. In: *Journal of Information Systems and Small Business* 1 (1-2), S. 41–58.

Chowdhury, Istehad; Zulkernine, Mohammad (2010): Can complexity, coupling, and cohesion metrics be used as early indicators of vulnerabilities? In: Sung Y. Shin, Sascha Ossowski, Michael Schumacher, Mathew J. Palakal, Chih-Cheng Hung, Istehad Chowdhury und Mohammad Zulkernine (Hg.): Proc. o. Symposium on Applied Computing. SAC. Sierre, Schweiz, 22.-26.03. ACM: ACM Press, S. 1963–1969.

Cooper, Harris M. (2006): Synthesizing research. A guide for literature reviews. 3. Aufl. Thousand Oaks, USA: SAGE (Applied social research methods series, 2).

Dallas, Ian; Wynn, Moe Thandar (2014): Business Process Management in Small Business: A Case Study. In: Jan Devos, Hendrik van Landeghem und Dirk Deschoolmeester (Hg.): Information Systems for Small and Medium-sized Enterprises. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 25–46.

Davenport, Thomas H. (1993): Process innovation. Reengineering work through information technology. Boston, Mass.: Harvard Business School Press.

Davis, D. (2013): 3rd Biennial PEX Network Report: State of the Industry. Trends and Success Fac-tors in Business Process Excellence. Process Excellence Network. Online verfügbar unter http://www.processexcellencenetwork.com/ business-transformation/white-papers/trends-and-success-factors-in-business-process-ex/.

Davis, Fred D. (1989): Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. In: *MIS Quarterly* 13 (3), S. 319.

Davis, Rob (2008): ARIS Design Platform. Advances Process Modelling and Administration. London: Springer London.

Davis, Rob; Brabänder, Eric (2007): ARIS design platform. Getting started with BPM. New York, London: Springer.

DeMarco, Tom; Boehm, Barry W. (1982): Controlling software projects. Englewood Cliffs, NJ: Yourdon Press (Yourdon Press computing series).

DESTATIS (2015): Report no. 52111-003. DESTATIS Genesis-Online. Online verfügbar unter https://www-genesis.destatis.de/genesis/online/link/tabellen/52111-0003, zuletzt geprüft am 30.03.2016.

Dongen, Boudewijn F. van; Medeiros, A.K.A de; Verbeek, H.M.W; Weijters, A.J.M.M; van der Aalst, Wil M. P. (2005): The ProM Framework. A New Era in Process Mining Tool Support. In: Gianfranco Ciardo und Philippe Darondeau (Hg.): 26th Int. Conf. on Applications and Theory of Petri Nets. ICATPN. Miami, USA, 20.06 - 25.06. Berlin: Springer (LNCS, 3536), S. 444–454. Online verfügbar unter http://link.springer.com/chapter/10.1007/11494744_25.

Dromey, R. Geoff (1996): Cornering the Chimera. In: *IEEE Software* 13 (1), S. 33–43.

El-Emam, K. (2001): Object-Oriented Metrics: A Review of Theory and Practice. NRC-CNRC (NRC, 44190).

Elliott, Robert; Fischer, Constance, T.; Rennie, David, L. (1999): Evolving guidelines for publication of qualitative research studies in psychology and related fields. In: *British Journal of Clinical Psychology* (38), S. 215–229.

Evermann, Joerg; Wand, Yair (2005): Ontology based object-oriented domain modelling: fundamental concepts. In: *Requirements Engineering* 10 (2), S. 146–160. Online verfügbar unter http://www.springerlink.com/content/y1klf42mp8dheafw/fulltext.pdf.

Evermann, Joerg; Wand, Yair (2009): Ontology Based Object-Oriented Domain Modeling. Representing Behavior. In: *Journal of Database Management* 20 (1), S. 48–77.

Falk, Thomas; Griesberger, Philipp; Leist, Susanne (2013): Patterns as an Artifact for Business Process Improvement - Insights from a Case Study. In: David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Hg.): Design Science at the Intersection of Physical and Virtual Design, Bd. 7939. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 88–104.

Fenton, Norman E.; Pfleeger, Shari Lawrence (1998): Software metrics. A rigorous and practical approach. 2. Aufl. Boston: PWS Publ.

Ferstl, Otto K.; Sinz, Elmar J. (2006): Modeling of Business Systems Using SOM. In: Peter Bernus, Kai Mertins und Günter Schmidt (Hg.): Handbook on Architectures of Information Systems. 2. Aufl. Berlin/Heidelberg: Springer-Verlag (International handbooks on information systems), S. 347–367.

Fettke, Peter; Houy, Constantin; Vella, Armella-Lucia; Loos, Peter (2012): Towards the Reconstruction and Evaluation of Conceptual Model Quality Discourses – Methodical Framework and Application in the Context of Model Understandability. In: Wil M. P. van der Aalst, John Mylopoulos, Michael Rosemann, Michael J. Shaw, Clemens Szyperski, Ilia Bider et al. (Hg.): Enterprise, Business-Process and Information Systems Modeling, Bd. 113. Berlin, Heidelberg: Springer (LNBIP), S. 406–421.

Fill, Hans-Georg; Johannsen, Florian (2016): A Knowledge Perspective on Big Data by Joining Enterprise Modeling and Data Analyses. In: IEEE (Hg.): 49th Hawaii International Conference on System Sciences. HICSS. Koloa, HI, USA, 5-8 Jan. IEEE.

Fischer, Christian; Gregor, Shirley Diane (2011): Forms of Reasoning in the Design Science Research Process. In: Hemant Jain, Atish P. Sinha und Padmal Vitharana (Hg.): Service-Oriented Perspectives in Design Science Research. 6th International Conference, Bd. 6629. DESRIST. Milwaukee, WI, USA, May 5-6. Berlin, Heidelberg: Springer (LNCS, 6629), S. 17–31.

Flood, Robert L.; Carson, Ewart R. (1993): Dealing with complexity. An introduction to the theory and application of systems science. 2. Aufl. New York: Plenum Press. Online verfügbar unter http://www.loc.gov/catdir/enhancements/fy0819/92040616-d.html.

Frank, Ulrich (1994): Multiperspektivische Unternehmensmodellierung. Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung. Zugl.: Marburg, Univ., Habil.-Schr., 1993. München, Wien: Oldenbourg (Berichte der Gesellschaft für Mathematik und Datenverarbeitung, 225). Online verfügbar unter https://www.wi-inf.uni-duisburg-essen.de/FGFrank/documents/Buecher/Habilitation.pdf.

Frank, Ulrich (1998): Evaluating Modelling Languages: Relevant Issues, Epistemological Challenges And A Preliminary Research Framework. Institut für Wirtschaftsinformatik, Universität Koblenz-Landau (Arbeitsberichte des Instituts für Wirtschaftsinformatik, 15). Online verfügbar unter http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.4836.

Gable, Guy G. (1994): Integrating case study and survey research methods. An example in information systems. In: *European Journal of Information Systems* 3 (2), S. 112–126.

Gemino, Andrew; Wand, Yair (2003): Evaluating modeling techniques based on models of learning. In: *Communications of the ACM* 46 (10), S. 79.

Gemino, Andrew; Wand, Yair (2004): A framework for empirical evaluation of conceptual modeling techniques. In: *Requirements Engineering* 9 (4), S. 248–260.

Gemino, Andrew; Wand, Yair (2005): Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties. In: *Quality in conceptual modelingFive examples of the state of artThe International Workshop on Conceptual Modeling Quality 2002 and 2003* 55 (3), S. 301–326.

Gericke, Anke (2009): Konstruktion situativer Artefakte. Beiträge zur Konstruktionsforschung und zur Artefaktkonstruktion. Dissertation. Universität St. Gallen Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften (HSG), St. Gallen. Institut für Wirtschaftsinformatik, zuletzt geprüft am 27.05.2014.

Gerth, Christian (2013): Business Process Models. Change Management. Berlin Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science, 7849). Online verfügbar unter http://dx.doi.org/10.1007/978-3-642-38604-6.

Goes, Paulo B. (2013): Commonalities Across IS Silos and IntradisciplinaryInformation Systems Research. In: *MIS Quarterly* 37 (2), S. iii–vii.

González, Laura Sánchez; Rubio, Félix García; González, Francisco Ruiz; Velthuis, Mario Piattini (2010): Measurement in business processes: a systematic review. In: *Business Process Management Journal* 16 (1), S. 114–134.

Green, Pam; Lane, Peter C.; Rainer, Austen; Scholz, Sven Bodo (2009): An Introduction to Slice-Based Cohesion and Coupling Metrics. University of Hertfordshire (Technical Report, 488), zuletzt geprüft am 14.08.2012.

Green, Peter; Rosemann, Michael (2000): Integrated Process Modeling. An ontological evaluation. In: *Information Systems and e-Business Management* 25 (2), S. 73–87.

Green, Peter; Rosemann, Michael (2001): Ontological Analysis of Integrated Process Models. Testing hypotheses. In: *Australasian Journal of Information Systems* 9 (1).

Gregor, Shirley Diane; Jones, David (2007): The Anatomy of a Design Theory. In: *Journal of the Association for Information Systems* 8 (5), S. 2.

Griesberger, Philipp (2014): Developing the Evaluation of a Pattern-Based Approach for Business Process Improvement. In: Monica Chiarini Tremblay, Debra VanderMeer, Marcus Rothenberger, Ashish Gupta und Victoria Yoon (Hg.): Advancing the Impact of Design Science. Moving from Theory to Practice, Bd. 8463. Cham: Springer (LNCS), S. 225–240.

Gruhn, Volker; Laue, Ralf (2006a): Adopting the Cognitive Complexity Measure for Business Process Models. In: 5th IEEE Int. Conf. on Cognitive Informatics. ICCI. Beijing, 2006. IEEE: IEEE Comput. Soc. Press, S. 236–241.

Gruhn, Volker; Laue, Ralf (2006b): Complexity Metrics for Business Process Models. In: Witold Abramowicz und Mayr, Heinrich, C. (Hg.): Proc. o. 9th int. Conf. on BIS. BIS 2006. Klagenfurt, Austria, 31 May - 2 June. Bonn: GI (GI-Edition Proceedings, 85), S. 1–12.

Gruhn, Volker; Laue, Ralf (2007): Approaches for Business Process Model Complexity Metrics. In: Witold Abramowicz und Mayr, Heinrich, C. (Hg.): Technologies for Business Information Systems. Berlin: Springer, S. 13–24.

Gui, Gui; Scott, Paul D. (2006): Coupling and cohesion measures for evaluation of component reusability. In: Stephan Diehl, Harald Gall und Ahmed E. Hassan (Hg.): Proc. o. Int. Workshop on Mining Software Repositories. MSR. Shanghai, China, 22.-23.03. ACM. New York, NY: ACM Press, S. 18–21.

Gui, Gui; Scott, Paul D. (2008): New Coupling and Cohesion Metrics for Evaluation of Software Component Reusability. In: 9th Int. Conf. for Young Computer Scientists. ICYCS. Zhang Jia Jie, China, 18.11. IEEE. Los Alamitos, USA: IEEE, S. 1181–1186. Online verfügbar unter http://origin-www.computer.org/plugins/dl/pdf/proceedings/icycs/2008/3398/00/3398b181.pdf?template=1&loginState=2&userData=University%2BLibrary%2Bof%2BRegensburg%253AUniversity%2BLibrary%2Bof%2BRegensburg%253AAddress%253A%2B132.199.123.228%252C%2B%255B140.98.196.191%252C%2B195.95.193.155%252C%2B132.199.123.228%255D.

Halstead, Maurice H. (1977): Elements of software science. New York: Elsevier.

Hammer, Michael (2010): What is Business Process Management? In: Jan Vom Brocke und Michael Rosemann (Hg.): Introduction, methods and information systems. New York: Springer (Handbook on business process management, 1), S. 3–16.

Hammer, Michael (2015): What is Business Process Management? In: Jan Vom Brocke und Michael Rosemann (Hg.): Handbook on Business Process Management 1. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 3–16.

Hammer, Michael; Champy, James (2002): Reengineering the corporation. A manifesto for business revolution: PerfectBound.

Harmon, Paul (2010): The Scope and Evolution of Business Process Management. In: Jan Vom Brocke und Michael Rosemann (Hg.): Introduction, methods and information systems. New York: Springer (Handbook on business process management, 1), S. 37–81.

Harmon, Paul (2016): The State of Business Process Management 2016. Hg. v. BPTrends. BPTrends. Online verfügbar unter http://www.bptrends.com/bpt/wp-content/uploads/2015-BPT-Survey-Report.pdf, zuletzt geprüft am 20.12.2016.

Harmon, Paul; Wolf, Celia (2014): The State of Business Process Management. BPTrends. Online verfügbar unter http://www.bptrends.com/bpt/wp-content/uploads/BPTrends-State-of-BPM-Survey-Report.pdf, zuletzt geprüft am 23.04.2015.

Harrison, R.; Counsell, S.; Nithi, R. (1998): Coupling metrics for object-oriented design. In: 5th Int. Software Metrics Symposium. METRICS. Bethesda, USA, 20.03- 21.03. IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 150–157. Online verfügbar unter http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=731240.

Henderson-Sellers, Brian (1996): Object-oriented metrics. Measures of complexity. Upper Saddle River, N.J: Prentice-Hall (Prentice Hall object-oriented series).

Henry, Sallie; Kafura, Dennis (1981): Software Structure Metrics Based on Information Flow. In: *IEEE Transactions on Software Engineering* 7 (5), S. 510–518.

Hevner, Alan R; Chatterjee, Samir (2010): Design Science Research Frameworks. In: Alan R Hevner und Samir Chatterjee (Hg.): Design Research in Information Systems, Bd. 22. Boston, MA: Springer (Integrated Series in Information Systems), S. 23–31.

Hevner, Alan R; March, Salvatore T.; Park, Jinsoo; Ram, Sudha (2004): Design Science in Information Systems Research. In: *MIS Quarterly* 28 (1), S. 75–105.

Hinrichs, Holger (2002): Datenqualitätsmanagement in Data Warehouse-Systemen. Dissertation. Universität Oldenburg, Oldenburg.

Hitz, Martin; Montazeri, Behzad (1995): Measuring coupling and cohesion in object-oriented systems. In: Proc. of 3rd Int. Symposium on Applied Corporate Computing. ISACC. Monterrey, Mexico, 25.10.-27.10. Online verfügbar unter http://ubdocs.uni-klu.ac.at/open/voll/fodok/inf/AC06311767.pdf.

Hoffmann, W.; Wein, R.; Scheer, August-Wilhelm (1993): Konzeption eines Steuerungsmodells für Informationssysteme - Basis für die Real-Time-Erweiterungen der EPK (rEPK). In: August-Wilhelm Scheer (Hg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Bd. 106. Institut für Wirtschaftsinforamtik, Universität des Saarlandes. Saarbrücken (Veröffentlichungen des Instituts für Wirtschaftsinformatik). Online verfügbar unter http://www.uni-saarland.de/fileadmin/user_upload/Professoren/fr13_ProfLoos/IWi-Hefte/IWi-Heft_Nr._106.pdf, zuletzt geprüft am 19.12.2016.

Houy, Constantin; Fettke, Peter; Loos, Peter (2010): Empirical research in business process management – analysis of an emerging field of research. In: *Business Process Management Journal* 16 (4), S. 619–661.

Houy, Constantin; Fettke, Peter; Loos, Peter (2012): Understanding Understandability of Conceptual Models. What Are We Actually Talking About? In: Paolo Atzeni, David Cheung und Sudha Ram (Hg.): Conceptual Modeling. 31st Int. Conf. ER2012, Bd. 7532. ER. Florence, Italy, 15 - 18.10. Berlin, Heidelberg: Springer (LNCS, 7532), S. 64–77.

Houy, Constantin; Fettke, Peter; Loos, Peter (2014): On the theoretical foundations of research into the understandability of business process modells. In: Michel Avital, Jan Marco Leimeister und Ulrike Schultze (Hg.): ECIS 2014 proceedings. 22th European Conference on Information Systems ; Tel Aviv, Israel, June 9-11, 2014. AIS Electronic Library, S. 7.

Huysmans, Philip; Bruyn, Peter de (2013): A Mixed Methods Approach To Combining Behavioral And Design Research Methods In Information Systems Research. In: AIS (Hg.): 21st European Conference on Information Systems. ECIS. Utrecht, NL, June 5-8.

Iivari, Juhani (2007): A Paradigmatic Analysis of Information Systems As a Design Science. In: *Scandinavian Journal of Information Systems* 19 (2), S. 5. Online verfügbar unter http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1018&context=sjis.

Jeanneret, Cédric; Glinz, Martin; Baar, Thomas (2012): Modeling the purposes of models. In: Elmar J. Sinz und Andy Schürr (Hg.): Modellierung 2012. 14. - 16. März 2012, Bamberg. Bonn: Ges. für Informatik (GI-Edition Proceedings, 201), S. 11–26.

Jeanneret, Cédric; Glinz, Martin; Baudry, Benoit (2011): Estimating footprints of model operations. In: Richard N. Taylor, Harald Gall und Nenad Medvidović (Hg.): Proceeding of the 33rd international conference on software engineering. ICSE. Waikiki, Honolulu, HI, USA, May 21-28. New York, NY: ACM Press, S. 601. Online verfügbar unter http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.386.4361&rep=rep1&type=pdf.

Johannsen, Florian; Leist, Susanne (2012a): Das Dekompositionsmodell nach Wand und Weber im Kontext der Prozessmodellierung. In: *WIRTSCHAFTSINFORMATIK* 54 (5), S. 263–280.

Johannsen, Florian; Leist, Susanne (2012b): Wand and Weber’s Decomposition Model in the Context of Business Process Modeling. In: *Business & Information Systems Engineering* 4 (5), S. 271–286.

Johannsen, Florian; Leist, Susanne; Braunnagel, Daniel (2014a): Testing the Impact of Wand and Weber's Decomposition Model on Process Model Understandability. In: AIS (Hg.): Proc. o.t. ICIS. ICIS. Auckland, NZ, Dec 14-17. AIS: AISeL. Online verfügbar unter http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1436&context=icis2014.

Johannsen, Florian; Leist, Susanne; Tausch, Reinhold (2014b): Wand and Weber's good decomposition conditions for BPMN. In: *Business Process Management Journal* 20 (5), S. 693–729.

Joshi, P.; Joshi, R.K (2010): Microscopic coupling metrics for refactoring. In: 10th Working Conf. on Source Code Analysis and Manipulation. SCAM. Timisoara, Rumania, 12.-13.09. IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 145–152. Online verfügbar unter http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1602366.

Jung, E. (2005): Das IT-Architekturmodell. Ein Rahmenwerkzeug fuer die Prozessoptimierung in der HypoVereinsbank. In: *Banking and Information Technology* (6), S. 62–64.

Jung, Jae-Yoon; Chin, Chang-Ho; Cardoso, Jorge (2011): An entropy-based uncertainty measure of process models. In: *Information Processing Letters* 111 (3), S. 135–141.

Kawalek, Peter; Kueng, Peter (1997): The Usefulness of Process Models: A Lifecycle Description of how Process Models are used in Modern Organisations. In: Keng Siau, Yair Wand und Jeffrey Parsons (Hg.): Proceedings of the Second CAiSE/IFIP8.1. EMMSAD. Barcelona, Spanien, 16.06, S. 1–12.

Kazemi, Ali; Azizkandi, Ali Nasirzadeh; Rostampour, Ali; Haghighi, Hassan; Jamshidi, Pooyan; Shams, Fereidoon (2011): Measuring the Conceptual Coupling of Services Using

Latent Semantic Indexing. In: Hans-Arno Jacobsen, Yang Wang und Patrick Hung (Hg.): IEEE Int. Conf. on Services Computing. SCC. Washington, USA, 04.06-09.06. IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 504–511.

Keller, G.; Nüttgens, Markus; Scheer, August-Wilhelm (1992): Semantische Prozeßmodellierung auf Grundlage "Ereignisgesteuerter Prozeßketten (EPK)". In: August-Wilhelm Scheer (Hg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Bd. 89. Institut für Wirtschaftsinforamtik, Universität des Saarlandes. Saarbrücken (Veröffentlichungen des Instituts für Wirtschaftsinformatik).

Khatri, Vijay; Vessey, Iris; Ramesh, V.; Clay, Paul; Park, S-J. (2006): Understanding Conceptual Schemas: Exploring the Role of Application and IS Domain Knowledge. In: *Information Systems Research* 17 (1). Online verfügbar unter https://www.researchgate.net/profile/Iris_Vessey2/publication/45256307_Understanding_Conceptual_Schemas_Exploring_the_Role_of_Application_and_IS_Domain_Knowledge/links/0deec51ca6bea7c482000000.pdf.

Khlif, Wiem; Makni, Lobna; Zaaboub, Nahla; Ben-Abdalla, Hanene (2009): Quality metrics for business process modeling. In: Roberto Revertia, Valeri Mladenov und Nikos Mastorakis (Hg.): Proceedings of the 9th WSEAS Int. Conf. on ACS. WSEAS ACS'. Genua, Italien, 17.10.2009. World Scientific and Engineering Academy and Society. USA: WSEAS, S. 195–200.

Khlif, Wiem; Zaaboub, Nahla; Ben-Abdalla, Hanene (2010): Coupling metrics for business process modeling. In: *WSEAS TRANSACTIONS on COMPUTERS* 9 (1), S. 31–41.

Klecun, Ela; Cornford, Tony (2005): A critical approach to evaluation. In: *European Journal of Information Systems* 14 (3), S. 229–243.

Koch, Susanne (2011): Methoden des Prozessmanagements. In: Susanne Koch (Hg.): Einführung in das Management von Geschäftsprozessen. Six Sigma, Kaizen und TQM. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, S. 47–114.

Kohlbacher, Markus (2010): The effects of process orientation. A literature review. In: *Business Process Management Journal* 16 (1), S. 135–152.

Kolář, Jiří (2014): The Adoption of Business Process Management in Small and MediumEnterprises. PHD Thesis. Masarykova Univerzita, Brün. Online verfügbar unter http://is.muni.cz/th/72655/fi_d/dissertation_draft-4.0.2_vbox_typo_fixes_FINAL.pdf, zuletzt geprüft am 16.12.2016.

Komus, Ayelt (2011): BPM Best Practice. Wie führende Unternehmen ihre Geschäftsprozesse managen. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg. Online verfügbar unter http://download-v2.springer.com/static/pdf/126/bok%253A978-3-642-16725-6.pdf?token2=exp=1431699138~acl=%2Fstatic%2Fpdf%2F126%2Fbok%25253A978-3-642-16725-6.pdf*~hmac=fc7f6b1368b8834196b4db5b33f-d5b81f4c4bc377a74174afb5e7718a50f06b0.

Krogstie, John; Sindre, Guttorm; Jørgensen, Håvard (2006): Process models representing knowledge for action: a revised quality framework. In: *European Journal of Information Systems* 15 (1), S. 91–102.

Kuechler, Bill; Vaishnavi, Vijay (2008): On theory development in design science research. Anatomy of a research project. In: *European Journal of Information Systems* 17 (5), S. 489–504.

La Rosa, Marcello; ter Hofstede, Arthur H. M.; Wohed, Petia; Reijers, Hajo A.; Mendling, Jan; van der Aalst, Wil M. P. (2011): Managing Process Model Complexity via Concrete Syntax Modifications. In: *IEEE Transactions on Industrial Informatics* 7 (2), S. 255–265.

Lara, Juan de; Guerra, Esther; Sánchez Cuadrado, Jesús (2013): Reusable abstractions for modeling languages. In: *Information Systems* 38 (8), S. 1128–1149.

Latva-Koivisto, Antti M. (2001): Finding a complexity measure for business process models. Helsinki University of Technology. Helsinki (Individual Research Projects in Applied Mathematics, Mat-2.108). Online verfügbar unter http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=0CB43E1590CC38E1ACB97174D8E45F4B?doi=10.1.1.25.2991&rep=rep1&type=pdf, zuletzt geprüft am 19.12.2016.

Lee, Aik; Chan, Chan Hock Chuan (2001): An Exploratory Analysis of Semantic Network Complexity for Data Modeling Performance. In: 5th Pacific Asia Conf. on Information Systems. PACIS. Seoul, Korea, 20. - 22.05.2001. AIS: AISeL.

Leist, Susanne; Johannsen, Florian (2012): Reflecting modeling languages regarding Wand and Weber's Decomposition Model. In: Elmar J. Sinz und Andy Schürr (Hg.): Modellierung 2012. Modellierung. Bamberg, 14-16 März. Gesellschaft für Informatik e.V. Bonn: Gesellschaft für Informatik (LNI, 201), S. 1–12. Online verfügbar unter http://subs.emis.de/LNI/Proceedings/Proceedings201/16.pdf.

Leist, Susanne; Zellner, Gregor (2008): Situational Architecture Engineering (SAE). Improving Strategic Change Through Architecture Methods. In: AIS (Hg.): Proceedings of the International Conference on Information Systems. ICIS. Paris, France, 14-17 December. AIS: AISeL, S. 2.

Ma, Qin; Kelsen, Pierre; Glodt, Christian (2015): A generic model decomposition technique and its application to the Eclipse modeling framework. In: *Software & Systems Modeling* 14 (2), S. 921–952.

Malinova, Monika; Leopold, Henrik; Mendling, Jan (2013): An Empirical Investigation on the Design of Process Architectures. In: Rainer Alt und Bogdan Franczyk (Hg.): Int. Conf. on Wirtschaftsinformatik // Proceedings of the 11th International Conference on Wirtschaftsinformatik (WI2013). Feb. 27 - Mar. 01.2013, Leipzig, Germany. WI2013. Leipzig, 27.02 - 01.03.2013. Internationale Tagung Wirtschaftsinformatik; Internationale Conference on Wirtschaftsinformatik; WI 2013. Leipzig: AISeL.

Malone, Thomas W.; Crowston, Kevin; Lee, Jintae; Pentland, Brian; Dellarocas, Chrysanthos; Wyner, George et al. (1999): Tools for Inventing Organizations: Toward a Handbook of Organizational Processes. In: *Management Science* 45 (3), S. 425–443.

March, Salvatore T.; Smith, Gerald F. (1995): Design and natural science research on information technology. In: *Decision Support Systems* 15 (4), S. 251–266.

Marcus, Andrian; Poshyvanyk, Denys (2005): The conceptual cohesion of classes. In: 21st IEEE Int. Conf. on Software Maintenance. ICSM. Budapest, Ungarn, 25.09 - 30.09.

IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 133–142. Online verfügbar unter http://www.cs.wayne.edu/~severe/publications/Marcus.ICSM.2005.ConceptualCohesionOfClasses.pdf.

Markovic, Ivan; Hasibether, Florian; Sukesh, Jain; Nenad, Stojanovic (2009): Process-oriented Semantic Business Modeling. In: Hans Robert Hansen, Dimitris Karagiannis und Hans-Georg Fill (Hg.): Business services. Konzepte, Technologien, Anwendungen;. Wirtschaftsinformatik. Wien, 25 - 27 February. Wien: Österr. Computer-Ges (Books@ocg.at, 247), S. 683–694.

Mayer, Richard E. (1989): Models for Understanding. In: *Review of Educational Research* 59 (1), S. 43. Online verfügbar unter https://www.uio.no/studier/emner/matnat/ifi/INF3280/v15/pensumliste/additionalliterature/mayer1989modelsforunderstanding.pdf.

McCabe, T.J (1976): A Complexity Measure. In: *IEEE Transactions on Software Engineering* 2 (4), S. 308–320. Online verfügbar unter http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1702388.

Mendling, Jan (2006): Testing Density as a Complexity Metric for EPCs. Vienna University of Economics and Business Administration (Technical Report, JM-2006-11-15).

Mendling, Jan (2008): Metrics for process models. Empirical foundations of verification, error prediction, and guidelines for correctness. Berlin: Springer (LNBIP, 6).

Mendling, Jan; Moser, M.; Neumann, Gustaf; Verbeek, H.M.W; Dongen, Boudewijn F. van; van der Aalst, Wil M. P. (2006): A quantitative analysis of faulty EPCs in the SAP reference model. Hg. v. BPMCenter.org (BPM Center Report, BPM-06-08). Online verfügbar unter http://bpmcenter.org/wp-content/uploads/reports/2006/BPM-06-08.pdf, zuletzt geprüft am 05.10.2012.

Mendling, Jan; Nüttgens, Markus (2006): EPC markup language (EPML). An XML-based interchange format for event-driven process chains (EPC). In: *Information Systems and e-Business Management* 4 (3), S. 245–263.

Mendling, Jan; Reijers, Hajo A.; Cardoso, Jorge (2007): What Makes Process Models Understandable? In: Gustavo Alonso, Peter Dadam und Michael Rosemann (Hg.): Business Process Management, Bd. 4714. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 48–63.

Mendling, Jan; Reijers, Hajo A.; van der Aalst, Wil M. P. (2010): Seven process modelling guidelines (7PMG). In: *Information and Software Technology* 52 (2), S. 127–136.

Mendling, Jan; Sánchez-González, Laura; García, Félix; La Rosa, Marcello (2012a): Thresholds for error probability measures of business process models. In: *Journal of Systems and Software* 85 (5), S. 1188–1197.

Mendling, Jan; Strembeck, Mark (2008): Influence Factors of Understanding Business Process Models. In: Witold Abramowicz und Dieter Fensel (Hg.): International Conference on Business Information Systems. BIS. Innsbruck, Austria, May 5-7: Springer Berlin Heidelberg, S. 142–153. Online verfügbar unter http://link.springer.com/content/pdf/10.1007%2F978-3-540-79396-0_13.pdf.

Mendling, Jan; Strembeck, Mark; Recker, Jan (2012b): Factors of process model comprehension—Findings from a series of experiments. In: *Decision Support Systems* 53 (1), S. 195–206.

Meyers, Timothy M.; Binkley, David (2007): An empirical study of slice-based cohesion and coupling metrics. In: *ACM Transactions on Software Engineering and Methodology* 17 (1), S. 1–27.

Milani, Fredrik; Dumas, Marlon; Matulevičius, Raimundas; Ahmed, Naved; Kasela, Silva (2016): Criteria and Heuristics for Business Process Model Decomposition. In: *Business & Information Systems Engineering* 58 (1), S. 7–17.

Minonne, Clemente; Colicchio, Carlo; Litzke, Matthias; Keller, Thomas (2011): Business Process Management 2011 - Status quo der Zukunft. Eine empirische Studie im deutschsprachigen Europa. 1. Aufl. Zürich: vdf Hochschulverlag AG an der ETH Zürich. Online verfügbar unter http://www.vdf.ethz.ch/info/showDetails.asp?isbnNr=3402.

Minonne, Clemente; Turner, Geoff (2012): Business Process Management-Are You Ready for the Future? In: *Knowledge and Process Management* 19 (3), S. 111–120.

Moody, Daniel L. (2005): Theoretical and practical issues in evaluating the quality of conceptual models. In: *Data & Knowledge Engineering* 55 (3), S. 243–276.

Nordsieck, Fritz (1962): Die schaubildliche Erfassung und Untersuchung der Betriebsorganisation. 6. Aufl. Stuttgart: Poeschel.

Nunamaker, Jay, F., Jr. (1990): Systems developement in information systems research. In: *Journal of Management Information Systems* 7 (3), S. 89–106.

Offermann, Philipp; Levina, Olga; Schönherr, Marten; Bub, Udo (2009): Outline of a design science research process. In: Vijay Vaishnavi und Sandeep Purao (Hg.): Proc. o. 4th DESRIST. DESRIST. Philadelphia, Pennsylvania, May 7-8. ACM. New York: ACM Press, S. 7.

Orme, A.M; Tao, H.; Etzkorn, L.H (2006): Coupling metrics for ontology-based system. In: *IEEE Software* 23 (2), S. 102–108. Online verfügbar unter http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1605186.

Österle, Hubert (1995): Business Engineering Prozeß- und Systementwicklung. Band 1: Entwurfstechniken. Berlin, Heidelberg, s.l.: Springer Berlin Heidelberg.

Overhage, Sven; Birkmeier, Dominik; Schlauderer, Sebastian (2012): Quality Marks, Metrics, and Measurement Procedures for Business Process Models - The 3QM-Framework. In: *Business & Information Systems Engineering* 4 (5), S. 229–246. Online verfügbar unter http://aisel.aisnet.org/bise/vol4/iss5/2.

Paulk, M. C.; Curtis, B.; Chrissis, M. B.; Weber, C. V. (1993): Capability maturity model, version 1.1. In: *IEEE Software* 10 (4), S. 18–27.

Peffers, Ken; Tuunanen, Tuure; Gengler, Charles, E.; Rossi, Matti; Hui, Wendy; Virtanen, Ville; Bragge, Johanna (2006): The Design Science Research Process. A Model for Producing and Presenting Information Systems Research. In: Proceedings of the first international conference on design science research in information systems and technology. DESRIST. Claremont, CA, 24-25 Feb, S. 83–106.

Peffers, Ken; Tuunanen, Tuure; Rothenberger, Marcus A.; Chatterjee, Samir (2007): A Design Science Research Methodology for Information Systems Research. In: *Journal of Management Information Systems* 24 (3), S. 45–77.

Perepletchikov, Mikhail; Ryan, Caspar; Frampton, Keith; Tari, Zahir (2007): Coupling Metrics for Predicting Maintainability in Service-Oriented Designs. In: 18th Australian Software Engineering Conf. ASWEC. Melbourne, Australia, 10.04- 13.04. IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 329–430.

Pieterse, Jurgens (2005): Aris Business process modelling tips. Hg. v. Toolbox.com. Online verfügbar unter http://it.toolbox.com/blogs/enterprise-design/aris-business-process-modelling-tips-5600, zuletzt geprüft am 19.12.2016.

Pinggera, Jakob; Soffer, Pnina; Fahland, Dirk; Weidlich, Matthias; Zugal, Stefan; Weber, Barbara et al. (2013): Styles in business process modeling: an exploration and a model. In: *Software & Systems Modeling*.

Polyvyanyy, A.; Smirnov, Sergey; Weske, Mathias (2008): Process Model Abstraction. A Slider Approach. In: Enterprise Distributed Object Computing Conf. EDOC. Munich, 15.09-19.09. IEEE: IEEE Comput. Soc. Press, S. 325–331.

Poshyvanyk, Denys; Marcus, A. (2006): The Conceptual Coupling Metrics for Object-Oriented Systems. In: 22nd IEEE Int. Conf. on Software Maintenance. ICSM. Philadelphia, USA, 24.09- 27.09. IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 469–478. Online verfügbar unter http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4021375.

Pries-Heje, Jan; Baskerville, Richard L (2008): The Design Theory Nexus. In: *MIS Quarterly* 32 (4), S. 731–755. Online verfügbar unter http://student.bus.olemiss.edu/files/Conlon/Others/Others/DesignScience_MISQ/THE%20DESIGN%20THEORY%20NEXUS.pdf.

Pries-Heje, Jan; Baskerville, Richard L; Venable, John R. (2007): Soft Design Science Research. Extending the Boundaries of Evaluation in Design Science Research. In: Proceedings of DESRIST 2007. International Conference on Design Science Research in Information Systems and Technology. Pasadena, S. 18–38.

Pries-Heje, Jan; Baskerville, Richard L; Venable, John R. (2008): Strategies for Design Science Research Evaluation. In: Willie Golden, Thomas Acton, Kieran Conboy, van der Heijden, Hans und Virpi Kristiina Tuunainen (Hg.): Proc. o. 16th ECIS. ECIS. Galway, Ireland, S. 255–266. Online verfügbar unter http://aisel.aisnet.org/ecis2008/87.

Purao, Sandeep; Vaishnavi, Vijay (2003): Product metrics for object-oriented systems. In: *ACM Computing Surveys* 35 (2), S. 191–221.

Qian, Kai; Liu, Jigang; Tsui, Frank (2006): Decoupling Metrics for Services Composition. In: Roger Lee und Naohiro Ishii (Hg.): Proc. o. t. 5th Annual IEEE/ACIS Int. Conf. on Computer and Information Science. ICIS-COMSAR. Honolulu, Hawaii, 10.07-12.07. IEEE; AIS. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 44–47.

Quynh, P.T; Thang, H.Q (2009): Dynamic coupling metrics for service-oriented software. In: *International Journal on Computer Science Engineering* (3), S. 282–287. Online verfügbar unter http://soict.hut.edu.vn/~thanghq/Papers/PTQuynh+HQ%20Thang%20-%202009.06.06.pdf.

Rajaraman, Chandrashekar; Lyu, M.R (1992): Reliability and maintainability related software coupling metrics in C++ programs. In: 3rd int. symposium on software reliability engineering. ISSRE. Research Triangle Park, USA, 07.10- 10.10. IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 303–311.

Recker, Jan Christian (2011): Evaluations of Process Modeling Grammars. Berlin, Heidelberg, Brisbane: Springer (LNBIP, 71). Online verfügbar unter http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10441764.

Recker, Jan Christian; Indulska, Marta; Rosemann, Michael; Green, Peter (2005): Do Process Modelling Techniques Get Better? A Comparative Ontological Analysis of BPMN. In: Bruce Campbell, Jim Underwood und Deborah Bunker (Hg.): 16th Australasian

Conference on Information Systems. ACIS. Sydney, Australia, November, 30 - December 2.

Recker, Jan Christian; Rosemann, Michael; Green, Peter; Indulska, Marta (2011): Do ontological deficiencies in modeling grammars matter? In: *MIS Quarterly* 35 (1), S. 57–79.

Recker, Jan Christian; Rosemann, Michael; Indulska, Marta; Green, Peter (2009): Business Process Modeling. A comparative analysis. In: *Journal of the american society for information science* 10 (4), S. 333–363.

Recker, Jan Christian; Safrudin, Norizan; Rosemann, Michael (2012): How novices design business processes. In: *Information Systems* 37 (6), S. 557–573.

1471, 2000: Recommended Practice for Architectural Description for Software-Intensive Systems. Online verfügbar unter http://standards.ieee.org/findstds/standard/1471-2000.html.

Reijers, H. A.; Mendling, J.; Dijkman, R. M. (2010): On the Usefulness of Subprocesses in Business Process Models. Hg. v. BPMCenter.org. Eindhoven (BPM Reports, 1003). Online verfügbar unter http://repository.tue.nl/ef25a703-a25e-4f49-a306-b6721d678fc3.

Reijers, Hajo A.; Mendling, Jan (2011): A Study Into the Factors That Influence the Understandability of Business Process Models. In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 41 (3), S. 449–462.

Reijers, Hajo A.; Mendling, Jan; Dijkman, R. M. (2011): Human and automatic modularizations of process models to enhance their comprehension. In: *Information Systems and e-Business Management* (36), S. 881–897.

Reijers, Hajo A.; Vanderfeesten, Irene T.P. (2004): Cohesion and coupling metrics for workflow process design. In: Jörg Desel, Barbara Pernici und Mathias Weske (Hg.): Business process management. BPM. Potsdam, 17.06. Berlin: Springer (LNCS, 3080), S. 290–305.

Roeser, Tobias; Kern, Eva-Maria (2015): Surveys in business process management – a literature review. In: *Business Process Management Journal* 21 (3), S. 692–718.

Rosemann, Michael (2006): Potential pitfalls of process modeling: part A. In: *Business Process Management Journal* 12 (2), S. 249–254.

Rosemann, Michael; Green, Peter (2002): Developing a meta model for the Bunge–Wand–Weber ontological constructs. In: *Information Systems* 27 (2), S. 75–91.

Rosemann, Michael; Vom Brocke, Jan (2015): The Six Core Elements of Business Process Management. In: Jan Vom Brocke und Michael Rosemann (Hg.): Handbook on Business Process Management 1: Springer Berlin Heidelberg (International handbooks on information systems), S. 105–122. Online verfügbar unter http://dx.doi.org/10.1007/978-3-642-45100-3_5.

Rosenberg, L.H; Hyatt, L.E (1997): Software quality metrics for object-oriented environments. In: *Crosstalk Jounal* 10 (4).

Ross, Jeanne W. (2003): Creating a Strategic IT Architecture Competency: Learning in Stages. Hg. v. Sloan School of Management. Sloan School of Management (MIT Sloan School of Management Working Paper, 4314-03). Online verfügbar unter http://dspace.mit.edu/bitstream/1721.1/3526/2/4314-03.pdf.

Rothenberg, Jeff (1989): The nature of modeling. In: Lawrence E. Widman (Hg.): Artificial intelligence, simulation, and modeling. New York u.a.: Wiley (A Wiley Interscience publication), S. 75–92.

Sandhu, Parvinder Singh; Singh, Hardeep (2005): A Critical Suggestive Evaluation of CK Metric. In: Pacific Asia Conf. on Information Systems. PACIS. Bangkok, Thailand, 07.- 10.07.2007. AIS: AISeL.

Sarshar, Kamyar; Loos, Peter (2005): Comparing the Control-Flow of EPC and Petri Net from the End-User Perspective. In: David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Hg.): Business Process Management, Bd. 3649. Berlin, Heidelberg: Springer (LNCS), S. 434–439.

Scheer, August-Wilhelm (2000): ARIS - business process modeling. With 179 figures. 3. Aufl. Berlin et al.: Springer.

Scheer, August-Wilhelm; Thomas, Oliver; Adam, Otmar (2005): Process Modeling Using Event-Driven Process Chains. In: Marlon Dumas, Wil M. P. van der Aalst und Arthur H. ter Hofstede (Hg.): Process-aware information systems: John Wiley and Sons, S. 119–146.

Schekkerman, Jaap (2006): How to survive in the jungle of enterprise architecture frameworks. Creating or choosing an enterprise architecture framework. 3. ed. Victoria: Trafford.

Schmelzer, Hermann J.; Sesselmann, Wolfgang (2008): Geschäftsprozessmanagement in der Praxis. Kunden zufrieden stellen, Produktivität steigern, Wert erhöhen ; [das Standardwerk]. 6., vollst. überarb. und erw. Aufl. München: Hanser.

Schütte, Reinhard (1998): Grundsätze ordnungsmäßiger Referenzmodellierung. Konstruktion konfigurations-und anpassungsorientierter Modelle: Gabler Verlag.

Schütte, Reinhard; Rotthowe, Thomas (1998): The Guidelines of Modeling. An Approach to Enhance the Quality in Information Models. In: Tok Wang Ling, Sudha Ram und Mong-Li Lee (Hg.): Proceedings ER '98. Proceedings. ER. Singapur, China, 16.11. Berlin et al.: Springer (LNCS, 1507), S. 240–254.

Sein, Maung K.; Henfridsson, Ola; Purao, Sandeep; Rossi, Matti; Lindgren, Rikard (2011): Action Design Research. In: *MIS Quarterly* 35 (1), S. 37–56. Online verfügbar unter http://content.ebscohost.com/pdf25_26/pdf/2011/MIS/01Mar11/59551157.pdf?T=P&P=AN&K=59551157&S=R&D=buh&EbscoContent=dGJyMNHr7ESep7M4yOvsOLCmr0yeprdSsay4S7OWxWXS&ContentCustomer=dGJyMPGrt0y1rLVMuePfgeyx44Dt6fIA.

Shadish, William R. (2002): Revisiting field experiments: Field notes for the future. In: *Psychological Methods* 7 (1), S. 3–18.

Shadish, William R.; Cook, Thomas D.; Campbell, Donald T. (2002): Experimental and quasi-experimental designs for generalized causal inference. Boston [u.a.]: Houghton Mifflin.

Sharp, Alec; McDermott, Patrick (2009): Workflow modeling. Tools for process improvement and applications development. 2. Aufl. Boston: Artech House. Online verfügbar unter http://site.ebrary.com/lib/academiccompletetitles/home.action.

Silver, Bruce (2008): Ten tips for effective process modeling. BPMInstitute.org. Online verfügbar unter http://www.bpminstitute.org/resources/articles/bpms-watch-ten-tips-effective-process-modeling, zuletzt geprüft am 09.10.2013.

Simon, Herbert A. (2008): The sciences of the artificial. 3. ed., [Nachdr.]. Cambridge, Mass: MIT Press.

Smirnov, Sergey; Reijers, Hajo A.; Weske, Mathias (2011): A Semantic Approach for Business Process Model Abstraction. In: Mouratidis Haralambos und Rolland Colette (Hg.): Proceedings of the 23rd international conference on Advanced information systems engineering. CAiSE. London, UK, June 20-24. Berlin, Heidelberg: Springer (LNCS, 6741), S. 497–511.

Söllner, René (2014): Die wirtschaftliche Bedeutung kleiner und mittlerer Unter- nehmen in Deutschland. Statistisches Bundesamt. Wiesbaden. Online verfügbar unter https://www.destatis.de/DE/Publikationen/WirtschaftStatistik/UnternehmenGewerbeanzeigen/BedeutungKleinerMittlererUnternehmen_12014.pdf;jsessionid=1799272A2D2BD96C09460DBD0F135EA2.cae1?__blob=publicationFile, zuletzt geprüft am 16.12.2016.

Sonnenberg, Christian; Vom Brocke, Jan (2012a): Evaluation Patterns for Design Science Research Artefacts. In: Markus Helfert und Brian Donnellan (Hg.): Practical Aspects of Design Science, Bd. 286. Berlin et al.: Springer (Communications in Computer and Information Science, 286), S. 71–83.

Sonnenberg, Christian; Vom Brocke, Jan (2012b): Evaluations in the Science of the Artificial. Reconsidering the Build-Evaluate Pattern in Design Science Research. In: Ken Peffers, Marcus A. Rothenberger und Bill Kuechler (Hg.): Proc. of DESRIST. Advances in Theory and Practice. DESRIST. Las Vegas, USA, May 14-15. Berlin, Heidelberg: Springer (LNCS, 7286), S. 381–397.

Stauss, Bernd; Seidel, Wolfgang (2004): Complaint management. The heart of CRM. Mason OH u.a.: Thomson/South-Western.

Sunju, Oh; Joongho, Ahn (2009): Ontology Module Metrics. In: 2009 IEEE Int. Conf. on e-Business Engineering. ICEBE. Macau, China, 21.10-23.10. IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press, S. 11–18. Online verfügbar unter http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5342136.

Sweller, John (1988): Cognitive Load During Problem Solving. Effects on Learning. In: *Cognitive Science* 12 (2), S. 257–285. Online verfügbar unter http://onlinelibrary.wiley.com/store/10.1207/s15516709cog1202_4/asset/s15516709cog1202_4.pdf?v=1&t=ivkothah&s=84038a991b6a322713305e7e09054b751236aad3.

Takeda, Hideaki; Veerkamp, Paul; Tomiyama, Tetsuo; Yoshikawa, Hiroyuki (1990): Modeling Design Process. In: *AI Magazine* 11 (4), S. 37–48.

Trkman, Peter (2010): The critical success factors of business process management. In: *International Journal of Information Management* 30 (2), S. 125–134.

Uhl, Jürgen (2004): „Unternehmensarchitekturen“ ist ein Dauerthema – aber die Ziele bzw. Motivation und damit Schwerpunkte ändern sich, vor allem mit wirtschaftlichen Randbedingungen. In: *WIRTSCHAFTSINFORMATIK* 46 (4), S. 317.

Újházi, Béla; Ferenc, Rudolf; Poshyvanyk, Denys; Gyimóthy, Tibor (2010): New Conceptual Coupling and Cohesion Metrics for Object-Oriented Systems. In: 10th Working Conf. on Source Code Analysis and Manipulation. SCAM. Timisoara, Rumania, 12.-13.09. IEEE. Los Alamitos, USA: IEEE Comput. Soc. Press.

Vaishnavi, Vijay; Kuechler, William (2007): Design science research methods and patterns. Innovating information and communication technology. Boca Raton: Auerbach Publications. Online verfügbar unter http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10193862.

van der Aalst, Wil M. P. (2012): A Decade of Business Process Management Conferences: Personal Reflections on a Developing Discipline. In: Alistair Barros, Avigdor Gal und Ekkart Kindler (Hg.): Business process management. 10th international conference, BPM 2012, Tallinn, Estonia, September 3 - 6, 2012 ; proceedings, Bd. 7481. Berlin: Springer (Lecture Notes in Computer Science, 7481), S. 1–16.

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

van der Aalst, Wil M. P.; Desel, Jörg; Kindler, Ekkart (2002): On the semantics of EPCs: A vicious circle. In: Markus Nüttgens und Frank J. Rump (Hg.): Proceedings des GI-Workshops und Arbeitskreistreffens EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten. EPK. Trier, November 2002. Gesellschaft für Informatik e.V.: Gesellschaft für Informatik, S. 71–80.

van der Aalst, Wil M. P.; Nakatumba, Joyce; Rozinat, Anne; Russell, Nick (2010): Business Process Simulation. In: Jan Vom Brocke und Michael Rosemann (Hg.): Handbook on Business Process Management 1. Berlin, Heidelberg: Springer, S. 313–318.

van Hee, Kees M.; Oanea, Olivia; Sidorova, Natalia (2005): Colored Petri Nets to Verify Extended Event-Driven Process Chains. In: Robert Meersman, Zahir Tari, Mohand-Said Hacid, John Mylopoulos, Barbara Pernici, Özalp Babaoglu et al. (Hg.): Confederated International Conferences CoopIS, DOA, and ODBASE. OTM. Agia Napa, Zypern, 31.10- 04.11. Berlin, Heidelberg: Springer (LNCS, 3760), S. 183–201. Online verfügbar unter http://www.dimap.ufrn.br/~jair/ES/artigos/ObjSysAna.pdf.

van Looy, Amy; Backer, Manu de; Poels, Geert (2011): Defining business process maturity. A journey towards excellence. In: *Total Quality Management & Business Excellence* 22 (11), S. 1119–1137.

Vanderfeesten, Irene T.P.; Cardoso, Jorge; Mendling, Jan; Reijers, Hajo A.; van der Aalst, Wil M. P. (2007a): Quality Metrics for Business Process Models. In: Layna Fischer (Hg.): BPM and workflow handbook. Methods, concepts, case studies and standards in business process management and workflow. Lighthouse Point, USA: Future Strategies, S. 179–191.

Vanderfeesten, Irene T.P.; Cardoso, Jorge; Reijers, Hajo A. (2007b): A weighted coupling metric for business process models. In: Johann Eder, Stein L. Tomassen und Andreas L Opdahl (Hg.): Proc. of the CAiSE'07 Forum at the 19th Int. Conf. on Advanced Information Systems Engineering. CAiSE. Trondheim, Norwegen, 11.06- 15.06: CEUR-WS (CEUR Workshop Proceedings, 247), S. 41–44.

Vanderfeesten, Irene T.P.; Reijers, Hajo A.; Mendling, Jan; van der Aalst, Wil M. P.; Cardoso, Jorge (2008a): On a quest for good process models. The cross-connectivity metric. In: Zohra Bellahsène und Michel Lénoard (Hg.): 20th Int. Conf. Advanced Information Systems Engineering. CAiSE. Montpeiller, Frankreich, 16.06.2008. Berlin, Heidelberg, New York, NY: Springer (LNCS, 5047), S. 480–494.

Vanderfeesten, Irene T.P.; Reijers, Hajo A.; Mendling, Jan; van der Aalst, Wil M. P.; Cardoso, Jorge (2008b): On a quest for good process models. The cross-connectivity metric. In: Zohra Bellahsène und Michel Lénoard (Hg.): 20th Int. Conf. Advanced Information Systems Engineering. CAiSE. Montpeiller, Frankreich, 16.06.2008. Berlin, Heidelberg, New York, NY: Springer (LNCS, 5047), S. 480–494.

Vanderfeesten, Irene T.P.; Reijers, Hajo A.; van der Aalst, Wil M. P. (2008c): Evaluating workflow process designs using cohesion and coupling metrics. In: *Computers in industry* 59 (5), S. 420–437.

Venable, John R.; Pries-Heje, Jan; Baskerville, Richard L (2012): A Comprehensive Framework for Evaluation in Design Science Research. In: Ken Peffers, Marcus A. Rothenberger und Bill Kuechler (Hg.): Proc. of DESRIST. Advances in Theory and Practice. DESRIST. Las Vegas, USA, May 14-15. Berlin, Heidelberg: Springer (LNCS, 7286), S. 423–438.

Venkatesh, Viswanath; Brown, Susan A.; Bala, Hillol (2013): Bridging the qualitative-quantitative divide. Guidelines for conducting mixed methods research in information systems. In: *MIS Quarterly* 37 (1), S. 21–54.

Vom Brocke, Jan; Simons, Alexander; Niehaves, Björn; Riemer, Kai; Plattfaut, Ralf; Cleven, Anne (2009): On the Importance of Rigour in Documenting the Literature Search Process. Reconstructing the Giant. In: Susan Newell, Edgar A. Whitley, Nancy Pouloudi, Jonathan Wareham und Lars Mathiassen (Hg.): 17th European Conf. on Information Systems, ECIS 2009. ECIS. Verony, Italy, S. 2206–2217.

Wahler, Ksenia; Küster, Jochen Malte (2008): Predicting Coupling of Object-Centric Business Process Implementations. In: Marlon Dumas, Manfred Reichert und Ming-Chien Shan (Hg.): Business process management. 6th International conference, BPM 2008, Milan, Italy, September 2-4, 2008 : proceedings. BPM. Milano, Italy, 02.09- 04.09. Berlin: Springer (LNCS, 5240), S. 148–163.

Walter, S. D.; Feinstein, A. R.; Wells, C. K. (1987): Coding Ordinal Independent Variables in Multiple Regression Analyses. In: *American Journal of Epidemiology* 125 (2), S. 319–323. Online verfügbar unter http://aje.oxfordjournals.org/content/125/2/319.full.pdf.

Wand, Y.; Weber, R. (1989a): A model of systems decomposition. In: Janice I. DeGross, John C. Henderson und Benn R. Konsynski (Hg.): Tenth International Conference on Information Systems (ICIS). ICIS. Boston, Massachusetts, United States, 1989: AISeL, S. 41–51.

Wand, Yair; Weber, Ron (1989b): A Model of Control and Audit Procedure Change in Evolving Data Processing Systems. In: *Accounting Review* 64 (1), S. 87–108. Online verfügbar unter http://content.ebscohost.com/pdf25_26/pdf/1989/ARW/01Jan89/4485928.pdf?T=P&P=AN&K=4485928&S=R&D=buh&EbscoContent=dGJyMNHr7ESeqLE40d-vuOLCmr0qep7VSr6i4Sa%2BWxWXS&ContentCustomer=dGJyMPGrt0y1rLV-MuePfgeyx44Dt6fIA.

Wand, Yair; Weber, Ron (1990a): An ontological model of an information systems. In: *IEEE Transactions on Software Engineering* 16 (11), S. 1282–1292. Online verfügbar unter http://www-public.it-sudparis.eu/~gibson/Teaching/CSC7302/ReadingMaterial/WandWeber90.pdf.

Wand, Yair; Weber, Ron (1990b): Mario Bunge's ontology as a formal foundation for information systems concepts. In: Paul Weingartner und Georg W. J. Dorn (Hg.): Studies on Mario Bunge's Treatise - Google Books. Atlanta: Rodopi. Online verfügbar unter http://books.google.de/books?hl=de&lr=&id=w-soo6UdwhAC&oi=fnd&pg=PA123&dq=mario+bunge%27s+ontology+as+a+formal+foundation+for+information+systems+concepts&ots=OhdbfLQtX1&sig=QkS4bifrEDbMk5NvYgv7yfqm08o#v=onepage&q=mario%20bunge%27s%20ontology%20as%20a%20formal%20foundation%20for%20information%20systems%20concepts&f=false.

Weber, Barbara; Reichert, Manfred; Mendling, Jan; Reijers, Hajo A. (2011): Refactoring large process model repositories. In: *Computers in industry* 62 (5), S. 467–486.

Weber, Ron (1997): Ontological foundations of information systems. Melbourne: Coopers & Lybrand (Accounting research methodology monograph, 4).

Weyuker, Elaine J. (1988): Evaluating software complexity measures. In: *IEEE Transactions on Software Engineering* 14 (9), S. 1357–1365. Online verfügbar unter http://origin-www.computer.org/plugins/dl/pdf/trans/ts/1988/09/e1357.pdf?template=1&loginState=2&userData=University%2BLibrary%2Bof%2BRegensburg%253AUniversity%2BLibrary%2Bof%2BRegensburg%253AAddress%253A%2B132.199.123.228%252C%2B%255B140.98.196.191%252C%2B132.199.123.228%252C%2B195.95.193.146%255D.

Wilde, Thomas; Hess, Thomas (2007): Forschungsmethoden der Wirtschaftsinformatik. Eine empirische Untersuchung. In: *WIRTSCHAFTSINFORMATIK* 49 (4), S. 280–287.

Wohlin, Claes; Runeson, Per; Höst, Martin; Ohlsson, Magnus, C.; Regnell, Björn; Wesslén, Anders (2012): Experimentation in software engineering. Berlin, New York: Springer.

Zachman, J. A. (1987): A framework for information systems architecture. In: *IBM Systems Journal* 26 (3), S. 276–292. Online verfügbar unter https://www.zachman.com/images/ZI_PIcs/ibmsj2603e.pdf.

Zugal, Stefan (2013): Applying Cognitive Psychology for Improving the Creation, Understanding and Maintenance of Business Process Models. Dissertation. University of Innsbruck, Innsbruck. Department of Computer Science, zuletzt geprüft am 15.04.2014.

Zugal, Stefan; Pinggera, Jakob; Weber, Barbara; Mendling, Jan; Reijers, Hajo A. (2011): Assessing the Impact of Hierarchy on Model Understandability. A Cognitive Perspective. In: Jörg Kienzle (Hg.): Models in Software Engineering. Workshops and Symposia at MODELS 2011. MODELS. Wellington, New Zealand, 16.10-21.10 (LNCS, 7167), S. 123–133.

Zugal, Stefan; Soffer, Pnina; Haisjackl, Cornelia; Pinggera, Jakob; Reichert, Manfred; Weber, Barbara (2013): Investigating expressiveness and understandability of hierarchy in declarative business process models. In: *Software & Systems Modeling*.